Generic Timer Module 104 (GTM104) Reference Manual

Document Number: GTM104RM
Rev 1, 3 Feb 2015

freescale™

Contents

Section number **Title** **Page**

Chapter 1
Introduction

Chapter 2
GTM Architecture

Chapter 3
Advanced Routing Unit (ARU)

Chapter 4
Broadcast (BRC) Module

Chapter 5
First In First Out Module (FIF0)

Chapter 6
AEI to FIFO Data Interface (AFD)

Chapter 7
FIFO-to-ARU (F2A) Unit

Chapter 8
Clock Management Unit (CMU)

Chapter 9
Time Base Unit (TBU)

Chapter 10
Timer Input Module (TIM)

Chapter 11
Timer Output Module (TOM)

Chapter 12
ARU-connected Timer Output Module (ATOM)

Chapter 13
Multi Channel Sequencer (MCS)

Chapter 14
Memory Configuration (MCFG)

Chapter 15
TIM0 Input Mapping Module (MAP)

Chapter 16
Digital PLL Module (DPLL)

Chapter 17
Sensor Pattern Evaluation (SPE)

Chapter 18
Interrupt Concentrator Module (ICM)

Chapter 19
Output Compare Unit (CMP)

Chapter 20
Monitor Unit (MON)

Chapter 1
Introduction

1.1 Overview

The Generic Timer Module (GTM) is composed of several submodules of different functionality. These submodules can be combined in a configurable manner to form a complex timer module that serves different application domains and different classes within one application domain. Because of this scalability and configurability the timer is called generic. The scalability and configurability is reached with an architecture philosophy where dedicated hardware submodules are located around a central routing unit called Advanced Routing Unit (ARU). The ARU can connect the submodules in a flexible manner. The connectivity is software programmable and can be configured during runtime.

Nevertheless, the GTM is designed to unload the CPU or a peripheral core from a high interrupt load. Most of the tasks inside the GTM can run (once set up by an external CPU) independent and in parallel to the software. There may be special situations, where the CPU has to take action but the goal of the GTM design was to reduce these situations to a minimum. In addition, the timer section of GTM runs at a fixed clock frequency (80 MHz) to allow real time period counting, but the interface to CPU can work with a frequency variable clock.

The GTM is designed to unload the CPU or a peripheral core from a high interrupt load. Most of the tasks inside the GTM can run, after setup by an external CPU, independently and in parallel to the software. There may be special situations, where the CPUhas to take action, but the goal of the GTM design is to reduce these situations to a minimum.

The hardware submodules have dedicated functionalities. By combining the functionality of several submodules through the ARU, complex functions can be established. For example, signals at an input module can be routed to a signal processing unit where intermediate values for the frequencies of the incoming signals can be calculated.

There are five groups of submodules:

1. Submodules that help to implement complex functionalities are called *infrastructural components*. These components are present in all GTM variants. However, the number of these components may vary from one GTM implementation to another.
2. Submodules that have general architectures that can fulfil typical timer functions, such as PWM generation units.
3. Submodules that fulfil a dedicated functionality for a certain application domain. For example, the DPLL serves engine management applications.
4. Submodules that are responsible for supporting the implementation of safety functions to fulfil a defined safety level.
5. The ICM module is responsible for interrupt services.

Each GTM implementation is built with submodules from those four groups. An application class is defined by which and how many of those submodules are integrated into the GTM implementation.

1.2 Features

The Generic Timer Module (GTM) has these general features:

- Time base generators
- Advanced data signal / time base router
- Input signal processors (up to 48 inputs)
- Simple output signal generators (up to 80 outputs)
- Microprocessed output signal generators (up to 72 outputs)
- Digital PLL
- BLDC support
- Safety features - GTM is a non-safety relevant module
  - Output compare unit
  - Monitoring unit
- CPU bus interface clock operates asynchronously to timer section clock, allowing CPU low power operation.

1.3 Document Structure

GTM Architecture Overview describes the dedicated GTM implementation for which this specification is written. It gives an overview of the implemented submodules, and their quantity, within this dedicated GTM implementation.

ARU Overview through TBU Overview deal with infrastructural components for routing, clock management, and common time base functions. TIM Overview through ATOM Overview describe the signal input and output modules.

MCS Overview explains the signal processing and generation submodule. MCFG Overview outlines a memory configuration module for the described signal processing and generation submodule.

The next sections provide detailed descriptions of application specific and safety related modules like the MAP, DPLL, SPE, CMP and MON submodules. ICM Overview describes a module that bundles several interrupts coming from other submodules, how they are bundled, and how the bundled result is externally connected.

The submodules are shown in Table 1-1.

Table 1-1. GTM submodules

Chapter	Submodule	Group
3	Advanced Routing Unit (ARU), see ARU Overview	Infrastructural components
4	Broadcast Module (BRC), see BRC Overview	Infrastructural components
5	First In First Out Module (FIFO), see FIFO Overview	Infrastructural components
6	AEI-to-FIFO Data Interface (AFD), see AFD Overview	Infrastructural components
7	FIFO-to-ARU Interface (F2A), see F2A Overview	Infrastructural components
8	Clock Management Unit (CMU), see CMU Overview	Infrastructural components
9	Time Base Unit (TBU), see TBU Overview	Infrastructural components
10	Timer Input Module (TIM), see TIM Overview	IO Modules
11	Timer Output Module (TOM), see TOM Overview	IO Modules
12	ARU-connected Timer Output Module (ATOM), see ATOM Overview	IO Modules
13	Multi Channel Sequencer (MCS), see MCS Overview	Signal generation and processing
14	Memory Configuration Module (MCFG), see MCFG Overview	Infrastructural component for MCS
15	TIM0 Input Mapping Module (MAP), see MAP Overview	Dedicated
16	Digital PLL (DPLL), see DPLL Overview	Dedicated
17	Sensor Pattern Evaluation Module (SPE), see SPE Overview	BLDC support
18	Interrupt Concentrator Module (ICM), see ICM Overview	Interrupt services
19	Output Compare Unit (CMP), see CMP Overview	Safety features
20	Monitoring Unit (MON), see MON Overview	Safety features

Chapter 2
GTM Architecture

2.1 GTM Architecture Overview

The GTM forms a generic timer platform that serves different application domains and different classes within those application domains. Depending on these multiple requirements of application domains, multiple device configurations with different quantities of submodules (i.e. ATOM, BRC, MCS, PSM, SPE, TIM, TOM) and a different count of channels per submodule (if applicable) are possible. In this document, an example of possible device configurations for the GTM realization is outlined. The architecture of the GTM is depicted in Figure 2-1. Please note, that the size of the submodules in the figure does not reflect the die size of the modules in the final RTL implementation. The device dependent configuration (i.e the count of submodules) is listed in the device specific Appendix B of this document.

2.1.1 GTM Architecture

As an example of a possible device configuration, the GTM104 realization is shown in the figure below.

The device dependent configuration (i.e. the count of submodules and channels per submodule) is listed in the device specific Appendix B of this document.

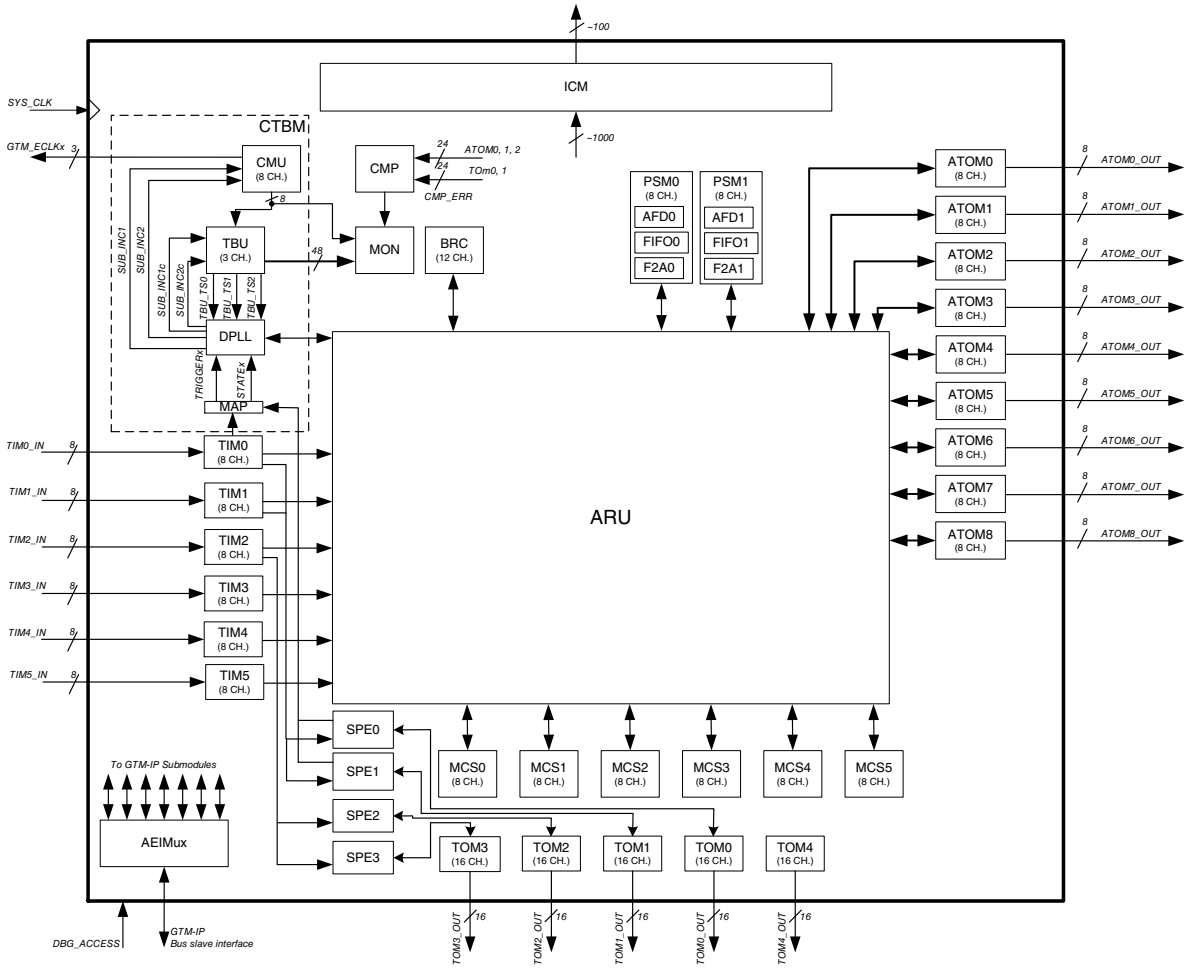


Figure 2-1. GTM104 architecture block diagram

2.1.2 Advanced Routing Unit (ARU) Brief Description

The Advanced Routing Unit (ARU) is the central component of the GTM. Most of the submodules are located around, and connected to, the ARU. The ARU forms, together with the Broadcast (BRC) module and the Parameter Storage Module (PSM), the infrastructural part of the GTM. The ARU uses a round-robin scheduling scheme of connected channels to route data from a connected source submodule to a connected destination submodule. The routing is done in a deterministic manner with a worst case round-trip time.

The routed data word size from the ARU is 53 bits. The data word, shown in the figure below, can be logically split into three parts.

1. ACB,
2. Data1, and
3. Data0.

The ACB part contains control bits to send control information from one submodule to another. The ARU Control Bits (ACB) can have a different meaning for different submodules.

The DATA0 and DATA1 parts typically hold data for the operation registers of the GTM. This data can be, for example, the duty cycle and period duration of a measured PWM input signal or the characteristic of an output PWM that is to be generated. Other possible content for Data0 and Data1 is two 24-bit values of the GTM time bases: TBU_TS0, TBU_TS1 and TBU_TS2.

It is possible to route data from a source to a destination, and later, that destination can act as a source for another destination. Routes through the GTM are called 'data streams'. For a detailed description of the ARU submodule, please refer to Chapter 3.

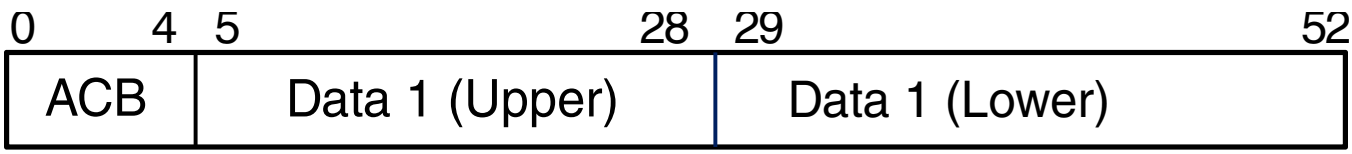


Figure 2-2. ARU data word

2.1.3 Broadcast Module (BRC) Brief Description

The Broadcast Module (BRC) can distribute data from one source submodule to more than one destination submodule, provided that the submodules are connected to the ARU.

2.1.4 Parameter Storage Module (PSM) Brief Description

The Parameter Storage Module (PSM) consists of three subunits:

1. AEI-to-FIFO Data Interface (AFD),
2. FIFO-to-ARU Interface (F2A), and
3. the FIFO itself.

The PSM can serve as a data storage for incoming data characteristics or as parameter storage for outgoing data. The data is stored in a RAM that is logically located inside the FIFO subunit, but physically implemented and integrated outside of the GTM. Accordingly, the GTM provides the interface to the RAM at its module boundary.

2.1.5 Timer Input Modules (TIMs) Brief Description

Signals are received into the GTM via the TIMs. The TIM s can filter the input signals and extrapolate additional information from them. Each channel is, for example, able to measure pulse high or low times and the period of a PWM signal in parallel and route the values to the ARU for further processing. The internal operation registers of the TIM submodule are 24 bits wide.

2.1.6 Clock Management Unit (CMU) Brief Description

The CMU provides up to 13 different clocks, and it can source up to three external clock pins, GTM_ECLK[0:2]. It acts as a clock divider for the system clock. The counters implemented inside other submodules are typically driven from the CMU.

NOTE

The CMU clocks are implemented as enable signals for the counters while the whole system runs with the GTM global clock, SYS_CLK.

2.1.7 Time Base Unit (TBU) Brief Description

The Time Base Unit (TBU) submodule provides three independent, common time bases for the GTM. In general, the number of time bases depends on the needs of the implemented device. If three time bases are implemented, two of the time bases can also be clocked with the sub_inc1c and sub_inc2c outputs from the Digital Phased Locked Loop (DPLL) submodule.

2.1.8 Digital Phase Locked Loop (DPLL) Brief Description

The DPLL generates higher frequency clock signals, sub_inc1, sub_inc2, sub_inc1c and sub_inc2c, based on the frequencies of up to two input signals. Input signals are filtered in the TIM0 submodule and then transferred to the MAP submodule, where two of six input signals are routed to the DPLL for further processing.

2.1.9 Timer Output Module (TOM) Brief Description

The TOM submodules and the ATOM submodules generate signal outputs. Each TOM is able to generate a PWM signal at its output. With the use of an integrated shadow register, multiple TOM channels can be used to generate complex PWM outputs, with the

parameters being serviced by the CPU. It is possible to trigger TOM channels for a successor TOM submodule through a trigger line between TOM(x)_CH(15) and TOM(x +1)_CH(0). However, to avoid long trigger paths, the trigger signal is saved in a register at the TOM submodule output, resulting in one SYS_CLK cycle delay of the trigger signal.

In addition, each TOM submodule can integrate functions to drive one Brushless Direct Current (BLDC) engine. BLDC support is established with a TOM, a TIM, and the Sensor Pattern Evaluation (SPE) submodule.

2.1.10 ARU-connected Timer Output Module (ATOM) Brief Description

ATOM submodules and TOM submodules generate output signals.

The ATOM submodules offer additional functionality over TOM submodules for generating complex output waveforms that do not require CPU interaction. The characteristics of those complex waveforms can be controlled by other submodules (that are connected to the ARU), such as the PSM or the Multi Channel Sequencer (MCS). While the internal operation and shadow registers of the TOM channels are 16 bits wide, the operation and shadow registers of the ATOM channels are 24 bits wide. The higher resolution of the ATOM channels allow comparison to time base values coming from the TBU.

It is possible to trigger ATOM channels for a successor ATOM submodule through a trigger line between ATOM(x)_CH(7) and ATOM(x+1)_CH(0). However, to avoid long trigger paths, the trigger signal is saved in a register at the output of each second ATOM submodule in the trigger line. This results in one SYS_CLK cycle delay of the trigger signal.

Together with the MCS, the ATOM is able to generate an arbitrary predefined output sequence at the GTM's external output pins. The output sequence is defined by instructions located in RAM that is connected to the MCS submodule. The instructions define the points were an output signal should change or react with other signal inputs. The output points can be one or two time stamps (or even an angle stamp in case of an engine management system) that are provided by the TBU. Since the MCS is able to read data from the ARU, it is also able to operate on incoming data routed from the TIM. Additionally, the MCS can process data that is located in its connected RAMs. Like the PSM, the MCS RAM is located logically inside the MCS, but the RAM is physically implemented and integrated outside of the GTM.

2.1.11 Interrupt Concentrator Module (ICM) Dreif Description

The Interrupt Concentrator Module (ICM) groups and concentrates interrupt sources from the other submodules, and provides interrupt requests at the GTM's boundary.

2.1.12 Compare (CMP) Module and Monitor (MON) Module Brief Description

The CMP and MON modules can implement safety related features. The CMP compares two functionally identical output channels from an ATOM or TOM and sends the result to the MON submodule, where any difference is signalled to the CPU. The MON submodule is also able to monitor activities in the ARU and CMU.

2.2 Signal Multiplex

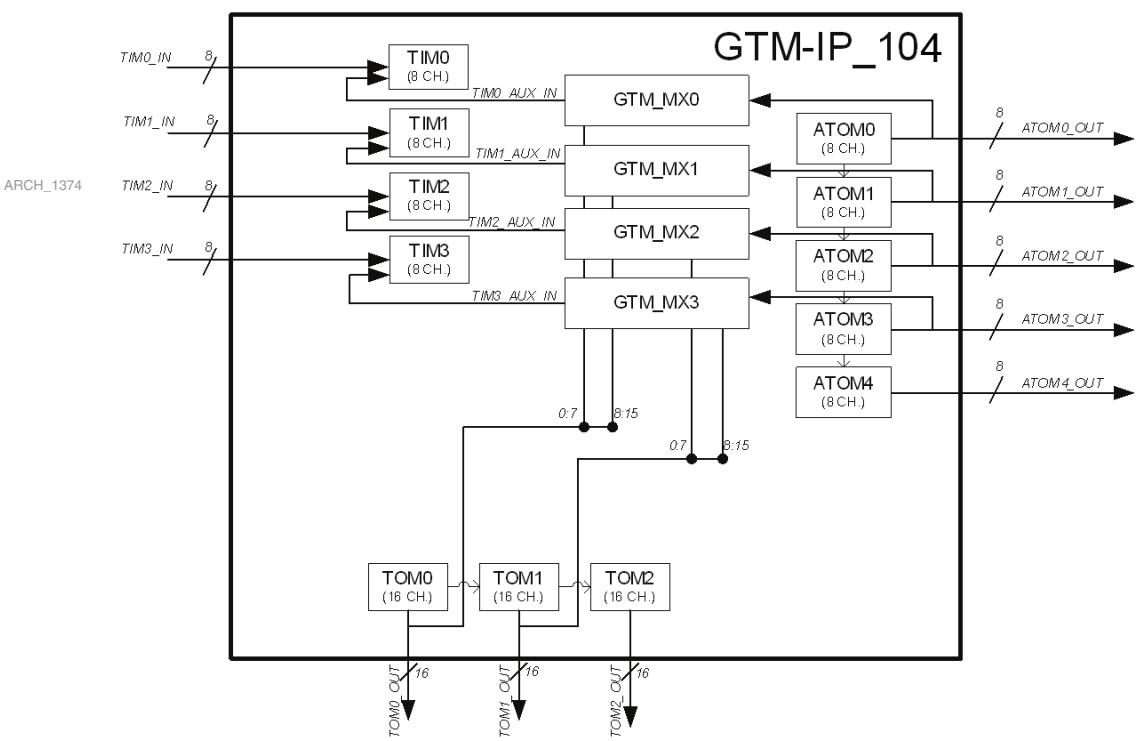


Figure 2-3. Signal Multiplexing

NOTE

GTM104 has more modules and more MXn's [6].

2.3 Signal Connectivity Overview

The source selection is defined with the bit SRXx in the register GTM_TIM[y]_AUX_IN_SRC .

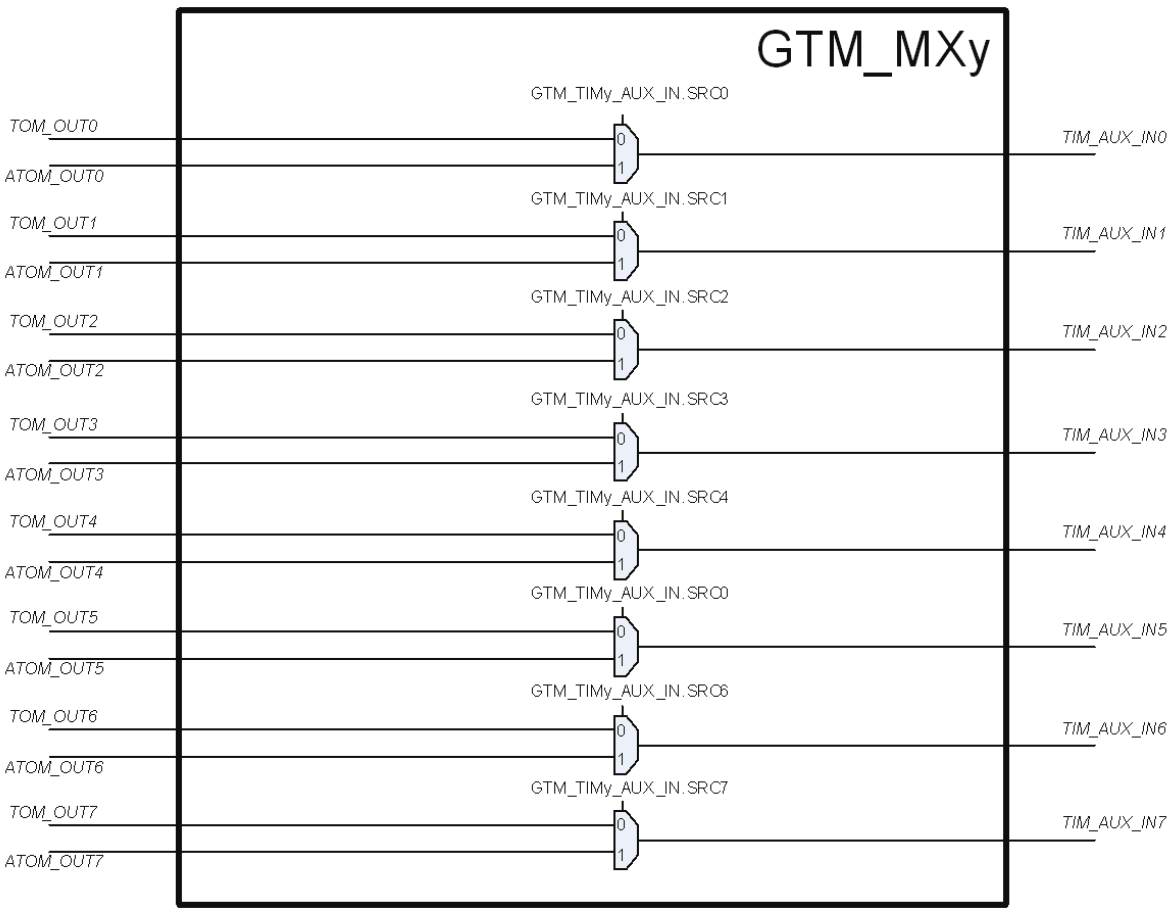


Figure 2-4. Signal Connectivity

2.4 GTM Interfaces

The GTM has four interface groups:

1. TIM input interface, through which incoming signals are received for analysis in a TIM,
2. TOM and ATOM output interfaces, through which outgoing signals, created in a TOM or ATOM , are sent,
3. Generic bus interface, through which the GTM can be connected to the SoC system bus (please see GTM Generic Bus Interface (AEI), and

4. ICM interface, through which internal interrupts that have been concentrated by the ICM (please see GTM Interrupt Concept) are sent to the dedicated CPU environment (where handling of each interrupt can be different).

2.4.1 GTM Generic Bus Interface (AEI)

The GTM has a generic bus interface, AE-Interface (AEI), that can be widely adapted to different SoC bus systems. The adaptation of the AEI to SoC buses is typically done with a bridge module translating the AEI signals to the SoC bus signals. The AEI bus signals are show in Table 2-1.

Table 2-1. AEI bus signals

Signal name	I/O	Description		Bit width
AEI_SEL	I	GTM select line		1
AEI_ADDR	I	GTM address		32
AEI_PIPE	I	AEI Address phase signal		1
AEI_W1R0	I	Read/Write access		1
AEI_WDATA	I	Write data bus		32
AEI_RDATA	O	Read data bus		32
AEI_READY	O	Data ready signal		1
AEI_STATUS	O	AEI Access status value:	AEI Access status description:	2
		00	No Error[1]	
		01	Illegal Byte Addressing[2]	
		10	Illegal Address Access[3]	
		11	Unsupported Address[4]	

1. The signal value 00 is driven if no error occurred during AEI access.
2. The signal value 01 is driven if the bus address is not an integer multiple of 4 (byte addressing).
3. The signal value 10 is driven if an illegal write access to one of the write protected registers is performed. A register is protected by setting its RF_PROT bit. In the case that an illegal write access is signaled by status "10", the register will not be modified.
4. The signal value 11 is driven if the address is not handled in the GTM.

Reading registers will never return status "10".

Write access to following registers returns status "10" under special conditions:

Table 2-2. Registers with write protected bit(s)

Registers		
ARU_IRQ_FORCINT	DPLL_IRQ_FORCINT	MCS[i]_CHn_PC
ATOM[i]_CHn_CM0	DPLL_ID_PMTR_n	MCS[i]_CHn_IRQ_FORCINT
ATOM[i]_CHn_CM1	DPLL_INC_CNT1	MCS[i]_CTRL

Table continues on the next page...

Table 2-2. Registers with write protected bit(s) (continued)

Registers		
ATOM[i]_CHn_SR0	DPLL_INC_CNT2	TBU_CHn_BASE
ATOM[i]_CHn_SR1	DPLL_TSACn	TBU_CHn_CTRL
ATOM[i]_CHn_RDADDR	DPLL_PSACn	MCS RAM during initialization
ATOM[i]_CHn_IRQ_FORCINT	DPLL_ACB_n	DPLL RAM during initialization
BRC_IRQ_FORCINT	F2A[i]_CHn_ARU_RD_FIFO	GTM_IRQ_MODE when written with non zero value
BRC_SRC_n_ADDR	F2A[i]_CHn_STR_CFG	ARU_IRQ_MODE when written with non zero value
CMP_IRQ_FORCINT	FIFO[i]_CHn_IRQ_FORCINT	BRC_IRQ_MODE when written with non zero value
CMU_GCLK_NUM	GTM_IRQ_FORCINT	FIFO[i]_CHn_IRQ_MODE when written with non zero value
CMU_GCLK_DEN	GTM_RST	TIM[i]_CHn_IRQ_MODE when written with non zero value
CMU_CLK_CTRLn	SPE[i]_IRQ_FORCINT	TOM[i]_CHn_IRQ_MODE when written with non zero value
CMU_ECLK_NUM	TIM[i]_CHn_CNTS	ATOM[i]_CHn_IRQ_MODE when written with non zero value
CMU_ECLK_DEN	TIM[i]_CHn_GPR1	MCS[i]_CHn_IRQ_MODE when written with non zero value
CMU_FXCLK_CTRL	TIM[i]_CHn_IRQ_FORCINT	DPLL_IRQ_MODE when written with non zero value
DPLL_ACT_STA	TOM[i]_CHn_IRQ_FORCINT	SPE[i]_IRQ_MODE when written with non zero value
DPLL_OSW	MCS[i]_CHn_CTRL	CMP_IRQ_MODE when written with non zero value

The AEI_STATUS bus signal is not visible but the AEI_STATUS signal '10' is known from the GTM_IRQ_NOTIFY register AEI Illegal Module address interrupt, AEI_IM_ADDR,

2.4.2 GTM Multi-master and multi-tasking support

To support multi-master and multi-task access to the registers of the GTM, a dedicated write-access scheme is used for critical control bits inside the GTM.

This scheme can be, for example, a shared register where more than one channel can be controlled globally by a single register write access.

Such shared register bits are implemented inside the GTM with a double bit mechanism, where the writing of:

- 00 has no effect,

- 01 sets the bit,
- 10 resets the bit, and
- 11 has no effect.

If the CPU wants to read the status of the bit, it always gets a:

- 00 if the bit is reset, and
- 11 if the bit is set.

2.5 ARU Routing Concept

The routing mechanism of the ARU submodule for data streams is a central concept of the GTM. Each data word transferred between the ARU and its connected submodule is 53 bits wide. Each module that is connected to the ARU can provide an arbitrary number of ARU write channels and an arbitrary number of ARU read channels. The ARU write channels are named data sources and the ARU read channels are named data destinations.

The ARU does not implement a switch matrix, but it implements a data router with serialized connectivity that provides the same interconnection flexibility. Principle of data routing using ARU describes the ARU data routing principle. Data sources are marked with a green rectangle and data destinations are marked with yellow rectangles. The dashed lines depict the configurable connections between data sources and data destinations. A connection between a data source and a data destination is also called a data stream.

2.5.1 Principle of data routing using ARU

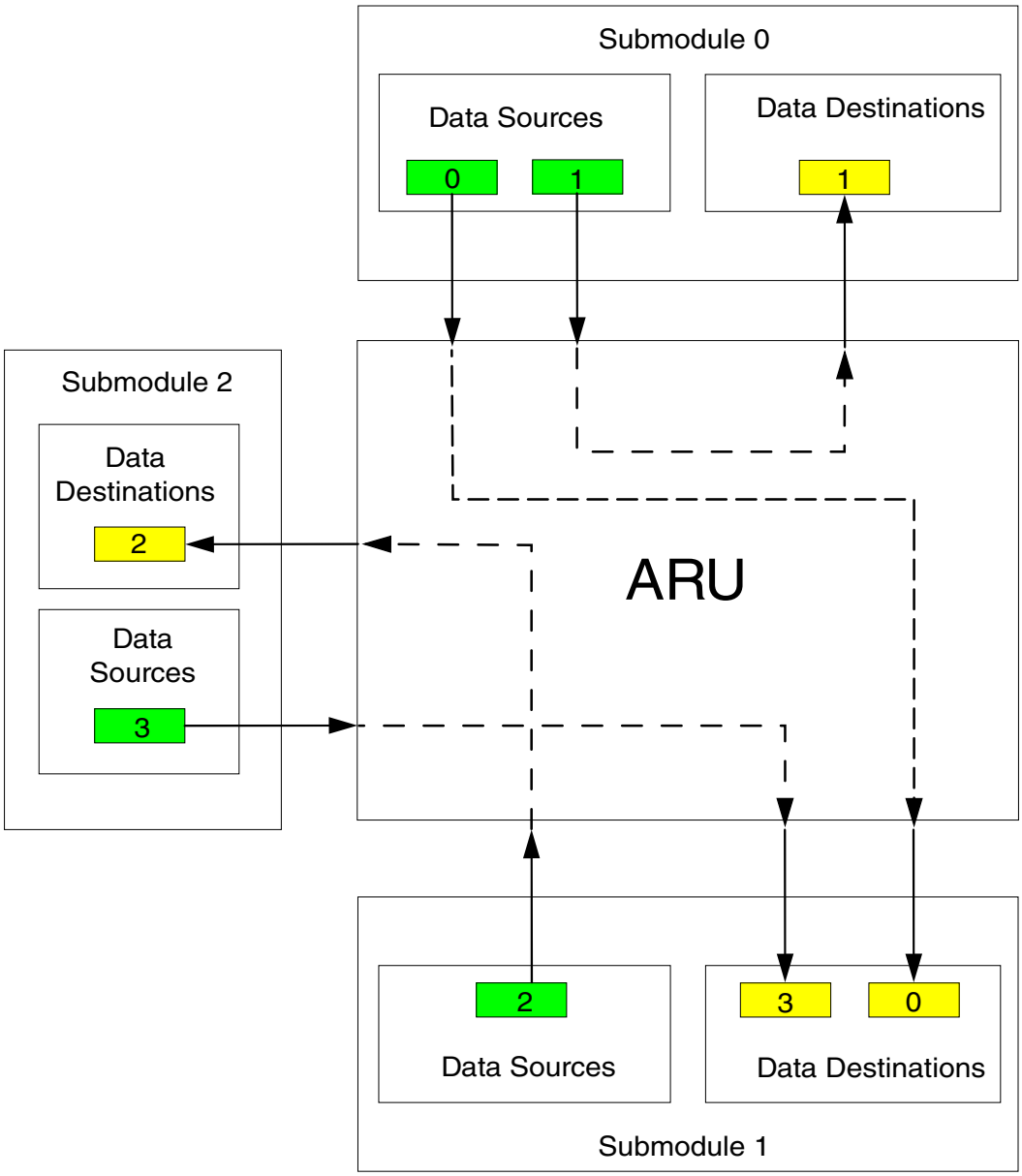


Figure 2-5. ARU data routing

Each data source has its fixed and unique source address. The fixed address of each data source is depicted by the numbers in the green boxes of Figure 2-5. The address definitions of all available data sources, in the GTM, can be obtained from TBD. The connection from a specific data source to a specific data destination is defined by configuring the corresponding address of a data source in the desired data destination. The configured address of each data destination is pointed out by the numbers in the yellow boxes of Figure 2-5.

Normally, the destination is idle and waits for data from the source. If the source offers new data, the destination does a destructive read, processes the data, and goes idle again. The same data is never read twice. However, this destructive read access does not hold for the BRC submodule when it is configured in Maximal Throughput Mode. For a description of the BRC submodule, please refer to BRC Overview.

The ARU sequentially polls the data destinations of the connected modules in a round-robin order. If a data destination requests new data from its configured data source and the data source has data available, the ARU delivers the data to the destination and it informs both, the data source and destination that the data is transferred. The data source marks the delivered ARU data as invalid, which means that the destination consumed the data.

NOTE

Each data source should only be connected to a single data destination. This is because the destinations consume the data. If two destinations reference the same source, one destination will consume the data before the other destination could consume it. Since the data transfers are blocking, the second destination will block until it receives new data from the source. The Broadcast (BRC) submodule must be used If a data source must be connected to more than one data destination. A transfer from a data source to the ARU is also blocking, which means that the source channel can only provide new data to the ARU when an old data word is consumed by a destination.

In order to speed up the latency of data transfers, the ARU handles two different data destinations in parallel. Therefore, a transfer between a source and a destination takes two cycles, but since the transfers are pipelined, these two cycles have only the latency of one ARU round trip (please see ARU Round Trip Time).

The quantity of data sources and destination for each applicable submodule is shown in Table 2-3.

Table 2-3. Data sources and destinations for the listed submodules

Submodule	Quantity of data sources	Quantity of data destinations
ARU	1	0
DPLL	24	24
TIM 0-3	32	0
MCS 0-3	96	32
BRC	22	12
TOM	0	0
ATOM 0-4	40	40

Table continues on the next page...

Table 2-3. Data sources and destinations for the listed submodules (continued)

Submodule	Quantity of data sources	Quantity of data destinations
PSM 0	8	8
CMP	0	0
MON	0	0
Total	223	116

2.5.2 ARU Round Trip Time

The ARU uses a round-robin arbitration scheme with a fixed round trip time for all connected data destinations. The time between two adjacent read requests always takes the round trip time, independent of whether the read request succeeds or fails.

2.5.3 ARU Blocking Mechanism

The ARU blocking mechanism is implemented for transferring data from a data source to a data destination. This mechanism is used by ARU-connected submodules to synchronize the submodules to the routed data streams (see Figure 2-6).

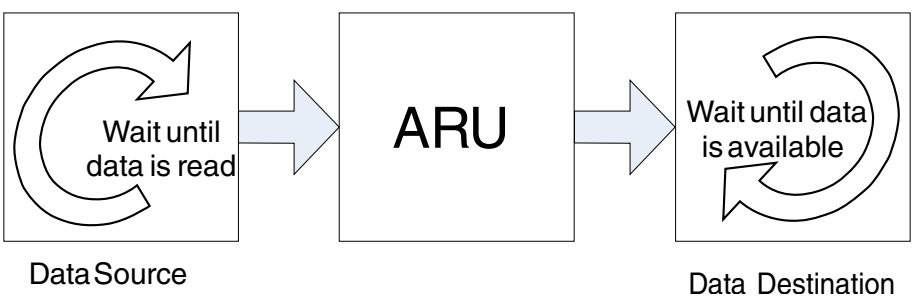


Figure 2-6. ARU blocking mechanism

If a data destination requests data from a data source over the ARU, but the data source does not have any data yet, the data destination has to wait until the data source provides new data. In this case, the submodule that owns the data destination may perform other tasks. When a data source produces new data faster than a data destination can consume it, the source raises an error interrupt to signal that its data could not be accepted in time by the destination. New data is then marked as valid for further transfers and the old data is overwritten.

The round trip time for the ARU is always fixed for a specific device configuration. Please refer to the device specific Appendix B of this specification for detailed information.

An exception is when the BRC submodule is configured in Maximal Throughput Mode (please see BRC Overview).

2.6 GTM Clock and Time Base Management (CTBM)

Clock and Time Base Management (CTBM) submodules, TBU, CMU, DPLL, and MAP (shown in the following figure) together provide clock and time base management of the entire GTM.

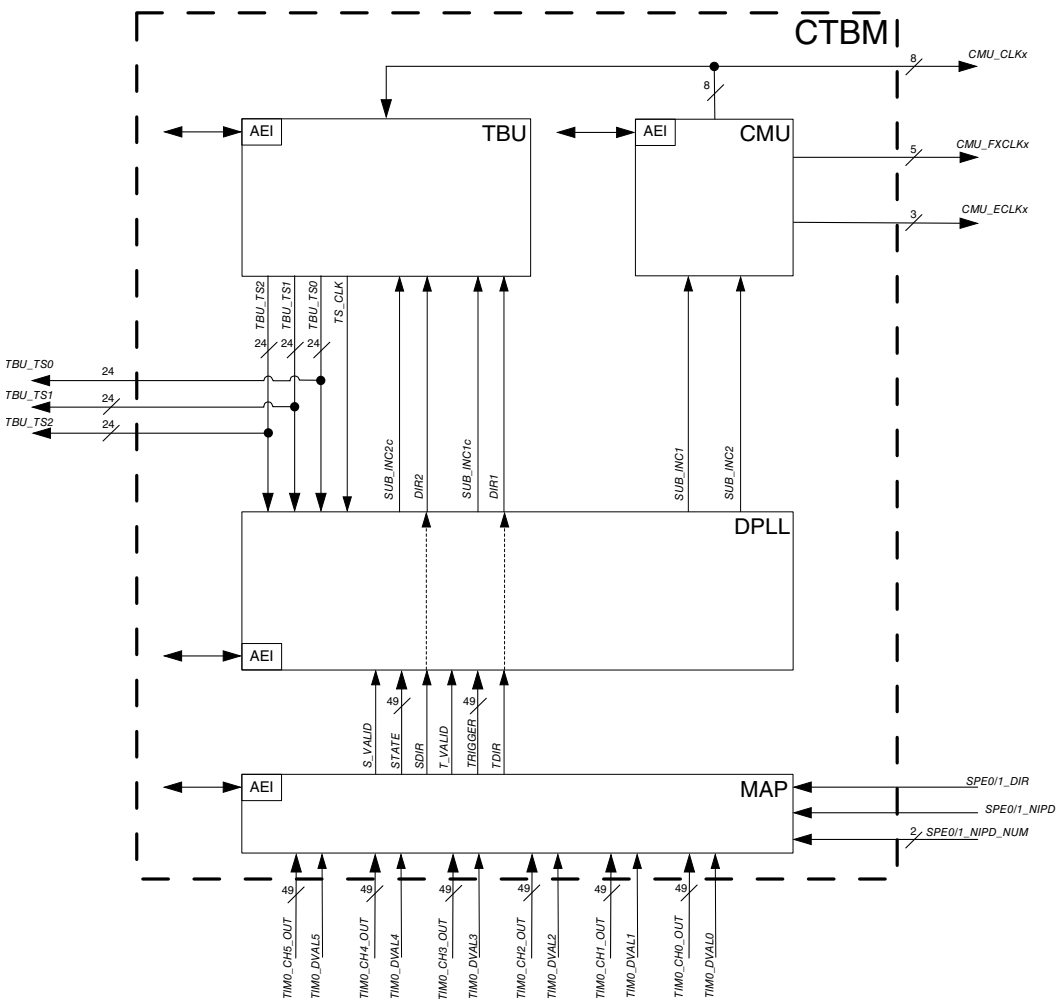


Figure 2-7. Clock and time base management diagram

The Clock Management Unit (CMU) generates up to 13 clocks for other GTM submodules and up to three GTM external clocks *CMU_ECLK[z]* (z: 0..2). For a description of CMU functionality and clocks please refer to CMU Overview.

Five CMU clocks, *CMU_FXCLK[y]* (y: 0..4), are used by the TOM submodule for PWM generation. A maximum of eight CMU clocks, *CMU_CLK[x]* (x: 0..7), are used by other GTM submodules for signal generation.

The Time Base Unit (TBU) can use one of the *CMU_CLK[x]* clocks per channel to generate a common time base for the GTM. The TBU submodule generates three time base signals, *TBU_TS0*, *TBU_TS1* and *TBU_TS2*, which are widely used inside the GTM as common time bases for signal characterization and generation.

Alternatively, the TBU can use compensated *SUB_INC[i]c* (i: 1,2) signals sourced by the DPLL submodule (please see DPLL Overview) for time base generation. This time base typically provides an angle clock for an engine management system. Two direction signal lines, *DIR[i]*, are used to cause the TBU time base to run forward or backward. TBU functionality is described in TBU Overview.

The DPLL submodule (please see DPLL Overview) multiples two input signal vectors, *TRIGGER* and *STATE,* (sourced by the MAP submodule) to source four clock signals, *SUB_INC[i]* (i: 1,2) and *SUB_INC[i]c* (i: 1,2).

The MAP submodule receives six signals sourced by the TIM0 submodule. The MAP then selects two of the six signals, generates a *TRIGGER* signal and a *STATE* signal, and sources them to the DPLL. The MAP also receives the SPE0 and SPE1 signals sourced by the SPE submodule, generates a *TDIR* (TRIGGER Direction) signal and a *SDIR* (STATE Direction) signal from a defined SPE0 and SPE1 input pattern, and provides them to both the DPLL and TBU submodules. For a description of the MAP submodule, please refer to MAP Overview.

2.7 GTM Interrupt Concept

GTM submodules can generate thousands of interrupts from internal events. The Interrupt Concentrator Module (ICM) combines these interrupts into interrupt groups, and then bundles the interrupt groups into a smaller quantity of interrupt sets. Out of the interrupt sets, a smaller quantity of interrupt signals is created and sourced external to the GTM. The external interrupt signals are named *GTM_<MOD>_IRQ,* where <MOD> is the name of the corresponding GTM submodule.

Each output signal, *GTM_<MOD>_IRQ*, has a corresponding input signal, *GTM_<MOD>_IRQ_CLR*, that can be used for clearing the associated interrupts. These input signals can be sourced by the surrounding CPU system, for example, to indicate:

- an acknowledge signal from a DMA controller,
- a validation signal from an ADC, or
- a clear signal from an GTM-external interrupt controller to do an atomic clear while executing an Interrupt Service Routine (ISR).

Each individual interrupt is controlled inside the submodules. If a submodule consists of several submodule channels that are most likely to work independently from each other (like TIM, PSM, MCS, TOM, and ATOM), each submodule channel has its own interrupt control and status register. Other submodules (like SPE, ARU, DPLL, BRC, CMP and global GTM functionality) have a common interrupt control and status register for the whole submodule. An interrupt control and status register, together, is called an interrupt set.

An interrupt set consists of four registers:

1. **IRQ_EN** register,
2. **IRQ_NOTIFY** register,
3. **IRQ_FORCINT** register, and
4. **IRQ_MODE** register.

The **IRQ_EN**, **IRQ_NOTIFY,** and **IRQ_FORCINT** registers show status and allow control of each individual interrupt source within an interrupt set. The **IRQ_MODE** register configures the interrupt mode that is applied to all interrupts in the set.

The **IRQ_EN** register controls the enabling and disabling of an individual interrupt. Only enabled interrupts, when they occur, can assert their associated *GTM_<MOD>_IRQ* external signals.

The **IRQ_NOTIFY** register collects the occurrence of interrupt events. The behavior for setting a bit in this register depends on the configured interrupt mode (see Level interrupt mode, Pulse interrupt mode, Pulse-notify interrupt mode, Single-pulse interrupt mode). Independent of the configured mode, any write access with value '1' to a bit in the **IRQ_NOTIFY** register always clears the corresponding **IRQ_NOTIFY** bit.

The enabling of a disabled interrupt source by a write access to the **IRQ_EN** register also clears the corresponding bit in the **IRQ_NOTIFY** register. However, if the enabling of a disabled interrupt is simultaneous to an incoming interrupt event, the interrupt event is dominant and the corresponding bit in the **IRQ_NOTIFY** register is not cleared. Each write access to the **IRQ_MODE** register, clears all bits in the **IRQ_NOTIFY** register. However, the clearing of IRQ_NOTIFY is applied indepentently of the written data (e.g. no mode change).

Thus, a secure way for reconfiguring the interrupt mode of an interrupt set is to:

- disable all interrupts of the interrupt set by writing to the IRQ_EN register,
- define the new interrupt mode by writing to the IRQ_MODE register, followed by
- enabling the desired interrupts by writing to the IRQ_EN register.

Thus, a secure way for reconfiguring the interrupt mode of an error interrupt set is to:

- disable all error interrupts of the error interrupt set by writing to the EIRQ_EN register,

- define the new interrupt mode by writing to the IRQ_MODE register, followed by
- enabling the desired error interrupts by writing to the EIRQ_EN register.

Software can trigger individual interrupts by writing a value '1' to the corresponding bit in the **IRQ_FORCINT** register. Since a write access to **IRQ_FORCINT** only generates a single pulse, **IRQ_FORCINT** is not implemented as a true register. Any read access to **IRQ_FORCINT** always returns a value of '0'.

NOTE

The mechanism for triggering interrupts with **IRQ_FORCINT** is globally disabled after reset. To use the mechanism, it has to be explicitly enabled first by clearing the **GTM_CTRL[RF_PROT**] bit.

For the AEI-bridge, BRC, FIFO, TIM, MCS, DPLL, SPE and CMP modules, each interrupt may configured to assert (instead of the normal interrupt) an error interrupt (if enabled by the corresponding error interrupt enable bit in the EIRQ_EN register).

NOTE

It is possible for a source to enable the normal interrupt and the error interrupt in parallel. Because both interrupt clear signals could reset the notify bit, this is expected to cause problems in a system and therefore it is strongly recommended to not enable both interrupt types at the same point in time.

Similar to enabling an interrupt, the enabling of a disabled error interrupt source with a write access to the EIRQ_EN register also clears the corresponding bit in the IRQ_NOTIFY register. However, if the enabling of a disabled error interrupt is simultaneous to an incoming interrupt event, the interrupt event is dominant and the IRQ_NOTIFY register is not cleared.

All enabled error interrupts are or-combined inside the ICM and assigned to the dedicated gtm_err_irq GTM port. A corresponding gtm_err_irq_clr input allows the reset of this error interrupt from outside the GTM (hardware clear).

To be able to detect the module source of the error interrupt, the ICM provides the ICM_IRQG_MEI register. The error interrupt-causing channel can be determined for the FIFO, TIM, and MCS modules by evaluating the ICM_IRQG_CE10 to ICM_IRQG_CE14 registers.

To support a wide variety of CPU architectures and interrupt systems, four GTM interrupt modes can be configured:

- Level interrupt mode,

- Pulse interrupt mode,
- Pulse-notify interrupt mode, and
- Single-pulse interrupt mode

2.7.1 Level interrupt mode

Level Interrupt Mode is the default interrupt mode for the GTM. Each interrupt event that has occurred is collected in the **IRQ_NOTIFY** register independent of the corresponding enable bit in the **IRQ_EN** register.

An interrupt event, which is defined as a pulse on the *int_out* signal (see Figure 2-8), may be triggered by:

- an interrupt source in the submodule, or
- software performing a write access to the corresponding **IRQ_FORCINT** register, but only when the **GTM_CTRL[RF_PROT]** bit is value '0'.

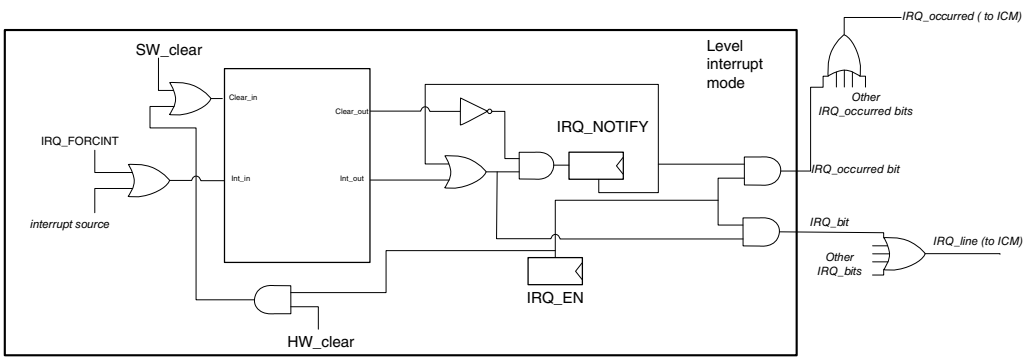


Figure 2-8. Level interrupt mode scheme

A collected interrupt bit in the **IRQ_NOTIFY** register may be cleared by a clear event, which is defined as a pulse on the *clr_out* signal . A clear event can be initiated by:

- writing value '1' to the corresponding bit in the **IRQ_NOTIFY** register, which then causes a pulse on the *SW_clear* signal, or
- assertion of an externally connected *GTM_<MOD>_IRQ_CLR* signal, which is internally routed to the *HW_clr* signal.

In the case of a simultaneous interrupt event and clear event, interrupt events are dominant as shown in Table 2-4.

Table 2-4. Events priority

int_in	Clear_in	int_out	Cear_out
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	0

An interrupt event that has occurred is signaled as a constant signal level with value '1' to the signal *IRQ_bit*, if the corresponding interrupt is enabled in the **IRQ_EN** register.

With the exception of the ARU and DPLL submodules, the *IRQ_bit* signal is OR-combined with the neighboring *IRQ_bit* signals from the same interrupt set and they are routed as a single *IRQ_line* signal to the interrupt concentrator module (ICM). The interrupt *IRQ_bit* signals from the DPLL and ARU submodules are routed directly as a *IRQ_line* signal to the ICM submodule. For the TOM and ATOM submodules, the ICM submodule may further OR-combine several *IRQ_line* signals to form an outgoing interrupt *GTM_<MOD>_IRQ* signal. In other cases, the ICM submodule directly connects the *IRQ_line* signals to the outgoing interrupt *GTM_<MOD>_IRQ* signals.

The signal *IRQ_occurred* is connected in a similar way as the *IRQ_line* signal, however this signal is used for monitoring the interrupt state of the **IRQ_NOTIFY** register in the ICM submodule.

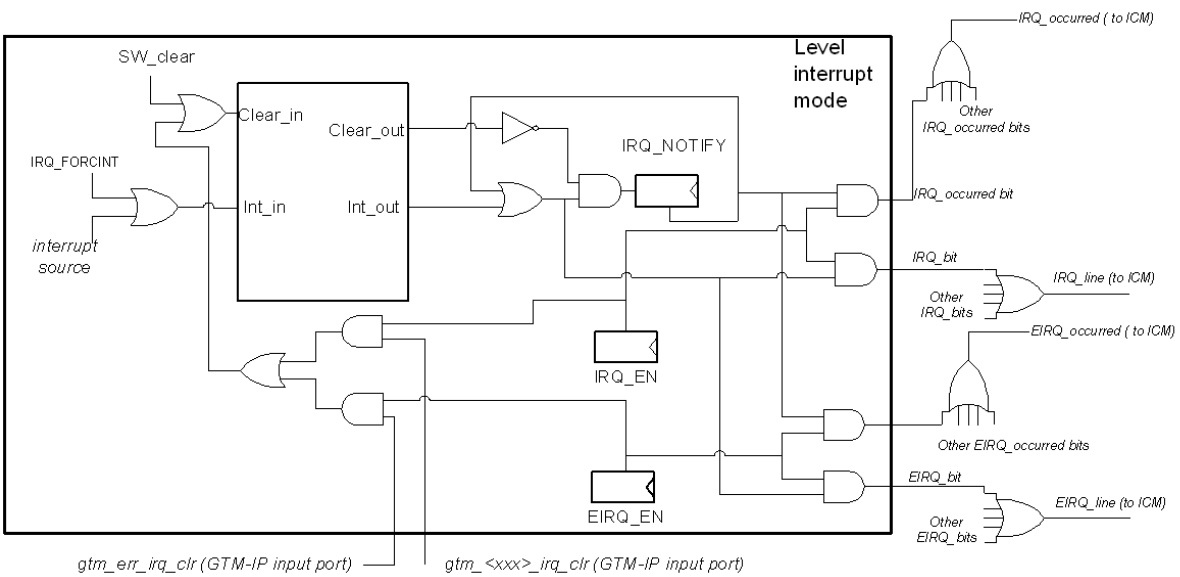


Figure 2-9. Level interrupt scheme for modules AEI-bridge, BRC, FIFO, TIM, MCS, DPLL, SPE, CMP

A collected interrupt bit in the IRQ_NOTIFY register may be cleared by a clear event, which is defined as a pulse on signal Clear_out in the figure above. A clear event can be performed with a write access with value '1' to the corresponding bit in the IRQ_NOTIFY register, leading to a pulse on signals SW_clear. A clear event may also result from externally connected signal gtm_<MOD>_irq_clr or gtm_err_irq_clr, which is routed as a HW_clear to the Clear_in signal in the figure above. However, the hardware clear mechanism is only possible if the corresponding interrupt or error interrupt is enabled by the IRQ_EN or EIRQ_EN registers

.As can be seen in the figure above, an occurred interrupt event is signalled as a constant signal level with value 1 to the signal IRQ_bit if the corresponding interrupt is enabled in the IRQ_EN register.

2.7.2 Pulse interrupt mode

In Pulse Interrupt Mode, each interrupt event will generate a pulse on the IRQ_bit signal line (see Figure 2-10), but only if the corresponding IRQ_EN bit is enabled.

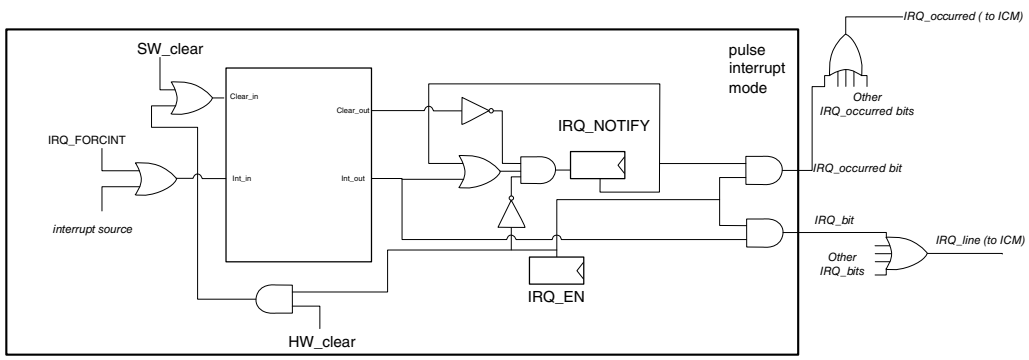


Figure 2-10. Pulse interrupt mode

In pulse interrupt mode:

- Each interrupt event that has occurred generates a pulse on the *IRQ_bit* signal, but only if the corresponding **IRQ_EN** bit is enabled.
- The interrupt bit in the **IRQ_NOTIFY** register is always cleared if the corresponding **IRQ_EN** bit is enabled. However, if the corresponding **IRQ_EN** bit is disabled, an interrupt event that has occurred is captured in the **IRQ_NOTIFY** register to allow software to poll for disabled interrupts. The **IRQ_NOTIFY** register may be cleared by an interrupt clear event.
- The IRQ_occurred signal is always value '0'.

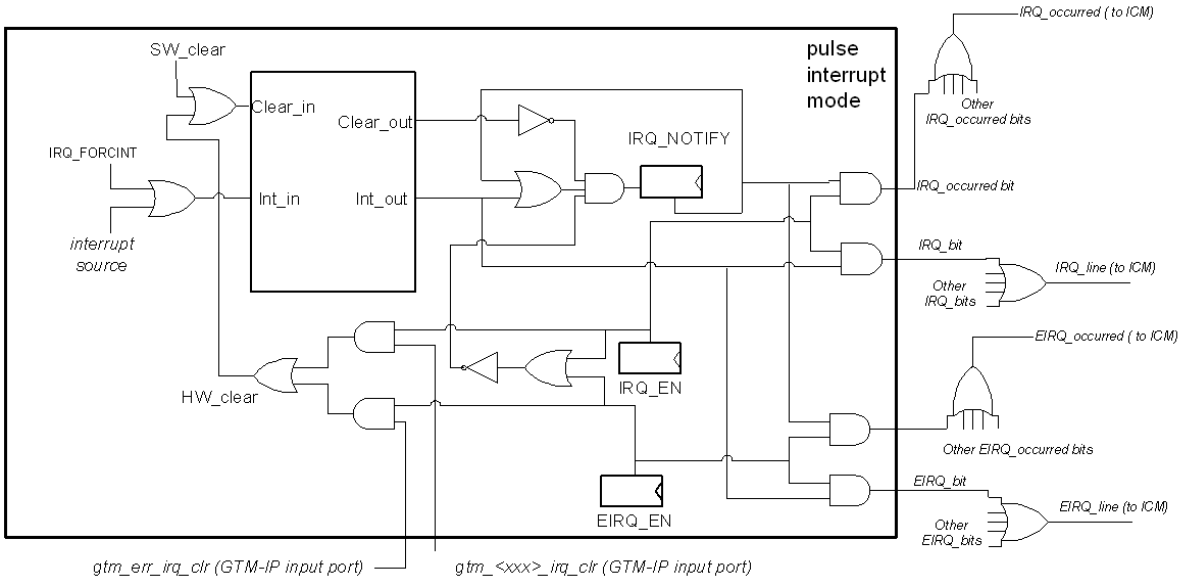


Figure 2-11. Pulse interrupt scheme for modules AEI-bridge, BRC, FIFO, TIM, MCS, DPLL, SPE, CMP

In Pulse Interrupt Mode, each Interrupt Event will generate a pulse on the EIRQ_bit signal if EIRQ_EN is enabled.

As it can be seen from the figure above, the interrupt bit in the IRQ_NOTIFY register is always cleared if EIRQ_EN or IRQ_EN are enabled.

However, if an error interrupt is disabled in the EIRQ_EN register, an occurred error interrupt event is captured in the IRQ_NOTIFY register in order to allow polling for disabled error interrupts by software.

Disabled error interrupts may be cleared by an error interrupt clear event. In Pulse interrupt mode, the signal EIRQ_occurred is always 0.

2.7.3 Pulse-notify interrupt mode

In Pulse-notify Interrupt mode (see Figure 2-12):

- all interrupt events that have occurred are captured in the **IRQ_NOTIFY** register,
- if an interrupt is enabled in the **IRQ_EN** register, each interrupt event that has occurred will generate a pulse on the *IRQ_bit* signal, and
- the *IRQ_occurred* signal will be high if the interrupt is enabled in the **IRQ_EN** register and if the corresponding bit of the **IRQ_NOTIFY** register is set.

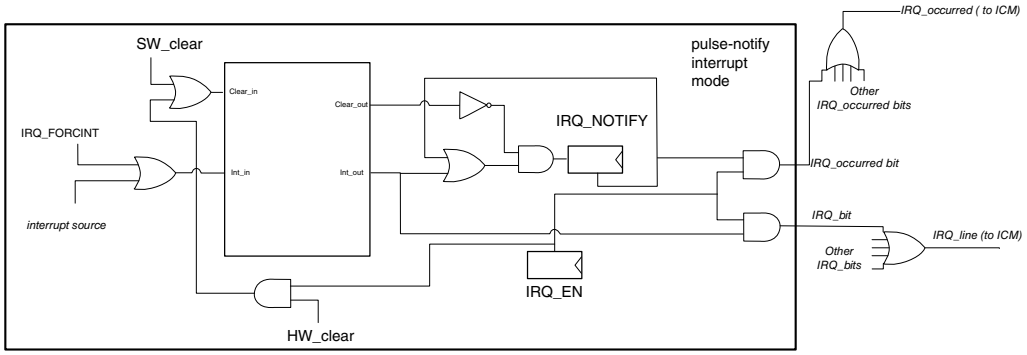


Figure 2-12. Pulse-notify mode

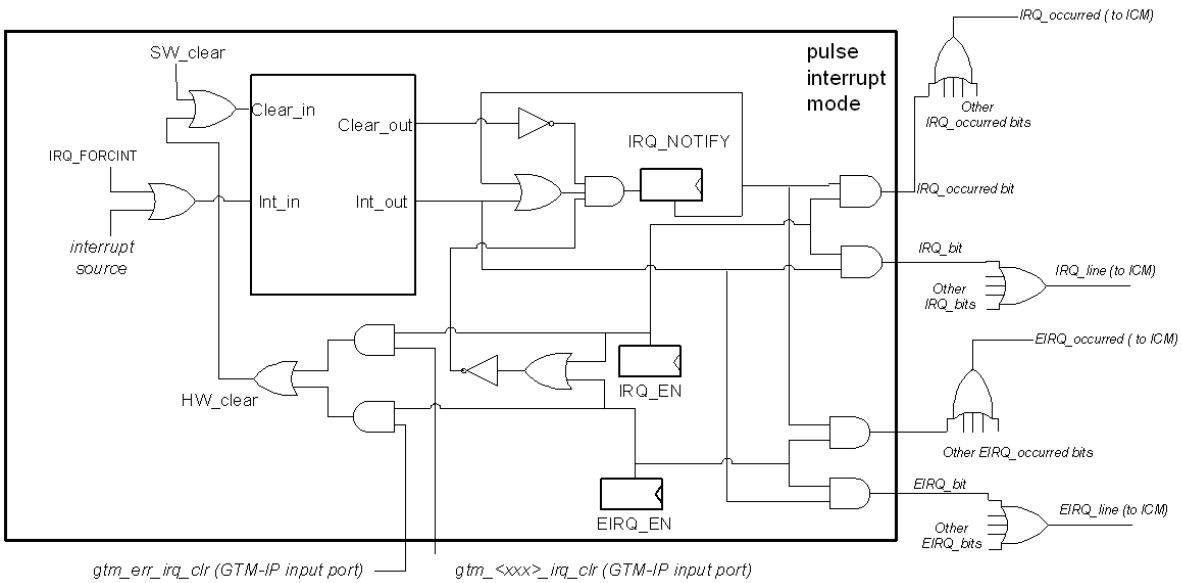


Figure 2-13. Pulse-notify interrupt scheme for modules AEI-bridge, BRC, FIFO, TIM, MCS, DPLL, SPE, CMP

In Pulse-notify Interrupt mode, all error interrupt events are captured in the IRQ_NOTIFY register. If an error interrupt is enabled by the EIRQ_EN register, each error interrupt event will also generate a pulse on the EIRQ_bit signal. The EIRQ_occurred signal will be high if the error interrupt is enabled in the EIRQ_EN register and the corresponding bit of the IRQ_NOTIFY register is set. The Pulse-notify interrupt mode for error interrupts is shown in the figure above.

2.7.4 Single-pulse interrupt mode

In Single-pulse Interrupt Mode (see Figure 2-14):

- An interrupt event is always captured in the **IRQ_NOTIFY** register, independent of the state of the **IRQ_EN** register. However, if an interrupt is enabled in the **IRQ_EN** register, only the first interrupt event that has occurred is forwarded to *IRQ_line* signal. Additional interrupt events that have occurred cannot generate pulses on the *IRQ_line* signal until the corresponding bits in **IRQ_NOTIFY** register are cleared by a clear event.
- The *IRQ_occurred* signal will be high, but only if the **IRQ_EN** and the **IRQ_NOTIFY** register bits are set.

The only exceptions are the ARU and DPLL submodules. In these submodules, the *IRQ_0ccurred* bit of each interrupt is directly connected (without OR-conjunction of neighboring *IRQ_occurred* bits) to the inverter so that further interrupt pulses are suppressed.

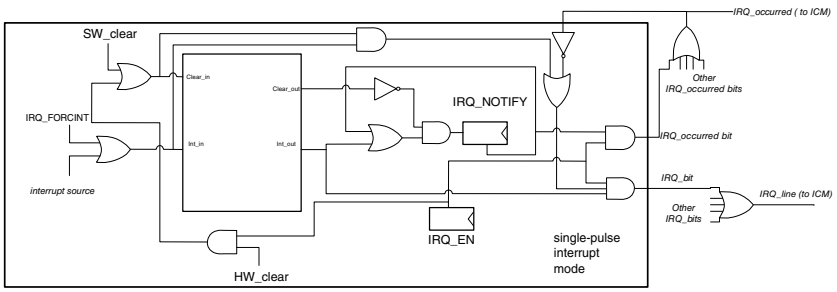


Figure 2-14. Single-pulse interrupt mode.

To avoid unexpected IRQ behavior in single-pulse mode:

- all desired interrupt sources should be enabled by a single write access to the **IRQ_EN** register, and
- all notification bits should be cleared by a single write access to the **IRQ_NOTIFY** register.

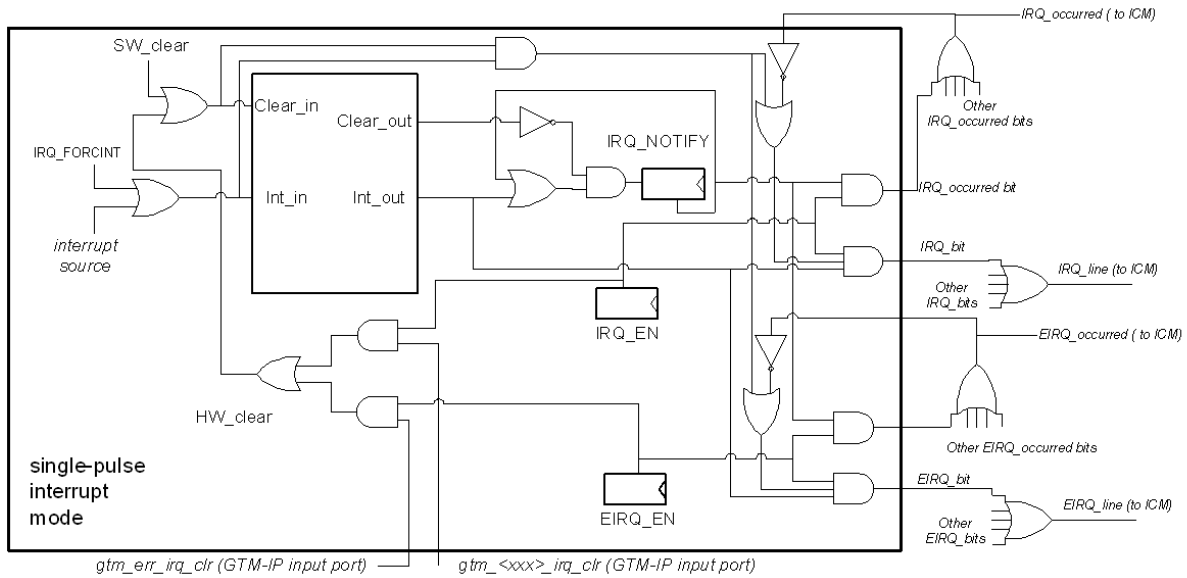


Figure 2-15. Single-pulse interrupt scheme for modules AEI-bridge, BRC, FIFO, TIM, MCS, DPLL, SPE, CMP

2.7.5 GTM Interrupt concentration method

Because of the grouping of interrupts inside the ICM submodule, it can be necessary for the software to access the ICM submodule first to determine the interrupt set that is responsible for an interrupt. A second access to the responsible **IRQ_NOTIFY** register is then necessary to identify the interrupt source, serve it, and then reset the interrupt flag in the **IRQ_NOTIFY**register. The interrupt flags are never reset by an access to the ICM submodule. For a description of the ICM submodule please refer to ICM Overview.

2.8 GTM Software Debugger Support

Status register bits must not be altered by a read access from a software debugger. If the values of status register bits must be altered, the CPU has to explicitly write values '1' to the register bits.

The following table describes the behavior of some special functionality GTM registers with regard to read accesses from the AEI bus interface.

Table 2-5. Register behavior for software debugger accesses

Module	Register	Description
AFD	AFD[i]_CH[x]_BUFFACC	The FIFO read access pointers are not altered on behalf of a debugger read access to these registers.

Table continues on the next page...

Table 2-5. Register behavior for software debugger accesses (continued)

Module	Register	Description
TIM	TIM[i]_CH[x]_GPR0/1	The overflow bit is not altered in case of a debugger read access to these registers.
ATOM	ATOM[i]_CH[x]_SR0/1	In SOMC mode, a read access to these registers by the debugger does not release the channel for a new compare/ match event.

Further on, some important states inside the GTM have to be signalled to the outside world, when reached and should (for example) trigger the software debugger to stop program execution. For this internal state signalling please refer to the *GTM integration guide*.

The GTM provides an external *gtm_halt* signal, which disables the *SYS_CLK* signal for debugging purposes. If the *SYS_CLK* signal is disabled, a connected debugger can use the AEI bus to read any GTM related registers and GTM internal RAMs. In order to enable advanced debugging features (e.g. modifications of register contents in single step mode), the debugger can perform write accesses to all GTM related registers and internal RAMs.

2.9 GTM Programming conventions

To serve different application domains, the GTM is a highly configurable module that has many configuration modes. In principle, the GTM submodules are intended to be configured at system startup to fulfil specific functionalities for the application domain.

For example, a TIM input channel can be used to monitor an application specific external signal, and this signal has to be filtered. Therefore, the configuration of the TIM channel filter mode will be specific to the external signal characteristic. Changing the filter mode during runtime should be avoided because doing so can lead to an unexpected behavior.

In general, the programmer has to be careful when reprogramming configuration registers of the GTM submodules during runtime. To avoid unexpected behavior of the GTM, it is recommended that channels be disabled before reconfiguration takes place .

2.10 Memory Map and Registers

The GTM Architecture registers are described as follows:

GTM memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	GTM Revision Register (GTM_REV)	32	R	See section	2.10.1/53
4	GTM Reset Register (GTM_RST)	32	R/W	0000_0000h	2.10.2/54
8	GTM Control Register (GTM_CTRL)	32	R/W	0000_0001h	2.10.3/54
C	GTM AEI Address Pointer Register (GTM_AEI_ADDR_XPT)	32	R	0000_0000h	2.10.4/56
10	GTM Interrupt Request Notification Register (GTM_IRQ_NOTIFY)	32	R/W	0000_0000h	2.10.5/57
14	GTM Interrupt Request Enable Register (GTM_IRQ_EN)	32	R/W	0000_0000h	2.10.6/58
18	GTM Force Interrupt Request Register (GTM_IRQ_FORCINT)	32	R/W	0000_0000h	2.10.7/59
1C	GTM Interrupt Request Mode Register (GTM_IRQ_MODE)	32	R/W	0000_0000h	2.10.8/60
20	GTM Error Interrupt Request Enable register (GTM_EIRQ_EN)	32	R/W	0000_0000h	2.10.9/61
30	GTM Bridge Mode Register (GTM_BRIDGE_MODE)	32	R/W	See section	2.10.10/62
34	GTM Bridge Pointer 1 Register (GTM_BRIDGE_PTR1)	32	R	See section	2.10.11/64
38	GTM Bridge Pointer 2 Register (GTM_BRIDGE_PTR2)	32	R	0000_0000h	2.10.12/65
40	GTM TIM0 Auxiliary Input Source register (GTM_TIM0_AUX_IN_SRC)	32	R/W	0000_0000h	2.10.13/65
44	GTM TIM1 Auxiliary Input Source register (GTM_TIM1_AUX_IN_SRC)	32	R/W	0000_0000h	2.10.14/66
48	GTM TIM2 Auxiliary Input Source register (GTM_TIM2_AUX_IN_SRC)	32	R/W	0000_0000h	2.10.15/67
4C	GTM TIM3 Auxiliary Input Source register (GTM_TIM3_AUX_IN_SRC)	32	R/W	0000_0000h	2.10.16/69
50	GTM TIM4 Auxiliary Input Source register (GTM_TIM4_AUX_IN_SRC)	32	R/W	0000_0000h	2.10.17/70
54	GTM TIM5 Auxiliary Input Source register (GTM_TIM5_AUX_IN_SRC)	32	R/W	0000_0000h	2.10.18/71

2.10.1 GTM Revision Register (GTM_REV)

NOTE

Reset values are GTM implementation dependent. They can be found in the corresponding 'GTM Integration Module Configuration' document.

Address: 0h base + 0h offset = 0h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	DEV_CODE2				DEV_CODE1				DEV_CODE0				MAJOR				MINOR				NO				STEP							
W																																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:

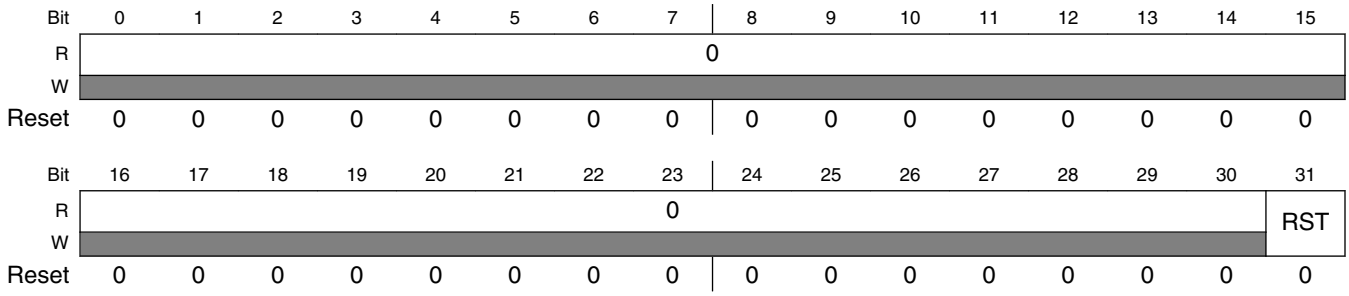
- Reset values are GTM implementation dependent. They can be found in the corresponding 'GTM Integration Module Configuration' document.

GTM_REV field descriptions

Field	Description
0–3 DEV_CODE2	Device encoding digit 2. **NOTE:** The numbers are encoded in BCD. Values "A" - "F" are characters.
4–7 DEV_CODE1	Device encoding digit 1. **NOTE:** The numbers are encoded in BCD. Values "A" - "F" are characters.
8–11 DEV_CODE0	Device encoding digit 0. **NOTE:** The numbers are encoded in BCD. Values "A" - "F" are characters.
12–15 MAJOR	Major version number Define major version number of GTM-IP specification.
16–19 MINOR	Minor version number Define minor version number of GTM-IP specification.
20–23 NO	Delivery number. Define delivery number of GTM-IP specification.
24–31 STEP	Release step. GTM Release step.

2.10.2 GTM Reset Register (GTM_RST)

Address: 0h base + 4h offset = 4h

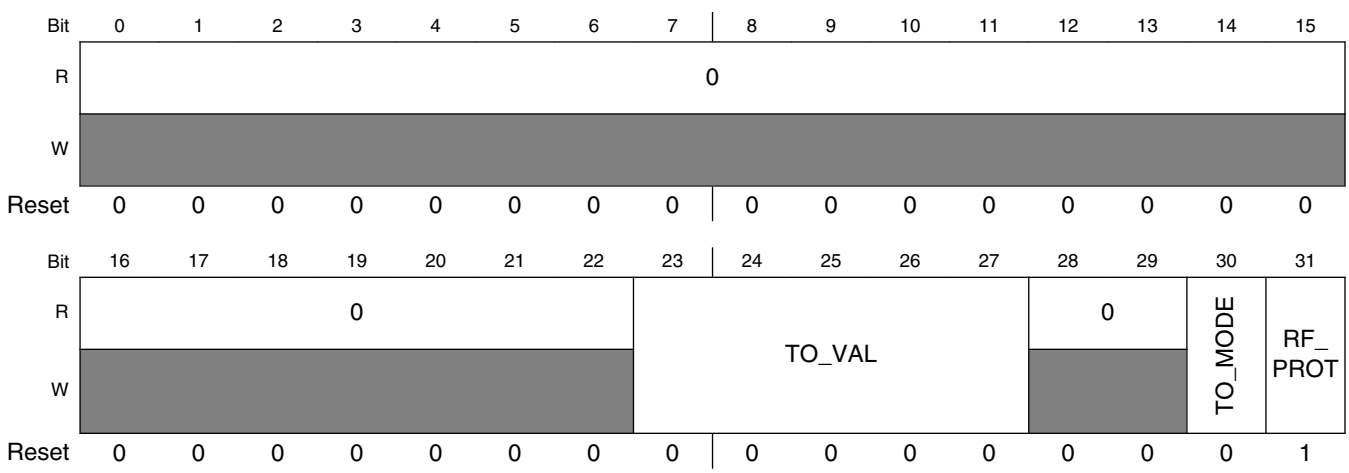


GTM_RST field descriptions

Field	Description
0–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
31 RST	GTM Reset. **NOTE:** This bit is automatically cleared by hardware after it was written. Therefore, the register is always read as zero (0) by the software. **NOTE:** This bit is write protected by bit GTM_CTRL[RF_PROT]. 0 No reset action. 1 Initiate reset action for all submodules.

2.10.3 GTM Control Register (GTM_CTRL)

Address: 0h base + 8h offset = 8h

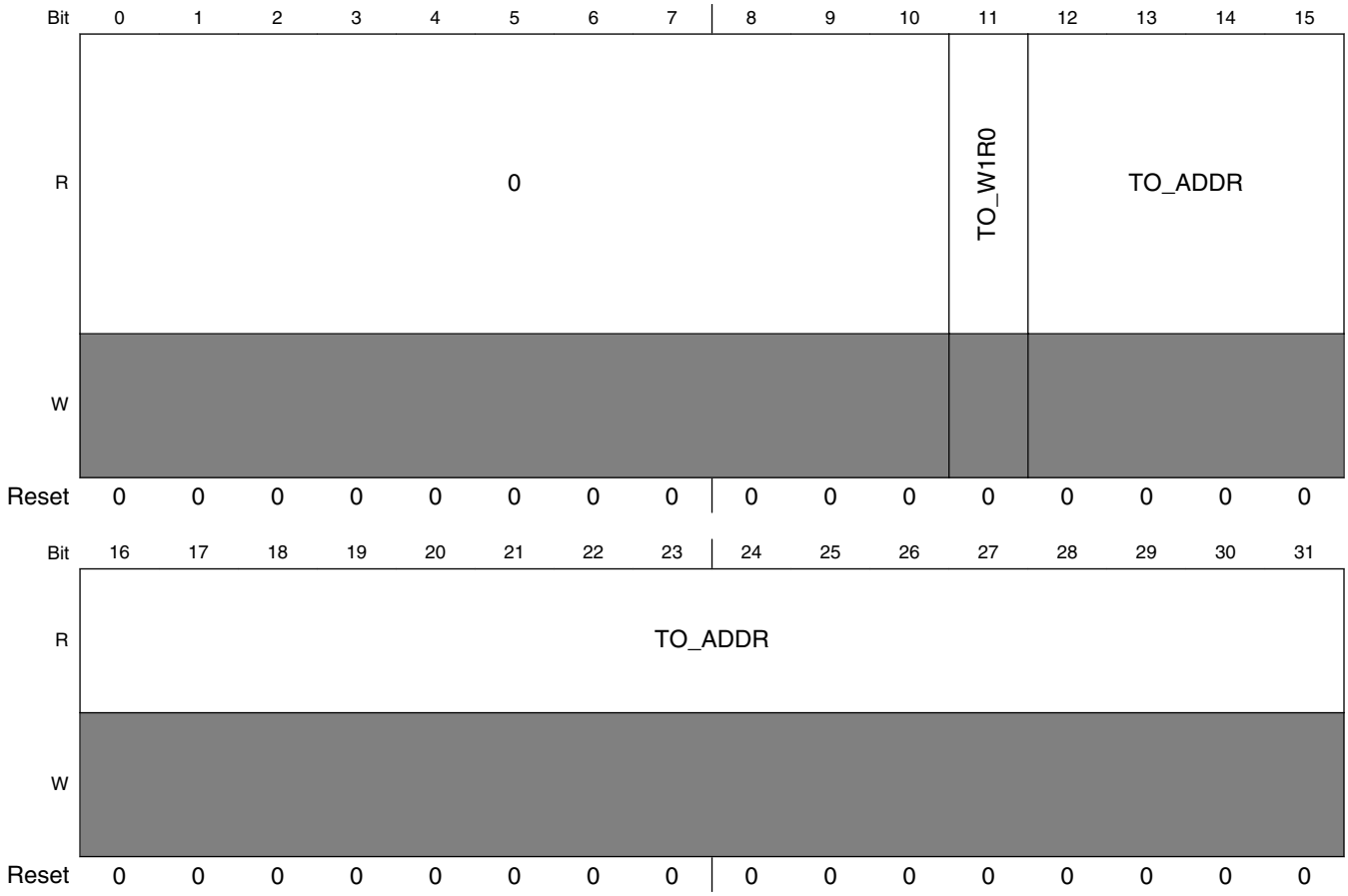


GTM_CTRL field descriptions

Field	Description
0–22 Reserved	Read as zero, should be written as zero. This field is reserved. This read-only field is reserved and always has the value 0.
23–27 TO_VAL	AEI Timeout value. These bits define the number of cycles after which a timeout event occurs. When TO_VAL equals zero (0), the AEI timeout functionality is disabled.
28–29 Reserved	Read as zero, should be written as zero. This field is reserved. This read-only field is reserved and always has the value 0.
30 TO_MODE	AEI Timeout mode. 0 Observe: If timeout_counter=0, the address and rw signal in addition with timeout flag will be stored to the GTM_AEI_ADDR_XPT register. Following timeout_counter=0, accesses will not overwrite the first entry in the aei_addr_timeout register. Clearing the timeout flag/aei_status error_code will reenable the storing of a next faulty access. 1 Abort: In addition to observe mode, the pending access will be aborted by signalling an illegal module access on aei_status and sending ready. In case of a read, deliver as data 0 by serving of next AEI accesses.
31 RF_PROT	RST and FORCINT protection. 0 SW RST (global), SW interrupt FORCINT, and SW RAM reset functionality is enabled. 1 SW RST (global), SW interrupt FORCINT, and SW RAM reset functionality is disabled.

2.10.4 GTM AEI Address Pointer Register (GTM_AEI_ADDR_XPT)

Address: 0h base + Ch offset = Ch

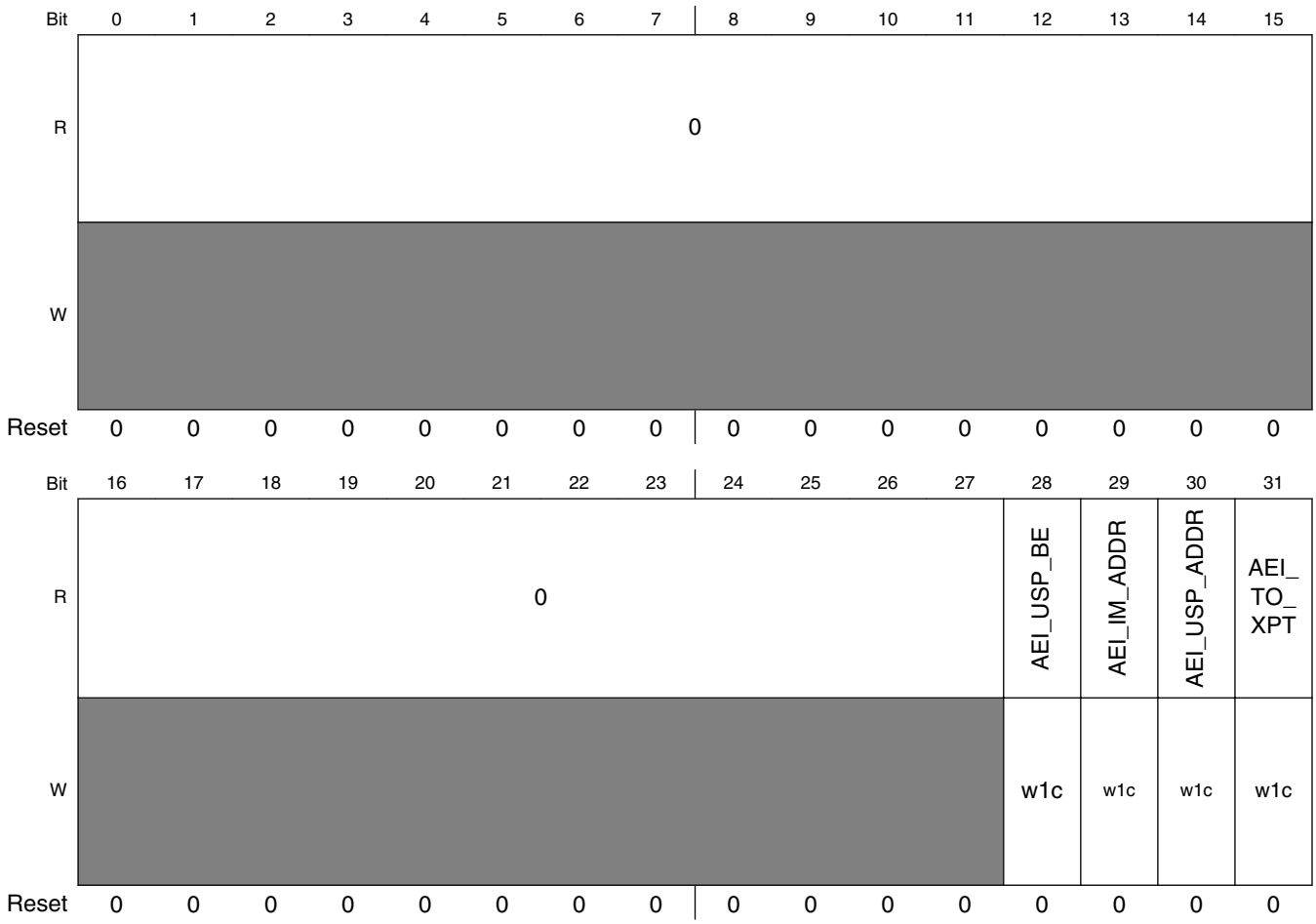


GTM_AEI_ADDR_XPT field descriptions

Field	Description
0–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 TO_W1R0	AEI Timeout Read/Write flag. This bit defines the AEI Read/Write flag for which the AEI timeout event occurred.
12–31 TO_ADDR	AEI Timeout address. This bit field defines the AEI address for which the AEI timeout event occurred.

2.10.5 GTM Interrupt Request Notification Register (GTM_IRQ_NOTIFY)

Address: 0h base + 10h offset = 10h



GTM_IRQ_NOTIFY field descriptions

Field	Description
0–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 AEI_USP_BE	AEI Unsupported byte enable interrupt. This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged. 0 No interrupt occurred. 1 AEI_USP_BE interrupt was raised by the AEI interface.
29 AEI_IM_ADDR	AEI Illegal Module address interrupt. This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
30 AEI_USP_ADDR	AEI Unsupported address interrupt. This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

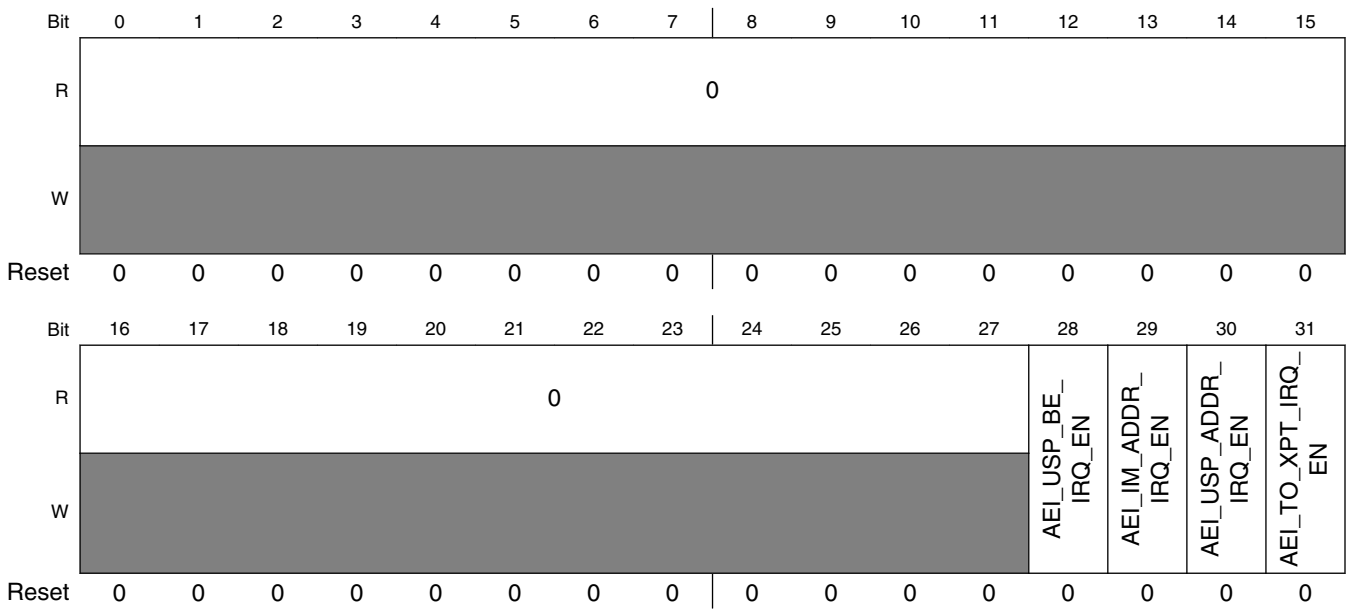
Table continues on the next page...

GTM_IRQ_NOTIFY field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 AEI_USP_ADDR interrupt was raised by the AEI interface.
31 AEI_TO_XPT	AEI Timeout exception occurred. This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged. 0 No interrupt occurred. 1 AEI_TO_XPT interrupt was raised by the AEI Timeout detection unit.

2.10.6 GTM Interrupt Request Enable Register (GTM_IRQ_EN)

Address: 0h base + 14h offset = 14h



GTM_IRQ_EN field descriptions

Field	Description
0–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 AEI_USP_BE_ IRQ_EN	AEI Unsupported Byte Enable Interrupt Enable. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
29 AEI_IM_ADDR_ IRQ_EN	AEI Illegal Module Address Interrupt Enable. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.

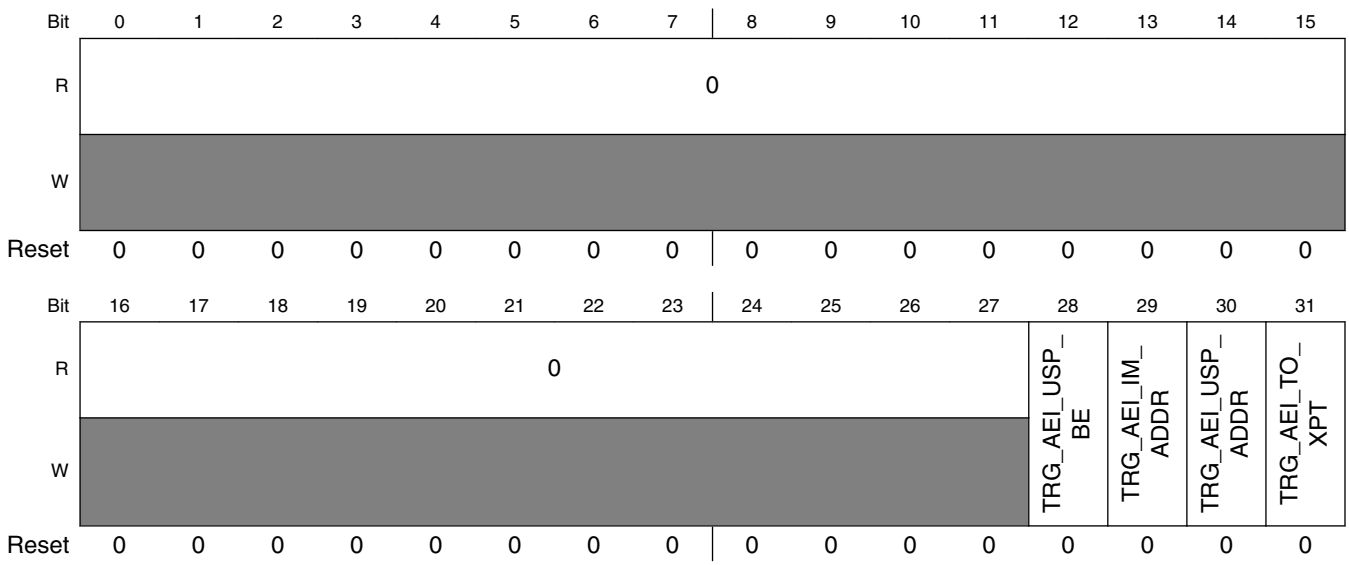
Table continues on the next page...

GTM_IRQ_EN field descriptions (continued)

Field	Description
30 AEI_USP_ ADDR_IRQ_EN	AEI Unsupported Address Interrupt Enable. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
31 AEI_TO_XPT_ IRQ_EN	AEI Timeout Exception Interrupt Enable. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.

2.10.7 GTM Force Interrupt Request Register (GTM_IRQ_FORCINT)

Address: 0h base + 18h offset = 18h



GTM_IRQ_FORCINT field descriptions

Field	Description
0–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 TRG_AEI_USP_ BE	Trigger AEI_USP_BE_IRQ interrupt by software. This bit is cleared automatically after write. This bit is write protected by bit GTM_CTRL[RF_PROT]. 0 No interrupt triggering. 1 Assert AEI_USP_BE_IRQ interrupt for one clock cycle.
29 TRG_AEI_IM_ ADDR	Trigger AEI_IM_ADDR_IRQ interrupt by software. This bit is cleared automatically after write. This bit is write protected by bit GTM_CTRL[RF_PROT].

Table continues on the next page...

GTM_IRQ_FORCINT field descriptions (continued)

Field	Description
	0 No interrupt triggering. 1 Assert AEI_IM_ADDR_IRQ interrupt for one clock cycle.
30 TRG_AEI_USP_ADDR	Trigger AEI_USP_ADDR_IRQ interrupt by software. This bit is cleared automatically after write. This bit is write protected by bit GTM_CTRL[RF_PROT]. 0 No interrupt triggering. 1 Assert AEI_USP_ADDR_IRQ interrupt for one clock cycle.
31 TRG_AEI_TO_XPT	Trigger AEI_TO_XPT_IRQ interrupt by software. This bit is cleared automatically after write. This bit is write protected by bit GTM_CTRL[RF_PROT]. 0 No interrupt triggering. 1 Assert AEI_TO_XPT_IRQ interrupt for one clock cycle.

2.10.8 GTM Interrupt Request Mode Register (GTM_IRQ_MODE)

Address: 0h base + 1Ch offset = 1Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

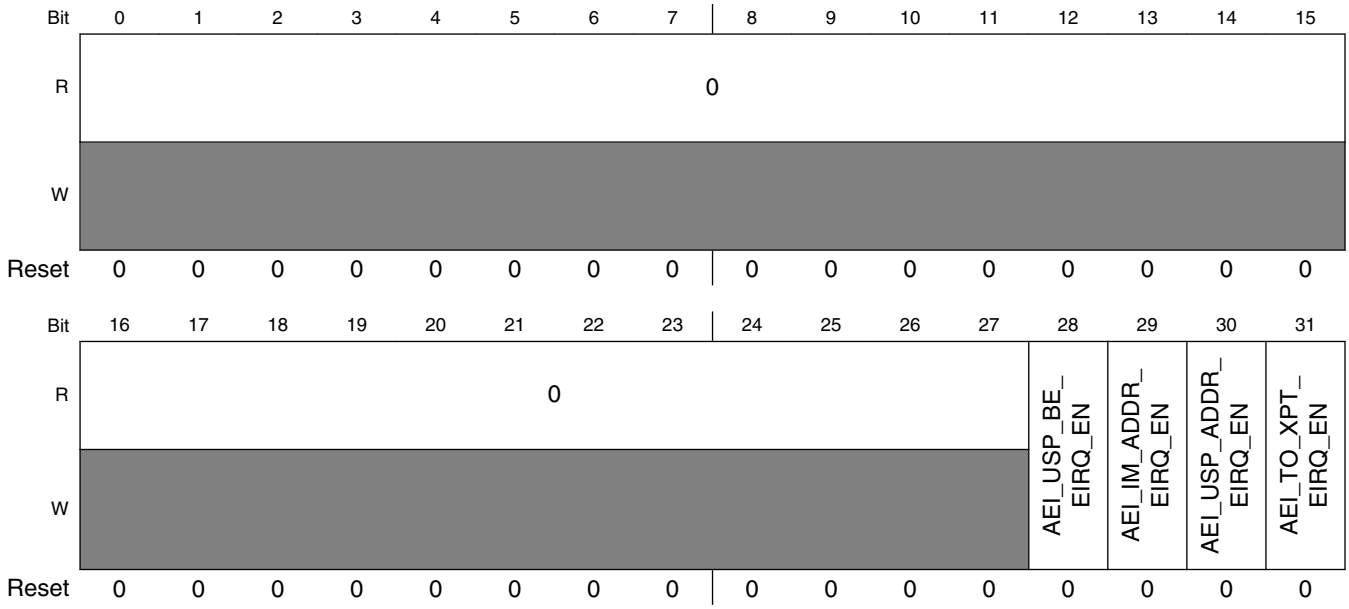
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0														IRQ_MODE	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GTM_IRQ_MODE field descriptions

Field	Description
0–29 Reserved	Read as zero, should be written as zero. This field is reserved. This read-only field is reserved and always has the value 0.
30–31 IRQ_MODE	Interrupt strategy mode selection for the AEI timeout and address monitoring interrupts. **NOTE:** The interrupt modes are described in section 2.5. 00 Level mode. 01 Pulse-Notify mode. 10 Pulse-Notify mode. 11 Single-Pulse mode.

2.10.9 GTM Error Interrupt Request Enable register (GTM_EIRQ_EN)

Address: 0h base + 20h offset = 20h

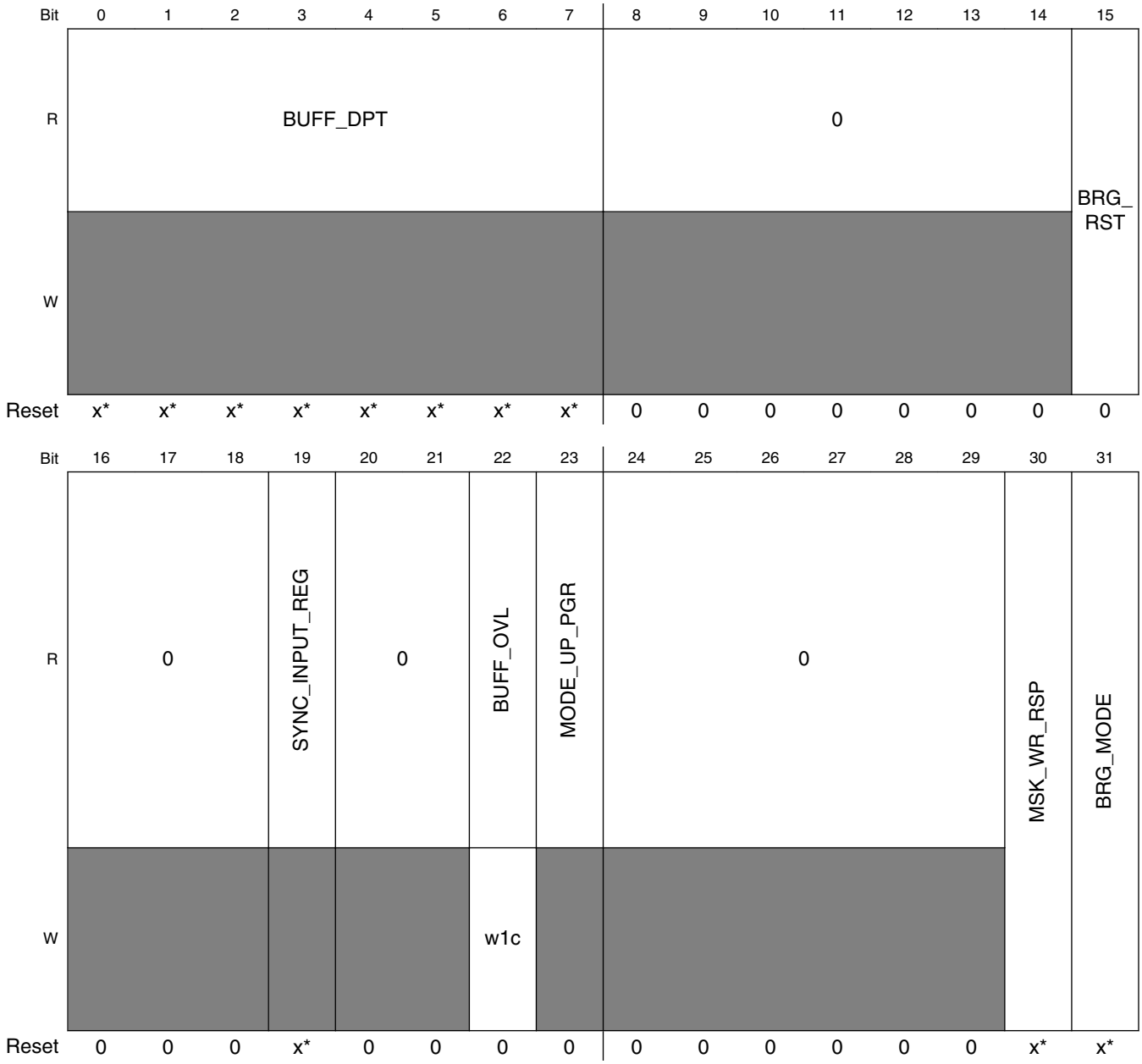


GTM_EIRQ_EN field descriptions

Field	Description
0–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 AEI_USP_BE_EIRQ_EN	AEI_USP_BE_EIRQ error interrupt enable. 0 Disable error interrupt, interrupt is not visible outside GTM-IP. 1 Enable error interrupt, interrupt is visible outside GTM-IP.
29 AEI_IM_ADDR_EIRQ_EN	AEI_IM_ADDR_EIRQ error interrupt enable. 0 Disable error interrupt, interrupt is not visible outside GTM-IP. 1 Enable error interrupt, interrupt is visible outside GTM-IP.
30 AEI_USP_ADDR_EIRQ_EN	AEI_USP_ADDR_EIRQ error interrupt enable. 0 Disable error interrupt, interrupt is not visible outside GTM-IP. 1 Enable error interrupt, interrupt is visible outside GTM-IP.
31 AEI_TO_XPT_EIRQ_EN	AEI_TO_XPT_EIRQ error interrupt enable. 0 Disable error interrupt, interrupt is not visible outside GTM-IP. 1 Enable error interrupt, interrupt is visible outside GTM-IP.

2.10.10 GTM Bridge Mode Register (GTM_BRIDGE_MODE)

Address: 0h base + 30h offset = 30h



* Notes:
- x = Undefined at reset.

GTM_BRIDGE_MODE field descriptions

Field	Description
0–7 BUFF_DPT	Buffer depth of AEI bridge.

Table continues on the next page...

GTM_BRIDGE_MODE field descriptions (continued)

Field	Description
	Signals the buffer depth of the GTM AEI bridge implementation.
8–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 BRG_RST	Bridge software reset. This bit is cleared automatically after write. 0 No bridge reset request. 1 Bridge reset request.
16–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 SYNC_INPUT_ REG	Additional pipelined stage in synchronous bridge mode. 0 No additional pipelined stage implemented. 1 Aadditional pipelined stage implemented. All accesses in synchronous mode will be increased by one clock cycle.
20–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 BUFF_OVL	Buffer overflow register. A buffer overflow can occur while multiple aborts are issued by the external bus or a pipelined instruction is started while FBC = 0 (see GTM_BRIDGE_PTR1 register). 0 No buffer overflow occurred. 1 Buffer overflow occurred.
23 MODE_UP_PGR	Mode update in progress. 0 No update in progress. 1 Update in progress.
24–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 MSK_WR_RSP	Mask write response. 0 Do not mask the write response. 1 Mask write response.
31 BRG_MODE	Defines the operation mode for the AEI bridge. 0 AEI bridge operates in sync_bridge mode. 1 AEI bridge operates in async_bridge mode.

2.10.11 GTM Bridge Pointer 1 Register (GTM_BRIDGE_PTR1)

This register operates on the AEI_CLK domain.

Address: 0h base + 34h offset = 34h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	RSP_TRAN_RDY						FBC						ABT_TRAN_PGR				TRAN_IN_PGR						FIRST_RSP_PTR					NEW_TRAN_PTR				
W																																
Reset	0	0	0	0	0	0	x*	x*	x*	x*	x*	x*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

* Notes:
- x = Undefined at reset.

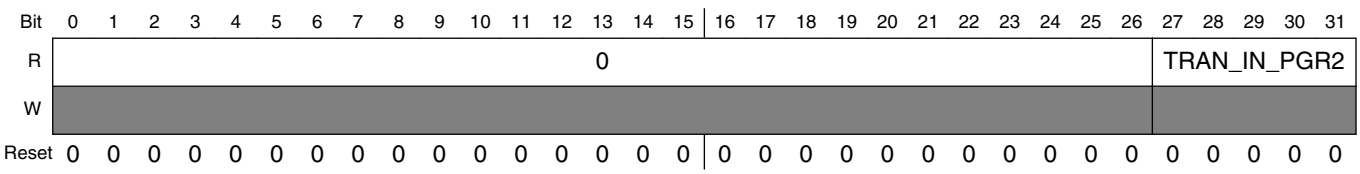
GTM_BRIDGE_PTR1 field descriptions

Field	Description
0–5 RSP_TRAN_RDY	Response transactions ready. Amount of ready response transactions.
6–11 FBC	Free buffer count. Number of free buffer entries. **NOTE:** Initial value depends on the hardware configuration choosen by licensee (see BUFF_DPT in GTM_BRIDGE_MODE register).
12–16 ABT_TRAN_PGR	Aborted transaction in progress pointer.
17–21 TRAN_IN_PGR	Transaction in progress pointer (acquire).
22–26 FIRST_RSP_PTR	First response pointer. Signals the actual value of first response pointer.
27–31 NEW_TRAN_PTR	New transaction pointer. Signals the actual value of the new transaction pointer.

2.10.12 GTM Bridge Pointer 2 Register (GTM_BRIDGE_PTR2)

This register operates on the GTM_CLK domain.

Address: 0h base + 38h offset = 38h

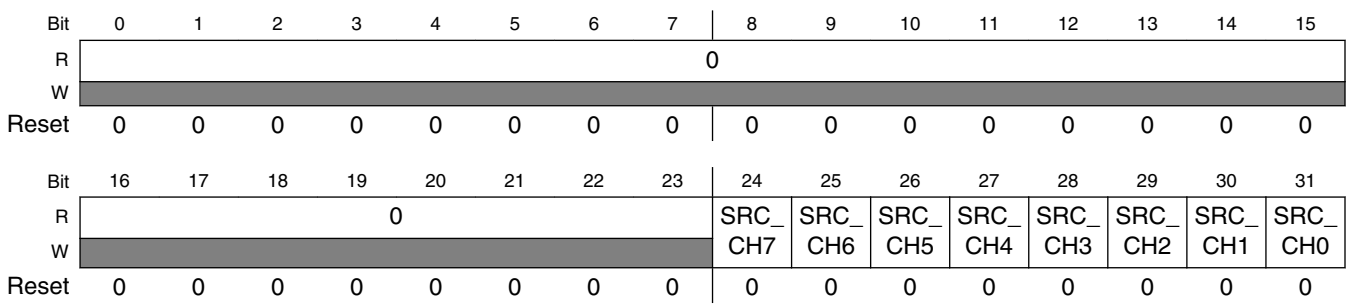


GTM_BRIDGE_PTR2 field descriptions

Field	Description
0–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–31 TRAN_IN_PGR2	Transaction in progress pointer (aquire2).

2.10.13 GTM TIM0 Auxiliary Input Source register (GTM_TIM0_AUX_IN_SRC)

Address: 0h base + 40h offset = 40h



GTM_TIM0_AUX_IN_SRC field descriptions

Field	Description
0–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 SRC_CH7	Defines AUX_IN source of TIM[i] channel 7. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 15) mod 16 and b = (i * 15) % 16. 1 ATOM Output selected ATOM[i] channel 7.
25 SRC_CH6	Defines AUX_IN source of TIM[i] channel 6.

Table continues on the next page...

GTM_TIM0_AUX_IN_SRC field descriptions (continued)

Field	Description
	0 TOM Output selected is TOM[a] channel [b] with a = (i * 14) mod 16 and b = (i * 14) % 16. 1 ATOM Output selected ATOM[i] channel 6.
26 SRC_CH5	Defines AUX_IN source of TIM[i] channel 5. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 13) mod 16 and b = (i * 13) % 16. 1 ATOM Output selected ATOM[i] channel 5.
27 SRC_CH4	Defines AUX_IN source of TIM[i] channel 4. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 12) mod 16 and b = (i * 12) % 16. 1 ATOM Output selected ATOM[i] channel 4.
28 SRC_CH3	Defines AUX_IN source of TIM[i] channel 3. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 11) mod 16 and b = (i * 11) % 16. 1 ATOM Output selected ATOM[i] channel 3.
29 SRC_CH2	Defines AUX_IN source of TIM[i] channel 2. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 10) mod 16 and b = (i * 10) % 16. 1 ATOM Output selected ATOM[i] channel 2.
30 SRC_CH1	Defines AUX_IN source of TIM[i] channel 1. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 9) mod 16 and b = (i * 9) % 16. 1 ATOM Output selected ATOM[i] channel 1.
31 SRC_CH0	Defines AUX_IN source of TIM[i] channel 0. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 8) mod 16 and b = (i * 8) % 16. 1 ATOM Output selected ATOM[i] channel 0.

2.10.14 GTM TIM1 Auxiliary Input Source register (GTM_TIM1_AUX_IN_SRC)

Address: 0h base + 44h offset = 44h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								SRC_ CH7	SRC_ CH6	SRC_ CH5	SRC_ CH4	SRC_ CH3	SRC_ CH2	SRC_ CH1	SRC_ CH0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GTM_TIM1_AUX_IN_SRC field descriptions

Field	Description
0–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

GTM_TIM1_AUX_IN_SRC field descriptions (continued)

Field	Description
24 SRC_CH7	Defines AUX_IN source of TIM[i] channel 7. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 15) mod 16 and b = (i * 15) % 16. 1 ATOM Output selected ATOM[i] channel 7.
25 SRC_CH6	Defines AUX_IN source of TIM[i] channel 6. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 14) mod 16 and b = (i * 14) % 16. 1 ATOM Output selected ATOM[i] channel 6.
26 SRC_CH5	Defines AUX_IN source of TIM[i] channel 5. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 13) mod 16 and b = (i * 13) % 16. 1 ATOM Output selected ATOM[i] channel 5.
27 SRC_CH4	Defines AUX_IN source of TIM[i] channel 4. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 12) mod 16 and b = (i * 12) % 16. 1 ATOM Output selected ATOM[i] channel 4.
28 SRC_CH3	Defines AUX_IN source of TIM[i] channel 3. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 11) mod 16 and b = (i * 11) % 16. 1 ATOM Output selected ATOM[i] channel 3.
29 SRC_CH2	Defines AUX_IN source of TIM[i] channel 2. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 10) mod 16 and b = (i * 10) % 16. 1 ATOM Output selected ATOM[i] channel 2.
30 SRC_CH1	Defines AUX_IN source of TIM[i] channel 1. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 9) mod 16 and b = (i * 9) % 16. 1 ATOM Output selected ATOM[i] channel 1.
31 SRC_CH0	Defines AUX_IN source of TIM[i] channel 0. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 8) mod 16 and b = (i * 8) % 16. 1 ATOM Output selected ATOM[i] channel 0.

2.10.15 GTM TIM2 Auxiliary Input Source register (GTM_TIM2_AUX_IN_SRC)

Address: 0h base + 48h offset = 48h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								SRC_CH7	SRC_CH6	SRC_CH5	SRC_CH4	SRC_CH3	SRC_CH2	SRC_CH1	SRC_CH0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GTM_TIM2_AUX_IN_SRC field descriptions

Field	Description
0–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 SRC_CH7	Defines AUX_IN source of TIM[i] channel 7. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 15) mod 16 and b = (i * 15) % 16. 1 ATOM Output selected ATOM[i] channel 7.
25 SRC_CH6	Defines AUX_IN source of TIM[i] channel 6. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 14) mod 16 and b = (i * 14) % 16. 1 ATOM Output selected ATOM[i] channel 6.
26 SRC_CH5	Defines AUX_IN source of TIM[i] channel 5. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 13) mod 16 and b = (i * 13) % 16. 1 ATOM Output selected ATOM[i] channel 5.
27 SRC_CH4	Defines AUX_IN source of TIM[i] channel 4. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 12) mod 16 and b = (i * 12) % 16. 1 ATOM Output selected ATOM[i] channel 4.
28 SRC_CH3	Defines AUX_IN source of TIM[i] channel 3. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 11) mod 16 and b = (i * 11) % 16. 1 ATOM Output selected ATOM[i] channel 3.
29 SRC_CH2	Defines AUX_IN source of TIM[i] channel 2. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 10) mod 16 and b = (i * 10) % 16. 1 ATOM Output selected ATOM[i] channel 2.
30 SRC_CH1	Defines AUX_IN source of TIM[i] channel 1. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 9) mod 16 and b = (i * 9) % 16. 1 ATOM Output selected ATOM[i] channel 1.
31 SRC_CH0	Defines AUX_IN source of TIM[i] channel 0. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 8) mod 16 and b = (i * 8) % 16. 1 ATOM Output selected ATOM[i] channel 0.

2.10.16 GTM TIM3 Auxiliary Input Source register (GTM_TIM3_AUX_IN_SRC)

Address: 0h base + 4Ch offset = 4Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								SRC_CH7	SRC_CH6	SRC_CH5	SRC_CH4	SRC_CH3	SRC_CH2	SRC_CH1	SRC_CH0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GTM_TIM3_AUX_IN_SRC field descriptions

Field	Description
0–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 SRC_CH7	Defines AUX_IN source of TIM[i] channel 7. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 15) mod 16 and b = (i * 15) % 16. 1 ATOM Output selected ATOM[i] channel 7.
25 SRC_CH6	Defines AUX_IN source of TIM[i] channel 6. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 14) mod 16 and b = (i * 14) % 16. 1 ATOM Output selected ATOM[i] channel 6.
26 SRC_CH5	Defines AUX_IN source of TIM[i] channel 5. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 13) mod 16 and b = (i * 13) % 16. 1 ATOM Output selected ATOM[i] channel 5.
27 SRC_CH4	Defines AUX_IN source of TIM[i] channel 4. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 12) mod 16 and b = (i * 12) % 16. 1 ATOM Output selected ATOM[i] channel 4.
28 SRC_CH3	Defines AUX_IN source of TIM[i] channel 3. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 11) mod 16 and b = (i * 11) % 16. 1 ATOM Output selected ATOM[i] channel 3.
29 SRC_CH2	Defines AUX_IN source of TIM[i] channel 2. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 10) mod 16 and b = (i * 10) % 16. 1 ATOM Output selected ATOM[i] channel 2.
30 SRC_CH1	Defines AUX_IN source of TIM[i] channel 1. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 9) mod 16 and b = (i * 9) % 16. 1 ATOM Output selected ATOM[i] channel 1.
31 SRC_CH0	Defines AUX_IN source of TIM[i] channel 0.

Table continues on the next page...

GTM_TIM3_AUX_IN_SRC field descriptions (continued)

Field	Description
	0 TOM Output selected is TOM[a] channel [b] with a = (i * 8) mod 16 and b = (i * 8) % 16. 1 ATOM Output selected ATOM[i] channel 0.

2.10.17 GTM TIM4 Auxiliary Input Source register (GTM_TIM4_AUX_IN_SRC)

Address: 0h base + 50h offset = 50h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								SRC_CH7	SRC_CH6	SRC_CH5	SRC_CH4	SRC_CH3	SRC_CH2	SRC_CH1	SRC_CH0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GTM_TIM4_AUX_IN_SRC field descriptions

Field	Description
0–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 SRC_CH7	Defines AUX_IN source of TIM[i] channel 7. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 15) mod 16 and b = (i * 15) % 16. 1 ATOM Output selected ATOM[i] channel 7.
25 SRC_CH6	Defines AUX_IN source of TIM[i] channel 6. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 14) mod 16 and b = (i * 14) % 16. 1 ATOM Output selected ATOM[i] channel 6.
26 SRC_CH5	Defines AUX_IN source of TIM[i] channel 5. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 13) mod 16 and b = (i * 13) % 16. 1 ATOM Output selected ATOM[i] channel 5.
27 SRC_CH4	Defines AUX_IN source of TIM[i] channel 4. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 12) mod 16 and b = (i * 12) % 16. 1 ATOM Output selected ATOM[i] channel 4.
28 SRC_CH3	Defines AUX_IN source of TIM[i] channel 3. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 11) mod 16 and b = (i * 11) % 16. 1 ATOM Output selected ATOM[i] channel 3.
29 SRC_CH2	Defines AUX_IN source of TIM[i] channel 2. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 10) mod 16 and b = (i * 10) % 16. 1 ATOM Output selected ATOM[i] channel 2.

Table continues on the next page...

GTM_TIM4_AUX_IN_SRC field descriptions (continued)

Field	Description
30 SRC_CH1	Defines AUX_IN source of TIM[i] channel 1. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 9) mod 16 and b = (i * 9) % 16. 1 ATOM Output selected ATOM[i] channel 1.
31 SRC_CH0	Defines AUX_IN source of TIM[i] channel 0. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 8) mod 16 and b = (i * 8) % 16. 1 ATOM Output selected ATOM[i] channel 0.

2.10.18 GTM TIM5 Auxiliary Input Source register (GTM_TIM5_AUX_IN_SRC)

Address: 0h base + 54h offset = 54h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								SRC_CH7	SRC_CH6	SRC_CH5	SRC_CH4	SRC_CH3	SRC_CH2	SRC_CH1	SRC_CH0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GTM_TIM5_AUX_IN_SRC field descriptions

Field	Description
0–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 SRC_CH7	Defines AUX_IN source of TIM[i] channel 7. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 15) mod 16 and b = (i * 15) % 16. 1 ATOM Output selected ATOM[i] channel 7.
25 SRC_CH6	Defines AUX_IN source of TIM[i] channel 6. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 14) mod 16 and b = (i * 14) % 16. 1 ATOM Output selected ATOM[i] channel 6.
26 SRC_CH5	Defines AUX_IN source of TIM[i] channel 5. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 13) mod 16 and b = (i * 13) % 16. 1 ATOM Output selected ATOM[i] channel 5.
27 SRC_CH4	Defines AUX_IN source of TIM[i] channel 4. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 12) mod 16 and b = (i * 12) % 16. 1 ATOM Output selected ATOM[i] channel 4.
28 SRC_CH3	Defines AUX_IN source of TIM[i] channel 3.

Table continues on the next page...

GTM_TIM5_AUX_IN_SRC field descriptions (continued)

Field	Description
	0 TOM Output selected is TOM[a] channel [b] with a = (i * 11) mod 16 and b = (i * 11) % 16. 1 ATOM Output selected ATOM[i] channel 3.
29 SRC_CH2	Defines AUX_IN source of TIM[i] channel 2. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 10) mod 16 and b = (i * 10) % 16. 1 ATOM Output selected ATOM[i] channel 2.
30 SRC_CH1	Defines AUX_IN source of TIM[i] channel 1. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 9) mod 16 and b = (i * 9) % 16. 1 ATOM Output selected ATOM[i] channel 1.
31 SRC_CH0	Defines AUX_IN source of TIM[i] channel 0. 0 TOM Output selected is TOM[a] channel [b] with a = (i * 8) mod 16 and b = (i * 8) % 16. 1 ATOM Output selected ATOM[i] channel 0.

Chapter 3
Advanced Routing Unit (ARU)

3.1 ARU Overview

The Advanced Routing Unit (ARU) transfers 53-bit wide data (five control bits and two 24-bit values) between several GTM submodules.

Since the ARU submodule concept has already been described in ARU Routing Concept, the following sections only describe additional ARU features, which can be used by software for configuring and debugging ARU-related data streams.

The definition of 'streams' and 'channels' in the ARU submodule context is found in ARU Routing Concept.

3.2 Special Data Sources

In addition to the addresses of the submodule related data sources described in the separate Appendix B, the ARU submodule provides three special data sources that can be used for the configuration of data streams:

- Address 0x1FF: Data source that always provides a 53-bit data word with zeros. A read access to this memory location will never block a requesting data destination.
- Address 0x1FE: Data source that never provides a data word. A read access to this memory location will always block a requesting data destination. This is the reset value of the read registers inside the data destinations.
- Address 0x000: This address is reserved. However, by writing the address 0x000 into the **ARU_ACCESS** register, data can be brought through the **ARU_DATA_H** and **ARU_DATA_L** registers into the system. This allows the CPU to bring software test data into the GTM.

3.3 ARU Access via AEI

In addition to the data transfers between connected submodules, there are two possibilities to access ARU submodule data via the AEI bus: Default ARU Access and Debug Access.

3.3.1 Default ARU Access

A default ARU access uses the **ARU_ACCESS** register, which is used for initiation of a read or write request, and the **ARU_DATA_H** and **ARU_DATA_L** registers, which provide an ARU data word that is to be transferred.

The status of a read or write transfer can be determined by polling the **ARU_ACCESS** register. To avoid data loss (e.g. on access cancellation), the **IRQ_NOTIFY***[acc_ack]* bit is set after a read or write access.

A pending read or write request may also be cancelled by clearing the associated bit. ARU_514. In the case of a read request, the AEI access behaves as a read request initiated by a data destination of a module. The read request is served by the ARU immediately when no other destination has a pending read request. This means, that an AEI read access does not take part in the scheduling of the destination channels and that the time between two consecutive read accesses is not limited by the round trip time.

However, an AEI bus access has the lowest priority behind the ARU scheduler that serves the destination channels. Worst case, the read request is served after one round trip of the ARU submodule, when all destination channels request data at the same time.

In the case of a write request, the ARU submodule provides write data at the address defined by the **ARU_ACCESS[ADDR]** bit field.

To avoid data loss, the reserved ARU address 0x0 has to be used to bring data into the system. Otherwise, in the case where the address specified inside the ADDR bit field is defined for another submodule that acts as a source, the ARU data loss may occur and no deterministic behaviour is guaranteed.

This is because the regular source submodule is not aware that its address is used by the ARU itself to provide data to a destination.

Configuring both read and write request bits results in a read request if the write request bit inside the register isn't already set. The read request bit will be set, but not the write request bit. The following table describes the important cases of the bit 12 (RREQ) and bit 13 (WREQ) of the ARU_ACCESS register.

Table 3-1. Bit 12 (RREQ) and bit 13 (WREQ) of the ARU_ACCESS register

AEI write access: aei_wdata (13:12)	Actual value of ARU_ACCESS(13:12)	Next value of ARU_ACCESS(13:12)	Comment
0:0	0:1	0:0	Cancel read request
0:0	1:0	0:0	Cancel write request
0:1	1:0	1:0	Unchanged register
1:0	0:1	0:1	Unchanged register
1:1	0:0	0:1	Both read and write requests result in a read request
1:1	1:0	1:0	As before, but WREQ bit is already set –> unchanged register

3.3.2 Debug Access

The debug access mode can be used to inspect routed data of configured data streams during runtime.

The ARU submodule provides two independent debug channels, where each is configured by a dedicated ARU read address in the **ARU_DBG_ACCESS0** register and the **ARU_DBG_ACCESS1** register.

The **ARU_DBG_DATA0_H**, **ARU_DBG_DATA0_L**, **ARU_DBG_DATA1_H**, and **ARU_DBG_DATA1_L**registers provide read access to the most recent data word that the corresponding data source sent through the ARU submodule.

Any time when data is transferred through the ARU submodule from a data source to the destination requesting the data, the *ARU_NEW_DATA0_IRQ* and *ARU_NEW_DATA1_IRQ* interrupt signals are asserted.

For advanced debugging purposes, software can write to the **ARU_IRQ_FORCINT** register to initiate assertion of an interrupt signal.

NOTE

The debug mechanism should not be used by the application when a HW-Debugger is used to trace ARU communication. In that case, the debug registers are used by the HW-Debugger to specify the ARU streams that are to be traced.

3.4 ARU Interrupt Signals

The ARU interrupt signals are described in Table 3-2.

Table 3-2. ARU interrupt signals

Signal	Description
ARU_NEW_DATA0_IRQ	Indicates that data is transferred through the ARU using the **ARU_DBG_ACCESS0** debug channel.
ARU_NEW_DATA1_IRQ	Indicates that data is transferred through the ARU using the **ARU_DBG_ACCESS1**debug channel.
ACC_ACK_IRQ	ARU access acknowledge interrupt request.

3.5 Memory Map and Registers

The Advanced Routing Unit (ARU) configuration registers are described as follows:

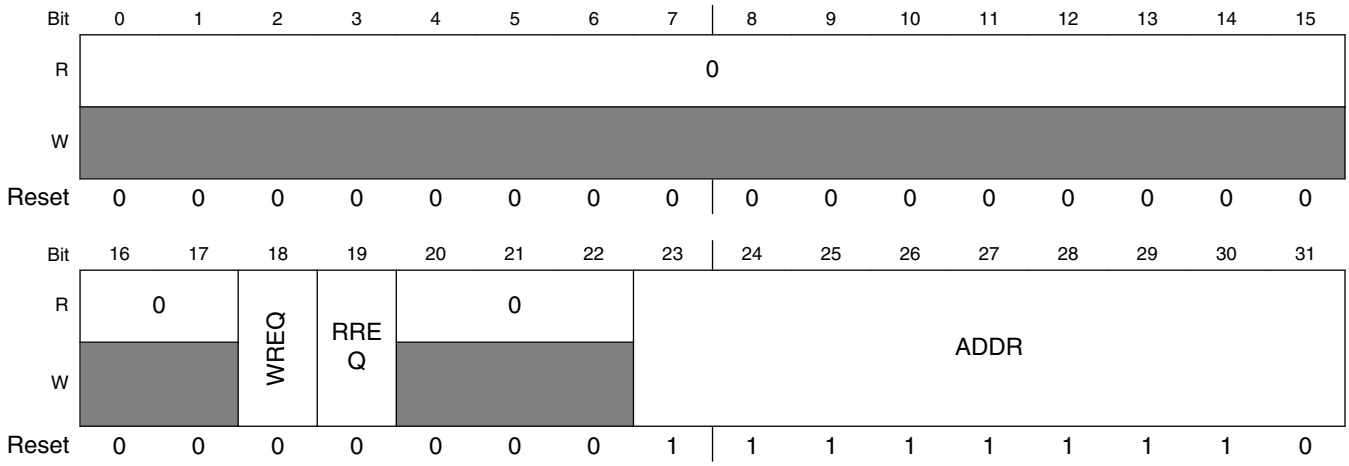
ARU memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	ARU Access Register (ARU_ACCESS)	32	R/W	0000_01FEh	3.5.1/77
4	ARU Data High Register (ARU_DATA_H)	32	R/W	0000_0000h	3.5.2/78
8	ARU Data Low Register (ARU_DATA_L)	32	R/W	0000_0000h	3.5.3/78
C	ARU Debug Access 0 Register (ARU_DBG_ACCESS0)	32	R/W	0000_01FEh	3.5.4/79
10	ARU Debug Data 0 High Register (ARU_DBG_DATA0_H)	32	R	0000_0000h	3.5.5/79
14	ARU Debug Data 0 Low Register (ARU_DBG_DATA0_L)	32	R	0000_0000h	3.5.6/80
18	ARU Debug Access 1 Register (ARU_DBG_ACCESS1)	32	R/W	0000_01FEh	3.5.7/80
1C	ARU Debug Data 1 High (ARU_DBG_DATA1_H)	32	R	0000_0000h	3.5.8/81
20	ARU Debug Data 1 Low Register (ARU_DBG_DATA1_L)	32	R	0000_0000h	3.5.9/81
24	ARU Interrupt Request Notification Register (ARU_IRQ_NOTIFY)	32	R/W	0000_0000h	3.5.10/82
28	ARU Interrupt Request Enable Register (ARU_IRQ_EN)	32	R/W	0000_0000h	3.5.11/83
2C	ARU Force Interrupt Request Register (ARU_IRQ_FORCINT)	32	R	0000_0000h	3.5.12/84
30	ARU Interrupt Request Mode Register (ARU_IRQ_MODE)	32	R/W	0000_0000h	3.5.13/85

3.5.1 ARU Access Register (ARU_ACCESS)

The register ARU_ACCESS can be used either for reading or for writing at the same point in time.

Address: 280h base + 0h offset = 280h



ARU_ACCESS field descriptions

Field	Description
0–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 WREQ	Initiate write request. This bit is cleared automatically after transaction. Moreover, it can be cleared by software to cancel a write request. The WREQ bit is only writable if the RREQ bit is zero, so to switch from WREQ to RREQ, a cancel request has to be performed beforehand. The data is provided at address ADDR. This address has to be programmed as the source address in the destination submodule channel. In worst case, the data is provided after one full ARU round trip. 0 No write request is pending. 1 Mark data in registers ARU_DATA_H and ARU_DATA_L as valid.
19 RREQ	Initiate read request. The RREQ bit is automatically cleared after transaction. Moreover, it can be cleared by software to cancel a read request. the RREQ bit is only writable if the WREQ bit is zero, so to switch from RREQ to WREQ, a cancel request has to be performed beforehand. when the RREQ bit is set by CPU, the ARU read request on address ADDR is served immediately when no other destination has a read request. In a worst case scenario, the read request is served after one round trip of the ARU, but this is only the case when every destination channel issues a read request at consecutive points in time. 0 No read request is pending. 1 Set read request to source channel addressed by ADDR.

Table continues on the next page...

ARU_ACCESS field descriptions (continued)

Field	Description
20–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–31 ADDR	ARU address. The ADDR bitfield defines the ARU address that is used for transferring data. For an ARU write request, the preferred address 0x0 has to be used. A write request to the address 0x1FF (always full address) or 0x1FE (always empty address) is ignored and doesn't have any effect. The ADDR bitfield is only writable if both of the RREQ and WREQ bits are zero.

3.5.2 ARU Data High Register (ARU_DATA_H)

Address: 280h base + 4h offset = 284h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0			DATA																												
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ARU_DATA_H field descriptions

Field	Description
0–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–31 DATA	Upper ARU data word. Transfer upper ARU data word addressed by ADDR. DATA bit[24:52] of an ARU word is mapped to DATA bit[0:28] of this register.

3.5.3 ARU Data Low Register (ARU_DATA_L)

Address: 280h base + 8h offset = 288h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0			DATA																												
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ARU_DATA_L field descriptions

Field	Description
0–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ARU_DATA_L field descriptions (continued)

Field	Description
3–31 DATA	Lower ARU data word. DATA holds the lower ARU data word that is addressed by ADDR. When data is read by the CPU, Data bit[0:23] of an ARU word is mapped to DATA bit[0:23] of this register and DATA bit[48:52] of an ARU word is mapped to DATA bit[24:28] of this register. When CPU writes data into the ARU, bit[24:28] of this register is not transferred to bit[48:52] of the ARU word. Only bit[0:23] is written to bit[0:23] of the ARU word

3.5.4 ARU Debug Access 0 Register (ARU_DBG_ACCESS0)

Address: 280h base + Ch offset = 28Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																ADDR															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0

ARU_DBG_ACCESS0 field descriptions

Field	Description
0–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–31 ADDR	ARU debugging address. Defines address of ARU debug channel 0.

3.5.5 ARU Debug Data 0 High Register (ARU_DBG_DATA0_H)

Address: 280h base + 10h offset = 290h

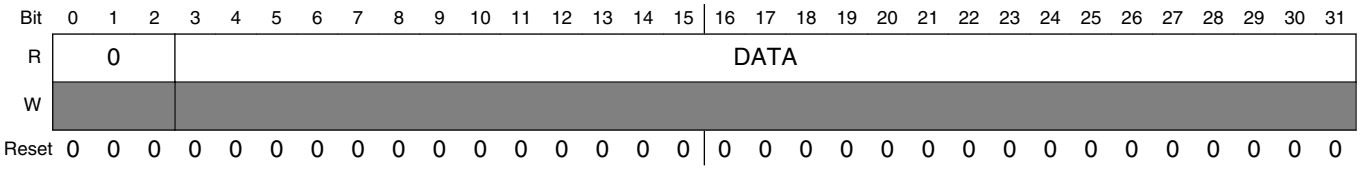
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0			DATA																												
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ARU_DBG_DATA0_H field descriptions

Field	Description
0–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–31 DATA	Upper debug data word. Transfer upper ARU data word addressed by register DBG_ACCESS0. DATA bit[24:52] of an ARU word are mapped to DATA bit[0:28] of this register. The interrupt ARU_NEW_DATA0_IRQ is raised if a new data word is available.

3.5.6 ARU Debug Data 0 Low Register (ARU_DBG_DATA0_L)

Address: 280h base + 14h offset = 294h

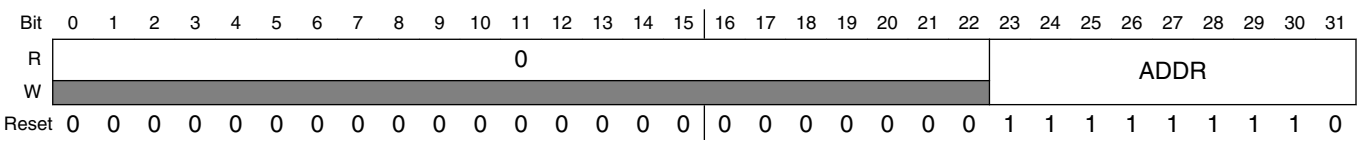


ARU_DBG_DATA0_L field descriptions

Field	Description
0–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–31 DATA	Lower debug data word. Transfer lower ARU data word addressed by register DBG_ACCESS0. DATA bit[0:23] of an ARU word is mapped to the DATA bit[0:23] of this register and DATA bit[48:52] of an ARU word is mapped to DATA bit[24:28] of this register. The interrupt request ARU_NEW_DATA0_IRQ is raised when a new data word is available.

3.5.7 ARU Debug Access 1 Register (ARU_DBG_ACCESS1)

Address: 280h base + 18h offset = 298h

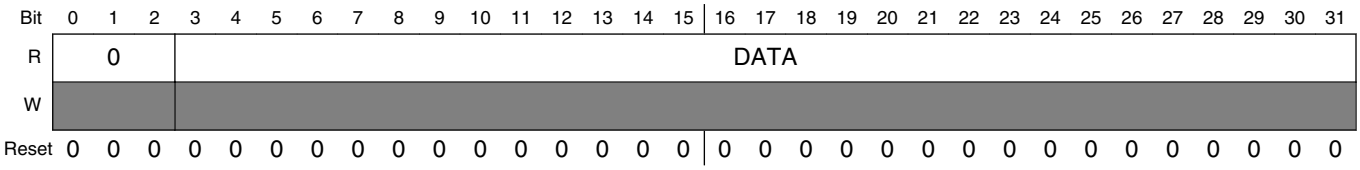


ARU_DBG_ACCESS1 field descriptions

Field	Description
0–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–31 ADDR	ARU debugging address. Defines address of ARU debug channel 1.

3.5.8 ARU Debug Data 1 High (ARU_DBG_DATA1_H)

Address: 280h base + 1Ch offset = 29Ch

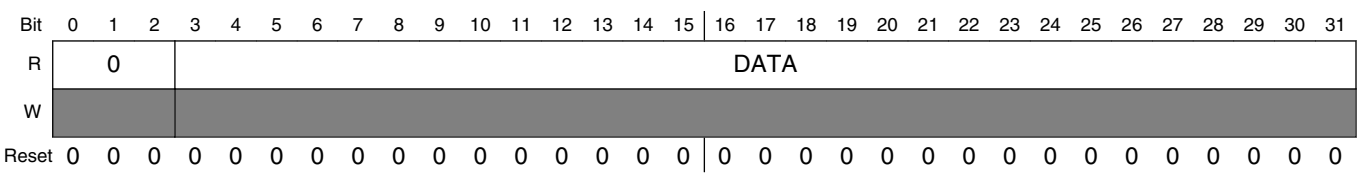


ARU_DBG_DATA1_H field descriptions

Field	Description
0–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–31 DATA	Upper debug data word Transfer upper ARU data word addressed by register DBG_ACCESS1. The DATA bit[24:52] of an ARU word are mapped to the DATA bit[0:28] of this register. The interrupt ARU_NEW_DATA1_IRQ is raised if a new data word is available.

3.5.9 ARU Debug Data 1 Low Register (ARU_DBG_DATA1_L)

Address: 280h base + 20h offset = 2A0h

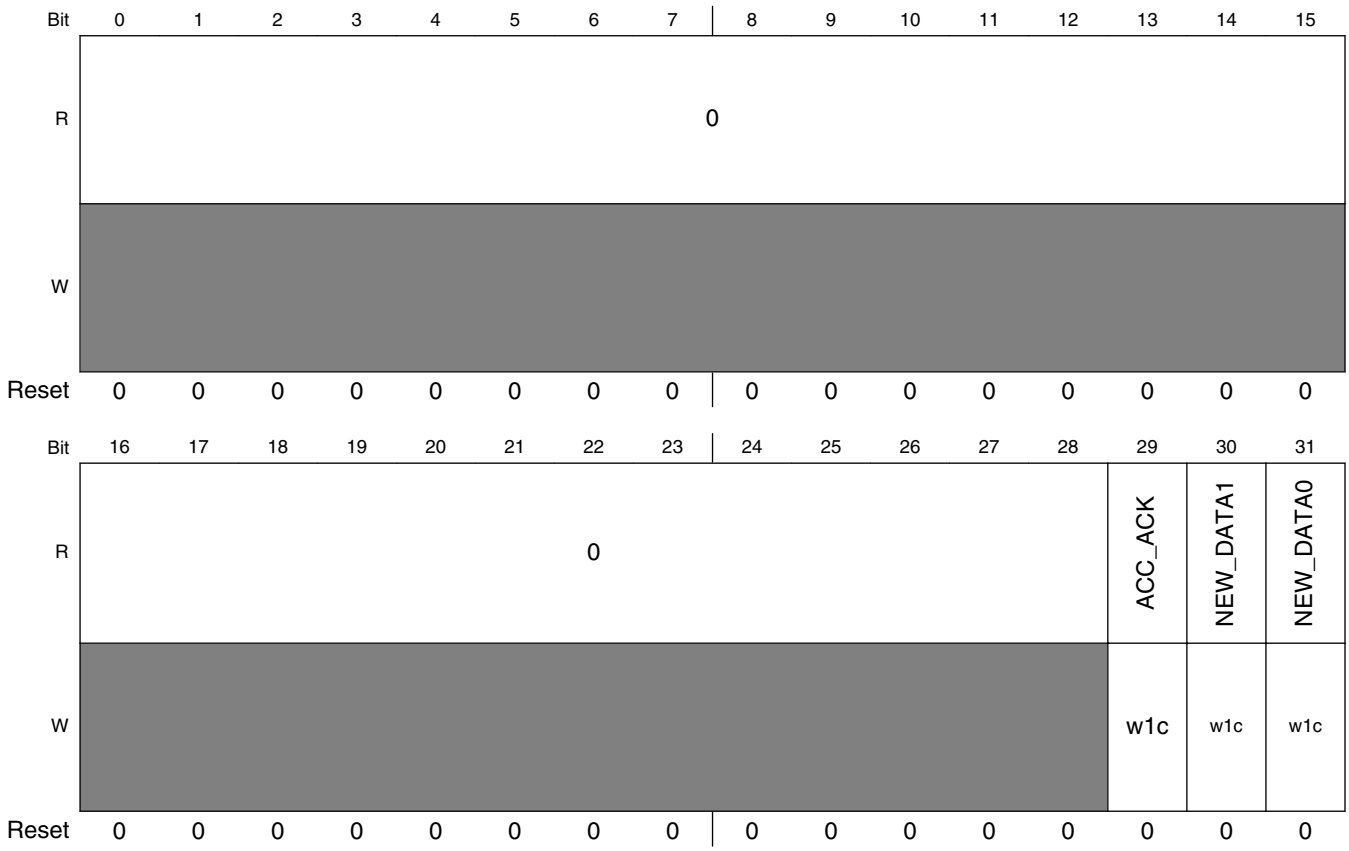


ARU_DBG_DATA1_L field descriptions

Field	Description
0–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–31 DATA	Lower debug data word. Transfer lower ARU data word addressed by register DBG_ACCESS1. DATA bit[0:23] of an ARU word is mapped to DATA bit[0:23] of this register and DATA bit[48:52] of an ARU word is mapped to DATA bit[24:28] of this register. The interrupt request ARU_NEW_DATA1_IRQ is raised when a new data word is available.

3.5.10 ARU Interrupt Request Notification Register (ARU_IRQ_NOTIFY)

Address: 280h base + 24h offset = 2A4h

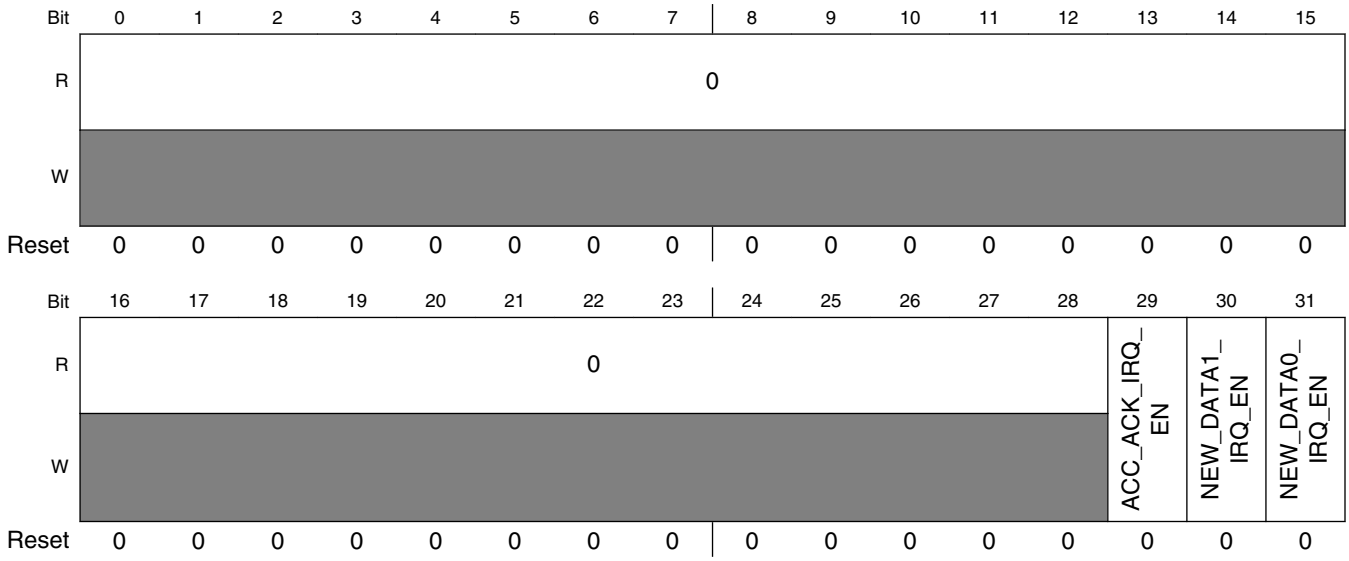


ARU_IRQ_NOTIFY field descriptions

Field	Description
0–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 ACC_ACK	On read access data valid, AEI to ARU access is complete. This bit is cleared by a CPU write access of value '1'. A read access leaves the bit unchanged.
30 NEW_DATA1	Data was transferred for address ARU_DBG_ACCESS1. This bit is cleared by a CPU write access of value '1'. A read access leaves the bit unchanged. 0 No interrupt occurred. 1 ARU_NEW_DATA1_IRQ interrupt was raised by the ARU.
31 NEW_DATA0	Data was transferred for address ARU_DBG_ACCESS0. This bit is cleared by a CPU write access of value '1'. A read access leaves the bit unchanged. 0 No interrupt occurred. 1 ARU_NEW_DATA0_IRQ interrupt was raised by the ARU.

3.5.11 ARU Interrupt Request Enable Register (ARU_IRQ_EN)

Address: 280h base + 28h offset = 2A8h

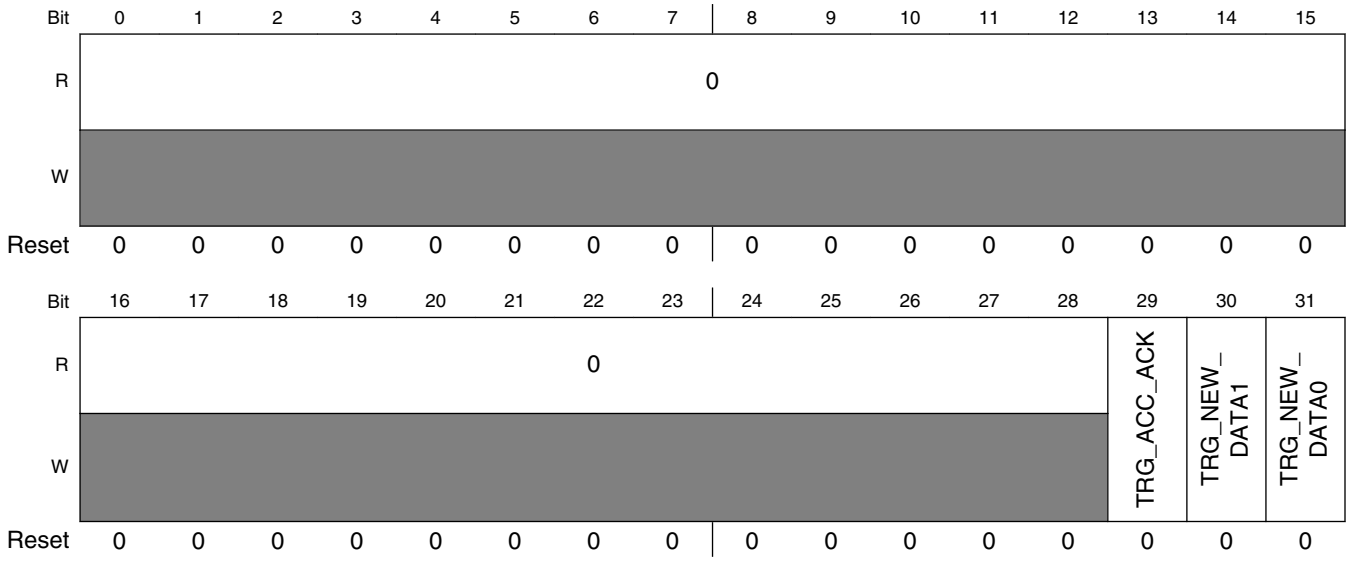


ARU_IRQ_EN field descriptions

Field	Description
0–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 ACC_ACK_IRQ_EN	ACC_ACK_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
30 NEW_DATA1_IRQ_EN	ARU_NEW_DATA1_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM
31 NEW_DATA0_IRQ_EN	ARU_NEW_DATA0_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.

3.5.12 ARU Force Interrupt Request Register (ARU_IRQ_FORCINT)

Address: 280h base + 2Ch offset = 2ACh



ARU_IRQ_FORCINT field descriptions

Field	Description
0–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 TRG_ACC_ACK	Trigger ACC_ACK interrupt. This bit is automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding bit in ARU_IRQ_NOTIFY register.
30 TRG_NEW_ DATA1	Trigger new data 1 interrupt. This bit is automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding bit in ARU_IRQ_NOTIFY register.
31 TRG_NEW_ DATA0	Trigger new data 0 interrupt. This bit is automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding bit in ARU_IRQ_NOTIFY register.

3.5.13 ARU Interrupt Request Mode Register (ARU_IRQ_MODE)

Address: 280h base + 30h offset = 2B0h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0														IRQ_MODE	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ARU_IRQ_MODE field descriptions

Field	Description
0–29 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
30–31 IRQ_MODE	IRQ mode select. The interrupt modes are described in section 2.5. 00 Level mode 01 Pulse mode 10 Pulse-Notify mode 11 Single-Pulse mode

Chapter 4
Broadcast (BRC) Module

4.1 BRC Overview

Each write address, for the submodule channels of the GTM that are able to write to the ARU submodule, can only be read by a single destination submodule. Exceptions are data sources that do not invalidate their data after the data is read by a destination, e.g. DPLL submodule.

For write addresses of submodule channels that must be read by multiple destinations, the broadcast (BRC) submodule has 12 input channels and 22 output channels that facilitate duplicate data streams, multiple times.

For an incoming data stream, the BRC submodule can clone the data and map it to any mix of zero through 22 output channels (e.g. output channels 3 through 12, output channels 4,15,19). When mapped to zero output channels, no channel is read.

The BRC submodule can delete an incoming data stream by setting the ***BRC_SRC_n_DEST****[EN_TRASHBIN]* bit.

4.2 BRC Configuration

The BRC input channels can read arbitrary ARU address locations and the BRC output channels provide broadcast data to fixed ARU write address locations. The associated write addresses for the BRC submodule are fixed (please see TBD).

The read address for each input channel is defined by the corresponding **BRC_SRC_n_ADDR** (n: 0..11) register. The mapping of an input channel to several output channels is defined by setting the appropriate bits in the **BRC_SRC_n_DEST** (x: 0..21) register. The address of each output channel is defined in TBD.

If no output channel bit is set in a **BRC_SRC_n_DEST** register, no data is provided to the corresponding ARU write address location from the defined read input specified by the **BRC_SRC_n_ADDR** register. As a result, the channel does not broadcast any data and it is disabled (reset state).

The **BRC_SRC_n_DEST[EN_TRASHBIN]** bit may be set to delete an incoming data stream. As a result, the data of an input channel defined by the **BRC_SRC_n_ADDR** register is deleted by the BRC submodule and it is not routed to any destination submodule. Also, the output channels defined in the **BRC_SRC_n_DEST** register are ignored and register bits 0 through 21 are set to zero.

The BRC submodule can work in two independent operation modes:

- *Data Consistency Mode* (DCM) guarantees data consistency since a BRC channel requests only new data from a source when all BRC destination channels have consumed the old data.
- *Maximal Throughput Mode* (MTM) where a BRC channel always requests data from a source and distributes it to the destinations regardless of whether all destinations have already consumed the old data or not. A destination that is still consuming old data will not accept new data that is presented to it. So, the newest available data is routed through the GTM, but data consistency is not guaranteed since some destination channels might still be processing old data while other destination channels accept new data and begin processing it. When this is the case, the *BRC_DID_IRQn* data inconsistency detected interrupt is asserted while the channel continues to operate. Also, a read channel is blocked from reading data twice.

The operation mode can be configured by writing to the **BRC_SRC_n_ADDR[BRC_MODE]** bit field. To avoid invalid configurations of the **BRC_SRC_n_DEST** registers, the BRC submodule implements a plausibility check for the configurations. If software assigns an already used output channel to a second input channel, the BRC submodule performs an auto correction of the last **BRC_SRC_n_DEST** register that was configured and it asserts the *BRC_DEST_ERR* interrupt.

As an example of how the auto correction mechanism works, assume the configuration of the 22 lower significant bits for the **BRC_SRC_n_DEST** registers to be as shown in Table 4-1.

Table 4-1. Example configurations

BRC_SRC_0_DEST:	00 0000 0000 1000 1000 0000 (binary)
BRC_SRC_1_DEST:	00 0000 0000 0100 0000 0100 (binary)
BRC_SRC_2_DEST:	00 0000 0000 0001 0100 0010 (binary)
BRC_SRC_3_DEST:	00 0000 0000 0010 0001 1001 (binary)

If software overwrites the following value for the BRC_SRC_2_DEST register:

BRC_SRC_2_DEST: 00 0000 0000 1001 0010 0010, where changed bits are underlined, the BRC asserts a BRC_DEST_ERR interrupt because bit 11 is already assigned in the **BRC_SRC_0_DEST** register. The auto correction mechanism forces bit 11 to be cleared. The modifications of bits 5 and 6 are accepted because there is no violation with previous configurations. So the result of the above write access results in the following modified register configuration:

BRC_SRC_2_DEST: 00 0000 0000 0001 0010 0010.

For debug purposes, the BRC_DEST_ERR interrupt can also be asserted by writing to the **BRC_IRQ_FORCINT** register. However, the interrupt has to be enabled in order for it to be visible external to the GTM.

4.3 BRC Interrupt Signals

BRC interrupt signals are shown in Table 4-2.

Table 4-2. BRC interrupts

Signal	Description
BRC_DEST_ERR_IRQ	Indicates configuration errors for BRC submodule
BRC_DID_IRQn	Data inconsistency occurred in MTM mode (n:0..11)

4.4 Memory Map and Registers

The Broadcast (BRC) module configuration registers are described as follows:

BRC memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	BRC Source n Address Register, n[0:11] (BRC_SRC0_ADDR)	32	R/W	0000_01FEh	4.4.1/91
4	BRC Source n to Destination Register, n[0:11] (BRC_SRC0_DEST)	32	R/W	0000_0000h	4.4.2/92
8	BRC Source n Address Register, n[0:11] (BRC_SRC1_ADDR)	32	R/W	0000_01FEh	4.4.1/91
C	BRC Source n to Destination Register, n[0:11] (BRC_SRC1_DEST)	32	R/W	0000_0000h	4.4.2/92

Table continues on the next page...

BRC memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
10	BRC Source n Address Register, n[0:11] (BRC_SRC2_ADDR)	32	R/W	0000_01FEh	4.4.1/91
14	BRC Source n to Destination Register, n[0:11] (BRC_SRC2_DEST)	32	R/W	0000_0000h	4.4.2/92
18	BRC Source n Address Register, n[0:11] (BRC_SRC3_ADDR)	32	R/W	0000_01FEh	4.4.1/91
1C	BRC Source n to Destination Register, n[0:11] (BRC_SRC3_DEST)	32	R/W	0000_0000h	4.4.2/92
20	BRC Source n Address Register, n[0:11] (BRC_SRC4_ADDR)	32	R/W	0000_01FEh	4.4.1/91
24	BRC Source n to Destination Register, n[0:11] (BRC_SRC4_DEST)	32	R/W	0000_0000h	4.4.2/92
28	BRC Source n Address Register, n[0:11] (BRC_SRC5_ADDR)	32	R/W	0000_01FEh	4.4.1/91
2C	BRC Source n to Destination Register, n[0:11] (BRC_SRC5_DEST)	32	R/W	0000_0000h	4.4.2/92
30	BRC Source n Address Register, n[0:11] (BRC_SRC6_ADDR)	32	R/W	0000_01FEh	4.4.1/91
34	BRC Source n to Destination Register, n[0:11] (BRC_SRC6_DEST)	32	R/W	0000_0000h	4.4.2/92
38	BRC Source n Address Register, n[0:11] (BRC_SRC7_ADDR)	32	R/W	0000_01FEh	4.4.1/91
3C	BRC Source n to Destination Register, n[0:11] (BRC_SRC7_DEST)	32	R/W	0000_0000h	4.4.2/92
40	BRC Source n Address Register, n[0:11] (BRC_SRC8_ADDR)	32	R/W	0000_01FEh	4.4.1/91
44	BRC Source n to Destination Register, n[0:11] (BRC_SRC8_DEST)	32	R/W	0000_0000h	4.4.2/92
48	BRC Source n Address Register, n[0:11] (BRC_SRC9_ADDR)	32	R/W	0000_01FEh	4.4.1/91
4C	BRC Source n to Destination Register, n[0:11] (BRC_SRC9_DEST)	32	R/W	0000_0000h	4.4.2/92
50	BRC Source n Address Register, n[0:11] (BRC_SRC10_ADDR)	32	R/W	0000_01FEh	4.4.1/91
54	BRC Source n to Destination Register, n[0:11] (BRC_SRC10_DEST)	32	R/W	0000_0000h	4.4.2/92
58	BRC Source n Address Register, n[0:11] (BRC_SRC11_ADDR)	32	R/W	0000_01FEh	4.4.1/91
5C	BRC Source n to Destination Register, n[0:11] (BRC_SRC11_DEST)	32	R/W	0000_0000h	4.4.2/92
60	BRC Interrupt Request Notification Register (BRC_IRQ_NOTIFY)	32	R/W	0000_0000h	4.4.3/94
64	BRC Interrupt Request Enable Register (BRC_IRQ_EN)	32	R/W	0000_0000h	4.4.4/96
68	BRC Force Interrupt Request Register (BRC_IRQ_FORCINT)	32	R	0000_0000h	4.4.5/98

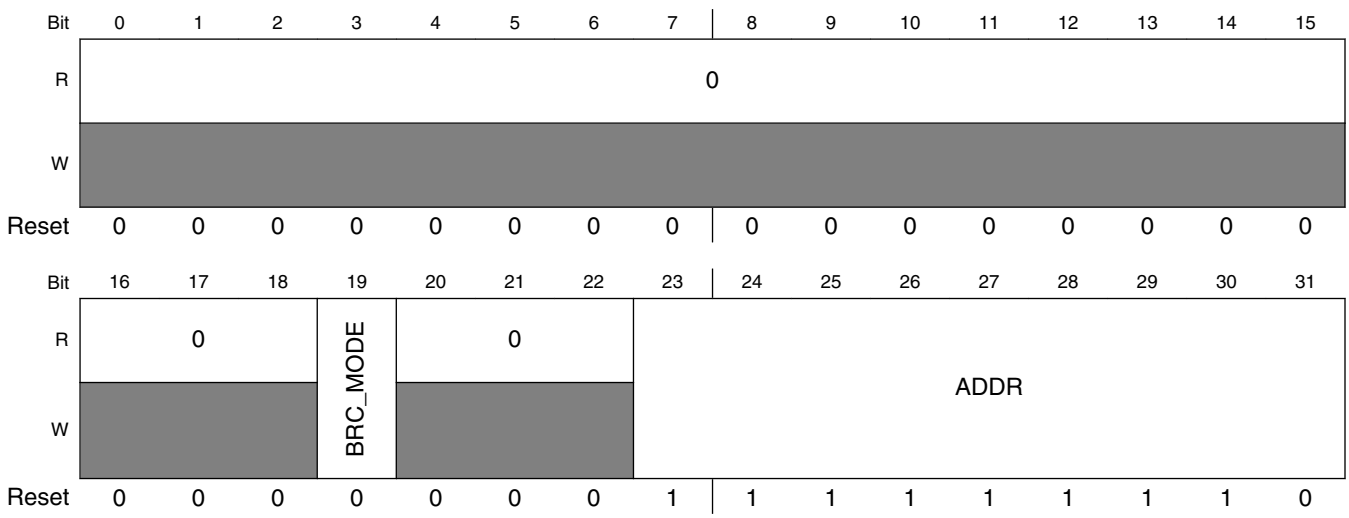
Table continues on the next page...

BRC memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
6C	BRC Interrupt Request Mode Register (BRC_IRQ_MODE)	32	R/W	See section	4.4.6/100
70	BRC Software Reset Register (BRC_RST)	32	R/W	0000_0000h	4.4.7/101
74	BRC Error Interrupt Request Enable Register (BRC_EIRQ_EN)	32	R/W	0000_0000h	4.4.8/102

4.4.1 BRC Source n Address Register, n[0:11] (BRC_SRC*n*_ADDR)

Address: 400h base + 0h offset + (8d × i), where i=0d to 11d



BRC_SRC*n*_ADDR field descriptions

Field	Description
0–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 BRC_MODE	BRC Operation mode select. This bit is writeable only if channel is disabled. 0 Consistency Mode (DCM) selected. 1 Maximum Throughput Mode (MTM) selected.
20–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–31 ADDR	Source ARU address. Defines the ARU read address used as data source for input channel n[0:11]. This bit field is writeable only if channel n is disabled.

4.4.2 BRC Source n to Destination Register, n[0:11] (BRC_SRC*n*_DEST)

Bit[10:31] is cleared by auto correction mechanism if a destination channel is assigned to multiple source channels.

When a BRC input channel is disabled (all EN_DEST[0:21] bits are reset to zero), the internal states are reset to their reset values.

Address: 400h base + 4h offset + (8d × i), where i=0d to 11d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0									EN_TRASHBIN	EN_DEST21	EN_DEST20	EN_DEST19	EN_DEST18	EN_DEST17	EN_DEST16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	EN_DEST15	EN_DEST14	EN_DEST13	EN_DEST12	EN_DEST11	EN_DEST10	EN_DEST9	EN_DEST8	EN_DEST7	EN_DEST6	EN_DEST5	EN_DEST4	EN_DEST3	EN_DEST2	EN_DEST1	EN_DEST0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BRC_SRC*n*_DEST field descriptions

Field	Description
0–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 EN_TRASHBIN	Control trash bin functionality. When EN_TRASHBIN is enabled, input channel n EN_DEST is ignored. Therefore, EN_DEST is set to zero (0) when trash bin functionality is enabled. 0 Trash bin functionality is disabled. 1 Trash bin functionality is enabled.
10 EN_DEST21	See EN_DEST0.
11 EN_DEST20	See EN_DEST0.
12 EN_DEST19	See EN_DEST0.
13 EN_DEST18	See EN_DEST0.
14 EN_DEST17	See EN_DEST0.

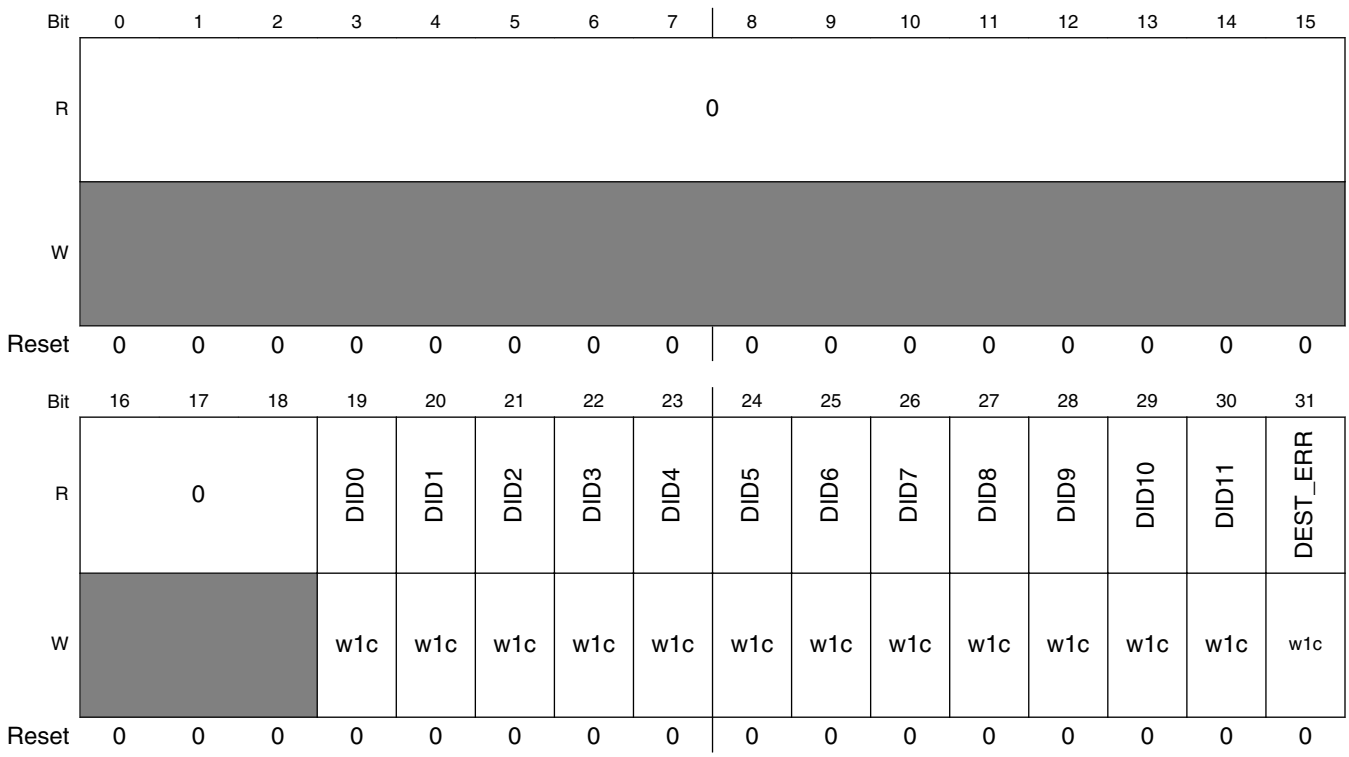
Table continues on the next page...

BRC_SRC*n*_DEST field descriptions (continued)

Field	Description
15 EN_DEST16	See EN_DEST0.
16 EN_DEST15	See EN_DEST0.
17 EN_DEST14	See EN_DEST0.
18 EN_DEST13	See EN_DEST0.
19 EN_DEST12	See EN_DEST0.
20 EN_DEST11	See EN_DEST0.
21 EN_DEST10	See EN_DEST0.
22 EN_DEST9	See EN_DEST0.
23 EN_DEST8	See EN_DEST0.
24 EN_DEST7	See EN_DEST0.
25 EN_DEST6	See EN_DEST0.
26 EN_DEST5	See EN_DEST0.
27 EN_DEST4	See EN_DEST0.
28 EN_DEST3	See EN_DEST0.
29 EN_DEST2	See EN_DEST0.
30 EN_DEST1	See EN_DEST0.
31 EN_DEST0	Enable BRC destination address. Each bit of EN_DEST provides an enable for a corresponding destination address. The destination addresses for a BRC channel is defined in section 21.2. 0 Destination address not mapped to source BRC_SRC_x_ADDR. 1 Destination address mapped to source BRC_SRC_x_ADDR.

4.4.3 BRC Interrupt Request Notification Register (BRC_IRQ_NOTIFY)

Address: 400h base + 60h offset = 460h



BRC_IRQ_NOTIFY field descriptions

Field	Description
0–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 DID0	Data inconsistency occurred in MTM mode. This DID bit corresponds to a channel 0. Each bit is cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
20 DID1	Data inconsistency occurred in MTM mode. This DID bit corresponds to a channel 1. Each bit is cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
21 DID2	Data inconsistency occurred in MTM mode. This DID bit corresponds to a channel 2. Each bit is cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
22 DID3	Data inconsistency occurred in MTM mode. This DID bit corresponds to a channel 3. Each bit is cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Table continues on the next page...

BRC_IRQ_NOTIFY field descriptions (continued)

Field	Description
23 DID4	Data inconsistency occurred in MTM mode. This DID bit corresponds to a channel 4. Each bit is cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
24 DID5	Data inconsistency occurred in MTM mode. This DID bit corresponds to a channel 5. Each bit is cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
25 DID6	Data inconsistency occurred in MTM mode. This DID bit corresponds to a channel 6. Each bit is cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
26 DID7	Data inconsistency occurred in MTM mode. This DID bit corresponds to a channel 7. Each bit is cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
27 DID8	Data inconsistency occurred in MTM mode. This DID bit corresponds to a channel 8. Each bit is cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
28 DID9	Data inconsistency occurred in MTM mode. This DID bit corresponds to a channel 9. Each bit is cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
29 DID10	Data inconsistency occurred in MTM mode. This DID bit corresponds to a channel 10. Each bit is cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
30 DID11	Data inconsistency occurred in MTM mode. This DID bit corresponds to a channel 11. Each bit is cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
31 DEST_ERR	Configuration error interrupt for BRC submodule. This bit is cleared by a CPU write access of value '1'. A read access leaves the bit unchanged. 0 No BRC configuration error occurred. 1 BRC configuration error occurred.

4.4.4 BRC Interrupt Request Enable Register (BRC_IRQ_EN)

Address: 400h base + 64h offset = 464h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0			DID_IRQ_EN0	DID_IRQ_EN1	DID_IRQ_EN2	DID_IRQ_EN3	DID_IRQ_EN4	DID_IRQ_EN5	DID_IRQ_EN6	DID_IRQ_EN7	DID_IRQ_EN8	DID_IRQ_EN9	DID_IRQ_EN10	DID_IRQ_EN11	DEST_ERR_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BRC_IRQ_EN field descriptions

Field	Description
0–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 DID_IRQ_EN0	BRC_DID_IRQ interrupt enable. Each DID_EN bit corresponds to a channel 0. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
20 DID_IRQ_EN1	BRC_DID_IRQ interrupt enable. Each DID_EN bit corresponds to a channel 1. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
21 DID_IRQ_EN2	BRC_DID_IRQ interrupt enable. Each DID_EN bit corresponds to a channel 2. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
22 DID_IRQ_EN3	BRC_DID_IRQ interrupt enable. Each DID_EN bit corresponds to a channel 3. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
23 DID_IRQ_EN4	BRC_DID_IRQ interrupt enable. Each DID_EN bit corresponds to a channel 4.

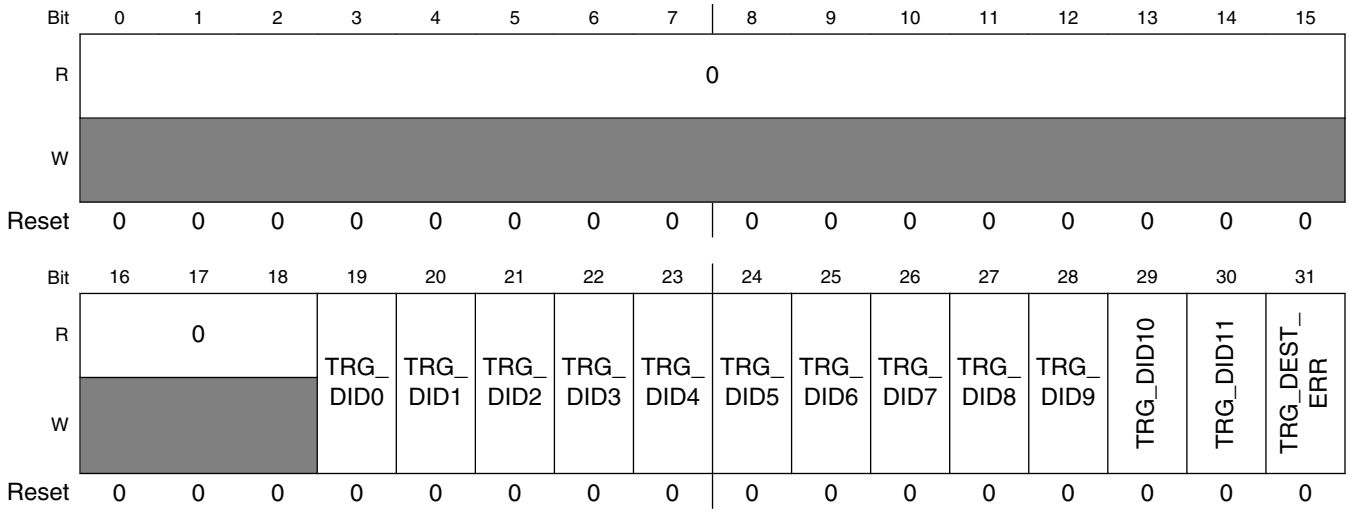
Table continues on the next page...

BRC_IRQ_EN field descriptions (continued)

Field	Description
	0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
24 DID_IRQ_EN5	BRC_DID_IRQ interrupt enable. Each DID_EN bit corresponds to a channel 5. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
25 DID_IRQ_EN6	BRC_DID_IRQ interrupt enable. Each DID_EN bit corresponds to a channel 6. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
26 DID_IRQ_EN7	BRC_DID_IRQ interrupt enable. Each DID_EN bit corresponds to a channel 7. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
27 DID_IRQ_EN8	BRC_DID_IRQ interrupt enable. Each DID_EN bit corresponds to a channel 8. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
28 DID_IRQ_EN9	BRC_DID_IRQ interrupt enable. Each DID_EN bit corresponds to a channel 9. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
29 DID_IRQ_EN10	BRC_DID_IRQ interrupt enable. Each DID_EN bit corresponds to a channel 10. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
30 DID_IRQ_EN11	BRC_DID_IRQ interrupt enable. Each DID_EN bit corresponds to a channel 11. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
31 DEST_ERR_EN	BRC_DEST_ERR_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM 1 Enable interrupt, interrupt is visible outside GTM

4.4.5 BRC Force Interrupt Request Register (BRC_IRQ_FORCINT)

Address: 400h base + 68h offset = 468h



BRC_IRQ_FORCINT field descriptions

Field	Description
0–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 TRG_DID0	Trigger data inconsistency error interrupt. This TRG_DID bit corresponds to a channel 0. The bit is cleared automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in BRC_IRQ_NOTIFY register.
20 TRG_DID1	Trigger data inconsistency error interrupt. This TRG_DID bit corresponds to a channel 1. The bit is cleared automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in BRC_IRQ_NOTIFY register.
21 TRG_DID2	Trigger data inconsistency error interrupt. This TRG_DID bit corresponds to a channel 2. The bit is cleared automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in BRC_IRQ_NOTIFY register.

Table continues on the next page...

BRC_IRQ_FORCINT field descriptions (continued)

Field	Description
22 TRG_DID3	Trigger data inconsistency error interrupt. This TRG_DID bit corresponds to a channel 3. The bit is cleared automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in BRC_IRQ_NOTIFY register.
23 TRG_DID4	Trigger data inconsistency error interrupt. This TRG_DID bit corresponds to a channel 4. The bit is cleared automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in BRC_IRQ_NOTIFY register.
24 TRG_DID5	Trigger data inconsistency error interrupt. This TRG_DID bit corresponds to a channel 5. The bit is cleared automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in BRC_IRQ_NOTIFY register.
25 TRG_DID6	Trigger data inconsistency error interrupt. This TRG_DID bit corresponds to a channel 6. The bit is cleared automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in BRC_IRQ_NOTIFY register.
26 TRG_DID7	Trigger data inconsistency error interrupt. This TRG_DID bit corresponds to a channel 7. The bit is cleared automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in BRC_IRQ_NOTIFY register.
27 TRG_DID8	Trigger data inconsistency error interrupt. This TRG_DID bit corresponds to a channel 8. The bit is cleared automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in BRC_IRQ_NOTIFY register.

Table continues on the next page...

BRC_IRQ_FORCINT field descriptions (continued)

Field	Description
28 TRG_DID9	Trigger data inconsistency error interrupt. This TRG_DID bit corresponds to a channel 9. The bit is cleared automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in BRC_IRQ_NOTIFY register.
29 TRG_DID10	Trigger data inconsistency error interrupt. This TRG_DID bit corresponds to a channel 10. The bit is cleared automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in BRC_IRQ_NOTIFY register.
30 TRG_DID11	Trigger data inconsistency error interrupt. This TRG_DID bit corresponds to a channel 11. The bit is cleared automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in BRC_IRQ_NOTIFY register.
31 TRG_DEST_ ERR	Trigger destination error interrupt. This bit is automatically cleared after a write access. This bit is write protected by GTM_CTRL[RF_PROT]. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in BRC_IRQ_NOTIFY register.

4.4.6 BRC Interrupt Request Mode Register (BRC_IRQ_MODE)

Address: 400h base + 6Ch offset = 46Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0														IRQ_MODE	
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- Bits 28-31 do not reset to a given value.

BRC_IRQ_MODE field descriptions

Field	Description
0–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–31 IRQ_MODE	IRQ mode select The interrupt modes are described in section 2.5. 00 Level mode 01 Pulse mode 10 Pulse-Notify mode 11 Single-Pulse mode

4.4.7 BRC Software Reset Register (BRC_RST)

Address: 400h base + 70h offset = 470h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

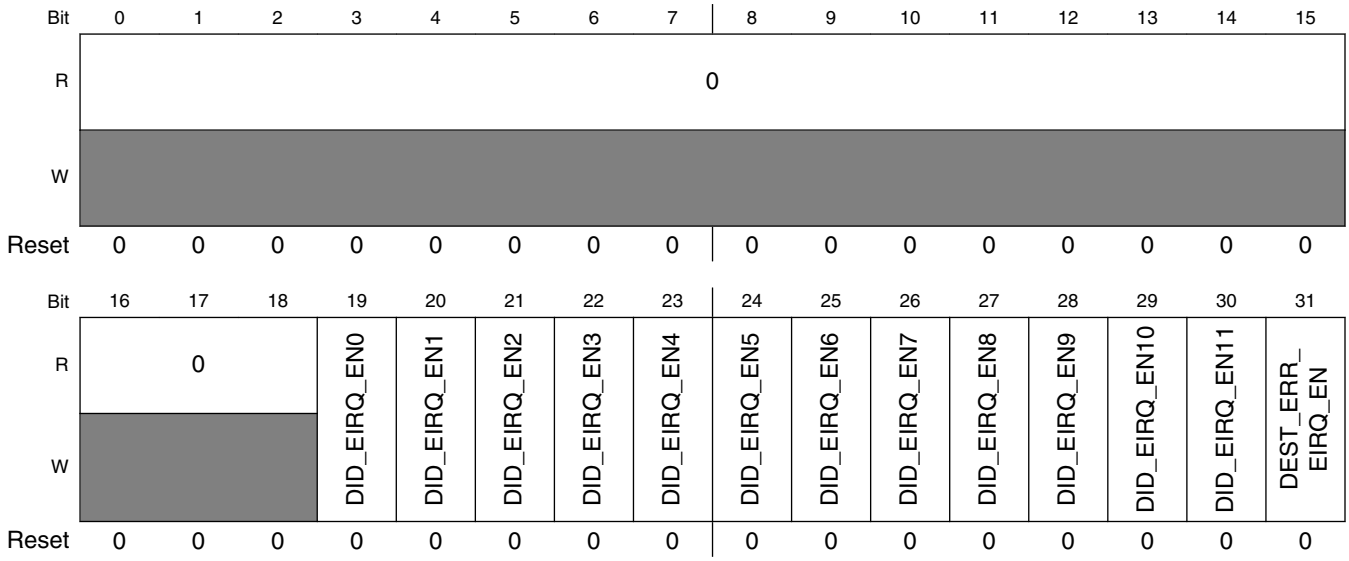
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0															RST
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BRC_RST field descriptions

Field	Description
0–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
31 RST	Software reset. This bit is automatically cleared after a write access by the CPU. The channel registers are set to their reset values and channel operation is stopped immediately. 0 No action 1 Reset BRC

4.4.8 BRC Error Interrupt Request Enable Register (BRC_EIRQ_EN)

Address: 400h base + 74h offset = 474h



BRC_EIRQ_EN field descriptions

Field	Description
0–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 DID_EIRQ_EN0	BRC_DID_EIRQ error interrupt enable. This bit corresponds to a channel 0. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.
20 DID_EIRQ_EN1	BRC_DID_EIRQ error interrupt enable. This bit corresponds to a channel 1. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.
21 DID_EIRQ_EN2	BRC_DID_EIRQ error interrupt enable. This bit corresponds to a channel 2. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.
22 DID_EIRQ_EN3	BRC_DID_EIRQ error interrupt enable. This bit corresponds to a channel 3. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.

Table continues on the next page...

BRC_EIRQ_EN field descriptions (continued)

Field	Description
23 DID_EIRQ_EN4	BRC_DID_EIRQ error interrupt enable. This bit corresponds to a channel 4. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.
24 DID_EIRQ_EN5	BRC_DID_EIRQ error interrupt enable. This bit corresponds to a channel 5. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.
25 DID_EIRQ_EN6	BRC_DID_EIRQ error interrupt enable. This bit corresponds to a channel 6. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.
26 DID_EIRQ_EN7	BRC_DID_EIRQ error interrupt enable. This bit corresponds to a channel 7. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.
27 DID_EIRQ_EN8	BRC_DID_EIRQ error interrupt enable. This bit corresponds to a channel 8. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.
28 DID_EIRQ_EN9	BRC_DID_EIRQ error interrupt enable. This bit corresponds to a channel 9. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.
29 DID_EIRQ_EN10	BRC_DID_EIRQ error interrupt enable. This bit corresponds to a channel 10. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.
30 DID_EIRQ_EN11	BRC_DID_EIRQ error interrupt enable. This bit corresponds to a channel 11. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.
31 DEST_ERR_ EIRQ_EN	BRC_DEST_ERR_EIRQ error interrupt enable. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP 1 Enable error interrupt, error interrupt is visible outside GTM-IP

Chapter 5
First In First Out Module (FIF0)

5.1 FIFO Overview

The FIFO unit is the storage part of the FIFO submodule. The FIFO-to-ARU Unit (F2A), described in F2A Overview, and the AEI-to-FIFO Data Interface Unit (AFD), described in AFD Overview, implement the interface part of the FIFO submodule. Each FIFO unit embeds eight logical FIFOs, where each logical FIFO is configurable for:

- size (defines start and end address),
- operation mode (normal or ring buffer),
- fill level control / memory region read protection, and
- normal FIFO operation mode or ring buffer operation mode.

Each logical FIFO represents a data stream between submodules of the GTM and the CPU, which connects to the AFD submodule via the AEI bus (see AFD Overview). The FIFO RAM has a depth of 1K words, where each word is 29 bits wide (24 data bits and five control bits).

For accessing its contents, the FIFO unit has three ports:

- F2A interface port,
- AFD interface port, and
- AEI bus interface port.

The AFD interface port always has the highest priority. The priority between the F2A interface port and the AEI bus interface port can be selected by software writing to the **FIFO[i]_CHn_CTRL** registers for all FIFO channels. Simultaneous accesses to the FIFO from the AFD interface port and the AEI bus interface port are prohibited because both ports are driven from the same AEI bus interface. The addresses for accessing the RAM via AEI can be found in the device-specific Appendix B for this document.

After reset, the FIFO RAM is not initialized by hardware. The FIFO channels can be flushed individually.

Beside the possibility of flushing each FIFO channel directly, a write access to FIFO[i]_CHn_END_ADDR or to FIFO[i]_CHn_START_ADDR will also flush the corresponding channel which means that the read and write pointer and also the fill level of the corresponding channel will be reset. In consequence of this, existing datums in the corresponding FIFO channel are no longer valid- thereafter the channel is empty.

5.2 Operation Modes

5.2.1 Normal Operation Mode

In the normal FIFO operation mode, the content of the FIFO is written and read in first-in first-out order, where data is deleted after it is forwarded to the AEI bus or to the F2A submodule (see F2A Overview).

The **FIFO[i]_CHn_UPPER_WM** and **FIFO[i]_CHn_LOWER_WM** upper and lower watermark registers, respectively, are used for controlling the FIFO's fill level. If the fill level declines below the lower watermark or it exceeds the upper watermark, a FIFO[i]_IRQ interrupt signal is asserted by the FIFO submodule (if the interrupt is enabled in the **FIFO[i]_IRQ_EN** register).

The FIFO interrupt signals are sent to the Interrupt Concentrator Module (ICM) (see ICM Overview).

5.2.2 Ring Buffer Operation Mode

In the ring buffer operation mode, the FIFO (without CPU interaction) provides a continuous data or configuration stream to other GTM submodules via the F2A submodule. The first word of the FIFO is delivered to the ARU first and after the last word is delivered to the ARU, the first word can be delivered again.

If in ring buffer mode, when the read pointer reaches the write pointer, it will be set again to the configured start address. So the read pointer always rotates cyclic between the configured start address of the regarding FIFO channel (first written data) and the write pointer which points to the last written data of the channel.

It is possible to add data via the AEI to FIFO interface (AFD) to the FIFO channel while running in ring buffer mode. The new written data will be add in the next ring buffer cycle.

Filling the FIFO channel first before enabling the data stream in the FIFO to ARU interface (F2A) is recommended.

Direct memory access provided by the FIFO AEI interface allows a change of data inside the continuous data stream to the GTM submodules or AEI bus.

5.3 FIFO Interrupt Signals

FIFO interrupt signals are shown in Table 5-1.

Table 5-1. FIFO interrupt signals

Signal	Description
FIFO[i]_CH[x]_EMPTY	Indicates empty FIFO x (x:0…7) was reached
FIFO[i]_CH[x]_FULL	Indicates full FIFO x (x:0…7) was reached
FIFO[i]_CH[x]_LOWER_WM	Indicates FIFO x (x:0…7) reached lower watermark.
FIFO[i]_CH[x]_UPPER_WM	Indicates FIFO x (x:0…7) reached upper watermark.

5.4 Memory Map and Registers

First In First Out (FIFO0) module configuration registers are described as follows:

FIFO_0 memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	FIFO0 Channel Control Register (FIFO_0_CH0_CTRL)	32	R/W	0000_0000h	5.4.1/112
4	FIFO0 Channel End Address Register (FIFO_0_CH0_END_ADDR)	32	R/W	See section	5.4.2/113
8	FIFO0 Channel Start Address Register (FIFO_0_CH0_START_ADDR)	32	R/W	See section	5.4.3/114
C	FIFO0 Channel Upper Watermark Register (FIFO_0_CH0_UPPER_WM)	32	R/W	0000_0060h	5.4.4/114
10	FIFO0 Channel Lower Watermark Register (FIFO_0_CH0_LOWER_WM)	32	R/W	0000_0020h	5.4.5/115
14	FIFO0 Channel Status Register (FIFO_0_CH0_STATUS)	32	R	0000_0005h	5.4.6/116

Table continues on the next page...

FIFO_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
18	FIFO0 Channel Fill Level Register (FIFO_0_CH0_FILL_LEVEL)	32	R	0000_0000h	5.4.7/117
1C	FIFO0 Channel Write Pointer Register (FIFO_0_CH0_WR_PTR)	32	R	See section	5.4.8/117
20	FIFO0 Channel Read Pointer (FIFO_0_CH0_RD_PTR)	32	R	See section	5.4.9/118
24	FIFO0 Channel IRQ Notification Register (FIFO_0_CH0_IRQ_NOTIFY)	32	R/W	0000_0005h	5.4.10/119
28	FIFO0 Channel IRQ Enable Register (FIFO_0_CH0_IRQ_EN)	32	R/W	0000_0000h	5.4.11/120
2C	FIFO0 Channel Force Interrupt Register (FIFO_0_CH0_IRQ_FORCINT)	32	R	0000_0000h	5.4.12/121
30	FIFO0 Channel IRQ Mode Register (FIFO_0_CH0_IRQ_MODE)	32	R/W	0000_0000h	5.4.13/122
34	FIFO0 Channel Error Interrupt Request Register (FIFO_0_CH0_EIRQ_EN)	32	R/W	0000_0000h	5.4.14/123
40	FIFO0 Channel Control Register (FIFO_0_CH1_CTRL)	32	R/W	0000_0000h	5.4.1/112
44	FIFO0 Channel End Address Register (FIFO_0_CH1_END_ADDR)	32	R/W	See section	5.4.2/113
48	FIFO0 Channel Start Address Register (FIFO_0_CH1_START_ADDR)	32	R/W	See section	5.4.3/114
4C	FIFO0 Channel Upper Watermark Register (FIFO_0_CH1_UPPER_WM)	32	R/W	0000_0060h	5.4.4/114
50	FIFO0 Channel Lower Watermark Register (FIFO_0_CH1_LOWER_WM)	32	R/W	0000_0020h	5.4.5/115
54	FIFO0 Channel Status Register (FIFO_0_CH1_STATUS)	32	R	0000_0005h	5.4.6/116
58	FIFO0 Channel Fill Level Register (FIFO_0_CH1_FILL_LEVEL)	32	R	0000_0000h	5.4.7/117
5C	FIFO0 Channel Write Pointer Register (FIFO_0_CH1_WR_PTR)	32	R	See section	5.4.8/117
60	FIFO0 Channel Read Pointer (FIFO_0_CH1_RD_PTR)	32	R	See section	5.4.9/118
64	FIFO0 Channel IRQ Notification Register (FIFO_0_CH1_IRQ_NOTIFY)	32	R/W	0000_0005h	5.4.10/119
68	FIFO0 Channel IRQ Enable Register (FIFO_0_CH1_IRQ_EN)	32	R/W	0000_0000h	5.4.11/120
6C	FIFO0 Channel Force Interrupt Register (FIFO_0_CH1_IRQ_FORCINT)	32	R	0000_0000h	5.4.12/121
70	FIFO0 Channel IRQ Mode Register (FIFO_0_CH1_IRQ_MODE)	32	R/W	0000_0000h	5.4.13/122
74	FIFO0 Channel Error Interrupt Request Register (FIFO_0_CH1_EIRQ_EN)	32	R/W	0000_0000h	5.4.14/123
80	FIFO0 Channel Control Register (FIFO_0_CH2_CTRL)	32	R/W	0000_0000h	5.4.1/112
84	FIFO0 Channel End Address Register (FIFO_0_CH2_END_ADDR)	32	R/W	See section	5.4.2/113

Table continues on the next page...

FIFO_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
88	FIFO0 Channel Start Address Register (FIFO_0_CH2_START_ADDR)	32	R/W	See section	5.4.3/114
8C	FIFO0 Channel Upper Watermark Register (FIFO_0_CH2_UPPER_WM)	32	R/W	0000_0060h	5.4.4/114
90	FIFO0 Channel Lower Watermark Register (FIFO_0_CH2_LOWER_WM)	32	R/W	0000_0020h	5.4.5/115
94	FIFO0 Channel Status Register (FIFO_0_CH2_STATUS)	32	R	0000_0005h	5.4.6/116
98	FIFO0 Channel Fill Level Register (FIFO_0_CH2_FILL_LEVEL)	32	R	0000_0000h	5.4.7/117
9C	FIFO0 Channel Write Pointer Register (FIFO_0_CH2_WR_PTR)	32	R	See section	5.4.8/117
A0	FIFO0 Channel Read Pointer (FIFO_0_CH2_RD_PTR)	32	R	See section	5.4.9/118
A4	FIFO0 Channel IRQ Notification Register (FIFO_0_CH2_IRQ_NOTIFY)	32	R/W	0000_0005h	5.4.10/119
A8	FIFO0 Channel IRQ Enable Register (FIFO_0_CH2_IRQ_EN)	32	R/W	0000_0000h	5.4.11/120
AC	FIFO0 Channel Force Interrupt Register (FIFO_0_CH2_IRQ_FORCINT)	32	R	0000_0000h	5.4.12/121
B0	FIFO0 Channel IRQ Mode Register (FIFO_0_CH2_IRQ_MODE)	32	R/W	0000_0000h	5.4.13/122
B4	FIFO0 Channel Error Interrupt Request Register (FIFO_0_CH2_EIRQ_EN)	32	R/W	0000_0000h	5.4.14/123
C0	FIFO0 Channel Control Register (FIFO_0_CH3_CTRL)	32	R/W	0000_0000h	5.4.1/112
C4	FIFO0 Channel End Address Register (FIFO_0_CH3_END_ADDR)	32	R/W	See section	5.4.2/113
C8	FIFO0 Channel Start Address Register (FIFO_0_CH3_START_ADDR)	32	R/W	See section	5.4.3/114
CC	FIFO0 Channel Upper Watermark Register (FIFO_0_CH3_UPPER_WM)	32	R/W	0000_0060h	5.4.4/114
D0	FIFO0 Channel Lower Watermark Register (FIFO_0_CH3_LOWER_WM)	32	R/W	0000_0020h	5.4.5/115
D4	FIFO0 Channel Status Register (FIFO_0_CH3_STATUS)	32	R	0000_0005h	5.4.6/116
D8	FIFO0 Channel Fill Level Register (FIFO_0_CH3_FILL_LEVEL)	32	R	0000_0000h	5.4.7/117
DC	FIFO0 Channel Write Pointer Register (FIFO_0_CH3_WR_PTR)	32	R	See section	5.4.8/117
E0	FIFO0 Channel Read Pointer (FIFO_0_CH3_RD_PTR)	32	R	See section	5.4.9/118
E4	FIFO0 Channel IRQ Notification Register (FIFO_0_CH3_IRQ_NOTIFY)	32	R/W	0000_0005h	5.4.10/119
E8	FIFO0 Channel IRQ Enable Register (FIFO_0_CH3_IRQ_EN)	32	R/W	0000_0000h	5.4.11/120
EC	FIFO0 Channel Force Interrupt Register (FIFO_0_CH3_IRQ_FORCINT)	32	R	0000_0000h	5.4.12/121

Table continues on the next page...

FIFO_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F0	FIFO0 Channel IRQ Mode Register (FIFO_0_CH3_IRQ_MODE)	32	R/W	0000_0000h	5.4.13/122
F4	FIFO0 Channel Error Interrupt Request Register (FIFO_0_CH3_EIRQ_EN)	32	R/W	0000_0000h	5.4.14/123
100	FIFO0 Channel Control Register (FIFO_0_CH4_CTRL)	32	R/W	0000_0000h	5.4.1/112
104	FIFO0 Channel End Address Register (FIFO_0_CH4_END_ADDR)	32	R/W	See section	5.4.2/113
108	FIFO0 Channel Start Address Register (FIFO_0_CH4_START_ADDR)	32	R/W	See section	5.4.3/114
10C	FIFO0 Channel Upper Watermark Register (FIFO_0_CH4_UPPER_WM)	32	R/W	0000_0060h	5.4.4/114
110	FIFO0 Channel Lower Watermark Register (FIFO_0_CH4_LOWER_WM)	32	R/W	0000_0020h	5.4.5/115
114	FIFO0 Channel Status Register (FIFO_0_CH4_STATUS)	32	R	0000_0005h	5.4.6/116
118	FIFO0 Channel Fill Level Register (FIFO_0_CH4_FILL_LEVEL)	32	R	0000_0000h	5.4.7/117
11C	FIFO0 Channel Write Pointer Register (FIFO_0_CH4_WR_PTR)	32	R	See section	5.4.8/117
120	FIFO0 Channel Read Pointer (FIFO_0_CH4_RD_PTR)	32	R	See section	5.4.9/118
124	FIFO0 Channel IRQ Notification Register (FIFO_0_CH4_IRQ_NOTIFY)	32	R/W	0000_0005h	5.4.10/119
128	FIFO0 Channel IRQ Enable Register (FIFO_0_CH4_IRQ_EN)	32	R/W	0000_0000h	5.4.11/120
12C	FIFO0 Channel Force Interrupt Register (FIFO_0_CH4_IRQ_FORCINT)	32	R	0000_0000h	5.4.12/121
130	FIFO0 Channel IRQ Mode Register (FIFO_0_CH4_IRQ_MODE)	32	R/W	0000_0000h	5.4.13/122
134	FIFO0 Channel Error Interrupt Request Register (FIFO_0_CH4_EIRQ_EN)	32	R/W	0000_0000h	5.4.14/123
140	FIFO0 Channel Control Register (FIFO_0_CH5_CTRL)	32	R/W	0000_0000h	5.4.1/112
144	FIFO0 Channel End Address Register (FIFO_0_CH5_END_ADDR)	32	R/W	See section	5.4.2/113
148	FIFO0 Channel Start Address Register (FIFO_0_CH5_START_ADDR)	32	R/W	See section	5.4.3/114
14C	FIFO0 Channel Upper Watermark Register (FIFO_0_CH5_UPPER_WM)	32	R/W	0000_0060h	5.4.4/114
150	FIFO0 Channel Lower Watermark Register (FIFO_0_CH5_LOWER_WM)	32	R/W	0000_0020h	5.4.5/115
154	FIFO0 Channel Status Register (FIFO_0_CH5_STATUS)	32	R	0000_0005h	5.4.6/116
158	FIFO0 Channel Fill Level Register (FIFO_0_CH5_FILL_LEVEL)	32	R	0000_0000h	5.4.7/117
15C	FIFO0 Channel Write Pointer Register (FIFO_0_CH5_WR_PTR)	32	R	See section	5.4.8/117
160	FIFO0 Channel Read Pointer (FIFO_0_CH5_RD_PTR)	32	R	See section	5.4.9/118

Table continues on the next page...

FIFO_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
164	FIFO0 Channel IRQ Notification Register (FIFO_0_CH5_IRQ_NOTIFY)	32	R/W	0000_0005h	5.4.10/119
168	FIFO0 Channel IRQ Enable Register (FIFO_0_CH5_IRQ_EN)	32	R/W	0000_0000h	5.4.11/120
16C	FIFO0 Channel Force Interrupt Register (FIFO_0_CH5_IRQ_FORCINT)	32	R	0000_0000h	5.4.12/121
170	FIFO0 Channel IRQ Mode Register (FIFO_0_CH5_IRQ_MODE)	32	R/W	0000_0000h	5.4.13/122
174	FIFO0 Channel Error Interrupt Request Register (FIFO_0_CH5_EIRQ_EN)	32	R/W	0000_0000h	5.4.14/123
180	FIFO0 Channel Control Register (FIFO_0_CH6_CTRL)	32	R/W	0000_0000h	5.4.1/112
184	FIFO0 Channel End Address Register (FIFO_0_CH6_END_ADDR)	32	R/W	See section	5.4.2/113
188	FIFO0 Channel Start Address Register (FIFO_0_CH6_START_ADDR)	32	R/W	See section	5.4.3/114
18C	FIFO0 Channel Upper Watermark Register (FIFO_0_CH6_UPPER_WM)	32	R/W	0000_0060h	5.4.4/114
190	FIFO0 Channel Lower Watermark Register (FIFO_0_CH6_LOWER_WM)	32	R/W	0000_0020h	5.4.5/115
194	FIFO0 Channel Status Register (FIFO_0_CH6_STATUS)	32	R	0000_0005h	5.4.6/116
198	FIFO0 Channel Fill Level Register (FIFO_0_CH6_FILL_LEVEL)	32	R	0000_0000h	5.4.7/117
19C	FIFO0 Channel Write Pointer Register (FIFO_0_CH6_WR_PTR)	32	R	See section	5.4.8/117
1A0	FIFO0 Channel Read Pointer (FIFO_0_CH6_RD_PTR)	32	R	See section	5.4.9/118
1A4	FIFO0 Channel IRQ Notification Register (FIFO_0_CH6_IRQ_NOTIFY)	32	R/W	0000_0005h	5.4.10/119
1A8	FIFO0 Channel IRQ Enable Register (FIFO_0_CH6_IRQ_EN)	32	R/W	0000_0000h	5.4.11/120
1AC	FIFO0 Channel Force Interrupt Register (FIFO_0_CH6_IRQ_FORCINT)	32	R	0000_0000h	5.4.12/121
1B0	FIFO0 Channel IRQ Mode Register (FIFO_0_CH6_IRQ_MODE)	32	R/W	0000_0000h	5.4.13/122
1B4	FIFO0 Channel Error Interrupt Request Register (FIFO_0_CH6_EIRQ_EN)	32	R/W	0000_0000h	5.4.14/123
1C0	FIFO0 Channel Control Register (FIFO_0_CH7_CTRL)	32	R/W	0000_0000h	5.4.1/112
1C4	FIFO0 Channel End Address Register (FIFO_0_CH7_END_ADDR)	32	R/W	See section	5.4.2/113
1C8	FIFO0 Channel Start Address Register (FIFO_0_CH7_START_ADDR)	32	R/W	See section	5.4.3/114
1CC	FIFO0 Channel Upper Watermark Register (FIFO_0_CH7_UPPER_WM)	32	R/W	0000_0060h	5.4.4/114
1D0	FIFO0 Channel Lower Watermark Register (FIFO_0_CH7_LOWER_WM)	32	R/W	0000_0020h	5.4.5/115

Table continues on the next page...

FIFO_0 memory map (continued)

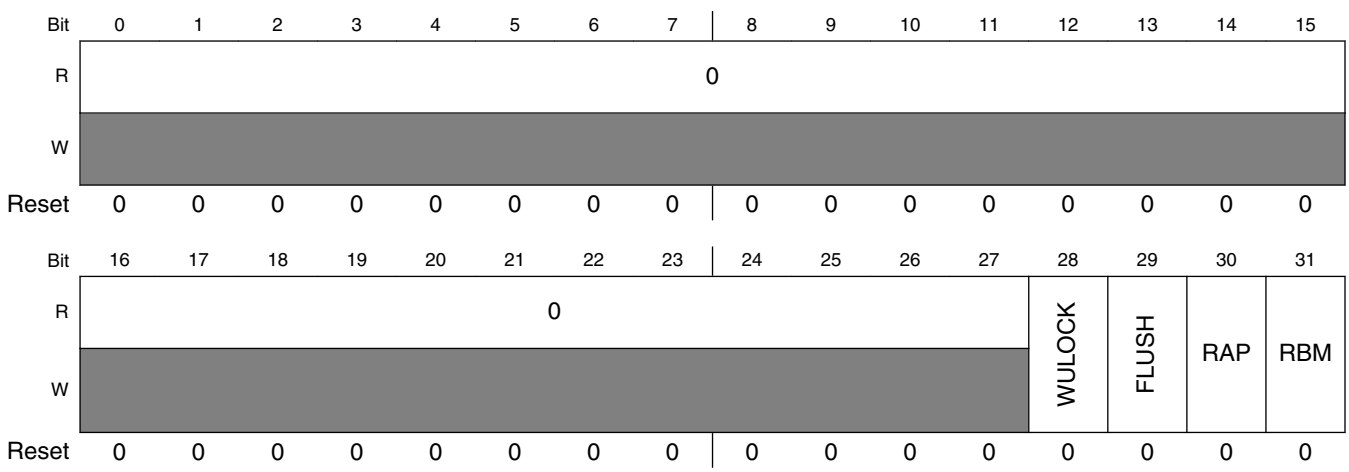
Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1D4	FIFO0 Channel Status Register (FIFO_0_CH7_STATUS)	32	R	0000_0005h	5.4.6/116
1D8	FIFO0 Channel Fill Level Register (FIFO_0_CH7_FILL_LEVEL)	32	R	0000_0000h	5.4.7/117
1DC	FIFO0 Channel Write Pointer Register (FIFO_0_CH7_WR_PTR)	32	R	See section	5.4.8/117
1E0	FIFO0 Channel Read Pointer (FIFO_0_CH7_RD_PTR)	32	R	See section	5.4.9/118
1E4	FIFO0 Channel IRQ Notification Register (FIFO_0_CH7_IRQ_NOTIFY)	32	R/W	0000_0005h	5.4.10/119
1E8	FIFO0 Channel IRQ Enable Register (FIFO_0_CH7_IRQ_EN)	32	R/W	0000_0000h	5.4.11/120
1EC	FIFO0 Channel Force Interrupt Register (FIFO_0_CH7_IRQ_FORCINT)	32	R	0000_0000h	5.4.12/121
1F0	FIFO0 Channel IRQ Mode Register (FIFO_0_CH7_IRQ_MODE)	32	R/W	0000_0000h	5.4.13/122
1F4	FIFO0 Channel Error Interrupt Request Register (FIFO_0_CH7_EIRQ_EN)	32	R/W	0000_0000h	5.4.14/123

5.4.1 FIFO0 Channel Control Register (FIFO_0_CH*n*_CTRL)

Only the WULOCK bit of register FIFO[i]_CHn_CTRL enables/disables the direct RAM write access. The WULOCK bits of the other channels are writeable but have no effect.

The RAP bit is only functional in register FIFO_CHn_CTRL. The priority is defined for all FIFO channels there.

Address: 1_8400h base + 0h offset + (64d × i), where i=0d to 7d



FIFO_0_CH*n*_CTRL field descriptions

Field	Description
0–27 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
28 WULOCK	RAM write unlock. Enable/disable direct RAM write access to the memory mapped FIFO region. **NOTE:** Only the FIFO[i]_CH0_CTRL[WULOCK] bit enables/disables the direct RAM write access for all FIFO channels (whole FIFO RAM). The WULOCK bits of the other channels are writeable, but have no effect. 0 Direct RAM write access disabled. 1 Direct RAM write access enabled.
29 FLUSH	FIFO Flush control. A FIFO Flush operation resets the FIFO[i]_CH[n]_FILL_LEVEL, FIFO[i]_CH[n]_WR_PTR and FIFO[i]_CH[n]_RD_PTR registers to their initial values. 0 Normal operation 1 Execute FIFO flush (bit is automatically cleared after flush).
30 RAP	RAM access priority. 0 FIFO ports have higher access priority than AEI-IF. 1 AEI-IF has higher access priority than FIFO ports.
31 RBM	Ring buffer mode enable. 0 Normal FIFO operation mode. 1 Ring buffer mode.

5.4.2 FIFO0 Channel End Address Register (FIFO_0_CH*n*_END_ADDR)

Address: 1_8400h base + 4h offset + (64d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																						ADDR									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FIFO_0_CH*n*_END_ADDR field descriptions

Field	Description
0–21 Reserved	Reserved. This field is reserved.

Table continues on the next page...

FIFO_0_CH*n*_END_ADDR field descriptions (continued)

Field	Description
	This read-only field is reserved and always has the value 0.
22–31 ADDR	End address for FIFO channel n. Initial value for ADDR is calculated as ADDR = 128*(n+1) -1. ADDR field value is relative to the FIFO RAM base address. **NOTE:** A write access will flush the corresponding channel.

5.4.3 FIFO0 Channel Start Address Register (FIFO_0_CH*n*_START_ADDR)

Address: 1_8400h base + 8h offset + (64d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																						ADDR									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FIFO_0_CH*n*_START_ADDR field descriptions

Field	Description
0–21 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
22–31 ADDR	Start address for FIFO channel n. Initial value for ADDR is calculated as ADDR = 128*n. ADDR field value is relative to the FIFO RAM base address. **NOTE:** A write access will flush the corresponding channel.

5.4.4 FIFO0 Channel Upper Watermark Register (FIFO_0_CH*n*_UPPER_WM)

Address: 1_8400h base + Ch offset + (64d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																						ADDR									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

FIFO_0_CH*n*_UPPER_WM field descriptions

Field	Description
0–21 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
22–31 ADDR	Upper watermark. The upper watermark is configured as a relative fill level of the FIFO. ADDR must be in range 0 ≤ ADDR ≤ FIFO[i]_CH[n]_END_ADDR minus FIFO[i]_CH[n]_START_ADDR.

5.4.5 FIFO0 Channel Lower Watermark Register (FIFO_0_CH*n*_LOWER_WM)

Address: 1_8400h base + 10h offset + (64d × i), where i=0d to 7d

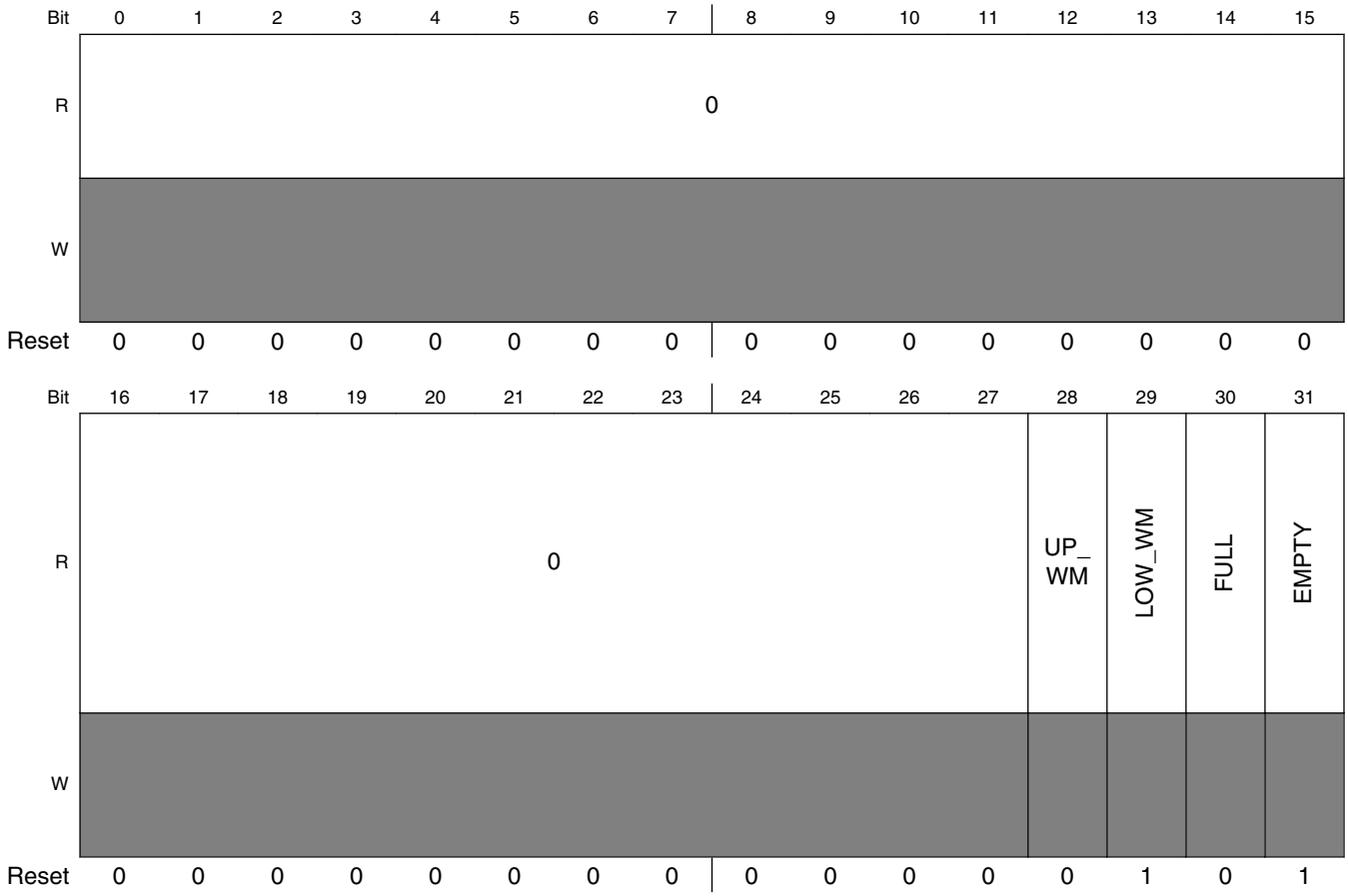
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																						ADDR									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

FIFO_0_CH*n*_LOWER_WM field descriptions

Field	Description
0–21 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
22–31 ADDR	Normal Operation mode: Lower watermark. Ring buffer operation mode: Gate pointer from LWU to UWE. The lower watermark is configured as a relative fill level of the FIFO. ADDR must be in range 0 ≤ ADDR ≤ FIFO[i]_CH[n]_END_ADDR minus FIFO[i]_CH[n]_START_ADDR.

5.4.6 FIFO0 Channel Status Register (FIFO_0_CH*n*_STATUS)

Address: 1_8400h base + 14h offset + (64d × i), where i=0d to 7d



FIFO_0_CH*n*_STATUS field descriptions

Field	Description
0–27 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
28 UP_WM	Upper watermark reached. Bit only applicable in normal mode. 0 Fill level less than upper watermark. 1 Fill level greater than or equal to upper watermark.
29 LOW_WM	Lower watermark reached. Bit only applicable in normal mode. 0 Fill level greater than lower watermark. 1 Fill level less than or equal to lower watermark.

Table continues on the next page...

FIFO_0_CH*n*_STATUS field descriptions (continued)

Field	Description
30 FULL	FIFO fill level status. Bit only applicable in normal mode 0 Fill level less than FIFO[i]_CH[x]_END_ADDR – FIFO[i]_CH[x]_START_ADDR + 1. 1 Fill level = FIFO[i]_CH[x]_END_ADDR – FIFO[i]_CH[x]_START_ADDR + 1
31 EMPTY	FIFO fill level status. Bit only applicable in normal mode 0 Fill level greater than 0. 1 Fill level = 0.

5.4.7 FIFO0 Channel Fill Level Register (FIFO_0_CH*n*_FILL_LEVEL)

Address: 1_8400h base + 18h offset + (64d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																					LEVEL										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIFO_0_CH*n*_FILL_LEVEL field descriptions

Field	Description
0–20 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
21–31 LEVEL	Fill level of the current FIFO. Register content is compared to the upper and lower watermark values for this channel to detect watermark over- and underflow. LEVEL is in range 0 ≤ LEVEL ≤ FIFO[i]_CH[n]_END_ADDR minus FIFO[i]_CH[n]_START_ADDR + 1.

5.4.8 FIFO0 Channel Write Pointer Register (FIFO_0_CH*n*_WR_PTR)

Address: 1_8400h base + 1Ch offset + (64d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																						ADDR									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FIFO_0_CH*n*_WR_PTR field descriptions

Field	Description
0–21 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
22–31 ADDR	Position of the write pointer. ADDR must be in range 0 less than or equal to ADDR less than or equal to 1023. Initial value for ADDR is defined as ADDR = FIFO[i]_CH[n]_START_ADDR.

5.4.9 FIFO0 Channel Read Pointer (FIFO_0_CH*n*_RD_PTR)

Address: 1_8400h base + 20h offset + (64d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																						ADDR									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

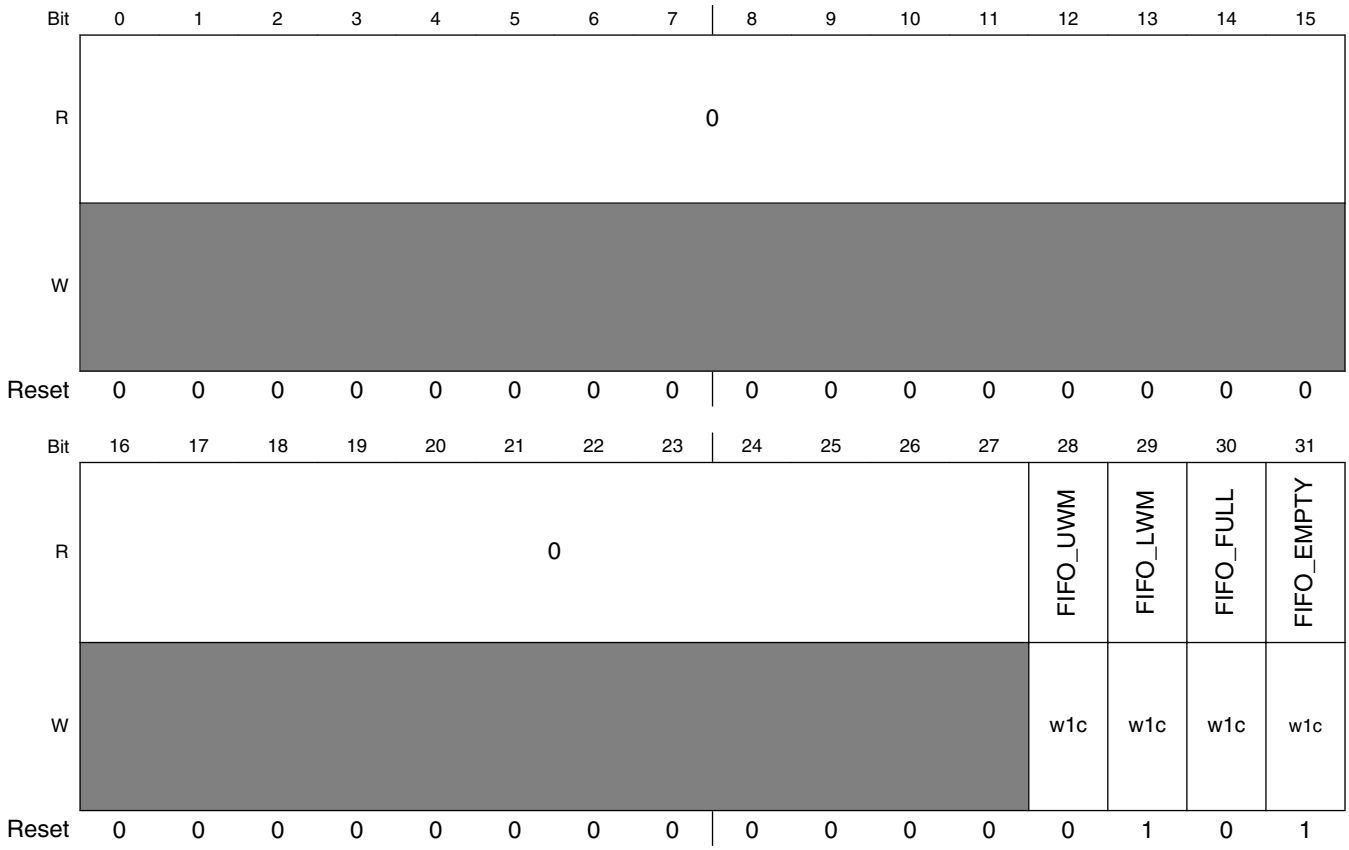
* Notes:
- x = Undefined at reset.

FIFO_0_CH*n*_RD_PTR field descriptions

Field	Description
0–21 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
22–31 ADDR	Position of the read pointer. ADDR must be in range 0 ≤ ADDR ≤ 1023. Initial value for ADDR is defined as ADDR = FIFO[i]_CH[n]_START_ADDR.

5.4.10 FIFO0 Channel IRQ Notification Register (FIFO_0_CH*n*_IRQ_NOTIFY)

Address: 1_8400h base + 24h offset + (64d × i), where i=0d to 7d



FIFO_0_CH*n*_IRQ_NOTIFY field descriptions

Field	Description
0–27 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
28 FIFO_UWM	FIFO Upper watermark was over-run. This bit is cleared by a CPU write access of value '1'. A read access leaves the bit unchanged. 0 No interrupt occurred. 1 FIFO interrupt occurred.
29 FIFO_LWM	FIFO Lower watermark was under-run. This bit is cleared by a CPU write access of value '1'. A read access leaves the bit unchanged. 0 No interrupt occurred. 1 FIFO interrupt occurred.

Table continues on the next page...

FIFO_0_CH*n*_IRQ_NOTIFY field descriptions (continued)

Field	Description
30 FIFO_FULL	FIFO is full. This bit is cleared by a CPU write access of value '1'. A read access leaves the bit unchanged. 0 No interrupt occurred. 1 FIFO is full interrupt occurred.
31 FIFO_EMPTY	FIFO is empty. This bit is cleared by a CPU write access of value '1'. A read access leaves the bit unchanged. 0 No interrupt occurred. 1 FIFO is empty interrupt occurred.

5.4.11 FIFO0 Channel IRQ Enable Register (FIFO_0_CH*n*_IRQ_EN)

Address: 1_8400h base + 28h offset + (64d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0												FIFO_UWM_IRQ_EN	FIFO_LWM_IRQ_EN	FIFO_FULL_IRQ_EN	FIFO_EMPTY_IRQ_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIFO_0_CH*n*_IRQ_EN field descriptions

Field	Description
0–27 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
28 FIFO_UWM_ IRQ_EN	FIFO Upper Water Mark Interrupt Enable. 0 Disable interrupt, interrupt is not visible outside GTM-IP. 1 Enable interrupt, interrupt is visible outside GTM-IP.
29 FIFO_LWM_ IRQ_EN	FIFO Lower Watermark Interrupt Enable.

Table continues on the next page...

FIFO_0_CH*n*_IRQ_EN field descriptions (continued)

Field	Description
	0 Disable interrupt, interrupt is not visible outside GTM-IP. 1 Enable interrupt, interrupt is visible outside GTM-IP.
30 FIFO_FULL_ IRQ_EN	FIFO Full Interrupt Enable. 0 Disable interrupt, interrupt is not visible outside GTM-IP. 1 Enable interrupt, interrupt is visible outside GTM-IP.
31 FIFO_EMPTY_ IRQ_EN	FIFO Empty Interrupt Enable. 0 Disable interrupt, interrupt is not visible outside GTM-IP. 1 Enable interrupt, interrupt is visible outside GTM-IP.

5.4.12 FIFO0 Channel Force Interrupt Register (FIFO_0_CH*n*_IRQ_FORCINT)

Address: 1_8400h base + 2Ch offset + (64d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0												TRG_FIFO_UWM	TRG_FIFO_LWM	TRG_FIFO_FULL	TRG_FIFO_EMPTY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIFO_0_CH*n*_IRQ_FORCINT field descriptions

Field	Description
0–27 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
28 TRG_FIFO_ UWM	Force interrupt of upper watermark. This bit is cleared automatically after write. 0 Corresponding bit in status register will not be forced 1 Assert corresponding field in FIFO[i]_CH[x]_IRQ_NOTIFY register.

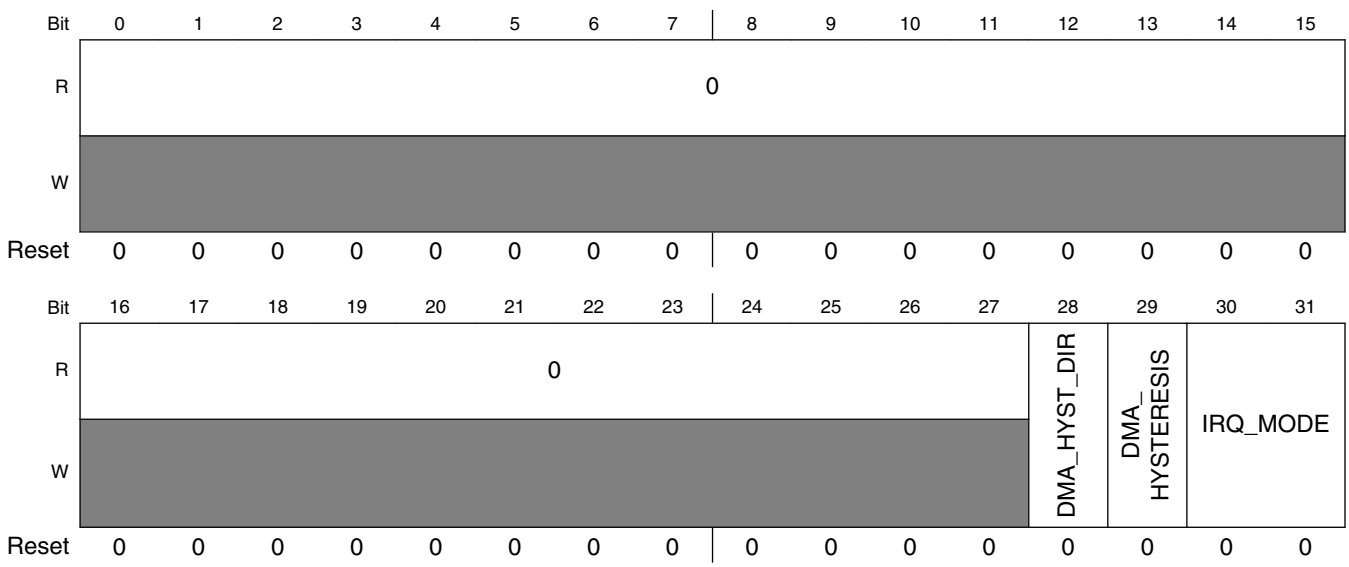
Table continues on the next page...

FIFO_0_CH*n*_IRQ_FORCINT field descriptions (continued)

Field	Description
29 TRG_FIFO_LWM	Force interrupt of lower watermark. This bit is automatically cleared after a write access. 0 Corresponding bit in status register will not be forced 1 Assert corresponding field in FIFO[i]_CH[x]_IRQ_NOTIFY register.
30 TRG_FIFO_ FULL	Force interrupt of FIFO full status. This bit is automatically cleared after a write access. 0 Corresponding bit in status register will not be forced 1 Assert corresponding field in FIFO[i]_CH[x]_IRQ_NOTIFY register.
31 TRG_FIFO_ EMPTY	Force interrupt of FIFO empty status. This bit is automatically cleared after a write access. 0 Corresponding bit in status register will not be forced 1 Assert corresponding field in FIFO[i]_CH[x]_IRQ_NOTIFY register.

5.4.13 FIFO0 Channel IRQ Mode Register (FIFO_0_CH*n*_IRQ_MODE)

Address: 1_8400h base + 30h offset + (64d × i), where i=0d to 7d



FIFO_0_CH*n*_IRQ_MODE field descriptions

Field	Description
0–27 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.

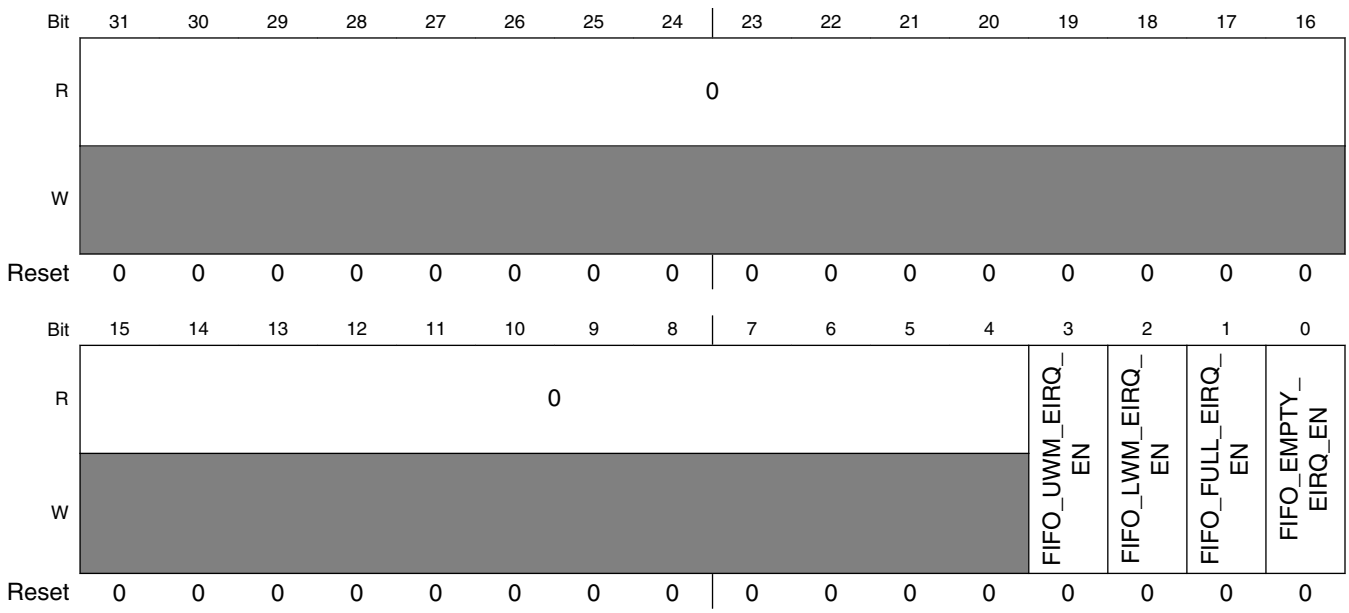
Table continues on the next page...

FIFO_0_CH*n*_IRQ_MODE field descriptions (continued)

Field	Description
28 DMA_HYST_DIR	DMA direction in hysteresis mode. In the case of DMA writing data to a FIFO, the DMA requests must be generated by the lower watermark. If the DMA hysteresis is enabled, the FIFO does not generate a new DMA request until the upper watermark is reached. In the case of DMA reading data from a FIFO, the DMA requests must be generated by the upper watermark. If the DMA hysteresis is enabled, the FIFO does not generate a new DMA request until the lower watermark is reached. 0 DMA direction read in hysteresis mode. 1 DMA direction write in hysteresis mode.
29 DMA_ HYSTERESIS	Enable DMA hysteresis mode. 0 Disable FIFO hysteresis for DMA access. 1 Enable FIFO hysteresis for DMA access.
30–31 IRQ_MODE	IRQ mode select. **NOTE:** The interrupt modes are described in section 2.5. 00 Level mode 01 Pulse mode 10 Pulse-Notify mode 11 Single-Pulse mode

5.4.14 FIFO0 Channel Error Interrupt Request Register (FIFO_0_CH*n*_EIRQ_EN)

Address: 1_8400h base + 34h offset + (64d × i), where i=0d to 7d



FIFO_0_CH*n*_EIRQ_EN field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 FIFO_UWM_ EIRQ_EN	FIFO Upper Water Mark Error Interrupt Request Enable. 0 Disable error interrupt, error interrupt is not visible outside GTM. 1 Enable error interrupt, error interrupt is visible outside GTM.
2 FIFO_LWM_ EIRQ_EN	FIFO Lower Water Mark Error Interrupt Request Enable. 0 Disable error interrupt, error interrupt is not visible outside GTM. 1 Enable error interrupt, error interrupt is visible outside GTM.
1 FIFO_FULL_ EIRQ_EN	FIFO Full Error Interrupt Request Enable. 0 Disable error interrupt, error interrupt is not visible outside GTM. 1 Enable error interrupt, error interrupt is visible outside GTM.
0 FIFO_EMPTY_ EIRQ_EN	FIFO Empty Error Interrupt Request Enable. 0 Disable error interrupt, error interrupt is not visible outside GTM. 1 Enable error interrupt, error interrupt is visible outside GTM.

Chapter 6
AEI to FIFO Data Interface (AFD)

6.1 AFD Overview

The AEI-to-FIFO Data Interface (AFD) submodule consists of eight logical FIFO channels that implement a data interface between the AEI bus and the FIFO submodule.

The AFD submodule provides eight buffer registers that are dedicated to the eight logical channels, respectively. A FIFO channel is accessed by reading or writing its respective **AFD[i]_CHn_BUF_ACC** (n:0..7) register.

If a FIFO channel is full, an AEI write access to the corresponding channel buffer register will be ignored. If a FIFO channel is empty, an AEI read access to the channel buffer register will result in a read of all zeros.

6.2 Memory Map and Registers

The AEI to FIFO Data Interface (AFD0) registers are described as follows:

AFD_0 memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	AFD0 FIFO Channel n Buffer Access Register, n=0:7 (AFD_0_CH0_BUF_ACC)	32	R/W	0000_0000h	6.2.1/126
10	AFD0 FIFO Channel n Buffer Access Register, n=0:7 (AFD_0_CH1_BUF_ACC)	32	R/W	0000_0000h	6.2.1/126
20	AFD0 FIFO Channel n Buffer Access Register, n=0:7 (AFD_0_CH2_BUF_ACC)	32	R/W	0000_0000h	6.2.1/126
30	AFD0 FIFO Channel n Buffer Access Register, n=0:7 (AFD_0_CH3_BUF_ACC)	32	R/W	0000_0000h	6.2.1/126

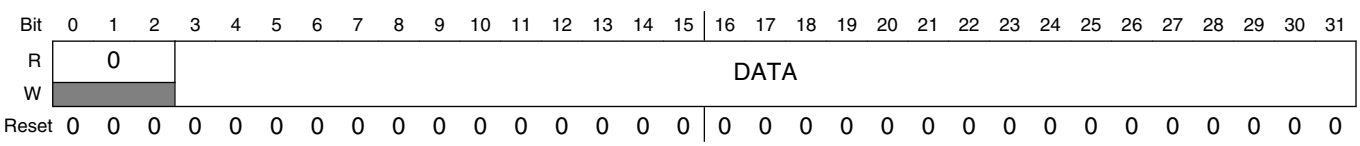
Table continues on the next page...

AFD_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
40	AFD0 FIFO Channel n Buffer Access Register, n=0:7 (AFD_0_CH4_BUF_ACC)	32	R/W	0000_0000h	6.2.1/126
50	AFD0 FIFO Channel n Buffer Access Register, n=0:7 (AFD_0_CH5_BUF_ACC)	32	R/W	0000_0000h	6.2.1/126
60	AFD0 FIFO Channel n Buffer Access Register, n=0:7 (AFD_0_CH6_BUF_ACC)	32	R/W	0000_0000h	6.2.1/126
70	AFD0 FIFO Channel n Buffer Access Register, n=0:7 (AFD_0_CH7_BUF_ACC)	32	R/W	0000_0000h	6.2.1/126

6.2.1 AFD0 FIFO Channel n Buffer Access Register, n=0:7 (AFD_0_CH*n*_BUF_ACC)

Address: 1_8000h base + 0h offset + (16d × i), where i=0d to 7d



AFD_0_CH*n*_BUF_ACC field descriptions

Field	Description
0–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–31 DATA	Read/write data from/to FIFO.

Chapter 7
FIFO-to-ARU (F2A) Unit

7.1 F2A Overview

The FIFO-to-ARU Unit (F2A) serves as the interface between the ARU submodule and the FIFO submodule. Since the data width of an ARU word is 53 bits (two 24-bit values and five control bits) and the data width of the FIFO is only 29 bits, the F2A submodule has to distribute the data from and to the FIFO channels in a configurable manner.

Data transfers between the FIFO submodule and the ARU submodule are handled by eight data/control streams that are connected to the eight channels of the corresponding FIFO submodule. So, a stream is a data/control flow from/to the ARU submodule, via the F2A submodule, to/from the FIFO submodule via a FIFO channel. Within a stream, the F2A submodule can transmit/receive the lower, upper, or lower and upper 24-bit data values of the ARU submodule along with the five ARU control bits. This is accomplished in accordance with the transfer mode that is configured (please see Transfer modes).

7.2 Transfer modes

The FIFO-to-ARU Unit (F2A) provides three transfer modes that map two 29-bit words (24-bit data word plus five control bits) from/to the FIFO into one 53-bit word (two 24-bit data words plus five control bits) to/from the ARU:

- transfer low data word (ARU data bits 23:0 plus control bits 48:52) from/to FIFO,
- transfer high data word (ARU data bits 47:24 plus control bits 48:52) from/to FIFO, or
- transfer both low and high data words, plus control bits, from/to FIFO.

The transfer mode can be selected by writing to the **F2A[i]_CHn_STR_CFG[TMODE]** bits (n: 0..7).

When both of the ARU data words plus control bits are transferred to the FIFO, they are stored one behind the other in the FIFO (if the FIFO is not full). If there is only space in the FIFO for one data word plus control bits, the F2A transfers it and then waits for an empty space before it transfers the second data word plus control bits. This must be accomplished before new data is requested from the ARU.

When two FIFO words (each having one 24-bit data word plus five control bits) are to be transferred into one ARU word (two 24-bit data words plus five control bits), the two FIFO words must be located one behind the other in the FIFO. The transfer of the two FIFO words to the ARU can proceed only when both FIFO words can be read out of the FIFO. If only one FIFO word is in the FIFO, the F2A waits until the second FIFO word is in the FIFO before transferring both to the ARU.

Figure 7-1 shows the ordering of the FIFO words when they are transferred to/from the ARU.

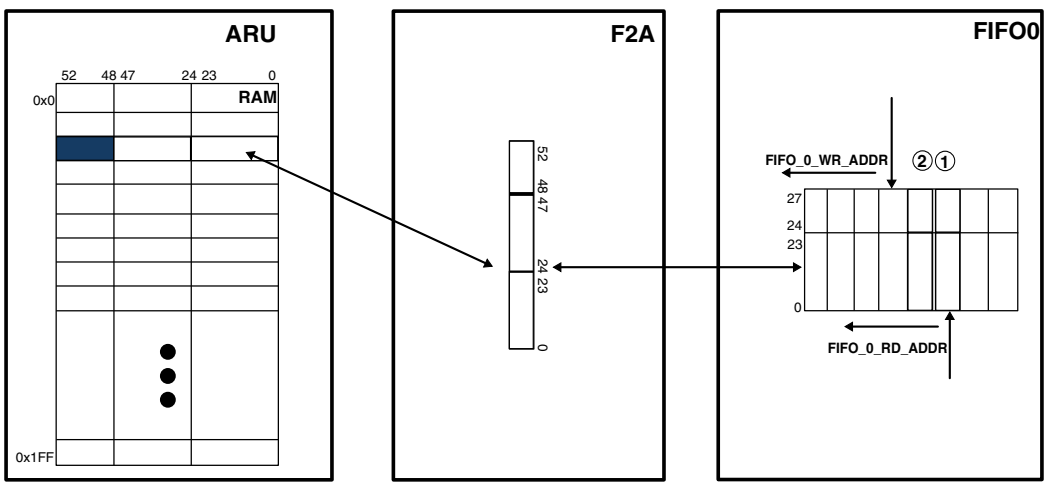


Figure 7-1. Transfers between FIFO and ARU

When reading from the ARU, the F2A first writes the lower data word (bits 0:23) plus control bits (bits 48:52) to the FIFO. Then, if space is available in the FIFO, the F2A writes the higher data word (bits 24:47) plus control bits (Bits 48:52) to the FIFO. Control bits 48:52 from the ARU are duplicated in control bits 24:28 of both FIFO words.

When writing to the ARU, the F2A first reads the lower 24-bit data word (bits 0:23) from the FIFO and writes it to ARU bits 0:23. Then the F2A reads the second 24-bit data word (bits 0:23) plus five control bits (bits 24:28) from the FIFO and:

- writes the 24-bit FIFO data word to ARU bits 24:47, and
- writes the five FIFO control bits to ARU bits 48:52.

7.3 Memory Map and Registers

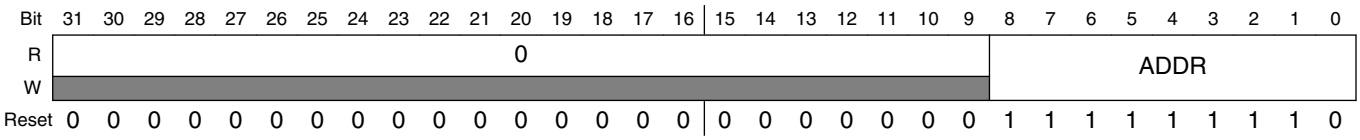
The FIFO to ARU Unit (F2A0) configuration registers are described as follows:

F2A_0 memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	F2A0 Read Channel n Address Register, n=0:7 (F2A_0_CH0_ARU_RD_FIFO)	32	R/W	0000_01FEh	7.3.1/130
4	F2A0 Read Channel n Address Register, n=0:7 (F2A_0_CH1_ARU_RD_FIFO)	32	R/W	0000_01FEh	7.3.1/130
8	F2A0 Read Channel n Address Register, n=0:7 (F2A_0_CH2_ARU_RD_FIFO)	32	R/W	0000_01FEh	7.3.1/130
C	F2A0 Read Channel n Address Register, n=0:7 (F2A_0_CH3_ARU_RD_FIFO)	32	R/W	0000_01FEh	7.3.1/130
10	F2A0 Read Channel n Address Register, n=0:7 (F2A_0_CH4_ARU_RD_FIFO)	32	R/W	0000_01FEh	7.3.1/130
14	F2A0 Read Channel n Address Register, n=0:7 (F2A_0_CH5_ARU_RD_FIFO)	32	R/W	0000_01FEh	7.3.1/130
18	F2A0 Read Channel n Address Register, n=0:7 (F2A_0_CH6_ARU_RD_FIFO)	32	R/W	0000_01FEh	7.3.1/130
1C	F2A0 Read Channel n Address Register, n=0:7 (F2A_0_CH7_ARU_RD_FIFO)	32	R/W	0000_01FEh	7.3.1/130
20	F2A0 Channel n Stream Configuration Register, n=0:7 (F2A_0_CH0_STR_CFG)	32	R/W	0000_0000h	7.3.2/130
24	F2A0 Channel n Stream Configuration Register, n=0:7 (F2A_0_CH1_STR_CFG)	32	R/W	0000_0000h	7.3.2/130
28	F2A0 Channel n Stream Configuration Register, n=0:7 (F2A_0_CH2_STR_CFG)	32	R/W	0000_0000h	7.3.2/130
2C	F2A0 Channel n Stream Configuration Register, n=0:7 (F2A_0_CH3_STR_CFG)	32	R/W	0000_0000h	7.3.2/130
30	F2A0 Channel n Stream Configuration Register, n=0:7 (F2A_0_CH4_STR_CFG)	32	R/W	0000_0000h	7.3.2/130
34	F2A0 Channel n Stream Configuration Register, n=0:7 (F2A_0_CH5_STR_CFG)	32	R/W	0000_0000h	7.3.2/130
38	F2A0 Channel n Stream Configuration Register, n=0:7 (F2A_0_CH6_STR_CFG)	32	R/W	0000_0000h	7.3.2/130
3C	F2A0 Channel n Stream Configuration Register, n=0:7 (F2A_0_CH7_STR_CFG)	32	R/W	0000_0000h	7.3.2/130
40	F2A0 Stream Activation Register (F2A_0_ENABLE)	32	R/W	0000_0000h	7.3.3/131

7.3.1 F2A0 Read Channel n Address Register, n=0:7 (F2A_0_CH*n*_ARU_RD_FIFO)

Address: 1_8080h base + 0h offset + (4d × i), where i=0d to 7d



F2A_0_CH*n*_ARU_RD_FIFO field descriptions

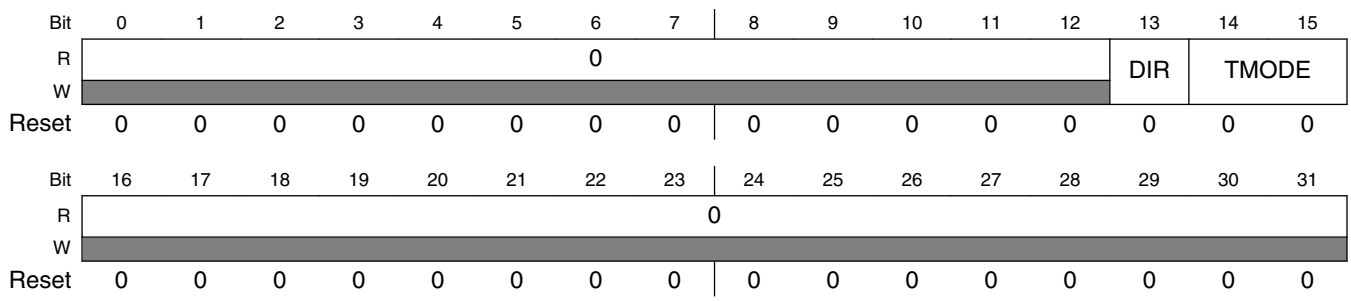
Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ADDR	ARU Read address. **NOTE:** This bit field is only writeable if channel is disabled.

7.3.2 F2A0 Channel n Stream Configuration Register, n=0:7 (F2A_0_CH*n*_STR_CFG)

NOTE

The write protected bits 13:15 are writable only if the corresponding enable bit F2A0_ENABLE[STRx_EN] is cleared.

Address: 1_8080h base + 20h offset + (4d × i), where i=0d to 7d



F2A_0_CH*n*_STR_CFG field descriptions

Field	Description
0–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

F2A_0_CH*n*_STR_CFG field descriptions (continued)

Field	Description
13 DIR	Data transfer direction. 0 Transport from ARU to FIFO 1 Transport from FIFO to ARU.
14–15 TMODE	Transfer mode for 53 bit ARU data from/to FIFO. 00 Transfer low word (ARU bits 23:0) from/to FIFO. 01 high word (ARU bits 47:24) from/to FIFO. 10 Transfer both words from/to FIFO. 11 Reserved.
16–31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

7.3.3 F2A0 Stream Activation Register (F2A_0_ENABLE)

Address: 1_8080h base + 40h offset = 1_80C0h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	STR7_EN		STR6_EN		STR5_EN		STR4_EN		STR3_EN		STR2_EN		STR1_EN		STR0_EN	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

F2A_0_ENABLE field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–17 STR7_EN	Enable/disable stream 7. Write of following double bit values is possible: 00 = Don’t care, bits 16:17 will not be changed. 01 =Stream 7 is disabled and internal states are reset. 10 = Stream 7 is enabled. 11 = Don’t care, bits 16:17 will not be changed Read of following double values means: 00 = Stream 7 disabled 11 = Stream 7 enabled
18–19 STR6_EN	Enable/disable stream 6. Write of following double bit values is possible:

Table continues on the next page...

F2A_0_ENABLE field descriptions (continued)

Field	Description
	00 = Don't care, bits 18:19 will not be changed. 01 =Stream 6 is disabled and internal states are reset. 10 = Stream 6 is enabled. 11 = Don't care, bits 18:19 will not be changed Read of following double values means: 00 = Stream 6 disabled 11 = Stream 6 enabled
20–21 STR5_EN	Enable/disable stream 5. Write of following double bit values is possible: 00 = Don't care, bits 20:21 will not be changed. 01 =Stream 5 is disabled and internal states are reset. 10 = Stream 5 is enabled. 11 = Don't care, bits 20:21 will not be changed Read of following double values means: 00 = Stream 5 disabled 11 = Stream 5 enabled
22–23 STR4_EN	Enable/disable stream 4. Write of following double bit values is possible: 00 = Don't care, bits 22:23 will not be changed. 01 =Stream 4 is disabled and internal states are reset. 10 = Stream 4 is enabled. 11 = Don't care, bits 22:23 will not be changed Read of following double values means: 00 = Stream 4 disabled 11 = Stream 4 enabled
24–25 STR3_EN	Enable/disable stream 3. Write of following double bit values is possible: 00 = Don't care, bits 24:25 will not be changed. 01 =Stream 3 is disabled and internal states are reset. 10 = Stream 3 is enabled. 11 = Don't care, bits 24:25 will not be changed Read of following double values means: 00 = Stream 3 disabled 11 = Stream 3 enabled
26–27 STR2_EN	Enable/disable stream 2. Write of following double bit values is possible:

Table continues on the next page...

F2A_0_ENABLE field descriptions (continued)

Field	Description
	00 = Don't care, bits 26:27 will not be changed. 01 =Stream 2 is disabled and internal states are reset. 10 = Stream 2 is enabled. 11 = Don't care, bits 26:27 will not be changed Read of following double values means: 00 = Stream 2 disabled 11 = Stream 2 enabled
28–29 STR1_EN	Enable/disable stream 1. Write of following double bit values is possible: 00 = Don't care, bits 28:29 will not be changed. 01 =Stream 1 is disabled and internal states are reset. 10 = Stream 1 is enabled. 11 = Don't care, bits 28:29 will not be changed Read of following double values means: 00 = Stream 1 disabled 11 = Stream 1 enabled
30–31 STR0_EN	STR0_EN Write of following double bit values is possible: 00 = Don't care, bits 30:31 will not be changed. 01 =Stream 0 is disabled and internal states are reset. 10 = Stream 0 is enabled. 11 = Don't care, bits 30:31 will not be changed Read of following double values means: 00 = Stream 0 disabled 11 = Stream 0 enabled

Chapter 8
Clock Management Unit (CMU)

8.1 CMU Overview

The CMU has three subunits that generate clocks or clock enable signals for GTM counters and registers:

- External Clock Generation Unit (EGU),
- Configurable Clock Generation Unit (CFGU), and
- Fixed Clock Generation Unit (FXU).

Figure 8-1 shows a block diagram of the CMU.

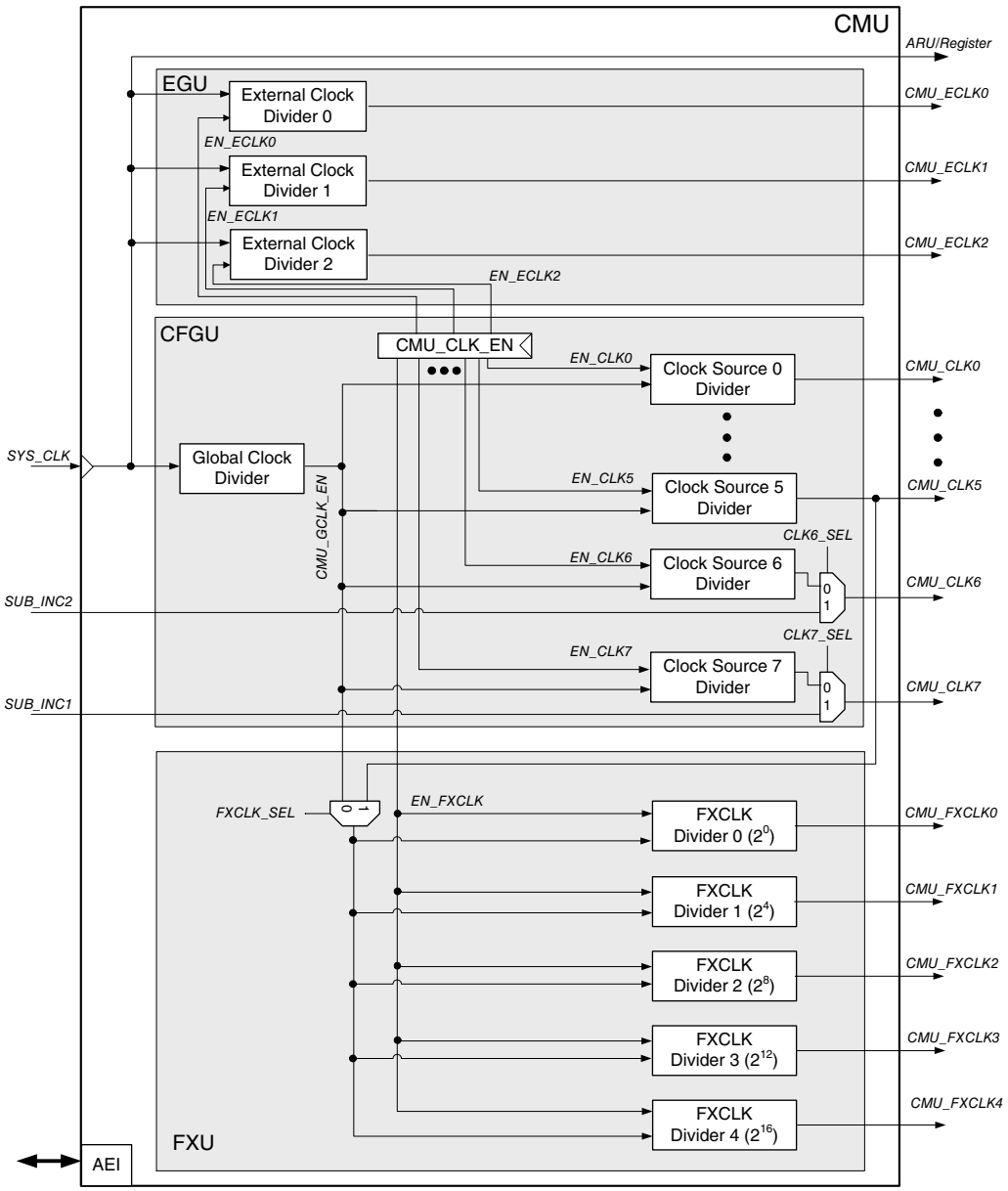


Figure 8-1. CMU block diagram

The Configurable Clock Generation Unit (CFGU) generates up to eight dedicated *CMU_CLKn* (n: 0..7) clock enable signals for the TIM, ATOM, TBU, and MON submodules, where each submodule can choose an arbitrary clock source to specify wide-ranging time bases.

The Fixed Clock Generation Unit (FXU) generates predefined, non-configurable *CMU_FXCLKn* (n: 0..4) clocks for the TOM and MON submodules.

NOTE

Clock enable signals can be used as clocks in some submodules. Conversely, clock signals can be used as clock enable signals in some modules. For example, the TOM submodule uses the CMU_FXCLKn signal as a clock, but also uses it as a clock enable for PWM generation.

The *CMU_CLKn* clock enables and CMU_FXCLKn clocks are derived from the *CMU_GCLK_EN* signal, which is generated by the Global Clock Divider (GCD) using division factors 2^0, 2^4, 2^8, 2^{12}, and 2^{16}.

The External Clock Generation Unit (EGU) generates up to three *CMU_ECLKn* (n: 0..2) clocks, where each clock has a duty cycle of approximately 50%.

SYS_CLK is the clock used by all GTM_IP registers.The *CMU_CLKn* (n: 0..7) and *CMU_FXCLKn (n:0..4)* signals are used as enable signals for all GTM_IP registers.

The four configurable clock enable signals, *CMU_CLK0*, *CMU_CLK1, CMU_CLK6* and *CMU_CLK7,* are used to enable the TIM filter counters.

8.2 Global Clock Divider

The Global Clock Divider (GCD) is used to divide the GTM global input clock, *SYS_CLK*, into a common subdivided clock signal, **CMU_GCLK_EN**. The CMU_CLK_EN block, clocked by **SYS_CLK**, distributes EN_CLKn (n:0..7) signals and EN_FXCLKn (n:0..4) signals to the clock source dividers and FXCLK dividers, respectively. These signals have the frequency of SYS_CLK. The dividers generate output clocks at user-specified frequencies.

The fractional divider, *Z/N,* in the equation

$T_{GCMU\_EN} = (Z / N) * T_{SYS\_CLK}$

is calculated by the following algorithm with *Z:CMU_GCLK_NUM(23:0); N:CMU_GCLK_DEN(23:0)*; *Z,N >0*:

1. Set remainder (*R*), operand1 (*OP1*) and operand2 (*OP2*) register during the initialization phase (with implicit conversion to signed):

   R = Z, OP1 = N, OP2 = N − Z;

2. After leaving the initialization phase and at least one **CMU_CLKn** has been generated, the sign of the remainder (R) for each *SYS_CLK* cycle will be checked:

- If R > 0, keep updating remainder and keep CMU_GCLK_EN = 0: R = R − OP1.
- If R < 0, update remainder and set CMU_GCLK_EN = 1: R = R − OP2

After at most (*Z / N*+1) subtractions, there will be a negative remainder *(R)* and an active phase of the generated clock enable (for one cycle) will be allowed. The remainder *(R)* is a measure for the distance to a real *Z / N* clock and will be regarded for the next generated clock enable cycle phase. The new remainder *(R)* value will be *R = R + (Z − N)*. In the worst case, the remainder *(R)* will sum up to an additional cycle in the generated clock enable period after *Z*-cycles. In other cases, equally distributed additional cycles will be inserted for the generated clock enable. If *Z* is an integer multiple of *N,* no additional cycles will be included for the generated clock enable.

For better resource sharing, all arithmetic has been reduced to subtractions and the initialization of the remainder *R* uses the complement of *Z − N*.

8.3 Configurable Clock Generation Subunit (CFGU)

The CFGU provides up to eight configurable clock divider blocks that divide the common **C***MU_GCLK_EN* signal into eight dedicated enable signals for the GTM submodules.

The duration of the eight dedicated output signals, CMU_CLKn (n: 0…7), depends on the configuration of the global clock enable signal, *CMU_GCLK_EN*, and the configuration of each Clock Source n Divider (which is accomplished by writing to the CMU_CLK_n_CTRL[CLK_CNT] bit field).

The frequency, $f_x = 1 / T_x$, of the corresponding **CMU_CLKn** output signal is determined by the unsigned representation the CMU_CLK_n_CTRL[CLK_CNT] bit field in equation:

$T_{CMU\_CLKn} = (CLK\_CNTn + 1) * T_{CMU\_GCLK\_EN}$.

The **CMU_CLK0** through CMU_CLK5 enable signals can be individually enabled by setting the corresponding CMU_CLK_EN[EN_CLKn] bit fields. The *CMU_CLK6* and *CMU_CLK7* enable signals are enabled by either the corresponding CMU_CLK_EN[EN_CLKn] bit fields or by the **CMU_CLK_6_CTRL[CLK6_SEL]** and **CMU_CLK_7_CTRL[CLK7_SEL]** bits, respectively. When the CLK6_SEL bit is 0, the divider sources the **CMU_CLK6** enable signal; when the CLK6_SEL bit is 1, the SUB_INC2 signal from the DPLL sources the CMU_CLK6 enable signal. When the CLK7_SEL bit is 0, the divider sources the **CMU_CLK7** enable signal; when the CLK7_SEL bit is 1, the SUB_INC1 signal from the DPLL sources the CMU_CLK7 enable signal.

To avoid unexpected hardware behavior, the configuration of a **CMU_CLK_n_CTRL** register can be changed only when the corresponding *CMU_CLKn* output clock is disabled. Furthermore, any changes to the **CMU_GCLK_NUM** and **CMU_GCLK_DEN** registers can be performed only when the *CMU_CLKn* output clocks are disabled and the **CMU_CLK_EN**[EN_FXCLK] bit is disabled.

The clock source signals, CMU_CLKn (n: 0..7) and CMU_FXCLK[y] are implemented in the form of enable signals for the corresponding registers, which means that the actual clock signals of all registers always uses the SYS_CLK signal.

All *CMU_CLKn* enable signals, which are enabled simultaneously by writing to the **CMU_CLK_EN[EN_CLKn]** bits, are synchronized to each other.

8.4 Wave Form of Generated Clock Signal CMU_CLKn

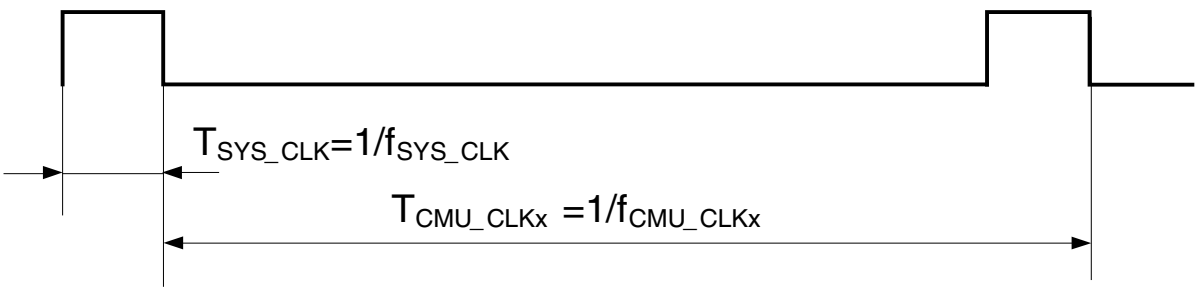


Figure 8-2. CMU_CLKn generated wave form

8.5 Fixed Clock Generation (FXU)

The FXU subunit generates fixed clock enables out of either the CMU_GCLK_EN or one of the eight CMU_CLK[x] enable signals depending on the CMU_FXCLK_CTRL[FXCLK_SEL] bit field. These clock enables are used for PWM generation inside the TOM submodules.

All *CMU_FXCLKn* clock enables can be simultaneously enabled or disabled by setting the appropriate **CMU_CLK_EN[EN_FXCLK]** bit fields.

The dividing factors are defined as 2^0, 2^4, 2^8, 2^{12}, and 2^{16}. The *CMU_FXCLKn* signals are implemented as enable signals for the corresponding registers.

8.6 External Generation Unit (EGU)

The EGU generates up to three separate *CMU_ECLKn* (n: 0..2) output clocks.

Each of these output clocks is derived from the corresponding External Clock n Divider, which is run by the *SYS_CLK* input clock.

In contrast to the *CMU_CLKn* and *CMU_FXCLKn* output signals, which are used as enable signals for registers, the *CMU_ECLKn* output clocks have a duty cycle of about 50% and are used as true clocks for external peripheral components.

Each of the CMU_ECLKn clocks is enabled or disabled by writing to the corresponding **CMU_CLK_EN**[EN_ECLKn] bit fields.

The frequencies, $f_{CMU\_ECLKn} = 1 / T_{CMU\_ECLKn,}$ of the external clocks are controlled by the **CMU_ECLK_n_NUM** and **CMU_ECLK_n_DEN** registers, where the equation

$T_{CMU\_ECLKn} = 2 * (\textbf{ECLKn\_NUM / ECLKn\_DEN}) * T_{SYS\_CLK}$

is implemented by the following algorithm, where *Z: CMU_ECLK_n_NUM(23:0) ; N: CMU_ECLK_n_DEN(23:0)* ; *Z,N > 0* ; *Z >= N ; CMU_ECLKn* = 0:

1. Set remainder (*R*), operand1 (*OP1*) and operand2 (*OP2*) registers during the initialization phase (with implicit conversion to signed):

   R = Z, OP1 = N, OP2 = N − Z.

2. After leaving the initialization phase (*CMU_ECLKn* clock has been enabled), the sign of remainder (R) for each *SYS_CLK* cycle will be checked:
   - If *R > 0,* keep updating remainder and keep the *CMU_ECLKn* clock enabled: *R = R − OP1*.

     If *R < 0,* update the remainder and toggle the enable of the *CMU_ECLKn* clock: *R = R − OP2*.

After at most *(Z / N*+1) subtractions, there will be a negative and an active phase of the generated clock enable will be triggered for one cycle. The remainder *R* is a measure of the distance to a real *Z / N* clock and it will be regarded for the next generated clock toggle phase. The new *R* value will be *R = R + (Z − N)*. Worst case, the remainder *R* will sum up to an additional cycle in the generated clock toggle period after *Z*-cycles. In other cases, equally distributed additional cycles will be inserted for the generated clock toggle. If *Z* is an integer multiple of *N*, no additional cycles will be included for the generated clock toggle.

For better resource sharing, all arithmetic has been reduced to subtractions, and the initialization of the remainder R uses the complement of $(Z - N)$.

The default value of a *CMU_ECLKn* clock output is 0..

8.7 Memory Map and Registers

The Clock Management Unit (CMU) configuration registers are described as follows:

CMU memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	CMU Clock Enable Register (CMU_CLK_EN)	32	R/W	0000_0000h	8.7.1/142
4	Global Clock Control Numerator Register (CMU_GCLK_NUM)	32	R/W	0000_0001h	8.7.2/146
8	Global Clock Control Denominator Register (CMU_GCLK_DEN)	32	R/W	0000_0001h	8.7.3/146
C	Control for Clock Source n register (CMU_CLK_0_CTRL)	32	R/W	0000_0000h	8.7.4/147
10	Control for Clock Source n register (CMU_CLK_1_CTRL)	32	R/W	0000_0000h	8.7.4/147
14	Control for Clock Source n register (CMU_CLK_2_CTRL)	32	R/W	0000_0000h	8.7.4/147
18	Control for Clock Source n register (CMU_CLK_3_CTRL)	32	R/W	0000_0000h	8.7.4/147
1C	Control for Clock Source n register (CMU_CLK_4_CTRL)	32	R/W	0000_0000h	8.7.4/147
20	Control for Clock Source n register (CMU_CLK_5_CTRL)	32	R/W	0000_0000h	8.7.4/147
24	Control for Clock Source 6 register (CMU_CLK_6_CTRL)	32	R/W	0000_0000h	8.7.5/147
28	Control for Clock Source 7 register (CMU_CLK_7_CTRL)	32	R/W	0000_0000h	8.7.6/148
2C	CMU External Clock n Numerator Register (CMU_ECLK0_NUM)	32	R/W	0000_0001h	8.7.7/149
30	CMU External Clock n Denominator Register (CMU_ECLK0__DEN)	32	R/W	0000_0001h	8.7.8/150
34	CMU External Clock n Numerator Register (CMU_ECLK1_NUM)	32	R/W	0000_0001h	8.7.7/149
38	CMU External Clock n Denominator Register (CMU_ECLK1__DEN)	32	R/W	0000_0001h	8.7.8/150
3C	CMU External Clock n Numerator Register (CMU_ECLK2_NUM)	32	R/W	0000_0001h	8.7.7/149
40	CMU External Clock n Denominator Register (CMU_ECLK2__DEN)	32	R/W	0000_0001h	8.7.8/150
44	CMU FX Clock Control register (CMU_FXCLK_CTRL)	32	R/W	0000_0000h	8.7.9/150

8.7.1 CMU Clock Enable Register (CMU_CLK_EN)

NOTE

Any read access to an EN_CLKx, EN_ECLKz or EN_FXCLK 2-bit field will always result in a value of 00 or 11, indicating the current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored.

NOTE

Any disabling to EN_CLKx will be reset internal counters for configurable clocks.

NOTE

Any disabling to EN_ECLKz will be reset internal counters for external clocks.

NOTE

An enable to EN_FXCLK from disable state will be reset internal fixed clock counters.

Address: 300h base + 0h offset = 300h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0								EN_FXCLK		EN_ECLK2		EN_ECLK1		EN_ECLK0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	EN_CLK7		EN_CLK6		EN_CLK5		EN_CLK4		EN_CLK3		EN_CLK2		EN_CLK1		EN_CLK0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMU_CLK_EN field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–9 EN_FXCLK	Enable all CMU_FXCLK. **NOTE:** An enable to EN_FXCLK from disable state will reset internal fixed clock counters. **NOTE:** Any read access to an EN_CLK[x], EN_ECLK[z] or EN_FXCLK bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. **NOTE:** Any disabling to EN_CLK[x] will be reset internal counters for configurable clocks. 00 Clock source is disabled (ignores write access). 01 Disable clock signal and reset internal states.

Table continues on the next page...

CMU_CLK_EN field descriptions (continued)

Field	Description
	10 Enable clock signal. 11 Clock signal enabled (ignores write access).
10–11 EN_ECLK2	Enable ECLK 2 generation subunit. **NOTE:** Any read access to an EN_CLK[x], EN_ECLK2 or EN_FXCLK bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. **NOTE:** Any disabling to EN_CLK[x] will be reset internal counters for configurable clocks. 00 Clock source is disabled (ignores write access). 01 Disable clock signal and reset internal states. 10 Enable clock signal. 11 Clock signal enabled (ignores write access).
12–13 EN_ECLK1	Enable ECLK 1 generation subunit **NOTE:** Any read access to an EN_CLK[x], EN_ECLK1 or EN_FXCLK bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. **NOTE:** Any disabling to EN_CLK[x] will be reset internal counters for configurable clocks. 0 Clock source is disabled (ignores write access). 0 Disable clock signal and reset internal states. 0 Enable clock signal. 0 Clock signal enabled (ignores write access).
14–15 EN_ECLK0	Enable ECLK 0 generation subunit. **NOTE:** Any read access to an EN_CLK[x], EN_ECLK0 or EN_FXCLK bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. **NOTE:** Any disabling to EN_CLK[x] will be reset internal counters for configurable clocks. 00 Clock source is disabled (ignores write access). 01 Disable clock signal and reset internal states. 10 Enable clock signal. 11 Clock signal enabled (ignores write access).
16–17 EN_CLK7	Enable clock source 7. **NOTE:** Any read access to an EN_CLK7, EN_ECLK[z] or EN_FXCLK bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. **NOTE:** Any disabling to EN_CLK7 will be reset internal counters for configurable clocks. 00 Clock source is disabled (ignores write access). 01 Disable clock signal and reset internal states. 10 Enable clock signal. 11 Clock signal enabled (ignores write access).

Table continues on the next page...

CMU_CLK_EN field descriptions (continued)

Field	Description
18–19 EN_CLK6	Enable clock source 6. **NOTE:** Any read access to an EN_CLK6, EN_ECLK[z] or EN_FXCLK bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. **NOTE:** Any disabling to EN_CLK6 will be reset internal counters for configurable clocks. 00 Clock source is disabled (ignores write access). 01 Disable clock signal and reset internal states. 10 Enable clock signal. 11 Clock signal enabled (ignores write access).
20–21 EN_CLK5	Enable clock source 5. **NOTE:** Any read access to an EN_CLK5, EN_ECLK[z] or EN_FXCLK bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. **NOTE:** Any disabling to EN_CLK5 will be reset internal counters for configurable clocks. 00 Clock source is disabled (ignores write access). 01 Disable clock signal and reset internal states. 10 Enable clock signal. 11 Clock signal enabled (ignores write access).
22–23 EN_CLK4	Enable clock source 4. **NOTE:** Any read access to an EN_CLK4, EN_ECLK[z] or EN_FXCLK bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. **NOTE:** Any disabling to EN_CLK4 will be reset internal counters for configurable clocks. 00 Clock source is disabled (ignores write access). 01 Disable clock signal and reset internal states. 10 Enable clock signal. 11 Clock signal enabled (ignores write access).
24–25 EN_CLK3	Enable clock source 3 **NOTE:** Any read access to an EN_CLK3, EN_ECLK[z] or EN_FXCLK bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. **NOTE:** Any disabling to EN_CLK3 will be reset internal counters for configurable clocks. 00 Clock source is disabled (ignores write access). 01 Disable clock signal and reset internal states. 10 Enable clock signal. 11 Clock signal enabled (ignores write access).
26–27 EN_CLK2	Enable clock source 2.

Table continues on the next page...

CMU_CLK_EN field descriptions (continued)

Field	Description
	NOTE: Any read access to an EN_CLK2, EN_ECLK[z] or EN_FXCLK bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. **NOTE:** Any disabling to EN_CLK2 will be reset internal counters for configurable clocks. 00 Clock source is disabled (ignores write access). 01 Disable clock signal and reset internal states. 10 Enable clock signal. 11 Clock signal enabled (ignore write access).
28–29 EN_CLK1	Enable clock source 1 **NOTE:** Any read access to an EN_CLK1, EN_ECLK[z] or EN_FXCLK bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. **NOTE:** Any disabling to EN_CLK1 will be reset internal counters for configurable clocks. 00 Clock source is disabled (ignores write access). 01 Disable clock signal and reset internal states. 10 Enable clock signal. 11 Clock signal enabled (ignores write access).
30–31 EN_CLK0	Enable clock source 0. **NOTE:** Any read access to an EN_CLK0, EN_ECLK[z] or EN_FXCLK bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. **NOTE:** Any disabling to EN_CLK0 will be reset internal counters for configurable clocks. 00 Clock source is disabled (ignores write access). 01 Disable clock signal and reset internal states. 10 Enable clock signal. 11 Clock signal enabled (ignores write access)

8.7.2 Global Clock Control Numerator Register (CMU_GCLK_NUM)

NOTE

The CMU hardware alters the content of CMU_GCLK_NUM and CMU_GCLK_DEN automatically to 0x1, if CMU_GCLK_NUM is specified less than CMU_GCLK_DEN or one of the values is specified with a value zero. Thus, a secure way for altering the values is writing twice to the register CMU_GCLK_NUM followed by a single write to register CMU_GCLK_DEN.

Address: 300h base + 4h offset = 304h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								GCLK_NUM																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

CMU_GCLK_NUM field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 GCLK_NUM	Numerator for global clock divider. Defines numerator of the fractional divider. **NOTE:** Value can only be modified when all clock enables EN_CLK[n] and the EN_FXCLK are disabled.

8.7.3 Global Clock Control Denominator Register (CMU_GCLK_DEN)

Address: 300h base + 8h offset = 308h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								GCLK_DEN																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

CMU_GCLK_DEN field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 GCLK_DEN	Denominator for global clock divider.

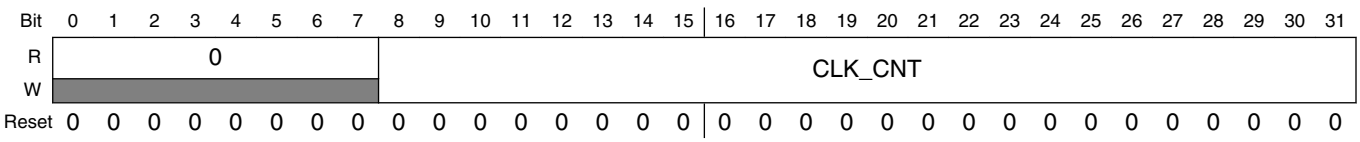
Table continues on the next page...

CMU_GCLK_DEN field descriptions (continued)

Field	Description
	Defines denominator of the fractional divider. **NOTE:** Value can only be modified when all clock enables EN_CLK[n] and the EN_FXCLK are disabled.

8.7.4 Control for Clock Source n register (CMU_CLK_*n*_CTRL)

Address: 300h base + Ch offset + (4d × i), where i=0d to 5d

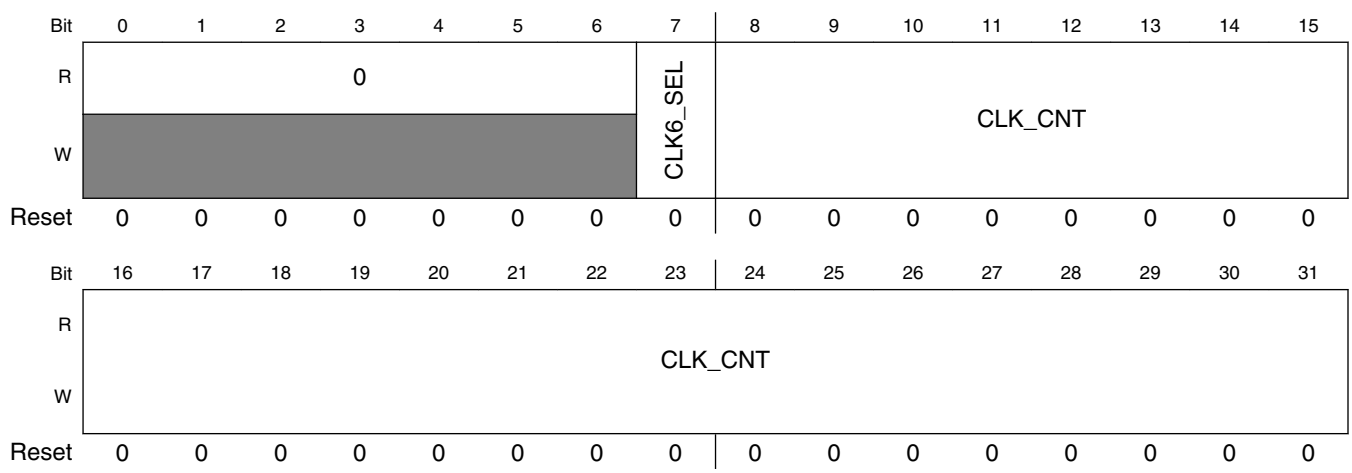


CMU_CLK_*n*_CTRL field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 CLK_CNT	Clock count. Defines count value for the clock divider of clock source CMU_CLK[0:4]. **NOTE:** Value can only be modified when corresponding clock enable EN_CLK[0:5] is disabled.

8.7.5 Control for Clock Source 6 register (CMU_CLK_6_CTRL)

Address: 300h base + 24h offset = 324h

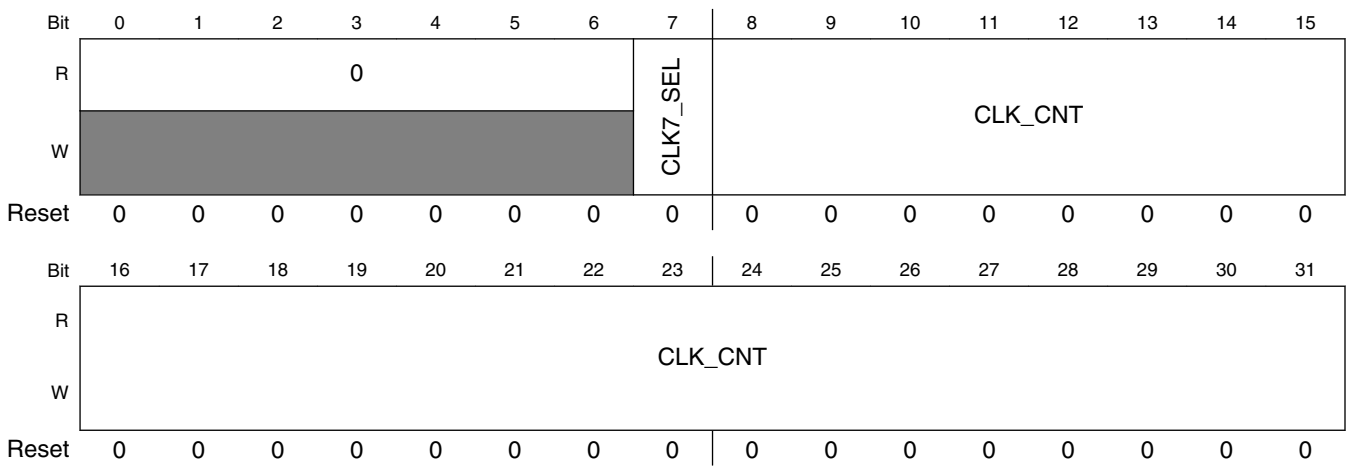


CMU_CLK_6_CTRL field descriptions

Field	Description
0–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CLK6_SEL	Clock source selection for CMU_CLK6. **NOTE:** Value can be modified only when EN_CLK6 is disabled. 0 Use Clock Source 6 Divider. 1 Use signal SUB_INC2 of submodule DPLL.
8–31 CLK_CNT	Clock count. Define count value for the clock divider of clock source CMU_CLK6. **NOTE:** Value can be modified only when EN_CLK6 is disabled.

8.7.6 Control for Clock Source 7 register (CMU_CLK_7_CTRL)

Address: 300h base + 28h offset = 328h



CMU_CLK_7_CTRL field descriptions

Field	Description
0–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CLK7_SEL	Clock source selection for CMU_CLK7. **NOTE:** Value can be modified only when EN_CLK7 is disabled. 0 Use Clock Source 7 Divider. 1 Use signal SUB_INC1 of submodule DPLL.
8–31 CLK_CNT	Clock count. Define count value for the clock divider of clock source CMU_CLK7.

Table continues on the next page...

CMU_CLK_7_CTRL field descriptions (continued)

Field	Description
	NOTE: Value can be modified only when EN_CLK7 is disabled.

8.7.7 CMU External Clock n Numerator Register (CMU_ECLK*n*_NUM)

NOTE

The CMU hardware alters the content of CMU_ECLK[n]_NUM and CMU_ECLK[n]_DEN automatically to 0x1if CMU_ECLK[n]_NUM is specified less than CMU_ECLK_[n]_DEN or one of the values is specified with a value zero. Thus, a secure way for altering the values is writing twice to the register CMU_ECLK[n]_NUM, followed by a single write to register CMU_ECLK[n]_DEN.

Address: 300h base + 2Ch offset + (8d × i), where i=0d to 2d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								ECLK_NUM																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

CMU_ECLK*n*_NUM field descriptions

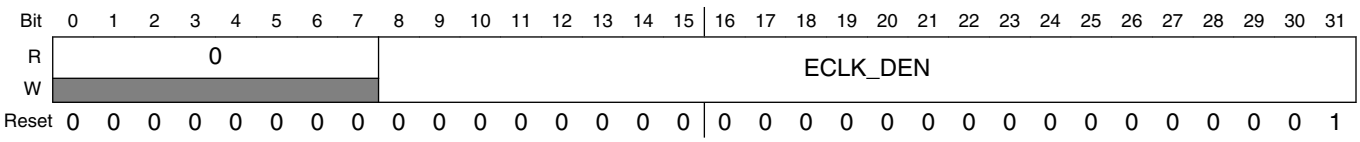
Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 ECLK_NUM	Numerator for external clock divider. Defines numerator of the fractional divider. **NOTE:** Value can only be modified when corresponding clock enable EN_ECLK[n] is disabled.

8.7.8 CMU External Clock n Denominator Register (CMU_ECLK*n*__DEN)

NOTE

The CMU hardware alters the content of CMU_ECLK[n]_NUM and CMU_ECLK[n]_DEN automatically to 0x1 if CMU_ECLK[n]_NUM is specified less than CMU_ECLK[n]_DEN or one of the values is specified with a value zero. Thus, a secure way for altering the values is writing twice to the register CMU_ECLK[n]_NUM followed by a single write to register CMU_ECLK[n]_DEN.

Address: 300h base + 30h offset + (8d × i), where i=0d to 2d

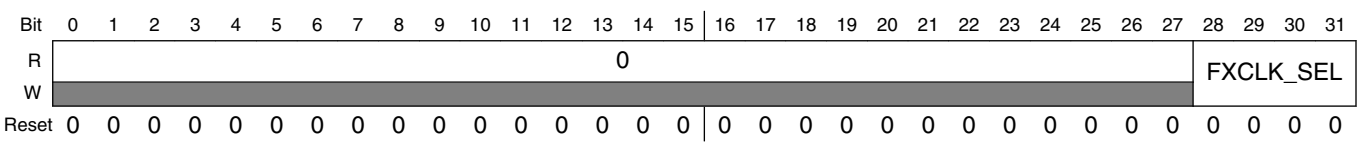


CMU_ECLK*n*__DEN field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 ECLK_DEN	Denominator for external clock divider. Defines denominator of the fractional dividerDefines denominator of the fractional divider **NOTE:** Value can only be modified when corresponding clock enable EN_ECLK[n] is disabled.

8.7.9 CMU FX Clock Control register (CMU_FXCLK_CTRL)

Address: 300h base + 44h offset = 344h



CMU_FXCLK_CTRL field descriptions

Field	Description
0–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

CMU_FXCLK_CTRL field descriptions (continued)

Field	Description
28–31 FXCLK_SEL	Input clock selection for EN_FXCLK line. **NOTE:** This value can only be written, when the CMU_FXCLK generation is disabled. See the CMU_CLK_EN register. **NOTE:** Other values for FXCLK_SEL are reserved and should not be used, but they behave like FXCLK_SEL = 0. 0000 CMU_GCLK_EN selected 0001 CMU_CLK0 selected 0010 CMU_CLK1 selected 0011 CMU_CLK2 selected 0100 CMU_CLK3 selected 0101 CMU_CLK4 selected 0110 CMU_CLK5 selected 0111 CMU_CLK6 selected 1000 CMU_CLK7 selected

Chapter 9
Time Base Unit (TBU)

9.1 TBU Overview

The TBU submodule, shown in Figure 9-1), provides common time bases for the GTM-IP. The TBU has three time base channels:

- Channel 0 uses the 27-bit **TBU_CH0_BASE** time base register, which is configurable to provide either its lower 24 bits or upper 24 bits to the GTM as the 24-bit *TBU_TS0* signal. The TBU_UP0_L signal goes high for one SYSCLK period when the lower 24-bits of the register are being updated; the TBU_UP0_H signal goes high for one SYSCLK period when the upper 24-bits of the register are being updated.
- Channel 1 uses the 24-bit TBU_CH1_BASE time base register, which provides the 24-bit TBU_TS1 signal to the GTM. The *TBU_UP1* signal goes high for a single SYS_CLK period whenever the corresponding signal TBU_TS1 is getting updated. .
- Channel 2 uses the 24-bit TBU_CH1_BASE time base register, which provides the 24-bit TBU_TS2 signal to the GTM. The *TBU_UP2* signal goes high for a single SYS_CLK period whenever the corresponding signal TBU_TS2 is getting updated. .

The TBU_UP0_L signal is set to high for a single SYS_CLK period if TBU_TS0 and TBU_TS0x are getting updated. The TBU_UP0_H signal is set to high for a single SYS_CLK period whenever the upper 24 bits of TBU_TS0 are updated.

NOTE

A fourth TBU time base channel (channel 3) can be implemented. It provides the same functionality as time base channel 1.

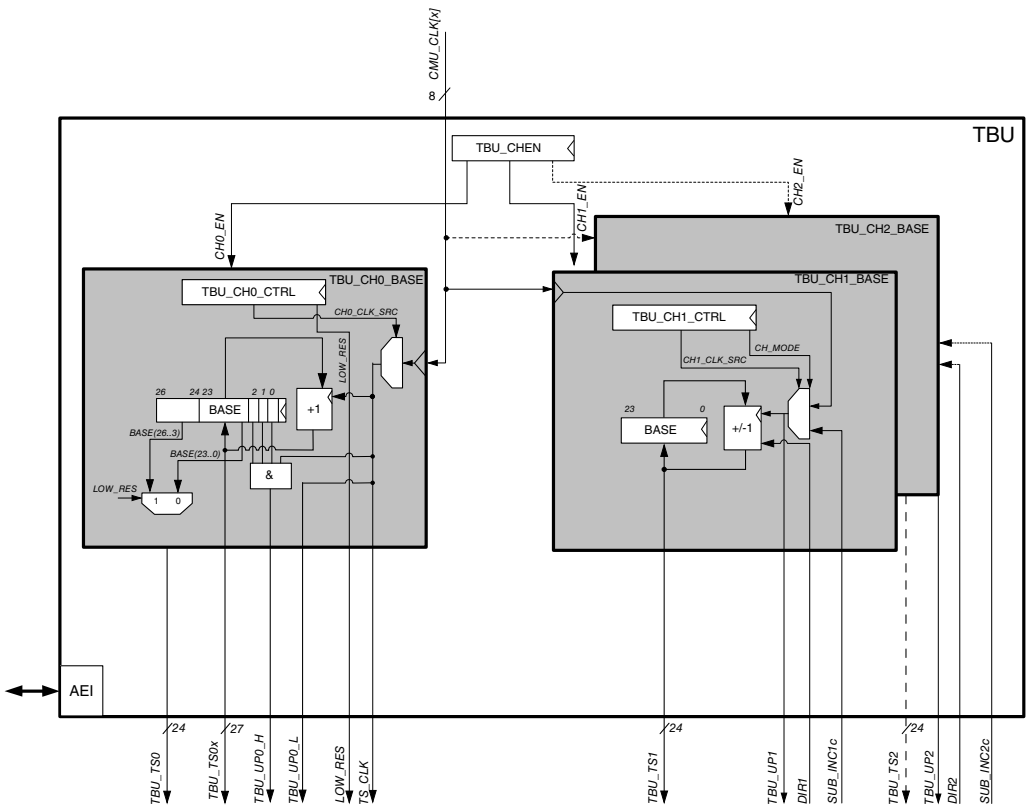


Figure 9-1. TBU block diagram

The time base channels can run independently of each other, and they can be synchronously enabled and disabled by control bit fields in the **TBU_CHEN** register.

Each channel is configurable via the AEI interface. Each channel can be configured to select one of the eight *CMU_CLKn* clock enable signals coming from the CMU submodule as an input clock source.

For channels 1 and 2 an, an additional clock enable signal, *SUB_INC[y]c* (y: 1, 2) coming from the DPLL submodule, can be selected as an input clock enable signal. The *DIRn (n: 1..2)* input signal determines the count direction.

The selected time stamp clock signal for channel 0 is routed to the DPLL submodule via the *TS_CLK* output signal (uses the same source as the TBU_UP0_L output signal).

9.2 TBU Channel Modes

The TBU has three configurable channel modes:

- free running,
- forward, and
- backward.

In all modes, the time base TBU_CHn_BASE registers can be initialized with start values (before enabling the corresponding TBU channels) and they can always be read to monitor current count values.

Channel 0 operates in the free running mode, where the TBU_CH0_BASE register is incremented on every SYS_CLK tick if enabled by the CMU_CLKn input signal.

Channels 1 and 2 can be independently configured to operate in any of the modes:

- Free running mode: the TBU_CH0_BASE register is incremented on every SYS_CLK tick if enabled by the CMU_CLKn input signal.
- Forward mode (DIRn = 0): the TBU_CH0_BASE register is incremented on every SYS_CLK tick if enabled by the SUB_INC[n]c (n:1..2) input signal.
- Backward mode (DIRn = 1): the TBU_CH0_BASE register is decremented on every SYS_CLK tick if enabled by the SUB_INC[n]c (n:1..2) input signal.

9.3 Memory Map and Registers

The Time Base Unit (TBU) configuration registers are described as follows:

TBU memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	TBU Global Channel Enable Register (TBU_CHEN)	32	R/W	0000_0000h	9.3.1/155
4	TBU Channel 0 Control Register (TBU_CH0_CTRL)	32	R/W	0000_0000h	9.3.2/157
8	TBU Channel 0 Base Register (TBU_CH0_BASE)	32	R/W	0000_0000h	9.3.3/158
C	TBU Channel 1 Control Register (TBU_CH1_CTRL)	32	R/W	0000_0000h	9.3.4/158
10	TBU Channel 1 Base Register (TBU_CH1_BASE)	32	R/W	0000_0000h	9.3.5/159
14	TBU Channel 2 Control Register (TBU_CH2_CTRL)	32	R/W	0000_0000h	9.3.6/160
18	TBU Channel 2 Base Register (TBU_CH2_BASE)	32	R/W	0000_0000h	9.3.7/161

9.3.1 TBU Global Channel Enable Register (TBU_CHEN)

Address: 100h base + 0h offset = 100h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0										ENDIS_CH2		ENDIS_CH1		ENDIS_CH0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TBU_CHEN field descriptions

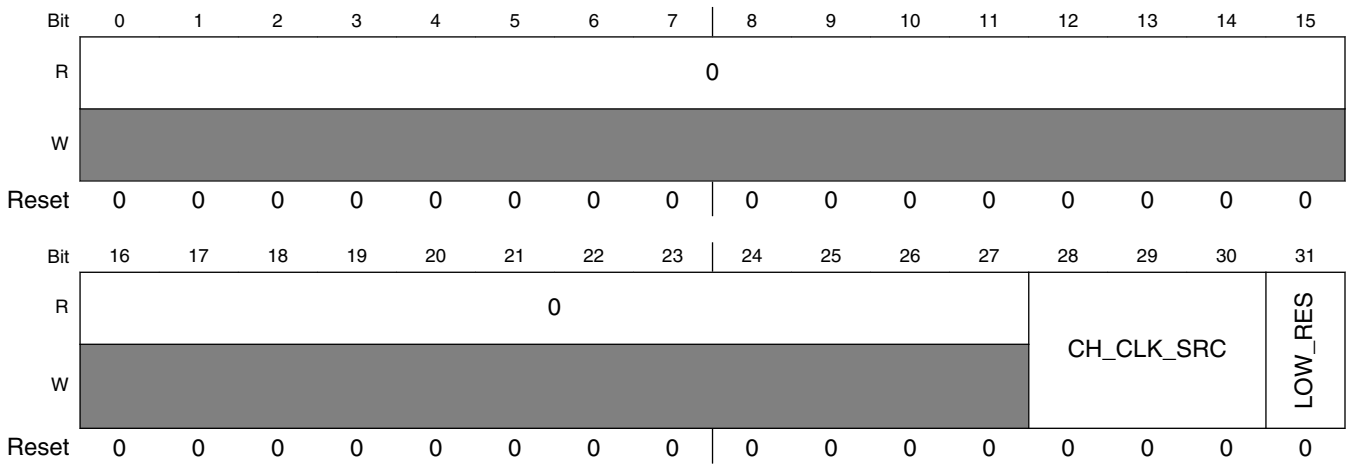
Field	Description
0–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–27 ENDIS_CH2	TBU channel 2 enable/disable control. Write of following double bit values is possible: 00 Don't care, bits 26:27 will not be changed. 01 Channel disabled: is read as 00. 10 Channel enabled: is read as 11. 11 Don't care, bits 26:27 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
28–29 ENDIS_CH1	TBU channel 1 enable/disable control. Write of following double bit values is possible: 00 Don't care, bits 28:29 will not be changed. 01 Channel disabled: is read as 00. 10 Channel enabled: is read as 11. 11 Don't care, bits 28:29 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
30–31 ENDIS_CH0	TBU channel 0 enable/disable control. Write of following double bit values is possible: 00 Don't care, bits 30:31 will not be changed. 01 Channel disabled: is read as 00. 10 Channel enabled: is read as 11. 11 Don't care, bits 30:31 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.

9.3.2 TBU Channel 0 Control Register (TBU_CH0_CTRL)

NOTE

Bit[28:31] can be modified only if channel 0 is disabled.

Address: 100h base + 4h offset = 104h

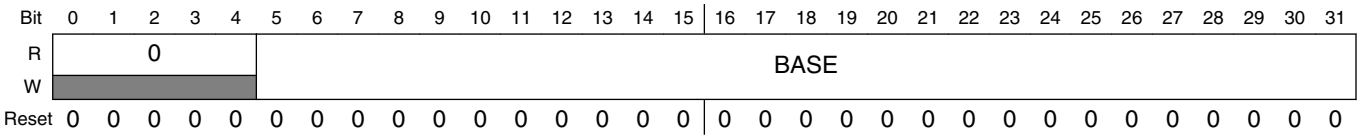


TBU_CH0_CTRL field descriptions

Field	Description
0–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–30 CH_CLK_SRC	Clock source for channel 0. 000 CMU_CLK0 selected. 001 CMU_CLK1 selected. 010 CMU_CLK2 selected. 011 CMU_CLK3 selected. 100 CMU_CLK4 selected. 101 CMU_CLK5 selected. 110 CMU_CLK6 selected. 111 CMU_CLK7 selected.
31 LOW_RES	TBU_CH0_BASE register resolution. **NOTE:** The two resolutions for TBU channel 0 can be used in the TIM channel 0 and the DPLL submodules. **NOTE:** This value can only be modified if TBU channel 0 is disabled. 0 TBU channel uses lower counter bits (bit 0 to 23). 1 TBU channel uses upper counter bits (bit 3 to 26).

9.3.3 TBU Channel 0 Base Register (TBU_CH0_BASE)

Address: 100h base + 8h offset = 108h

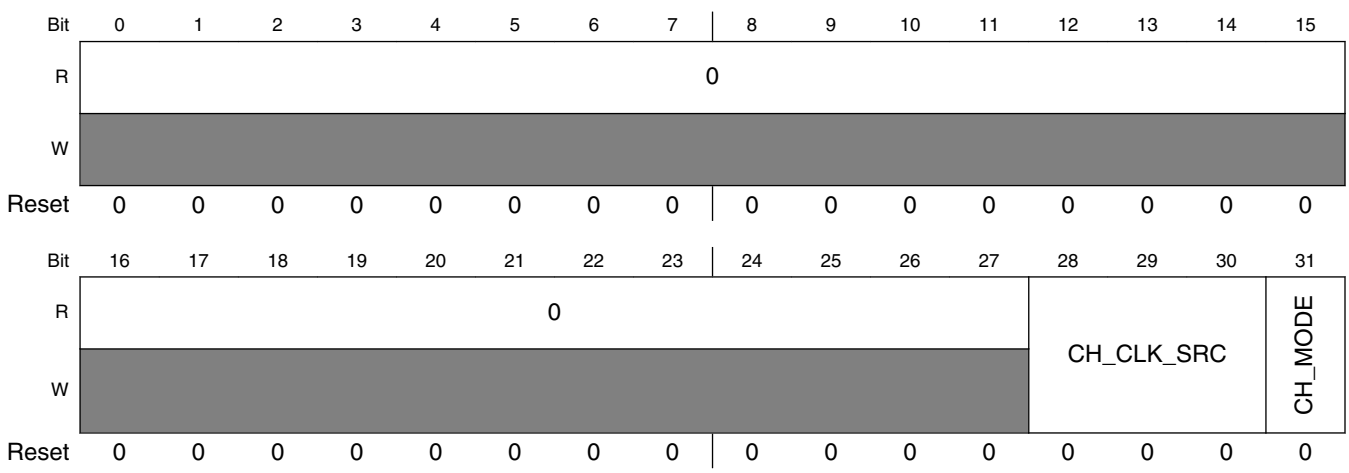


TBU_CH0_BASE field descriptions

Field	Description
0–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–31 BASE	Time base value for channel 0. **NOTE:** The value of BASE can be written only if TBU channel 0 is disabled. **NOTE:** If channel 0 is enabled, a read access to this register provides the current value of the underlying 27 bit counter.

9.3.4 TBU Channel 1 Control Register (TBU_CH1_CTRL)

Address: 100h base + Ch offset = 10Ch



TBU_CH1_CTRL field descriptions

Field	Description
0–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–30 CH_CLK_SRC	Clock source for channel time base counter. **NOTE:** This value can be modified only if this channel is disabled:

Table continues on the next page...

TBU_CH1_CTRL field descriptions (continued)

Field	Description
	000 CMU_CLK0 selected. 001 CMU_CLK1 selected. 010 CMU_CLK2 selected. 011 CMU_CLK3 selected. 100 CMU_CLK4 selected. 101 CMU_CLK5 selected. 110 CMU_CLK6 selected. 111 CMU_CLK7 selected.
31 CH_MODE	Channel mode. **NOTE:** This value can only be modified if this channel is disabled. In Free running counter mode the CMU clock source specified by CH_CLK_SRC is used for the counter. In Forward/Backward counter mode the SUB_INC[n] clock signal in combination with the DIR[n] input signal is used to determine the counter direction and clock frequency. 0 Free running counter mode. 1 Forward/backward counter mode.

9.3.5 TBU Channel 1 Base Register (TBU_CH1_BASE)

Address: 100h base + 10h offset = 110h

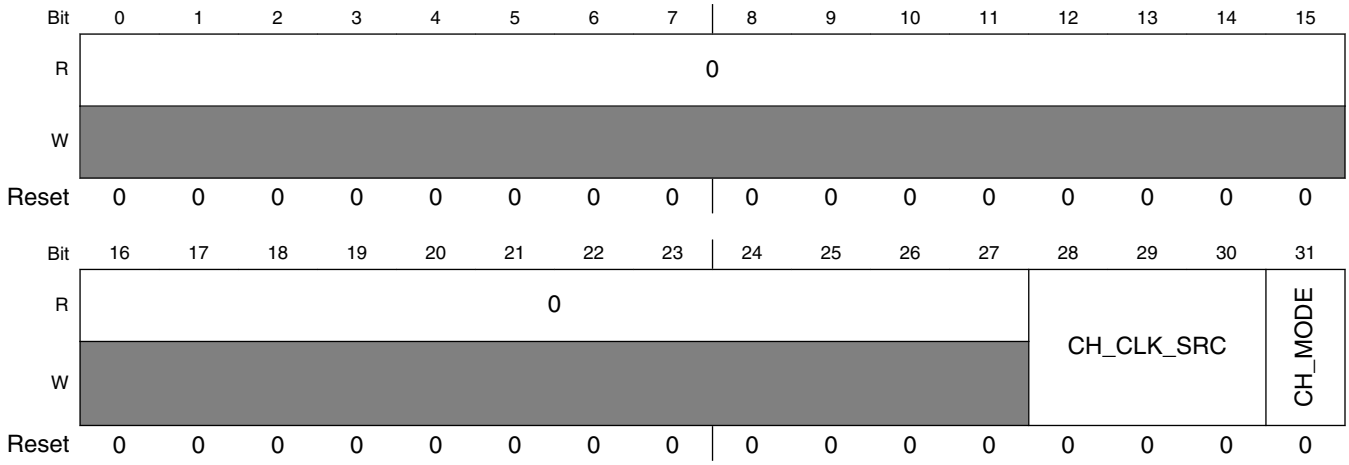
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								BASE																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TBU_CH1_BASE field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 BASE	Time base value for the channel. **NOTE:** The value of BASE can only be written if the corresponding TBU channel is disabled **NOTE:** If the corresponding channel is enabled, a read access to this register provides the current value of the underlying counter.

9.3.6 TBU Channel 2 Control Register (TBU_CH2_CTRL)

Address: 100h base + 14h offset = 114h



TBU_CH2_CTRL field descriptions

Field	Description
0–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–30 CH_CLK_SRC	Clock source for channel time base counter. **NOTE:** This value can be modified only if this channel is disabled: 000 CMU_CLK0 selected. 001 CMU_CLK1 selected. 010 CMU_CLK2 selected. 011 CMU_CLK3 selected. 100 CMU_CLK4 selected. 101 CMU_CLK5 selected. 110 CMU_CLK6 selected. 111 CMU_CLK7 selected.
31 CH_MODE	Channel mode. **NOTE:** This value can only be modified if this channel is disabled. In Free running counter mode the CMU clock source specified by CH_CLK_SRC is used for the counter. In Forward/Backward counter mode the SUB_INC[n] clock signal in combination with the DIR[n] input signal is used to determine the counter direction and clock frequency. 0 Free running counter mode. 1 Forward/backward counter mode.

9.3.7 TBU Channel 2 Base Register (TBU_CH2_BASE)

Address: 100h base + 18h offset = 118h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								BASE																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TBU_CH2_BASE field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 BASE	Time base value for the channel. **NOTE:** The value of BASE can only be written if the corresponding TBU channel is disabled **NOTE:** If the corresponding channel is enabled, a read access to this register provides the current value of the underlying counter.

Chapter 10
Timer Input Module (TIM)

10.1 TIM Overview

The Timer Input Module (TIM) is responsible for filtering and capturing input signals of the GTM. Several characteristics of the input signals can be measured inside the TIM channels. For advanced data processing the detected input characteristics of the TIM module can be routed through the ARU to subsequent processing units of the GTM.

Input characteristics are either:

- time stamp values of detected input rising or falling edges together with the new signal level, or
- the number of edges received since the channel was enabled together with the actual time stamp, or
- PWM signal durations for a whole PWM period.

The architecture of the TIM is shown in Figure 10-1.

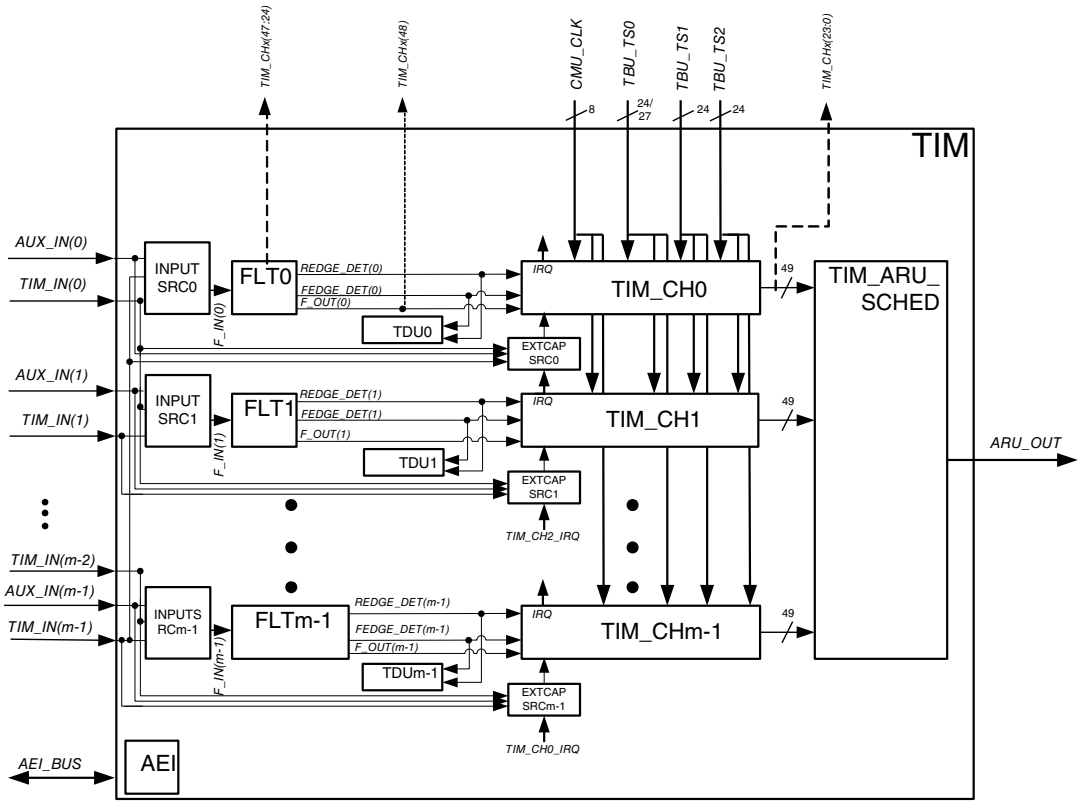


Figure 10-1. TIM block diagram

The number of channels, n, inside a TIM submodule depends on the device.

Each of the dedicated TIM_INn input signals is filtered inside the FLTx subunit of the TIM Module.

NOTE

Incoming TIM_INn input signals are synchronized to SYS_CLK, resulting in a delay of two SYS_CLK periods for the incoming signals.

The TIM submodule provides two filter mechanisms, which are described in FLT Overview. After filtering, the signal is routed to the corresponding TIM channel.

The measurement values can be directly read by the CPU via the AEI-Bus or they can be routed through the ARU to other submodules of the GTM.

Each individual channel has a Timeout Detection Unit (TDU) for timeout detection of an incoming signal (no subsequent edge detected during a specified duration).

Two adjacent channels can be combined by setting the **TIM[i]_CHn_CTRL[CICTRL]** bit field. This allows a combination of complex measurements on one TIM_INn input signal using two TIM channels.

For the TIM0 submodule only, the dashed *TIM[i]_CHn(23:0)*, *TIM[i]_CHn(47:24)* and *TIM[i]_CHn(48)* signal outputs come from TIM0 submodule channels zero through five, and are routed to the MAP submodule. There, they are used for further processing before routing to the DPLL.

The time bases coming from the TBU are connected to the TIM channels to annotate time stamps to incoming TIM_INn signals.

For the TIM0 submodule, the extended 27 bit width TBU_TS0 time base is routed to the TIM channels. The CPU has to configure whether the lower 24 bits (*TBU_TS0n(23..0)*) or the higher 24 bits (*TBU_TS0n(26..3)*) are stored in the **GPRn** registers.

10.2 TIM_Filter_Function_FLT

10.2.1 FLT Overview

The TIM submodule provides a configurable filter mechanism for each input signal. The filter mechanism is in the FLT subunit. The FLT architecture is shown in FLT Architecture .

The filter includes:

- a clock synchronization unit (CSU),
- an edge detection unit (EDU), and
- a filter counter that is associated with the filter unit (FLTU).

The CSU synchronizes the incoming F_IN signal to the CMU_CLKn enable signal, which is selected by writing to the **TIM[i]_CHn_CTRL[FLT_CNT_FRQ]** bit field. The resulting synchronized *F_IN_SYNC* input signal is used for further processing within the filter.

NOTE

Glitches with a duration less than the period of the selected CMU_CLKn enable signal are lost.

The filter modes can be applied individually to the falling and rising edges of an input signal. The following filter modes are available:

- immediate edge propagation mode,
- individual de-glitch time mode (up/down counter), and
- individual de-glitch time mode (hold counter).

10.2.1.1 FLT Architecture

The architecture of the FLT subunit is shown in Figure 10-2 .

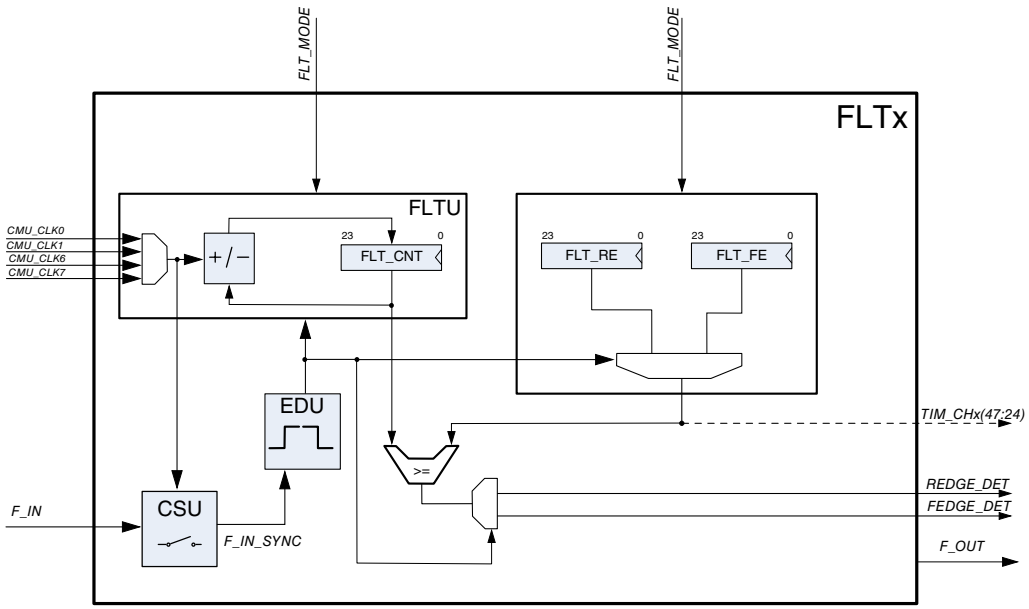


Figure 10-2. FLT architecture

The filter parameters (de-glitch and acceptance time) for the rising or falling edge of the F_IN input signal are configured by writing to either the **FLT_RE** (rising edge) filter parameter register or the **FLT_FE** (falling edge) filter parameter register, respectively. The exact meaning of the parameters depend on the filter mode.

The delay time, T, of both the **FLT_RE** and FLT_FE filter parameters can be determined by:

$T = (FLT\_E + 1) * T_{FLT\_CLK}$,

where T_{FLT_CLK} is the period of the CMU_CLKn enable signal, which is selected by the **TIM[i]_CH[x]_CTRL[FLT_CNT_FRQ]** bit field.

When a glitch is detected on an F_IN input signal, the GLITCHDET status flag is set inside the **TIM[i]_CH[x]_IRQ_NOTIFY** register.

Filter Parameter summary for the different Filter Modes gives an overview of the definitions of the **FLT_RE** and **FLT_FE** registers.

In the individual de-glitch time modes, the actual filter threshold for a detected regular edge is provided on the *TIM[i]_CHn, bits 47:24* output. For immediate edge propagation mode, a value of zero is provided on the *TIM[i]_CHn, bits 47:24* output.

The *TIM[i]_CHn, bits 47:24)* output is used by the MAP submodule for further processing (see MAP Overview).

10.2.1.2 Filter Parameter summary for the different Filter Modes

A filter parameter summary for different filter modes is shown in Table 10-1.

Table 10-1. Filter parameter summary

Filter mode	Meaning of FLT_RE	Meaning of FLT_FE
Immediate edge propagation	Acceptance time for rising edge	Acceptance time for falling edge
Individual de-glitch time (up/down counter)	De-glitch time for rising edge	De-glitch time for falling edge
Individual de-glitch time (hold counter)	De-glitch time for rising edge	De-glitch time for falling edge

A **FLT_CNT** counter is used to measure the glitch and acceptance times.

The frequency of the **FLT_CNT** counter is configurable by writing to the **TIM[i]_CHn_CTRL[FLT_CNT_FRQ]** bit field.

The **FLT_CNT** counter can be clocked by either of the *CMU_CLK0, CMU_CLK1, CMU_CLK6* or *CMU_CLK7* input signals (these clock signals are generated in the CMU submodule).

The **FLT_CNT**, **FLT_FE** and **FLT_RE** registers are each 24 bits wide. For example, when the resolution of the *CMU_CLK0* clock enable signal is 50ns, this allows maximal filter de-glitch and acceptance times of about 838ms.

10.2.2 TIM Filter Modes

10.2.2.1 Immediate Edge Propagation Mode

After detection of an edge, the *F_IN_SYNC* new signal level is propagated to the *F_OUT* output signal with a delay of one T_{FLT_CLK} period. The new signal level remains unchanged until the configured acceptance time expires.

For each edge type, the acceptance time can be configured separately in the **FLT_RE** and **FLT_FE** registers.

Each signal change on the *F_IN_SYNC* signal, during the duration of the acceptance time, has no effect on the level of the *F_OUT* output signal of the filter. However, the signal change does cause the **TIM[i]_CH[x]_IRQ_NOTIFY[GLITCHDET]** glitch bit to be set..

After the acceptance time expires, the *F_IN_SYNC* input signal is monitored. When a change in the level of the *F_IN_SYNC* input signal is detected, the filter raises a new detected edge and propagates the new signal level to the *F_OUT* signal output.

When the acceptance time expires, the value of the *F_OUT* signal is always set to the value of the *F_IN_SYNC* signal.

10.2.2.2 Immediate Edge Propagation Mode in the case of a rising edge

Figure 10-3 shows an example of how the immediate propagation mode responds to a rising edge of the F_IN_SYNC signal.

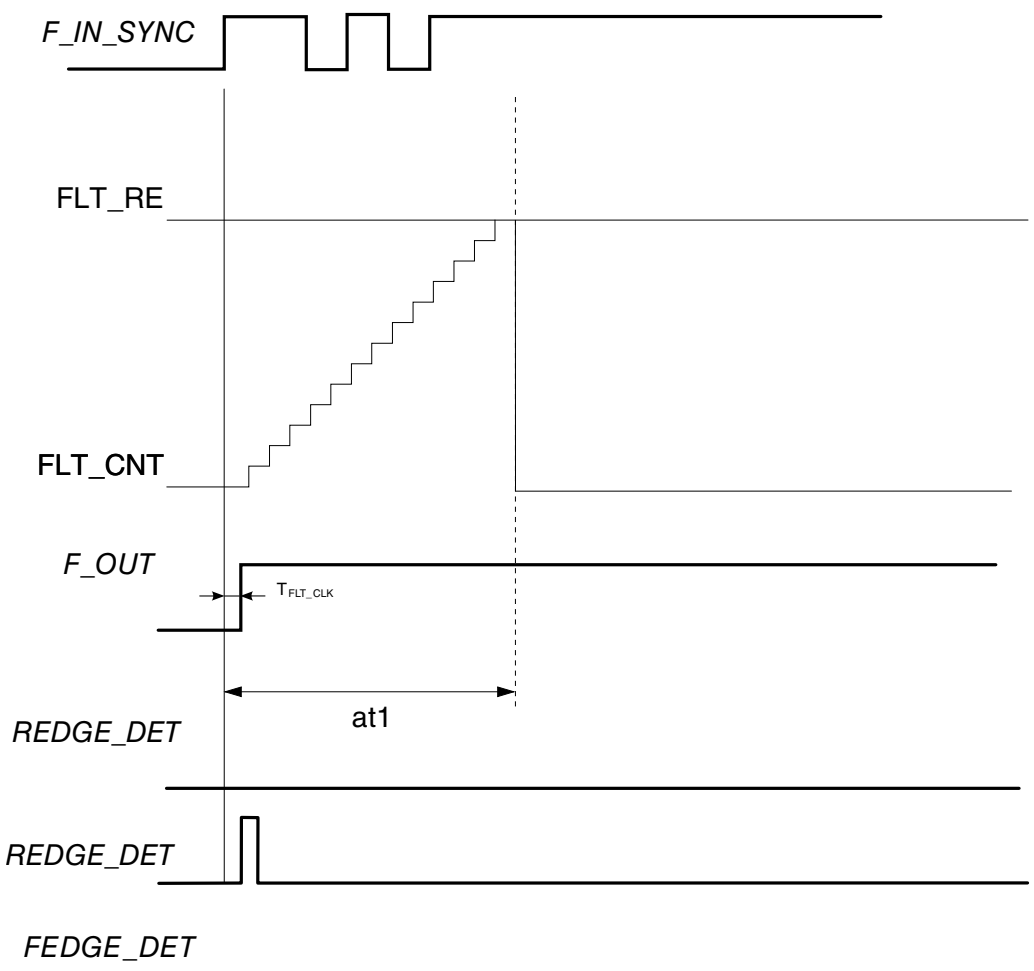


Figure 10-3. Immediate edge propagation mode, rising edge

In the immediate edge propagation mode, the glitch measurement mechanism is not applied to the edge detection. During the acceptance time, detected edges on F_IN_SYNC are transferred directly to F_OUT. When the acceptance time expires, the value of the F_OUT signal is always set to the value of the F_IN_SYNC signal.

The **FLT_CNT** counter is incremented until the acceptance time threshold is reached.

10.2.2.3 Immediate Edge Propagation Mode in the case of a rising and falling edge

Figure 10-4 shows a more complex example of the TIM filter, in which both, rising and falling edges are configured in immediate edge propagation mode.

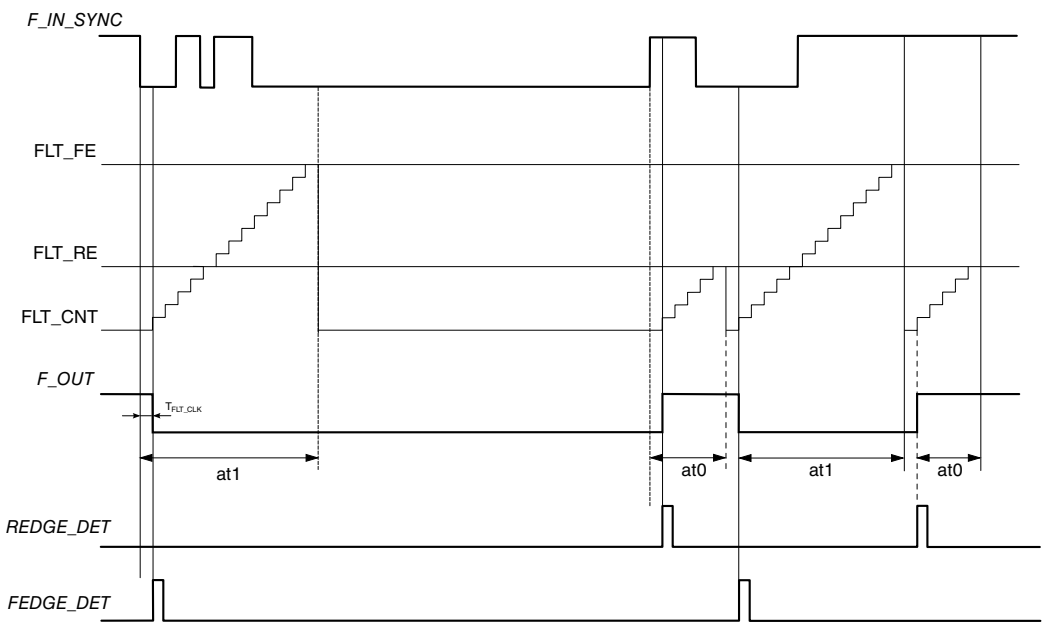


Figure 10-4. Immediate edge propagation mode, rising and falling edges

If:

- the **FLT_CNT** counter has reached the end of the acceptance time for a specific F_IN_SYNC signal edge, and
- the *F_IN_SYNC* signal has already changed to the opposite level of the *F_OUT* signal,

then:

- the opposite *F_IN_SYNC* signal level is sent to the *F_OUT* signal, and
- the acceptance time measurement is immediately started again.

10.2.2.4 Individual De-Glitch Time Mode (up/down counter)

Each edge of an input signal can be filtered with an individual de-glitch threshold filter value that is described in the **FLT_RE** and **FLT_FE** registers, respectively.

The **FLT_CNT** filter counter register is:

- incremented when the *F_IN_SYNC* signal level is unequal to the *F_OUT* signal level, and
- decremented if the *F_IN_SYNC* signal level equals the *F_OUT* signal level

After the **FLT_CNT** counter decrements to zero, the counter is immediately stopped.

If a glitch is detected, the **TIM[i]_CHn_IRQ_NOTIFY[GLITCHDET]** glitch detection bit is set.

The detected edge signal and its signal level is propagated to the *F_OUT* signal after the individual de-glitch threshold is reached as shown in Figure 10-5.

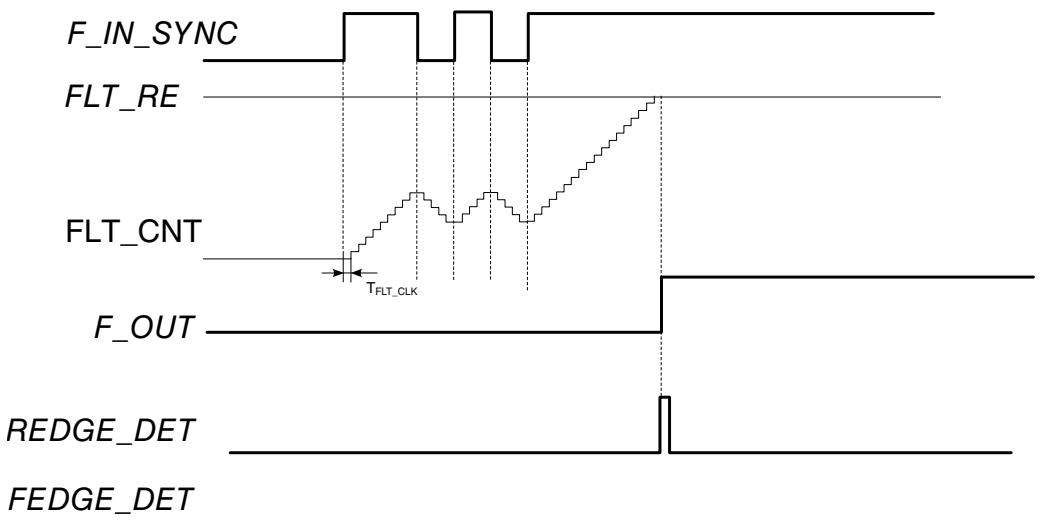


Figure 10-5. Individual De-Glitch Time Mode (up/down counter with rising edge)

10.2.2.5 Individual De-Glitch Time Mode (hold counter)

In individual de-glitch time mode (hold counter) each edge of an input signal can be filtered with an individual de-glitch threshold filter value mentioned in the registers FLT_RE and FLT_FE, respectively.

The filter counter register FLT_CNT is incremented when the signal level on F_IN_SYNC is unequal to the signal level on F_OUT and the counter value of FLT_CNT is hold if F_IN equals F_OUT. If a glitch is detected the glitch detection bit GLITCHDET is set in the TIM[i]_CH[x]_IRQ_NOTIFY register. The detected edge

signal together with the new signal level is propagated to F_OUT after the individual de-glitch threshold is reached. The following figure shows the behavior of the filter in individual de-glitch time (hold counter) mode in the case of the rising edge detection.

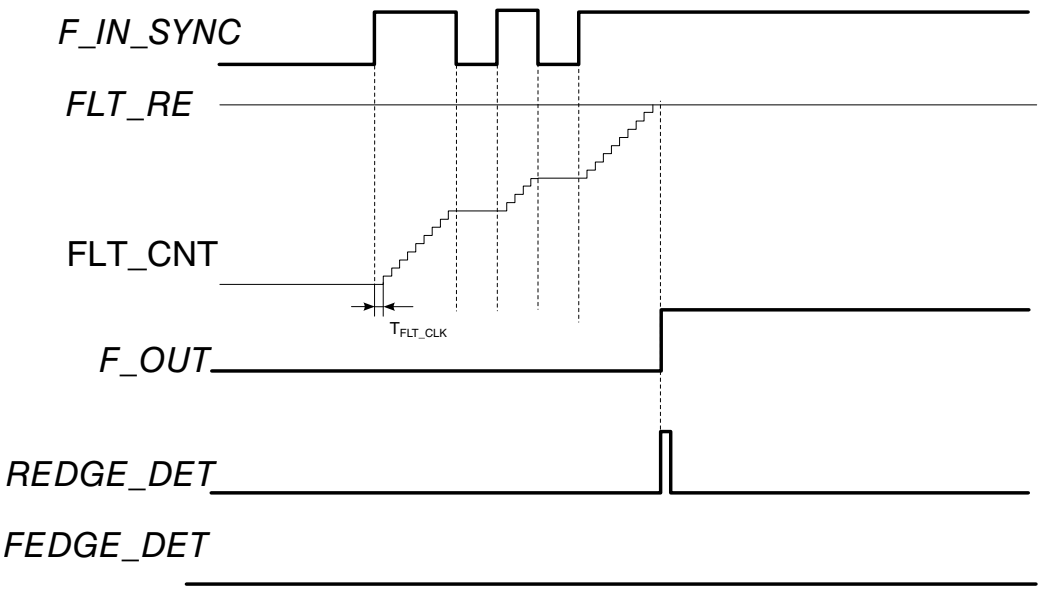


Figure 10-6. Individual De-Glitch Time Mode (hold counter) in the case of a rising edge

10.2.2.6 Immediate Edge Propagation and Individual De-Glitch Mode

The three different filter modes can be applied individually to each edge of the measured signal.

However, if one edge is configured with the immediate edge propagation mode and the other edge is configured with an individual de-glitch mode (whether up/down counter or hold counter), a special consideration has to be applied:

Assume that the rising edge is configured for the immediate edge propagation mode and the falling edge is configured for the individual de-glitch mode (up/down counter) as shown in Figure 10-7.

If the falling edge of the incoming signal already occurs during the measuring of the acceptance time of the rising edge, the measurement of the de-glitch time for the falling edge is starts immediately after the acceptance time measurement phase of the rising edge has finished.

Consequently, the de-glitch counter can not measure the time T_{ERROR}.

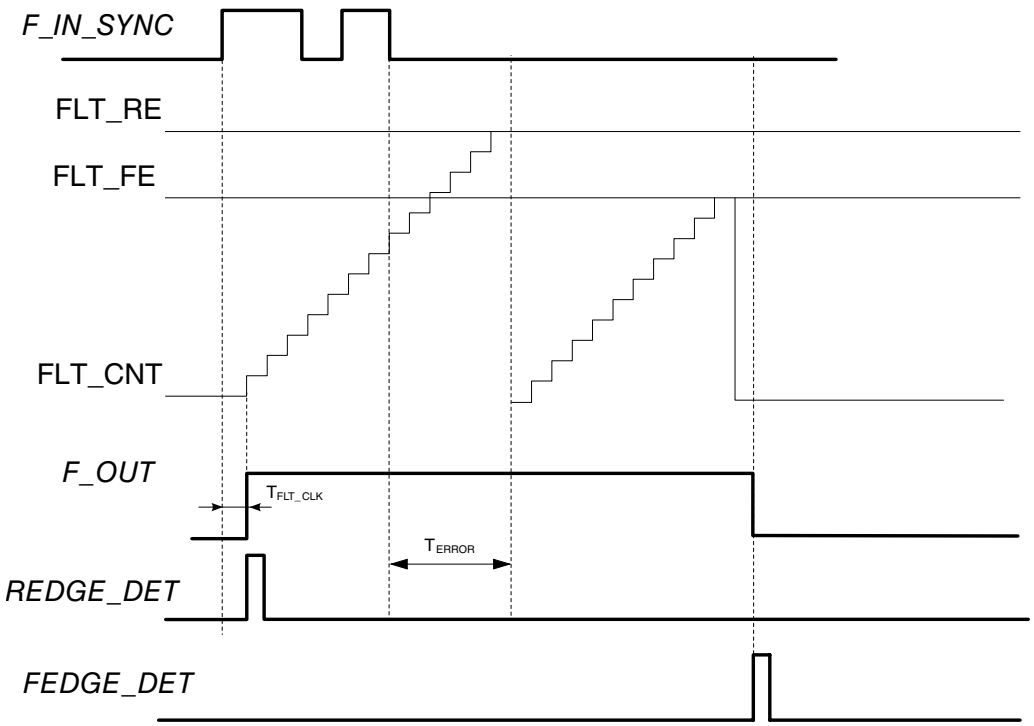


Figure 10-7. Mixed mode measurement

10.2.3 TIM Filter reconfiguration

If FLT_EN=1:

- A change of FLT_RE or FLT_FE will take place immediately.
- A change of FLT_MODE_RE or FLT_MODE_FE will be used with the next occuring corresponding edge. If the mode is changed while the filter unit is processing a certain mode, it will end this edge filtering in the mode as started.
- A change of FLT_CTR_RE or FLT_CTR_FE will take place immediately.

10.3 Timeout Detection Unit (TDU)

The Timeout Detection Unit (TDU) is responsible for timeout detection of the TIM input signals.

Each channel of the TIM submodule has its own Timeout Detection Unit (TDU), where a timeout event can be set up on the filtered input signal of the corresponding channel.

The TDU subunit architecture is shown in Figure 10-8.

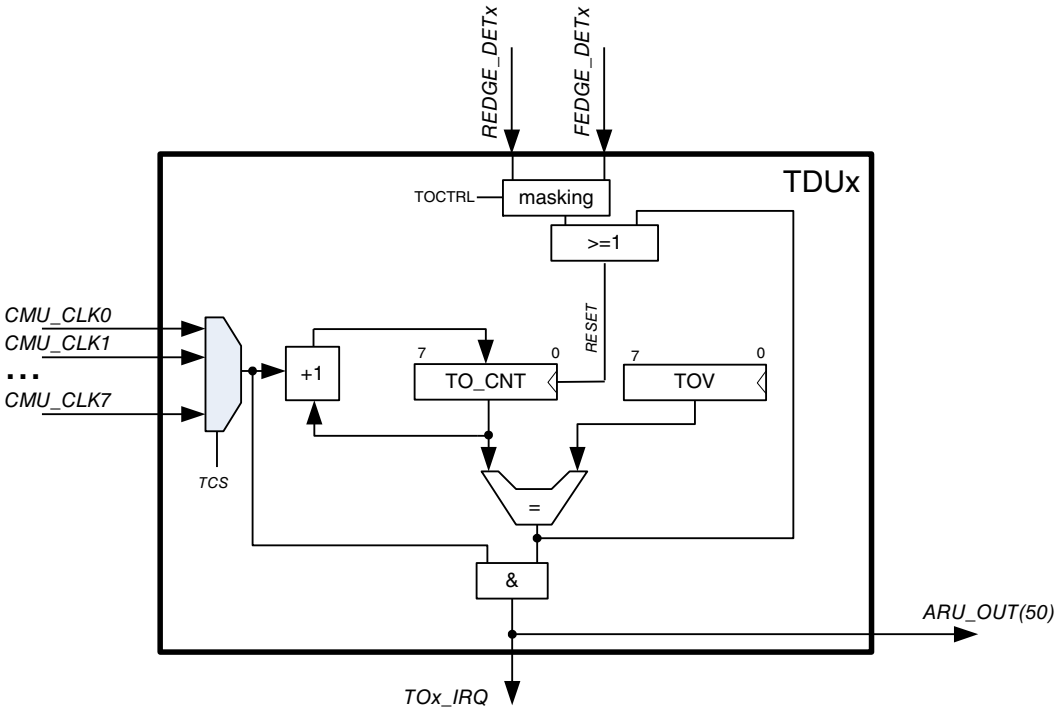


Figure 10-8. TDU Subnit Architecture

It is possible to detect timeouts with the resolution of the specified *CMU_CLKn* input clock enable signal, which is configured by writing to the **TIM[i]_CHn_TDUV[TCS]** bit field. The individual timeout values have to be specified in 'number of ticks' of the selected input clock enable signal. The 'number of ticks' is configured by writing to the **TIM[i]_CHn_TDUV[TOV]** bit field.

The exact time out value, T_{TDU}, is calculated by the equation:

$T_{TDU} = (TOV + 1) * T_{CMU\_GCLKn}$,

where T_{CMU_GCLKn} is the period of the selected CMU_CLKn input clock enable signal.

Timeout detection can be enabled or disabled by writing to the TIM[i]_CHn_CTRL[TOCTRL] bit field.

Timeout detection can be enabled to be sensitive to falling, rising or both edges of the input signal by writting the corresponding values to the TIM[i]_CHn_CTRL[TOCTRL] bit field.

The **TIM[i][_CHn_TDUC[TO_CNT]** counter is reset by:

- each detected valid input edge coming from the filtered input signal, or
- the **TIM[i]_CHn_TDUV[TOV]** timeout value being reached in the **TIM[i] [_CHn_TDUC[TO_CNT]** counter.

After such a reset, or by enabling the channel by writing to the **TIM[i]_CHn_CTRL[TOCTRL]** Timeout Control bit, the **TIM[i][_CHn_TDUC[TO_CNT]** counter (when enabled by the specified input clock enable signal) starts counting again from zero.

The TDU generates a *TIM_TODETn_IRQ* interrupt signal whenever:

- a timeout is detected for an individual input signal, and
- the **TIM[i]_CHn_IRQ_NOTIFY[TODET]** bit is set.

In addition, when ARU access is enabled by the CPU setting the **TIM[i]_CHn_CTRL[ARU_EN]** bit, and if a timeout event occurs, the actual values stored inside the **TIM[i]_CHn_GPRz** registers are sent together with the last stored signal level to the ARU.

To signal that a timeout occurred, the ARU_OUT[ACB(2)] bit, is set.

As a result, a destination can determine if a timeout occurred at the TIM input by evaluating ACB bit 2.

Since the TIM channel still monitors its input even though the timeout occurred, a valid edge could occur at the input while the timeout information is still valid at the ARU. In that case:

- the new edge associated data is stored inside the **TIM[i]_CHn_GPRz** registers,
- GPRz overflow detected sets the ARU_OUT[ACB(1)] bit with the timeout ARU_OUT[ACB(2)] bit, and
- the values are marked as valid to the ARU.

The ARU_OUT[ACB(2)] bit is cleared, when a successful ARU write access by the TIM channel takes place.

Table 10-2 clarifies the meaning of the ACB bits for valid data provided by a TIM channel:

Table 10-2. Meaning of the ACB Bits for valid data provided by a TIM channel

ACB4/3	ACB2	ACB1	ACB0	Description
dc[1]	0	0	SL[2]	Valid edge detected
dc	0	1	SL	Input edge overwritten by subsequent edge
dc	1	0	SL	Timeout detected without valid edge
dc	1	1	SL	Timeout detected with subsequent valid edge detected

1. don't care
2. Signal Level

10.4 TIM Channel Architecture

10.4.1 TIM Channel Overview

Each TIM channel consists of:

- an **ECNT** input edge counter,
- a Signal Measurement Unit (SMU) with a counter **CNT**,
- a counter shadow register **CNTS** for the SMU counter, and
- two general purpose registers, **GPR0** and **GPR1**, for value storage.

The **TOV** value from the **TIM[i]_CHn_TDU** timeout register is provided to the TDU subunit of each individual channel for timeout measurement. The architecture of a TIM channel is shown in Figure 10-9.

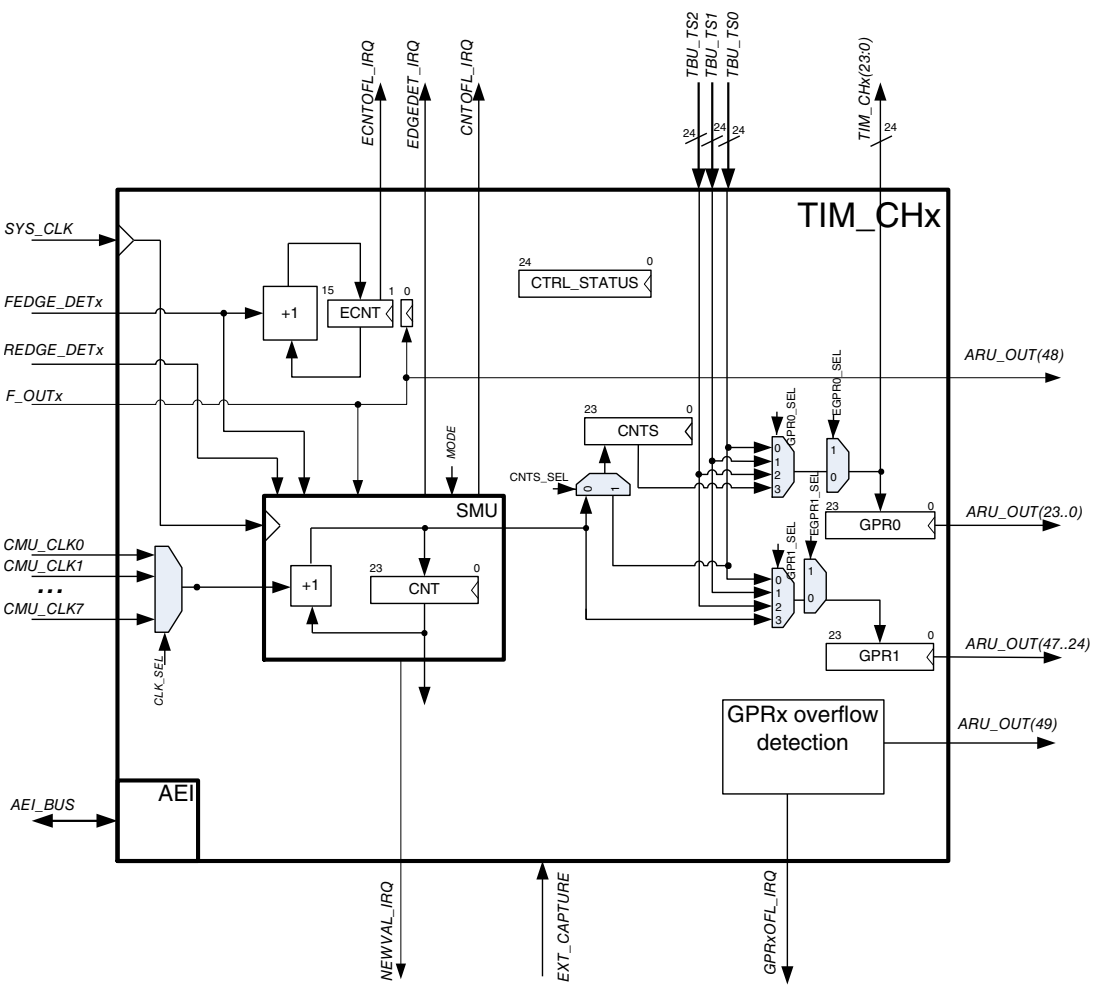


Figure 10-9. TIM channel architecture

Each TIM channel receives the *REDGE_DETx* and *FEDGE_DETx* input trigger signals (that are generated by the corresponding filter module) echo the *F_INn* input signal. The filtered *F_INn* input signal is routed to the *F_OUTn* signal.

The eight bit **ECNT** edge counter counts every incoming filtered edge (rising and falling). The counter value is uneven in the case of a detected rising filtered edge, and even in the case of a detected falling filtered edge.

As a result, the input signal level is part of the **ECNT** counter and it can be obtained from bit 0. (The actual counter implementation counts only falling edges on the ECNT[n:1] bits and it generates the **ECNT** output by concatenating the F_OUTx signal (ECNT bit 0) with the ECNT[n:1] bits.)

The whole ECNT counter value is always odd when a positive edge was received and always even when a negative edge was received.

The current **ECNT[7:0]** register content is made visible on bits 31 through 24 of the **GPRz** and **CNTS** registers. This allows the CPU to detect inconsistent read accesses to the **GPR0**, **GPR1**, and **CNTS** registers. However, the update strategy of these registers depends on the selected TIM modes. Therefore, a consistency check has to be adapted carefully. It can be chosen with the FR_ECNT_OFL bit field when an ECNT overflow is signaled on ECNTOFL. An ECNT overflow can be signaled on 8-bit or full range resolution.

While reading the TIM[i]_CH[x]_ECNT register, the ECNT[0] bit shows the F_OUTx input signal value independent of the state (enabled / disabled) of the channel. If a channel gets disabled (OSM mode or resetting TIM_EN), the content of TIM[i]_CH[x]_ECNT will be frozen until a read of the register takes place. This read will reset the ECNT counter. Continuing reads will show the input signal value in bit ECNT[0] again.

When new data is written into the **GPR0** and **GPR1** registers, the **TIM[i]_CHn_IRQ_NOTIFY[NEWVAL]** bit is set and the *NEWVALn_IRQ* interrupt is asserted (if enabled).

Each TIM input channel has an ARU connection for providing data, via the ARU, to other GTM submodules. The data provided to the ARU depends on the TIM channel mode and its corresponding adjustments (e.g. multiplexer configuration).

The **TIM[i]_CHn_CTRL[ARU_EN]** bit enables the measurement results of the **GPRz** registers to be consumed by:

- another submodule via the ARU (**ARU_EN** = 1), or
- the CPU via the AEI (**ARU_EN** = 0).

To guarantee a consistent delivery of data from the **GPR0** and **GPR1** registers to the ARU or the CPU, each TIM channel has to ensure that the data is consumed before it is overwritten with new values.

If new data was produced by the TIM channel (as indicated by the **TIM[i]_CHn_IRQ_NOTIFY**[NEWVAL] bit being set) while the old data is not consumed by the ARU (ARU_EN = 1) or CPU (ARU_EN = 0) , the TIM channel sets the **TIM[i]_CHn_IRQ_NOTIFY**[GPRzOFL] bit and overwrites the data inside the **GPRz** registers.

If the CPU is selected as the consumer for the **GPRz** registers (**ARU_EN** = 0), the acknowledge for reading out data from the registers is signaled by a read access to the **GPR0** register So, the **GPR1** register should always be read before the **GPR0** register is read.

If the ARU is selected as consumer for the **GPRz** registers (**ARU_EN** = 1), the acknowledge for reading out data from the registers is performed by the ARU itself. However, the GPRz registers could be read by the CPU without giving an acknowledge.

10.4.1.1 Input source selection INPUTSRCx

The source that is used for processing in the FLT, TDU, and TIM_CH units can be configured by writing to the TIM[i]_IN_SRC[MODE_x, VAL_x] fields and the TIM[i]_CH[x]_CTRL[CICTRL] field. The input source block diagram is shown in the following figure.

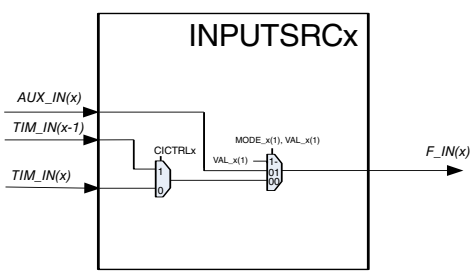


Figure 10-10. INPUTSRC Block Diagram

In a certain MODE_x, VAL_x combination, the input signal, F_IN(x), can be driven by VAL_x(1) with 0 or 1 directly. Due to the fact that all 8 channels are bundled in the TIM[i]_IN_SRC register, a synchronous control of all 8 input channels is possible.

Two adjacent channels can be combined by setting the corresponding TIM[i]_CH[x]_CTRL[CICTRL] field. This allows for a combination of complex measurements on one input signal using two TIM channels.

The AUX_IN_x signal can be selected as an input. The source of this signal is defined in Signal Multiplex.

10.4.1.2 External capture source selection EXTCAPSRCx

Each channel can operate on an EXT_CAPTURE external capture signal. The source to use for this signal can be configured by writing to the TIM[i]_CH[x]_ECTRL[EXT_CAP_SRCx] bit field. The following figure shows the EXTCAPSRC Block Diagram.

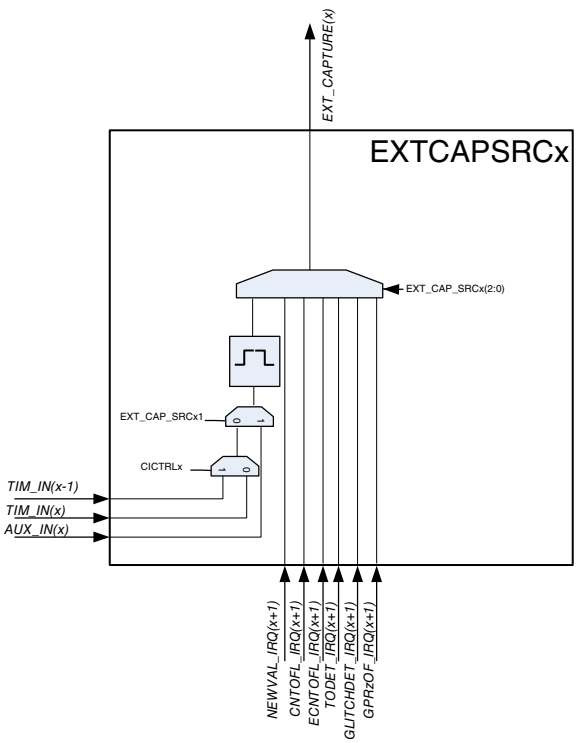


Figure 10-11. EXTCAPSRC Block Diagram

The external capture functionality can be enabled for the TIM channel x with the bit EXT_CAP_EN in the register TIM[i]_CH[x]_CTRL, it will trigger on each rising edge. A pulse generation for each rising edge of the selected input signal TIM_IN[x] and AUX_IN[x] is applied.

The six TIM channel interrupt sources can be triggered by the operation in the certain TIM channel modes. Alternatively they can be issued by a soft trigger using the corresponding bits in the register TIM[i]_CH[x+1]_FORCINT.

10.4.2 TIM Channel Modes

The TIM submodule provides six different measurement modes:

- TIM PWM Measurement Mode (TPWM),
- TIM Pulse Integration Mode (TPIM),
- TIM Input Event Mode (TIEM),
- TIM Input Prescaler Mode (TIPM)
- TIM Bit Compression Mode (TBCM), and
- TIM Gated Periodic Sampling Mode (TGPS).

The modes can be configured by writing to the **TIM[i]_CHn_CTRL[TIM_MODE]** bit field.

Other configuration bits in the **TIM[i]_CHn_CTRL** register provide detailed control over each mode. The detailed control bits are:

- **DSL**: Controls the signal level for the measurement modes (e.g. if a measurement is started with a rising edge or a falling edge, or if high level pulses or low level pulses are measured).
- EGPR0_SEL, GPR0_SEL and EGPR1_SEL, GPR1_SEL: Controls the actual content of the **GPR0** and **GPR1** registers after a measurement has completed.
- **CNTS_SEL**: Controls the content of the **CNTS** register. The actual time for updating the **CNTS** register is mode dependent.
- **OSM**: Activates measurement in one-shot mode or continuous mode. In one-shot mode, only one measurement cycle is performed and then the channel is disabled.
- **NEWVAL**: Shows the status of the NEWVAL IRQ interrupt. The interrupt is triggered at the end of a measurement cycle, signalling that the GPR0 and GPR1 registers are updated.
- **ARU_EN**: Enables sending of the **GPR0** and **GPR1** registers together with the actual signal level in bit 48, the overflow GPROFL signal in bit 49, and the timeout status information in bit 50 to the ARU.
- EXT_CAP_EN: Forces an update of the GPR0 and GPR1 registers only on each rising edge of the EXT_CAPTURE signal and asserts a NEWVAL IRQ interrupt. If this mode is disabled, the NEWVAL IRQ interrupt is asserted at the end of each measurement cycle.

  For each channel, the source of the EXT_CAPTURE signal can be configured by writing to the TIM[i]_CHn_ECTRL[EXT_CAP_SRC] bit fields.

10.4.2.1 TIM PWM Measurement Mode (TPWM)

In TIM PWM Measurement Mode, the TIM channel measures the duty cycle and period of an incoming PWM signal. The **TIM[i]_CHn_CTRL[DSL]** bit defines the polarity of the PWM signal.

When the measurements of pulse high time and period are requested (PWM with a high level duty cycle, **TIM[i]_CHn_CTRL[DSL**=1]), the channel starts measuring after the first rising edge is detected by the channel filter.

Measurement is accomplished by the **CNT** counter counting (when enabled by the **CMU_CLKn** input clock enable signal) until a falling edge is detected. The counter value is then stored in the **CNTS** shadow register (if **TIM[i]_CHn_CTRL[CNTS_SEL** = 0] and the **CNT** counter continues counting until the next rising edge is detected.

On this rising edge, the content of the **CNTS** register is transferred to the **GPR0** register and the content of the **CNT** counter is transferred to the **GPR1**register (if **TIM[i]_CHn_CTRL[GPR0_SEL**=3, **GPR1_SEL**=3]). As a result, the **GPR0** register then contains the duty cycle length and the **GPR1** register then contains the period.

NOTE

Bits one through seven of the **ECNT** counter output may be used to check data consistency of the **GPR0** and **GPR1** registers.

In addition, the **CNT**counter is cleared, the **TIM[i]_CHn_IRQ_NOTIFY[NEWVAL]** status bit is set, and the *TIM_NEWVALx_IRQ* interrupt is raised (if enabled).

The CNTS register update is not performed until the measurement is started (first edge defined by DSL is detected). Afterwards each edge leaving the level defined by DSL is perfoming a CNTS register update.

If a PWM with a low level duty cycle is to be measured (**TIM[i]_CHn_CTRL[DSL** = 0]), the channel waits for a falling edge until measurement is started. On this falling edge:

1. the low level duty cycle time is stored in the **CNTS** register,
2. then the low level duty cycle time is stored in the **GPR0** register, and
3. then the period is stored in the **GPR1** register.

When a PWM period is successfully measured, the data in the **GPR0** and **GPR1** registers is marked as valid for reading by the ARU when:

- the **TIM[i]_CHn_CTRL[ARU_EN]** bit is set,
- the **TIM[i]_CHn_IRQ_NOTIFY[NEWVAL]** status bit is set, and
- a new measurement is started.

If the preceding PWM values were not consumed by an ARU destination (**TIM[i]_CHn_CTRL[ARU_EN]** bit is set) or by the CPU, the TIM channel:

- sets the **TIM[i]_CHn_IRQ_NOTIFY[GPROFL]** status bit,
- triggers the *GPROFL_IRQ* interrupt (if enabled), and
- and overwrites the old values in **GPR0** and **GPR1** registers.

Afterward, a new measurement is started.

If the **CNT** counter produces an overflow during the measurement, the **TIM[i]_CHn_IRQ_NOTIFY[CNTOFL]** bit is set, and the *TIM_CNTOFLn_IRQ* interrupt (if enabled) is triggered.

If the **ECNT** counter produces an overflow during the measurement, the **TIM[i]_CH[x]_IRQ_NOTIFY[ECNTOFL]** bit is set and the *TIM_ECNTOFLn_IRQ* interrupt (if enabled) is triggered.

TIM_ECNTOFL[x]_IRQ is asserted depending on the corresponding interrupt enable condition.

10.4.2.1.1 External capture TIM PWM measurement mode (TPWM)

If external capture is enabled, the PWM measurement is done continuously. The actual measurement values are captured to GPRx if an external capture events occurs.

Operation is done depending on the
- CMU clock,
- ISL bit,
- DSL bit, and
- the F_OUTx input signal value as defined in the following table.

Table 10-3. External capture

F_OUTx input signal	Selected CMU clock	External capture	ISL	DSL	Action description
0	1	0	–	0	CNT++
1	1	0	–	0	None
Rising edge	–	0	0	0	Capture CNT value in CNTS
Falling edge	–	0	0	0	CNT = 0
Rising edge	–	0	1	0	None
Falling edge	–	0	1	0	Capture CNT value in CNTS; CNT = 0
1	1	0	–	1	CNT++
0	1	0	–	1	None
Falling edge	–	0	0	1	Capture CNT value in CNTS
Rising edge	–	0	0	1	CNT = 0

Table continues on the next page...

Table 10-3. External capture (continued)

F_OUTx input signal	Selected CMU clock	External capture	ISL	DSL	Action description
Falling edge	–	0	1	1	None
Rising edge	–	0	1	1	Capture CNT value in CNTS; CNT=0
–	–	Rising edge	–	–	Do GPRx capture; assert NEWVAL_IRQ interrupt
–	0	0	–	–	None

10.4.2.2 TIM Pulse Integration Mode (TPIM)

In TIM Pulse Integration Mode and depending on the configuration of the **TIM[i]_CHn_CTRL[DSL]** bit, each TIM channel is able to measure a sum of pulse high or low times of an input signal.

The pulse times are measured by incrementing the CNT channel counter whenever the pulse has the signal level specified by the **DSL** bit. The CNT counter is stopped whenever the input signal has the opposite signal level of that specified by the DSL bit.

The CNT counter counts with SYS_CLK when enabled by the *CMU_CLKn* input clock enable signal (specified the **TIM[i]_CHn_CTRL[CLK_SEL]** bit).

The **CNT** counter is reset at the time the channel is activated and it accumulates pulses while the channel remains enabled.

Whenever the CNT counter is stopped, the **CNTS**, **GPR0** and **GPR1** registers are updated according to the **TIM[i]_CHn_CTRL[GPR0_SEL,CNTS_SEL]** bit configurations for their corresponding input multiplexers.

NOTE

Bits one through seven of the **ECNT** register can be used to check data consistency of the **GPR0** and **GPR1** registers.

When the **TIM[i]_CHn_CTRL[ARU_EN]** bit is set, the measurement results of the **GPR0** and **GPR1** registers can be routed to destination submodules that are attached to the ARU.

10.4.2.2.1 External capture TIM pulse integration mode (TPIM)

If external capture is enabled, the PWM measurement is done continuously. The actual measurement values are captured to GPRx if an external capture events occurs.

Operation is done depending on the

- CMU clock,
- ISL bit,
- DSL bit, and
- the F_OUTx input signal value as defined in the following table.

Table 10-4. External capture

F_OUTx input signal	Selected CMU clock	External capture	ISL	DSL	Action description
0	1	0	–	0	CNT++
1	1	0	–	0	None
1	1	0	–	1	CNT++
0	1	0	–	1	None
–	–	Rising edge	–	–	Do capture; issue NEWVAL_IRQ; CNT=0
–	0	0	–	–	None

10.4.2.3 TIM Input Event Mode (TIEM)

In TIM Input Event Mode, the TIM channel is able to count edges.

Counting on the rising, falling, or rising and falling edges can be configured by writing to the **TIM[i]_CHn_CTRL[DSL, ISL]** bits.

In addition, a *TIM[i]_NEWVALn_IRQ* interrupt (if enabled) is raised when the configured edge is received.

The CNT counter is used to count the number of received edges.

The TIM[i]_CHn_CTRL[**GPR0_SEL**, **GPR1_SEL**, **CNTS_SEL]** bits are used to configure the desired update values for the **GPR0**, **GPR1**, and **CNTS** registers. These registers are updated whenever the CNT counter is incremented (because a configured edge is received).

If the preceding data was not consumed by a reader attached to the ARU, or by the CPU, the TIM channel sets the GPROFL status bit and asserts a GPROFL[x]_IRQ (if it is enabled in TIM[i]_CH[x]_IRQ_EN register) and overwrites the old values in GPR0 and GPR1 with the new values.

If the **CNT** counter produces an overflow during the measurement, the **TIM[i]_CHn_IRQ_NOTIFY[CNTOFL]** bit is set and the*TIM_CNTOFLn_IRQ* interrupt (if enabled) is raised.

If the **ECNT** register produces an overflow during the measurement, the **TIM[i]_CH[x]_IRQ_NOTIFY**[ECNTOFL] bit is set and the *TIM_ECNTOFL[x]_IRQ* interrupt (if enabled) is raised.

If measured data that is routed to the CPU, or through the ARU to a destination, is not consumed by the time that new measured data is available, the TIM channel:

- sets the **TIM[i]_CH[x]_IRQ_EN[GPRzOFL]** status bit,
- raises the *GPRzOFL[x]_IRQ* interrupt (if enabled), and
- overwrites the old measured data in the **GPR0** and **GPR1** registers with the new measured data.

The TIM Input Event Mode does not use the **TIM[i]_CHn_CTRL[CLK_SEL]** bit field.

10.4.2.3.1 External capture TIM input event mode (TIEM)

If external capture is enabled, capturing is done depending on the:

- DSL bit,
- ISL bit, and
- the input signal value

as defined in the following table.

Table 10-5. External capture

F_OUTx Input signal	External capture	ISL	DSL	Action description
–	Rising edge	1	–	Do capture; issue NEWVAL_IRQ; CNT++
–	0	1	–	None
1	Rising edge	0	1	Do capture; issue NEWVAL_IRQ; CNT++
0	–	0	1	None
0	Rising edge	0	0	Do capture; issue NEWVAL_IRQ; CNT++
1	–	0	0	None

10.4.2.4 TIM Input Prescaler Mode (TIPM)

In the TIM Input Prescaler Mode, the number of received edges that are detected before a *TIM[i]_NEWVALn_IRQ* interrupt is raised is configurable (by the CPU writing a value to the **CNTS** register.

A value of zero in the **CNTS** register means that the interrupt is raised after one edge is detected. A value of one in the **CNTS** register means that the interrupt is raised after two edges are detected, and so on.

The **TIM[i]_CHn_CTRL[DSL, ISL]** bits are used to configure whether rising edges, falling edges, or rising and falling edges are detected.

With each raised interrupt, the **GPR0** and **GPR1** registers are updated according to the TIM[i]_CHn_CTRL[GPR0_SEL, **GPR1_SEL**] bits.

If the **ECNT** register produces an overflow during the measurement, the **TIM[i]_CHn_IRQ_NOTIFY[ECNTOFL]** bit is set and the *TIM_ECNTOFLn_IRQ* interrupt is raised (if enabled).

The TIM Input Prescaler Mode does not use the **TIM[i]_CHn_CTR[CLK_SEL]** bit field.

10.4.2.4.1 External capture TIM input prescaler mode (TIPM)

If external capture is enabled, the external capture events are counted instead of the input signal edges.

Operation is done depending on:
- the external capture signal,
- DSL bit, and
- ISL bit

as per the following table.

Table 10-6. External Capture

F_OUTx Input signal	External capture	ISL	DSL	Action description
–	Rising edge	1	–	If CNT == CNTS: do capture, issue NEWVAL_IRQ, CNT=0; else CNT++ endif
–	0	1	–	None
1	Rising edge	0	1	If CNT == CNTS: do capture, issue NEWVAL_IRQ, CNT=0; else CNT++ endif
0	–	0	1	None
0	Rising edge	0	0	If CNT == CNTS: do capture, issue NEWVAL_IRQ, CNT=0; else CNT++ endif
1	–	0	0	None

10.4.2.5 TIM Bit Compression Mode (TBCM)

The TIM Bit Compression Mode (TBCM) can be used to:

- combine all of the filtered TIM_INn input signals of a TIM submodule into a parallel m-bit data word, where m is the number of channels available in the TIM submodule, and
- route that data word to the ARU.

Since this mode uses all available input signals with its input filters, it is only available within TIM channel 0 of each TIM submodule.

As shown in Figure 10-12, TIM channel 0 is used to combine the eight signals.

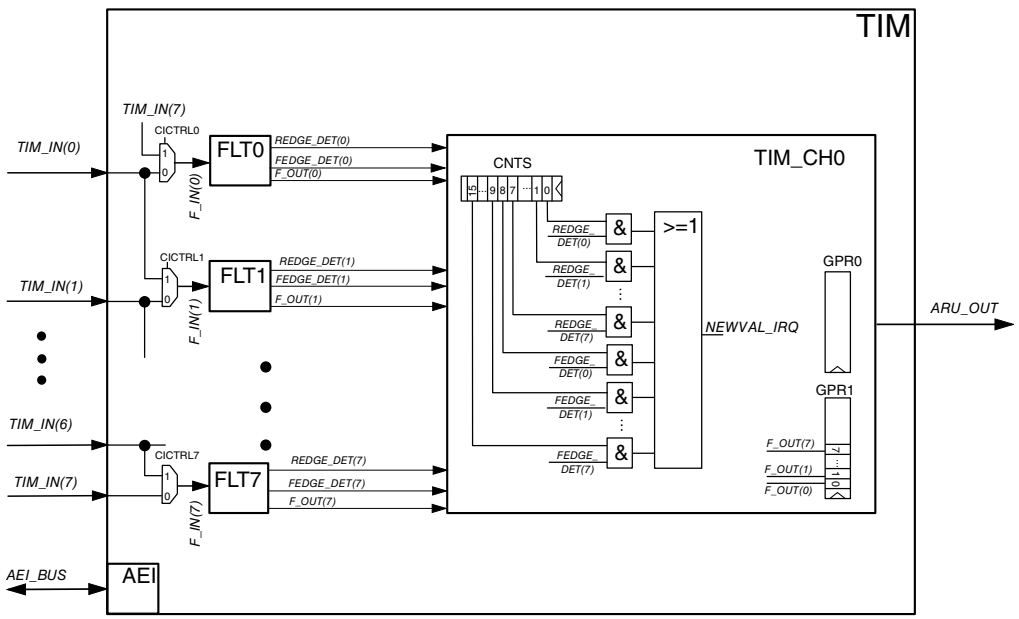


Figure 10-12. Compression mode architecture

The CNTS register of TIM channel 0 is used to configure the event that releases the NEWVAL_IRQ and samples the input signals F_IN(0) to F_IN(m-1) in ascending order as a parallel data word in GPR1.

Bits zero through m – 1 of the **CNTS** register are used to select the *REDGE_DET* signals of TIM filters zero through m – 1, respectively, as a sampling event. Bits eight through 7 + m of the **CNTS** register are used to select the *FEDGE_DET* signals of the TIM filters zero through m – 1, respectively, as a sampling event. If the *REDGE_DET* signals and the *FEDGE_DET* signals are selected as a sampling event, the events are OR-combined.

The TIM_CH0_CTRL[**GRP0_SEL]** bit selects the time stamp value that is routed through the ARU. **The TIM_CH0_CTRL[GRP1_SEL]**bit is not used in TBCM mode.

If the **TIM[i]_CH0_CTRL[ARU_EN]** bit is set and when the NEWVAL_IRQ interrupt is triggered, the sampled data from the **GPR1** register is routed, together with the time stamp value from the **GPR0** register, to the ARU.

The **ECNT** counter increments on each triggering of a *NEWVAL_IRQ* interrupt. The value in the ECNT counter can depend on all of the m filtered input signals. However, the LSB of the **ECNT** counter does not reflect the actual level of the filtered input signal.

If the **ECNT** counter produces an overflow during the measurement, the **TIM[i]_CHn_IRQ_NOTIFY[ECNTOFL]** bit is set and the *TIM_ECNTOFLn_IRQ* interrupt (if enabled) is triggered.

The TIM Bit Compression Mode does not use the **TIM[i]_CHn_CTRL[CLK_SEL]** bit.

To set up the TIM Bit Compression Mode, the CPU must:

- configures all input filters,
- enable TIM channel 0 in bit compression mode, and
- disable TIM channels one through seven.

10.4.2.5.1 External capture bit Compression mode (TBCM)

If external capture is enabled, capturing is done depending on:

- the input signal value,
- DSL bit, and
- ISL bit

per the following table

Table 10-7. External capture

F_OUTx Input signal	External capture	ISL	DSL	Action description
–	Rising edge	1	–	Do capture; issue NEWVAL_IRQ; CNT++
–	0	1	–	None
1	Rising edge	0	1	Do capture; issue NEWVAL_IRQ; CNT++
0	–	0	1	None
0	Rising edge	0	0	Do capture; issue NEWVAL_IRQ; CNT++
1	–	0	0	None

10.4.2.6 TIM Gated Periodic Sampling Mode (TGPS)

In the TIM Gated Periodic Sampling Mode, the number of CMU clock cycles that should elapse before capturing and TIM[i]_NEWVALn_IRQ interrupt assertion is programmable. In the TIM Gated Periodic Sampling Mode (TGPS) mode, the TIM[i]_NEWVALn_IRQ interrupt is asserted only after a number of CMU clock cycles have occurred as specified in the CNTS register.

A value of 0 in TIM[i]_CHn_CNTS register means that a trigger/interrupt is asserted after one CLK_SEL edge; a value of 1 means that a trigger/interrupt is asserted after two CLK_SEL edges; and so on.

In the TIM[i]_CHn_CNT register, the elapsed cycles were incremented and compared against TIM[i]_CHn_CNTS register. If the TIM[i]_CHn_CNT register value is ≥ to the TIM[i]_CHn_CNTS register value, a trigger will be asserted. So, the actual period time can be changed on the fly by writing a value to TIM[i]_CHn_CNTS register.

Operation is done depending on CMU clock, DSL, ISL bit and the input signal value defined in the following table:

Table 10-8. Operation

Input signal F_OUTx	Selected CMU Clock	External capture	ISL	DSL	Action description
–	1	0	1	–	If CNT == CNTS, then do capture; issue NEWVAL_IRQ; CNT = 0 else CNT++ endif
0	0	0	0	1	No
1	1	0	0	1	If CNT == CNTS, then do capture; issue NEWVAL_IRQ; CNT = 0 else CNT++ endif
0	0	–	0	1	No
0	1	0	0	0	If CNT == CNTS, then do capture; issue NEWVAL_IRQ; CNT = 0 else CNT++ endif
1	0	0	0	0	No
–	0	0	–	–	No

In this mode. the TIM[i]_CHn_GPR1 operates as a shadow register for TIM[i]_CHn_CNTS. This allows the period for the next sampling period to be specified. The update of TIM[i]_CHn_CNTS will only take place once on a trigger if the TIM[i]_CHn_GPR1 was written by the CPU. This means that the captured value from the previous trigger can be read by the CPU from TIM[i]_CHn_GPR1 and, afterwards, the new sampling period for the next sampling period (the one after the actual sampling period) can be written.

With each asserted interrupt, the GPR0 and GPR1 registers are updated according to the GPR0_SEL, GPR1_SEL, EGPR0_SEL and EGPR1_SEL bits.

When selecting ECNT as a source for the capture registers, GPRx will show the edge count and the input signal value at point of capture. Selecting GPR0_SEL = '11' and EGPR0_SEL = '0' for TIM channel 0, all 8 TIM input signals will be captured to GPR0[7:0].

The TIM[i]_CHn_CTRL[CLK_SEL] bit field will select which CMU clock is used.

The behavior of the ECNT counter register is configurable by writing to the TIM[i]_CHn_CTRL[ECNT_RESET] bit. If set to 1 when each interrupt (period expired) is asserted, the ECNT will be reset. Otherwise, the ECNT counter operates in wrap around mode.

If the ECNT counter register produces an overflow during the measurement, the TIM[i]_CH[x]_IRQ_NOTIFY[ECNTOFL] bit is set and the TIM_ECNTOFLn_IRQ interrupt is asserted (if enabled).

10.4.2.6.1 External capture TIM gated periodic sampling mode (TGPS)

If external capture is enabled, the external capture events will capture GPRx, reset the CNT counter, and assert a NEWVAL_IRQ interrupt.

Operation is done depending on the:

- cmu clock,
- external capture signal,
- DSL bit, and
- ISL bit

per the following table.

Table 10-9. External capture

F_OUTx input signal	Selected CMU clock	External capture	ISL	DSL	Action description
–	1	0	1	–	If CNT == CNTS: do capture ,issue NEWVAL_IRQ, CNT=0; else CNT++ endif
0	0	0	0	1	None
1	1	0	0	1	If CNT == CNTS: do capture, issue NEWVAL_IRQ, CNT=0; else CNT++ endif
0	0	–	0	1	None
0	1	0	0	0	If CNT == CNTS: do capture, issue NEWVAL_IRQ, CNT=0; else CNT++ endif
1	0	0	0	0	None
–	0	0	–	–	None

10.5 MAP Submodule Interface

The GTM provides one dedicated TIM submodule, TIM0 to the MAP submodule. TIM0 channels zero through five are connected to the MAP submodule (see MAP Overview). Each TIM0 channel provides a 49-bit data signal (shown in Table 10-10) to the corresponding input of the MAP. Each 49-bit data signal is marked as valid with a separate valid signal, tim0_map_dval[n] (n:0..5).

Table 10-10. TIM0 49-bit output data signals to MAP input signals

49-bit data signal composition for each TIM0 channel	Signal description
tim0_map_data0(48)	signal level bit from tim0_ch0
tim0_map_data0(47:24)	actual filter value: TIM0_CH0_FLT_RE or TIM0_CH0_FLT_FE
tim0_map_data0(23:0)	time stamp value from CH0 GRP0 register
tim0_map_dval0	mark tim0_map_data0 valid for one clock cycle
tim0_map_data1(48)	signal level bit from tim0_ch1
tim0_map_data1(47:24)	actual filter value: TIM0_CH1_FLT_RE or TIM0_CH1_FLT_FE
tim0_map_data1(23:0)	time stamp value from CH1 GRP0 register
tim0_map_dval1	mark tim0_map_data0 valid for one clock cycle
tim0_map_data2(48)	signal level bit from tim0_ch2
tim0_map_data2(47:24)	actual filter value: TIM0_CH2_FLT_RE or TIM0_CH2_FLT_FE
tim0_map_data2(23:0)	time stamp value from CH2 GRP0 register
tim0_map_dval2	mark tim0_map_data2 valid for one clock cycle
tim0_map_data3(48)	signal level bit from tim0_ch3
tim0_map_data3(47:24)	actual filter value: TIM0_CH3_FLT_RE or TIM0_CH3_FLT_FE
tim0_map_data3(23:0)	time stamp value from CH3 GRP0 register
tim0_map_dval3	mark tim0_map_data3 valid for one clock cycle
tim0_map_data4(48)	signal level bit from tim0_ch4
tim0_map_data4(47:24)	actual filter value: TIM0_CH4_FLT_RE or TIM0_CH4_FLT_FE
tim0_map_data4(23:0)	time stamp value from CH4 GRP0 register
tim0_map_dval4	mark tim0_map_data4 valid for one clock cycle
tim0_map_data5(48)	signal level bit from tim0_ch5
tim0_map_data5(47:24)	actual filter value: TIM0_CH5_FLT_RE or TIM0_CH5_FLT_FE
tim0_map_data5(23:0)	time stamp value from CH5 GRP0 register
tim0_map_dval5	mark tim0_map_data5 valid for one clock cycle

NOTE

With TIM_EN=1, the MAP interface starts operation. It is not dependent on the setting of the bitfields TIM_MODE, ISL, DSL.

NOTE

While the MAP interface is in use, the following guidelines have to be fullfilled, otherwise inconsistent filter values can be transfered:

- Change TIM0_CH[n]_FLT_RE only between occurance of rising and falling edge.
- Change TIM0_CH[n]_FLT_FE only between occurance of falling and rising edge.

10.6 TIM Interrupt Signals

The TIM submodule provides six interrupt signals per channel, as shown in Table 10-11, where m is the number of channels.

Table 10-11. TIM interrupt signals

Signal	Description
NEWVAL_IRQ	New measurement value detected by SMU of channel n (n:0…m–1)
ECNTOFL_IRQ	ECNT counter overflow of channel n (n:0…m–1)
CNTOFL_IRQ	SMU CNT counter overflow of channel n (n:0…m–1)
GPROFL_IRQ	GPR0 and GPR1 data overflow, old data was not read out before new data arrived at input of channel n (n:0…m–1)
TODET_IRQ	Time out reached for input signal of channel n (n:0…m–1)
GLITCHDET_IRQ	A glitch was detected by the TIM filter of channel n (n:0…m–1)

10.7 Memory Map and Registers

The Timer Input Module (TIM0) registers are described as follows:

TIM_0 memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	TIM0 Channel n General Purpose Register 0, n=0:7 (TIM_0_CH0_GPR0)	32	R	0000_0000h	10.7.1/197
4	TIM0 Channel n General Purpose Register 1, n=0:7 (TIM_0_CH0_GPR1)	32	R	0000_0000h	10.7.2/198
8	TIM0 Channel n Count Register, n=0:7 (TIM_0_CH0_CNT)	32	R	0000_0000h	10.7.3/198
C	TIM0 Channel n Edge Counter Register, n=0:7 (TIM_0_CH0_ECNT)	32	R	0000_0000h	10.7.4/199
10	TIM0 Channel n Counter Shadow Register, n=0:7 (TIM_0_CH0_CNTS)	32	R/W	0000_0000h	10.7.5/199
14	TIM0 Channel n TDUC Register, n=0:7 (TIM_0_CH0_TDUC)	32	R	0000_0000h	10.7.6/200
18	TIM0 Channel n TDUV Register, n=0:7 (TIM_0_CH0_TDUV)	32	R/W	0000_0000h	10.7.7/200
1C	TIM0 Channel n Filter Parameter 0 Register, n=0:7 (TIM_0_CH0_FLT_RE)	32	R/W	0000_0000h	10.7.8/201
20	TIM0 Channel n Filter Parameter 1 Register, n=0:7 (TIM_0_CH0_FLT_FE)	32	R/W	0000_0000h	10.7.9/201
24	TIM0 Channel n Control Register, n=0:7 (TIM_0_CH0_CTRL)	32	R/W	0000_0000h	10.7.10/202
28	TIM0 Channel n External Control Register, n=0:7 (TIM_0_CH0_ECTRL)	32	R/W	0000_0000h	10.7.11/206
2C	TIM0 Channel n Interrupt Request Notification Register, n=0:7 (TIM_0_CH0_IRQ_NOTIFY)	32	w1c	0000_0000h	10.7.12/207
30	TIM0 Channel n Interrupt Request Enable Register, n=0:7 (TIM_0_CH0_IRQ_EN)	32	R/W	0000_0000h	10.7.13/208
34	TIM0 Channel n Force Interrupt Request Register, n=0:7 (TIM_0_CH0_IRQ_FORCINT)	32	R/W	0000_0000h	10.7.14/210
38	TIM0 Channel n Interrupt Request Mode Register, n=0:7 (TIM_0_CH0_IRQ_MODE)	32	R/W	See section	10.7.15/211
3C	TIM0 Channel Error Interrupt Request Enable Register, n=0:7 (TIM_0_CH0_EIRQ_EN)	32	R/W	0000_0000h	10.7.16/213
78	TIM0 Input Source Register (TIM_0_IN_SRC)	32	R/W	0000_0000h	10.7.17/214
7C	TIM0 Channel Reset Register (TIM_0_RST)	32	R/W	0000_0000h	10.7.18/221
80	TIM0 Channel n General Purpose Register 0, n=0:7 (TIM_0_CH1_GPR0)	32	R	0000_0000h	10.7.1/197
84	TIM0 Channel n General Purpose Register 1, n=0:7 (TIM_0_CH1_GPR1)	32	R	0000_0000h	10.7.2/198
88	TIM0 Channel n Count Register, n=0:7 (TIM_0_CH1_CNT)	32	R	0000_0000h	10.7.3/198
8C	TIM0 Channel n Edge Counter Register, n=0:7 (TIM_0_CH1_ECNT)	32	R	0000_0000h	10.7.4/199
90	TIM0 Channel n Counter Shadow Register, n=0:7 (TIM_0_CH1_CNTS)	32	R/W	0000_0000h	10.7.5/199
94	TIM0 Channel n TDUC Register, n=0:7 (TIM_0_CH1_TDUC)	32	R	0000_0000h	10.7.6/200
98	TIM0 Channel n TDUV Register, n=0:7 (TIM_0_CH1_TDUV)	32	R/W	0000_0000h	10.7.7/200
9C	TIM0 Channel n Filter Parameter 0 Register, n=0:7 (TIM_0_CH1_FLT_RE)	32	R/W	0000_0000h	10.7.8/201

Table continues on the next page...

TIM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
A0	TIM0 Channel n Filter Parameter 1 Register, n=0:7 (TIM_0_CH1_FLT_FE)	32	R/W	0000_0000h	10.7.9/201
A4	TIM0 Channel n Control Register, n=0:7 (TIM_0_CH1_CTRL)	32	R/W	0000_0000h	10.7.10/202
A8	TIM0 Channel n External Control Register, n=0:7 (TIM_0_CH1_ECTRL)	32	R/W	0000_0000h	10.7.11/206
AC	TIM0 Channel n Interrupt Request Notification Register, n=0:7 (TIM_0_CH1_IRQ_NOTIFY)	32	w1c	0000_0000h	10.7.12/207
B0	TIM0 Channel n Interrupt Request Enable Register, n=0:7 (TIM_0_CH1_IRQ_EN)	32	R/W	0000_0000h	10.7.13/208
B4	TIM0 Channel n Force Interrupt Request Register, n=0:7 (TIM_0_CH1_IRQ_FORCINT)	32	R/W	0000_0000h	10.7.14/210
B8	TIM0 Channel n Interrupt Request Mode Register, n=0:7 (TIM_0_CH1_IRQ_MODE)	32	R/W	See section	10.7.15/211
BC	TIM0 Channel Error Interrupt Request Enable Register, n=0:7 (TIM_0_CH1_EIRQ_EN)	32	R/W	0000_0000h	10.7.16/213
100	TIM0 Channel n General Purpose Register 0, n=0:7 (TIM_0_CH2_GPR0)	32	R	0000_0000h	10.7.1/197
104	TIM0 Channel n General Purpose Register 1, n=0:7 (TIM_0_CH2_GPR1)	32	R	0000_0000h	10.7.2/198
108	TIM0 Channel n Count Register, n=0:7 (TIM_0_CH2_CNT)	32	R	0000_0000h	10.7.3/198
10C	TIM0 Channel n Edge Counter Register, n=0:7 (TIM_0_CH2_ECNT)	32	R	0000_0000h	10.7.4/199
110	TIM0 Channel n Counter Shadow Register, n=0:7 (TIM_0_CH2_CNTS)	32	R/W	0000_0000h	10.7.5/199
114	TIM0 Channel n TDUC Register, n=0:7 (TIM_0_CH2_TDUC)	32	R	0000_0000h	10.7.6/200
118	TIM0 Channel n TDUV Register, n=0:7 (TIM_0_CH2_TDUV)	32	R/W	0000_0000h	10.7.7/200
11C	TIM0 Channel n Filter Parameter 0 Register, n=0:7 (TIM_0_CH2_FLT_RE)	32	R/W	0000_0000h	10.7.8/201
120	TIM0 Channel n Filter Parameter 1 Register, n=0:7 (TIM_0_CH2_FLT_FE)	32	R/W	0000_0000h	10.7.9/201
124	TIM0 Channel n Control Register, n=0:7 (TIM_0_CH2_CTRL)	32	R/W	0000_0000h	10.7.10/202
128	TIM0 Channel n External Control Register, n=0:7 (TIM_0_CH2_ECTRL)	32	R/W	0000_0000h	10.7.11/206
12C	TIM0 Channel n Interrupt Request Notification Register, n=0:7 (TIM_0_CH2_IRQ_NOTIFY)	32	w1c	0000_0000h	10.7.12/207
130	TIM0 Channel n Interrupt Request Enable Register, n=0:7 (TIM_0_CH2_IRQ_EN)	32	R/W	0000_0000h	10.7.13/208
134	TIM0 Channel n Force Interrupt Request Register, n=0:7 (TIM_0_CH2_IRQ_FORCINT)	32	R/W	0000_0000h	10.7.14/210
138	TIM0 Channel n Interrupt Request Mode Register, n=0:7 (TIM_0_CH2_IRQ_MODE)	32	R/W	See section	10.7.15/211
13C	TIM0 Channel Error Interrupt Request Enable Register, n=0:7 (TIM_0_CH2_EIRQ_EN)	32	R/W	0000_0000h	10.7.16/213

Table continues on the next page...

TIM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
180	TIM0 Channel n General Purpose Register 0, n=0:7 (TIM_0_CH3_GPR0)	32	R	0000_0000h	10.7.1/197
184	TIM0 Channel n General Purpose Register 1, n=0:7 (TIM_0_CH3_GPR1)	32	R	0000_0000h	10.7.2/198
188	TIM0 Channel n Count Register, n=0:7 (TIM_0_CH3_CNT)	32	R	0000_0000h	10.7.3/198
18C	TIM0 Channel n Edge Counter Register, n=0:7 (TIM_0_CH3_ECNT)	32	R	0000_0000h	10.7.4/199
190	TIM0 Channel n Counter Shadow Register, n=0:7 (TIM_0_CH3_CNTS)	32	R/W	0000_0000h	10.7.5/199
194	TIM0 Channel n TDUC Register, n=0:7 (TIM_0_CH3_TDUC)	32	R	0000_0000h	10.7.6/200
198	TIM0 Channel n TDUV Register, n=0:7 (TIM_0_CH3_TDUV)	32	R/W	0000_0000h	10.7.7/200
19C	TIM0 Channel n Filter Parameter 0 Register, n=0:7 (TIM_0_CH3_FLT_RE)	32	R/W	0000_0000h	10.7.8/201
1A0	TIM0 Channel n Filter Parameter 1 Register, n=0:7 (TIM_0_CH3_FLT_FE)	32	R/W	0000_0000h	10.7.9/201
1A4	TIM0 Channel n Control Register, n=0:7 (TIM_0_CH3_CTRL)	32	R/W	0000_0000h	10.7.10/202
1A8	TIM0 Channel n External Control Register, n=0:7 (TIM_0_CH3_ECTRL)	32	R/W	0000_0000h	10.7.11/206
1AC	TIM0 Channel n Interrupt Request Notification Register, n=0:7 (TIM_0_CH3_IRQ_NOTIFY)	32	w1c	0000_0000h	10.7.12/207
1B0	TIM0 Channel n Interrupt Request Enable Register, n=0:7 (TIM_0_CH3_IRQ_EN)	32	R/W	0000_0000h	10.7.13/208
1B4	TIM0 Channel n Force Interrupt Request Register, n=0:7 (TIM_0_CH3_IRQ_FORCINT)	32	R/W	0000_0000h	10.7.14/210
1B8	TIM0 Channel n Interrupt Request Mode Register, n=0:7 (TIM_0_CH3_IRQ_MODE)	32	R/W	See section	10.7.15/211
1BC	TIM0 Channel Error Interrupt Request Enable Register, n=0:7 (TIM_0_CH3_EIRQ_EN)	32	R/W	0000_0000h	10.7.16/213
200	TIM0 Channel n General Purpose Register 0, n=0:7 (TIM_0_CH4_GPR0)	32	R	0000_0000h	10.7.1/197
204	TIM0 Channel n General Purpose Register 1, n=0:7 (TIM_0_CH4_GPR1)	32	R	0000_0000h	10.7.2/198
208	TIM0 Channel n Count Register, n=0:7 (TIM_0_CH4_CNT)	32	R	0000_0000h	10.7.3/198
20C	TIM0 Channel n Edge Counter Register, n=0:7 (TIM_0_CH4_ECNT)	32	R	0000_0000h	10.7.4/199
210	TIM0 Channel n Counter Shadow Register, n=0:7 (TIM_0_CH4_CNTS)	32	R/W	0000_0000h	10.7.5/199
214	TIM0 Channel n TDUC Register, n=0:7 (TIM_0_CH4_TDUC)	32	R	0000_0000h	10.7.6/200
218	TIM0 Channel n TDUV Register, n=0:7 (TIM_0_CH4_TDUV)	32	R/W	0000_0000h	10.7.7/200
21C	TIM0 Channel n Filter Parameter 0 Register, n=0:7 (TIM_0_CH4_FLT_RE)	32	R/W	0000_0000h	10.7.8/201
220	TIM0 Channel n Filter Parameter 1 Register, n=0:7 (TIM_0_CH4_FLT_FE)	32	R/W	0000_0000h	10.7.9/201

Table continues on the next page...

TIM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
224	TIM0 Channel n Control Register, n=0:7 (TIM_0_CH4_CTRL)	32	R/W	0000_0000h	10.7.10/202
228	TIM0 Channel n External Control Register, n=0:7 (TIM_0_CH4_ECTRL)	32	R/W	0000_0000h	10.7.11/206
22C	TIM0 Channel n Interrupt Request Notification Register, n=0:7 (TIM_0_CH4_IRQ_NOTIFY)	32	w1c	0000_0000h	10.7.12/207
230	TIM0 Channel n Interrupt Request Enable Register, n=0:7 (TIM_0_CH4_IRQ_EN)	32	R/W	0000_0000h	10.7.13/208
234	TIM0 Channel n Force Interrupt Request Register, n=0:7 (TIM_0_CH4_IRQ_FORCINT)	32	R/W	0000_0000h	10.7.14/210
238	TIM0 Channel n Interrupt Request Mode Register, n=0:7 (TIM_0_CH4_IRQ_MODE)	32	R/W	See section	10.7.15/211
23C	TIM0 Channel Error Interrupt Request Enable Register, n=0:7 (TIM_0_CH4_EIRQ_EN)	32	R/W	0000_0000h	10.7.16/213
280	TIM0 Channel n General Purpose Register 0, n=0:7 (TIM_0_CH5_GPR0)	32	R	0000_0000h	10.7.1/197
284	TIM0 Channel n General Purpose Register 1, n=0:7 (TIM_0_CH5_GPR1)	32	R	0000_0000h	10.7.2/198
288	TIM0 Channel n Count Register, n=0:7 (TIM_0_CH5_CNT)	32	R	0000_0000h	10.7.3/198
28C	TIM0 Channel n Edge Counter Register, n=0:7 (TIM_0_CH5_ECNT)	32	R	0000_0000h	10.7.4/199
290	TIM0 Channel n Counter Shadow Register, n=0:7 (TIM_0_CH5_CNTS)	32	R/W	0000_0000h	10.7.5/199
294	TIM0 Channel n TDUC Register, n=0:7 (TIM_0_CH5_TDUC)	32	R	0000_0000h	10.7.6/200
298	TIM0 Channel n TDUV Register, n=0:7 (TIM_0_CH5_TDUV)	32	R/W	0000_0000h	10.7.7/200
29C	TIM0 Channel n Filter Parameter 0 Register, n=0:7 (TIM_0_CH5_FLT_RE)	32	R/W	0000_0000h	10.7.8/201
2A0	TIM0 Channel n Filter Parameter 1 Register, n=0:7 (TIM_0_CH5_FLT_FE)	32	R/W	0000_0000h	10.7.9/201
2A4	TIM0 Channel n Control Register, n=0:7 (TIM_0_CH5_CTRL)	32	R/W	0000_0000h	10.7.10/202
2A8	TIM0 Channel n External Control Register, n=0:7 (TIM_0_CH5_ECTRL)	32	R/W	0000_0000h	10.7.11/206
2AC	TIM0 Channel n Interrupt Request Notification Register, n=0:7 (TIM_0_CH5_IRQ_NOTIFY)	32	w1c	0000_0000h	10.7.12/207
2B0	TIM0 Channel n Interrupt Request Enable Register, n=0:7 (TIM_0_CH5_IRQ_EN)	32	R/W	0000_0000h	10.7.13/208
2B4	TIM0 Channel n Force Interrupt Request Register, n=0:7 (TIM_0_CH5_IRQ_FORCINT)	32	R/W	0000_0000h	10.7.14/210
2B8	TIM0 Channel n Interrupt Request Mode Register, n=0:7 (TIM_0_CH5_IRQ_MODE)	32	R/W	See section	10.7.15/211
2BC	TIM0 Channel Error Interrupt Request Enable Register, n=0:7 (TIM_0_CH5_EIRQ_EN)	32	R/W	0000_0000h	10.7.16/213
300	TIM0 Channel n General Purpose Register 0, n=0:7 (TIM_0_CH6_GPR0)	32	R	0000_0000h	10.7.1/197

Table continues on the next page...

TIM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
304	TIM0 Channel n General Purpose Register 1, n=0:7 (TIM_0_CH6_GPR1)	32	R	0000_0000h	10.7.2/198
308	TIM0 Channel n Count Register, n=0:7 (TIM_0_CH6_CNT)	32	R	0000_0000h	10.7.3/198
30C	TIM0 Channel n Edge Counter Register, n=0:7 (TIM_0_CH6_ECNT)	32	R	0000_0000h	10.7.4/199
310	TIM0 Channel n Counter Shadow Register, n=0:7 (TIM_0_CH6_CNTS)	32	R/W	0000_0000h	10.7.5/199
314	TIM0 Channel n TDUC Register, n=0:7 (TIM_0_CH6_TDUC)	32	R	0000_0000h	10.7.6/200
318	TIM0 Channel n TDUV Register, n=0:7 (TIM_0_CH6_TDUV)	32	R/W	0000_0000h	10.7.7/200
31C	TIM0 Channel n Filter Parameter 0 Register, n=0:7 (TIM_0_CH6_FLT_RE)	32	R/W	0000_0000h	10.7.8/201
320	TIM0 Channel n Filter Parameter 1 Register, n=0:7 (TIM_0_CH6_FLT_FE)	32	R/W	0000_0000h	10.7.9/201
324	TIM0 Channel n Control Register, n=0:7 (TIM_0_CH6_CTRL)	32	R/W	0000_0000h	10.7.10/202
328	TIM0 Channel n External Control Register, n=0:7 (TIM_0_CH6_ECTRL)	32	R/W	0000_0000h	10.7.11/206
32C	TIM0 Channel n Interrupt Request Notification Register, n=0:7 (TIM_0_CH6_IRQ_NOTIFY)	32	w1c	0000_0000h	10.7.12/207
330	TIM0 Channel n Interrupt Request Enable Register, n=0:7 (TIM_0_CH6_IRQ_EN)	32	R/W	0000_0000h	10.7.13/208
334	TIM0 Channel n Force Interrupt Request Register, n=0:7 (TIM_0_CH6_IRQ_FORCINT)	32	R/W	0000_0000h	10.7.14/210
338	TIM0 Channel n Interrupt Request Mode Register, n=0:7 (TIM_0_CH6_IRQ_MODE)	32	R/W	See section	10.7.15/211
33C	TIM0 Channel Error Interrupt Request Enable Register, n=0:7 (TIM_0_CH6_EIRQ_EN)	32	R/W	0000_0000h	10.7.16/213
380	TIM0 Channel n General Purpose Register 0, n=0:7 (TIM_0_CH7_GPR0)	32	R	0000_0000h	10.7.1/197
384	TIM0 Channel n General Purpose Register 1, n=0:7 (TIM_0_CH7_GPR1)	32	R	0000_0000h	10.7.2/198
388	TIM0 Channel n Count Register, n=0:7 (TIM_0_CH7_CNT)	32	R	0000_0000h	10.7.3/198
38C	TIM0 Channel n Edge Counter Register, n=0:7 (TIM_0_CH7_ECNT)	32	R	0000_0000h	10.7.4/199
390	TIM0 Channel n Counter Shadow Register, n=0:7 (TIM_0_CH7_CNTS)	32	R/W	0000_0000h	10.7.5/199
394	TIM0 Channel n TDUC Register, n=0:7 (TIM_0_CH7_TDUC)	32	R	0000_0000h	10.7.6/200
398	TIM0 Channel n TDUV Register, n=0:7 (TIM_0_CH7_TDUV)	32	R/W	0000_0000h	10.7.7/200
39C	TIM0 Channel n Filter Parameter 0 Register, n=0:7 (TIM_0_CH7_FLT_RE)	32	R/W	0000_0000h	10.7.8/201
3A0	TIM0 Channel n Filter Parameter 1 Register, n=0:7 (TIM_0_CH7_FLT_FE)	32	R/W	0000_0000h	10.7.9/201
3A4	TIM0 Channel n Control Register, n=0:7 (TIM_0_CH7_CTRL)	32	R/W	0000_0000h	10.7.10/202

Table continues on the next page...

TIM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3A8	TIM0 Channel n External Control Register, n=0:7 (TIM_0_CH7_ECTRL)	32	R/W	0000_0000h	10.7.11/206
3AC	TIM0 Channel n Interrupt Request Notification Register, n=0:7 (TIM_0_CH7_IRQ_NOTIFY)	32	w1c	0000_0000h	10.7.12/207
3B0	TIM0 Channel n Interrupt Request Enable Register, n=0:7 (TIM_0_CH7_IRQ_EN)	32	R/W	0000_0000h	10.7.13/208
3B4	TIM0 Channel n Force Interrupt Request Register, n=0:7 (TIM_0_CH7_IRQ_FORCINT)	32	R/W	0000_0000h	10.7.14/210
3B8	TIM0 Channel n Interrupt Request Mode Register, n=0:7 (TIM_0_CH7_IRQ_MODE)	32	R/W	See section	10.7.15/211
3BC	TIM0 Channel Error Interrupt Request Enable Register, n=0:7 (TIM_0_CH7_EIRQ_EN)	32	R/W	0000_0000h	10.7.16/213

10.7.1 TIM0 Channel n General Purpose Register 0, n=0:7 (TIM_0_CH*n*_GPR0)

Address: 1000h base + 0h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	ECNT								GPR0																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIM_0_CH*n*_GPR0 field descriptions

Field	Description
0–7 ECNT	Edge counter. The ECNT counts every incoming filtered edge (rising and falling). The counter value is uneven in case of detected rising, and even in case of detected falling edge. Thus, the input signal level is part of the counter and can be obtained by bit 0 of ECNT. **NOTE:** The ECNT register is reset to its initial value when the channel is enabled. Please note, bit 0 depends on the input level coming from the filter unit and defines the reset value immediately.
8–31 GPR0	Input signal characteristic parameter 0. **NOTE:** The content of this register has different meaning for the TIM channels modes. The content directly depends on the bit field TIM[i]_CH[n]_CTRL[GPR0_SEL].

10.7.2 TIM0 Channel n General Purpose Register 1, n=0:7 (TIM_0_CH*n*_GPR1)

Address: 1000h base + 4h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	ECNT								GPR1																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIM_0_CH*n*_GPR1 field descriptions

Field	Description
0–7 ECNT	Edge counter. The ECNT counts every incoming filtered edge (rising and falling). The counter value is uneven in case of detected rising, and even in case of detected falling edge. Thus, the input signal level is part of the counter and can be obtained by bit 0 of ECNT. **NOTE:** The ECNT register is reset to its initial value when the channel is enabled. Please note, that bit 0 depends on the input level coming from the filter unit and defines the reset value immediately.
8–31 GPR1	Input signal characteristic parameter 1. **NOTE:** The content of this register has different meaning for the TIM channels modes. The content directly depends on the bit field TIM[i]_CH[n]_CTRL[GPR1_SEL]. **NOTE:** The content of this bit field can only be written in TIM channel mode TGPS.

10.7.3 TIM0 Channel n Count Register, n=0:7 (TIM_0_CH*n*_CNT)

Address: 1000h base + 8h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								CNT																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIM_0_CH*n*_CNT field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 CNT	Actual SMU counter value. The meaning of this value depends on the configured mode: TPWM Actual duration of PWM signal. TPIM Actual duration of all pulses (sum of pulses).

Table continues on the next page...

TIM_0_CH*n*_CNT field descriptions (continued)

Field	Description
	TIEM Actual number of received edges. TIPM Actual number of received edges.

10.7.4 TIM0 Channel n Edge Counter Register, n=0:7 (TIM_0_CH*n*_ECNT)

Address: 1000h base + Ch offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																ECNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIM_0_CH*n*_ECNT field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–31 ECNT	Edge counter. **NOTE:** If TIM channel is disabled, the content of ECNT gets frozen. A read will auto clear the bits [15:1]. Further read accesses to ECNT will show on bit 0 the actual input signal value of the channel.

10.7.5 TIM0 Channel n Counter Shadow Register, n=0:7 (TIM_0_CH*n*_CNTS)

Address: 1000h base + 10h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	ECNT								CNTS																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIM_0_CH*n*_CNTS field descriptions

Field	Description
0–7 ECNT	Edge counter. The ECNT counts every incoming filtered edge (rising and falling). The counter value is uneven in case of detected rising, and even in case of detected falling edge. Thus, the input signal level is part of the counter and can be obtained by bit 0 of ECNT.

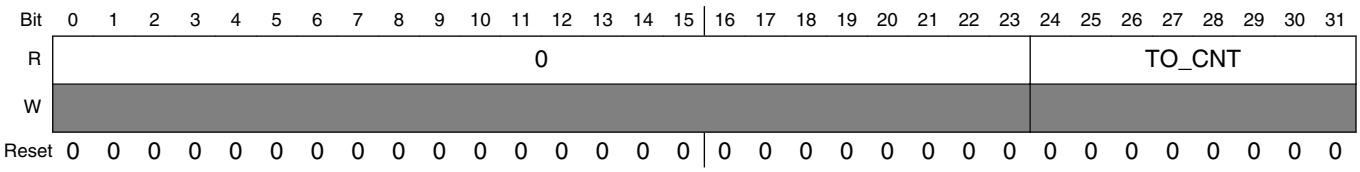
Table continues on the next page...

TIM_0_CH*n*_CNTS field descriptions (continued)

Field	Description
	NOTE: The ECNT register is reset to its initial value when the channel is enabled. Please note, that bit 0 depends on the input level coming from the filter unit and defines the reset value immediately.
8–31 CNTS	Counter shadow register. The content of this register has different meaning for the TIM channels modes. The content depends directly on the bit field CNTS_SEL of register TIM[i]_CH[n]_CTRL. **NOTE:** The register TIM[i]_CH[n]_CNTS is only writable in TIPM and TBCM modes.

10.7.6 TIM0 Channel n TDUC Register, n=0:7 (TIM_0_CH*n*_TDUC)

Address: 1000h base + 14h offset + (128d × i), where i=0d to 7d

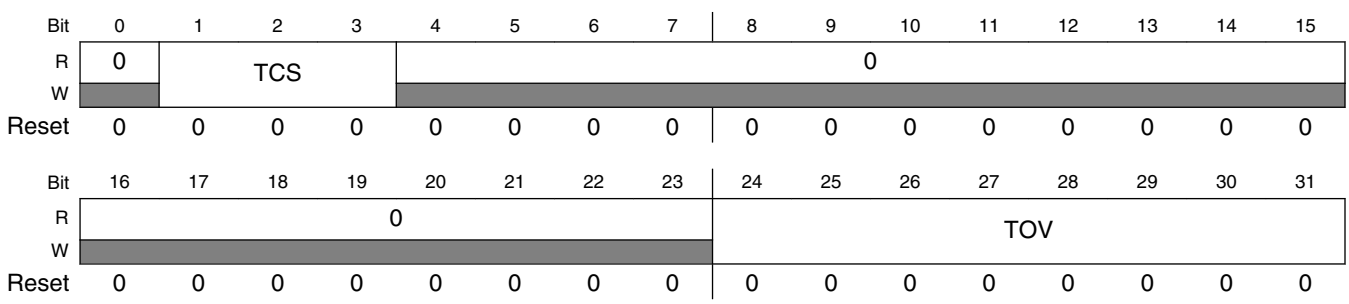


TIM_0_CH*n*_TDUC field descriptions

Field	Description
0–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24–31 TO_CNT	Current Timeout value for channel n (n:0...m-1).

10.7.7 TIM0 Channel n TDUV Register, n=0:7 (TIM_0_CH*n*_TDUV)

Address: 1000h base + 18h offset + (128d × i), where i=0d to 7d



TIM_0_CH*n*_TDUV field descriptions

Field	Description
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

TIM_0_CH*n*_TDUV field descriptions (continued)

Field	Description
1–3 TCS	Timeout Clock selection. 000 CMU_CLK0 selected 001 CMU_CLK2 selected 010 CMU_CLK2 selected 011 CMU_CLK3 selected 100 CMU_CLK4 selected 101 CMU_CLK5 selected 110 CMU_CLK6 selected 111 CMU_CLK7 selected
4–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24–31 TOV	Time out duration for channel n (n:0...m-1).

10.7.8 TIM0 Channel n Filter Parameter 0 Register, n=0:7 (TIM_0_CH*n*_FLT_RE)

Address: 1000h base + 1Ch offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								FLT_RE																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIM_0_CH*n*_FLT_RE field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 FLT_RE	Filter parameter for rising edge. This register has different meanings in the various filter modes: Immediate edge propagation mode = acceptance time for rising edge. Individual deglitch time mode = deglitch time for rising edge.

10.7.9 TIM0 Channel n Filter Parameter 1 Register, n=0:7 (TIM_0_CH*n*_FLT_FE)

Address: 1000h base + 20h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								FLT_FE																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIM_0_CH*n*_FLT_FE field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 FLT_FE	Filter parameter for falling edge. This register has different meanings in the various filter modes: Immediate edge propagation mode = acceptance time for falling edge. Individual deglitch time mode = deglitch time for falling edge.

10.7.10 TIM0 Channel n Control Register, n=0:7 (TIM_0_CH*n*_CTRL)

Address: 1000h base + 24h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	TOCTRL		EGPR1_SEL	EGPR0_SEL	FR_ECNT_OFL	CLK_SEL			FLT_CTR_FE	FLT_MODE_FE	FLT_CTR_RE	FLT_MODE_RE	EXT_CAP_EN	FLT_CNT_FRQ		FLT_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	ECNT_RESET	ISL	DSL	CNTS_SEL	GPR1_SEL		GPR0_SEL		TBU0_SEL	CICTRL	ARU_EN	OSM	TIM_MODE			TIM_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIM_0_CH*n*_CTRL field descriptions

Field	Description
0–1 TOCTRL	Timeout control 00 Timeout feature disabled 01 Timeout feature enabled for rising edge only 10 Timeout feature enabled for falling edge only 11 Timeout feature enabled for both edges
2 EGPR1_SEL	Extension of GPR1_SEL bit field. Details are described in the GPR1_SEL bit field.
3 EGPR0_SEL	Extension of GPR0_SEL bit field. Details are described in the GPR0_SEL bit field.
4 FR_ECNT_OFL	Extended Edge counter overflow behavior. 0 Overflow will be signaled on ECNT bit width = 8 1 Overflow will be signaled on EECNT bit width (full range)

Table continues on the next page...

TIM_0_CH*n*_CTRL field descriptions (continued)

Field	Description
5–7 CLK_SEL	CMU clock source select for channel. 000 CMU_CLK0 selected 001 CMU_CLK1 selected 010 CMU_CLK2 selected 011 CMU_CLK3 selected 100 CMU_CLK4 selected 101 CMU_CLK5 selected 110 CMU_CLK6 selected 111 CMU_CLK7 selected
8 FLT_CTR_FE	Filter counter mode for falling edge. 0 Up/Down Counter. **NOTE:** This bit is only applicable in Individual Deglitch Time Mode 1 Hold Counter. **NOTE:** This bit is only applicable in Individual Deglitch Time Mode
9 FLT_MODE_FE	Filter mode for falling edge. 0 Immediate edge propagation mode. 1 Individual de-glitch mode.
10 FLT_CTR_RE	Filter counter mode for rising edge. **NOTE:** This bit is only applicable in Individual Deglitch Time Mode. 0 Up/Down Counter. 1 Hold Counter.
11 FLT_MODE_RE	Filter mode for rising edge. 0 Immediate edge propagation mode. 1 Individual de-glitch mode.
12 EXT_CAP_EN	Enables external capture mode. The selected TIM mode is only sensitive to external capture pulses, the input event changes are ignored. 0 External capture disabled 1 External capture enabled
13–14 FLT_CNT_FRQ	Filter counter frequency select. 00 FLT_CNT counts with CMU_CLK0 01 FLT_CNT counts with CMU_CLK1 10 FLT_CNT counts with CMU_CLK6 11 FLT_CNT counts with CMU_CLK7
15 FLT_EN	Filter enable for channel n. **NOTE:** If the filter is disabled, all filter related units (including CSU) are bypassed, which means that the F_IN signal is directly routed to F_OUT signal.

Table continues on the next page...

TIM_0_CH*n*_CTRL field descriptions (continued)

Field	Description
	0 Filter disabled and internal states are reset. 1 Filter enabled.
16 ECNT_RESET	Enables resetting the ECNT counter in periodic sampling mode. 0 ECNT counter operating in wrap around mode 1 ECNT counter is reset with periodic sampling
17 ISL	Ignore Signal Level. **NOTE:** This bit is only applicable in Input Event mode (TIEM and TIPM). 0 Use DSL bit for selecting active signal level 1 Ignore DSL and treat both edges as active-edge
18 DSL	Signal level control. 0 Measurement starts with falling edge (low level measurement). 1 Measurement starts with rising edge (high level measurement).
19 CNTS_SEL	Selection for CNTS register. **NOTE:** The functionality of the CNTS_SEL bit is disabled in the TIPM and TBCM modes. 0 Use CNT register as input. 1 Use TBU_TS0 as input.
20–21 GPR1_SEL	Selection for GPR1 register. **NOTE:** If a reserved value is written to the EGPR1_SEL, GPR1_SEL bitfields , the hardware will use TBU_TS0 input. If EGPR1_SEL =0: 00 Use TBU_TS0 as input 01 Use TBU_TS1 as input 10 Use TBU_TS2 as input 11 Use CNT as input If EGPR1_SEL =1: 00 Use ECNT as input 01 Reserved 10 Reserved 11 Reserved
22–23 GPR0_SEL	Selection for GPR0 register. **NOTE:** If a reserved value is written to the EGPR0_SEL, GPR0_SEL bit fields , the hardware will use TBU_TS0 input. If EGPR0_SEL =0: 00 Use TBU_TS0 as input 01 Use TBU_TS1 as input

Table continues on the next page...

TIM_0_CH*n*_CTRL field descriptions (continued)

Field	Description
	10 Use TBU_TS2 as input 11 Use CNT as input If EGPR0_SEL =1: 00 Use ECNT as input 01 Reserved 10 Reserved 11 Reserved
24 TBU0_SEL	TBU_TS0 bits input select for TIM0_CHn_GPRz (z: 0, 1) **NOTE:** This bit is only applicable for TIM0 0 Use TBU_TS0(23..0) to store in TIM0_CH[x]_GPR0 and TIM0_CH[x]_GPR1. 1 Use TBU_TS0(26..3) to store in TIM0_CH[x]_GPR0 and TIM0_CH[x]_GPR1.
25 CICTRL	Channel Input Control. 0 Use signal TIM_IN(x) as input for channel x. 1 Use signal TIM_IN(x-1) as input for channel x (or TIM_IN(7) if x is 0).
26 ARU_EN	GPR0 and GPR1 register values routed to ARU. 0 Registers content not routed. 1 Registers content routed.
27 OSM	One-shot mode. 0 Continuous operation mode. 1 One-shot mode. **NOTE:** After finishing the action in one-shot mode the TIM_EN bit is cleared automatically.
28–30 TIM_MODE	TIM channel n mode. **NOTE:** The Bit Compression Mode is only available in TIM channel 0. If this mode is selected in any other channels, the TIM_MODE = 000 (TPWM mode) is used instead. However, the register TIM_MODE is read with value 100. **NOTE:** If an undefined value is written to the TIM_MODE register, the hardware switches automatically to TIM_MODE = 000 (TPWM mode). **NOTE:** The TIM_MODE register should not be changed while the TIM channel is enabled. **NOTE:** If the TIM channel is enabled and operating in TPWM or TPIM mode after the first valid edge defined by DSL has occured, a reconfiguration of DSL,ISL,TIM_MODE will not change the channel behaviour. Reading these fields after reconfiguration will show the newly configured settings but the inital channel behaviour will not change. Only a disabling of the TIM channel by setting TIM_EN= 0 will stop the channel operation. 000 PWM Measurement Mode (TPWM). 001 Pulse Integration Mode (TPIM). 010 Input Event Mode (TIEM). 011 Input Prescaler Mode (TIPM). 100 Bit Compression Mode (TBCM).

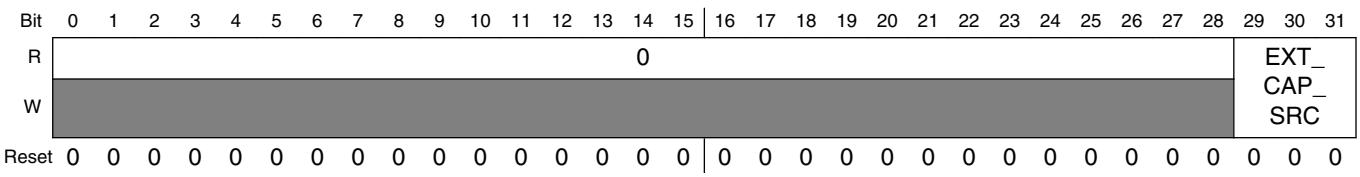
Table continues on the next page...

TIM_0_CH*n*_CTRL field descriptions (continued)

Field	Description
	NOTE: Bit Compression Mode is only available in TIM channel 0. If this mode is selected in any other channels, the TIM_MODE = 000 (TPWM mode) is used instead. However, the register TIM_MODE is read with value 100.
31 TIM_EN	TIM channel n enable. **NOTE:** Enabling of the channel resets the registers ECNT, TIM[i]_CH[n]_CNT, TIM[i]_CH[n]_GPR0, and TIM[i]_CH[n]_GPR1 to their reset values. **NOTE:** After finishing the action in one-shot mode the TIM_EN bit is cleared automatically. Otherwise, the bit must be cleared manually. 0 Channel disabled. 1 Channel enabled. **NOTE:** Enabling of the channel resets the ECNT, TIM[i]_CH[x]_CNT, TIM[i]_CH[x]_GPR0, and TIM[i]_CH[x]_GPR1registers to their reset values. **NOTE:** After finishing the action in one-shot mode, the TIM_EN bit is automatically cleared. Otherwise, the bit must be manually cleared.

10.7.11 TIM0 Channel n External Control Register, n=0:7 (TIM_0_CH*n*_ECTRL)

Address: 1000h base + 28h offset + (128d × i), where i=0d to 7d

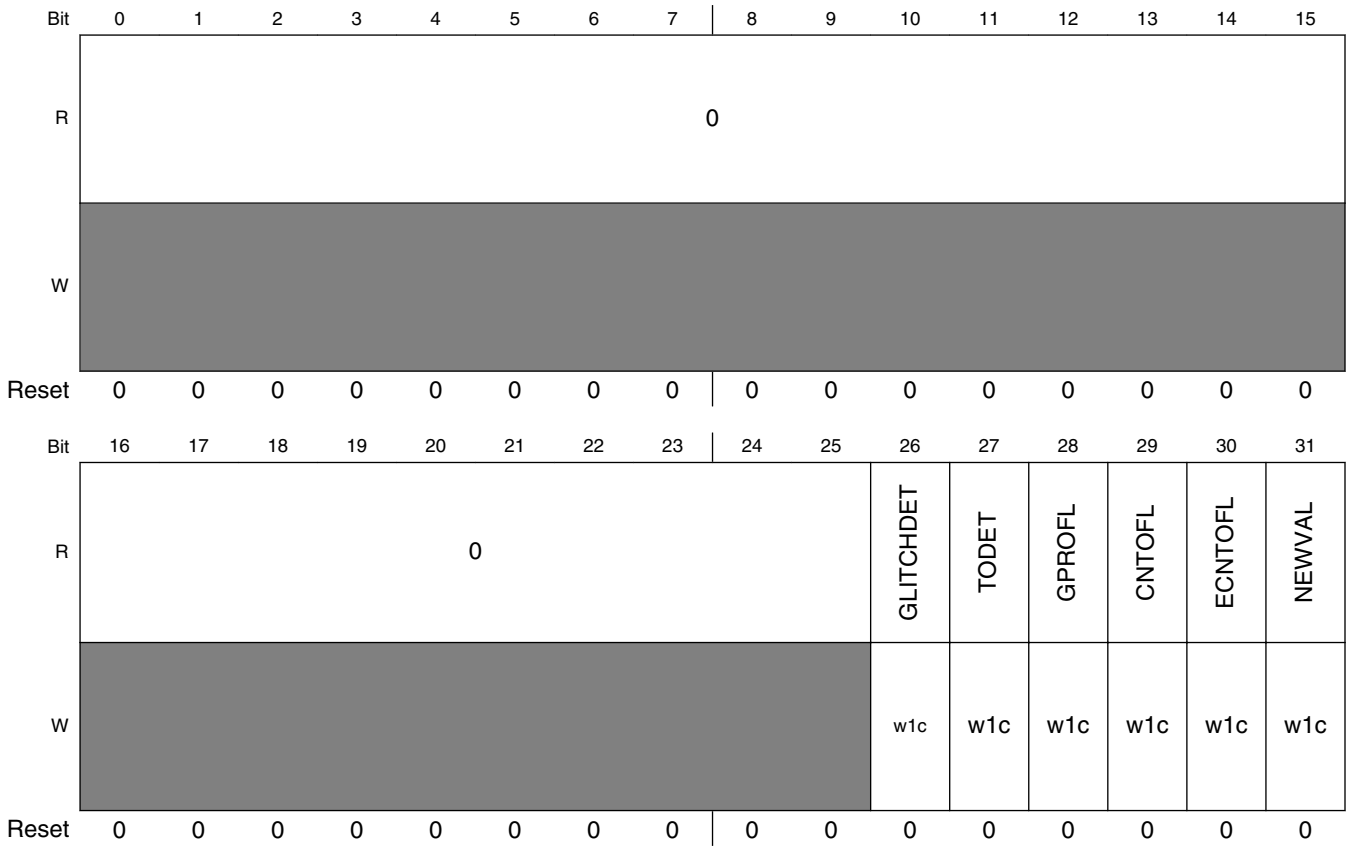


TIM_0_CH*n*_ECTRL field descriptions

Field	Description
0–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–31 EXT_CAP_SRC	Defines selected source for triggering the EXT_CAPTURE functionality. 000 NEW_VAL_IRQ of following channel selected. 001 AUX_IN selected. 010 CNTOFL_IRQ of following channel selected. 011 If CICTRL = 0, use signal TIM_IN(x-1) as input for channel n (or TIM_IN(m-1) if x is 0). If CICTRL = 1, use signal TIM_IN(x) as input for channel n. 100 ECNTOFL_IRQ of following channel selected. 101 TODET_IRQ of following channel selected. 110 GLITCHDET_IRQ of following channel selected. 111 GPROF_IRQ of following channel selected.

10.7.12 TIM0 Channel n Interrupt Request Notification Register, n=0:7 (TIM_0_CH*n*_IRQ_NOTIFY)

Address: 1000h base + 2Ch offset + (128d × i), where i=0d to 7d



TIM_0_CH*n*_IRQ_NOTIFY field descriptions

Field	Description
0–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 GLITCHDET	Glitch detected on channel n. This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged. 0 No glitch detected for last edge. 1 Glitch detected for last edge.
27 TODET	Timeout reached for input signal of channel n. This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged. 0 No event has occurred. 1 NEWVAL occurred on the TIM channel.
28 GPROFL	GPR0 and GPR1 data overflow, old data not read out before new data has arrived at input pin. This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Table continues on the next page...

TIM_0_CH*n*_IRQ_NOTIFY field descriptions (continued)

Field	Description
	0 No event has occurred. 1 Overflow occurred on the TIM channel.
29 CNTOFL	SMU CNT counter overflow of channel n. This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged. 0 No event has occurred. 1 Overflow occurred on the TIM channel.
30 ECNTOFL	ECNT counter overflow of channel n. This bit will be cleared on a CPU 0 No event has occurred. 1 Overflow occurred on the TIM channel.
31 NEWVAL	New measurement value detected by in channel n This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged. 0 No event has occurred. 1 NEWVAL occurred on the TIM channel.

10.7.13 TIM0 Channel n Interrupt Request Enable Register, n=0:7 (TIM_0_CH*n*_IRQ_EN)

Address: 1000h base + 30h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

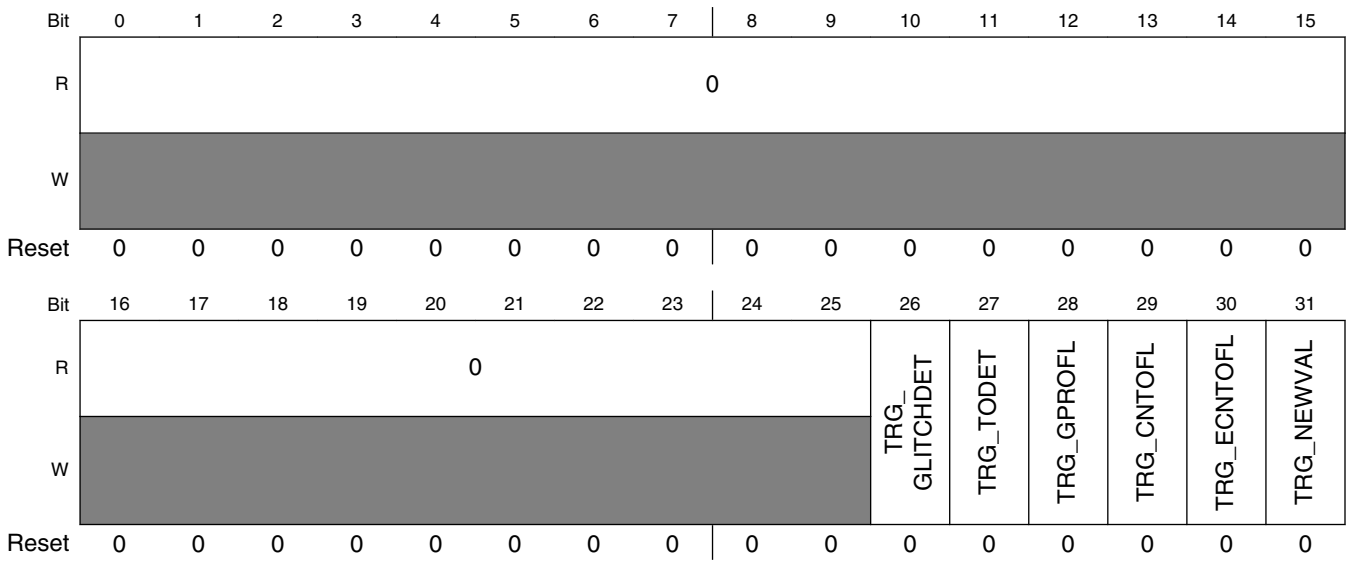
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0										GLITCHDET_IRQ_EN	TODET_IRQ_EN	GPROFL_IRQ_EN	CNTOFL_IRQ_EN	ECNTOFL_IRQ_EN	NEWVAL_IRQ_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIM_0_CH*n*_IRQ_EN field descriptions

Field	Description
0–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 GLITCHDET_IRQ_EN	TIM_GLITCHDETx_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM 1 Enable interrupt, interrupt is visible outside GTM
27 TODET_IRQ_EN	TIM_TODETn_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM 1 Enable interrupt, interrupt is visible outside GTM
28 GPROFL_IRQ_EN	TIM_GPROFL_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM 1 Enable interrupt, interrupt is visible outside GTM
29 CNTOFL_IRQ_EN	TIM_CNTOFLn_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM 1 Enable interrupt, interrupt is visible outside GTM
30 ECNTOFL_IRQ_EN	TIM_ECNTOFLn_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM 1 Enable interrupt, interrupt is visible outside GTM
31 NEWVAL_IRQ_EN	TIM_NEWVALn_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM 1 Enable interrupt, interrupt is visible outside GTM

10.7.14 TIM0 Channel n Force Interrupt Request Register, n=0:7 (TIM_0_CH*n*_IRQ_FORCINT)

Address: 1000h base + 34h offset + (128d × i), where i=0d to 7d



TIM_0_CH*n*_IRQ_FORCINT field descriptions

Field	Description
0–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 TRG_ GLITCHDET	Trigger bit TIM_CHn_IRQ_NOTIFY[GLITCHDET] by software. This bit is cleared automatically after write **NOTE:** This bit is write protected by bit GTM_CTRL[RF_PROT]. 0 No interrupt triggering. 1 Assert bit TIM_CHx_IRQ_NOTIFY[GLITCHDET].
27 TRG_TODET	Trigger bit TIM_CHn_IRQ_NOTIFY[TODET] by software. This bit is cleared automatically after write **NOTE:** This bit is write protected by bit GTM_CTRL[RF_PROT]. 0 No interrupt triggering. 1 Assert bit TIM_CHn_IRQ_NOTIFY[TODET].
28 TRG_GPROFL	Trigger bit TIM_CHn_IRQ_NOTIFY[GPROFL] by software. This bit is cleared automatically after write **NOTE:** This bit is write protected by bit GTM_CTRL[RF_PROT]. 0 No interrupt triggering. 1 Assert bit TIM_CHn_IRQ_NOTIFY[GPROFL].

Table continues on the next page...

TIM_0_CH*n*_IRQ_FORCINT field descriptions (continued)

Field	Description
29 TRG_CNTOFL	Trigger bit TIM_CHn_IRQ_NOTIFY[CNTOFL] by software. This bit is cleared automatically after write **NOTE:** This bit is write protected by bit GTM_CTRL[RF_PROT]. 0 No interrupt triggering. 1 Assert bit TIM_CHn_IRQ_NOTIFY[CNTOFL].
30 TRG_ECNTOFL	Trigger bit TIM_CHn_IRQ_NOTIFY[ECNTOFL] by software. This bit is cleared automatically after write **NOTE:** This bit is write protected by bit GTM_CTRL[RF_PROT]. 0 No interrupt triggering. 1 Assert bit TIM_CHn_IRQ_NOTIFY[ECNTOFL].
31 TRG_NEWVAL	Trigger bit TIM_CHn_IRQ_NOTIFY[NEWVAL] by software. This bit is cleared automatically after write **NOTE:** This bit is write protected by bit GTM_CTRL[RF_PROT]. 0 No interrupt triggering. 1 Assert bit TIM_CHn_IRQ_NOTIFY[NEWVAL].

10.7.15 TIM0 Channel n Interrupt Request Mode Register, n=0:7 (TIM_0_CH*n*_IRQ_MODE)

Address: 1000h base + 38h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0														IRQ_MODE	
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- Bits 28-31 do not reset to a given value.

TIM_0_CH*n*_IRQ_MODE field descriptions

Field	Description
0–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

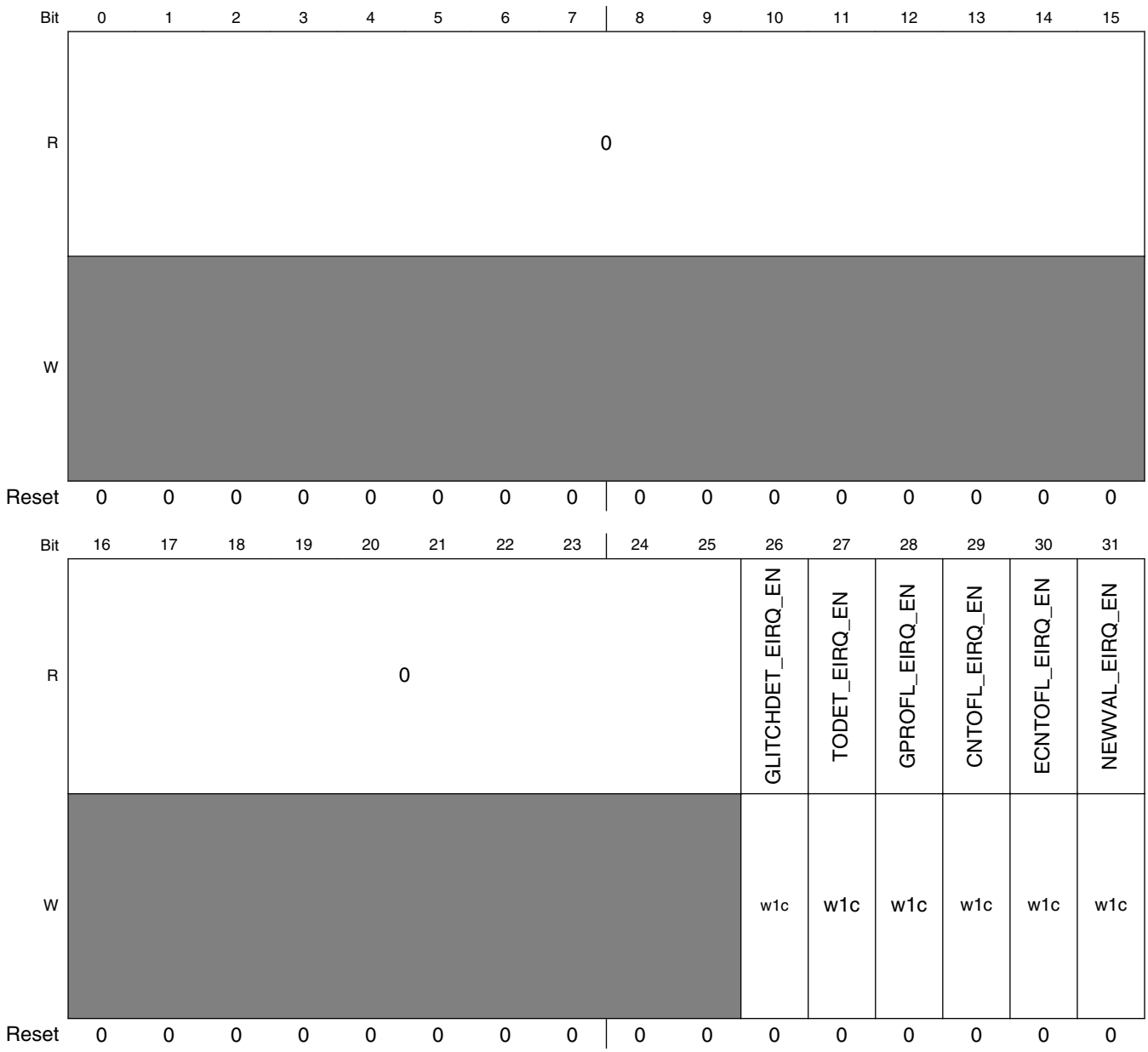
Table continues on the next page...

TIM_0_CH*n*_IRQ_MODE field descriptions (continued)

Field	Description
30–31 IRQ_MODE	IRQ mode selection. **NOTE:** The interrupt modes are described in section 2.5. 00 Level mode 01 Pulse mode 10 Pulse-Notify mode 11 Single-Pulse mode

10.7.16 TIM0 Channel Error Interrupt Request Enable Register, n=0:7 (TIM_0_CH*n*_EIRQ_EN)

Address: 1000h base + 3Ch offset + (128d × i), where i=0d to 7d



TIM_0_CH*n*_EIRQ_EN field descriptions

Field	Description
0–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

TIM_0_CH*n*_EIRQ_EN field descriptions (continued)

Field	Description
26 GLITCHDET_ EIRQ_EN	TIM_GLITCHDETx_IRQ interrupt enable. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP 1 Enable error interrupt, error interrupt is visible outside GTM-IP
27 TODET_EIRQ_ EN	TIM_TODETx_IRQ interrupt enable. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP 1 Enable error interrupt, error interrupt is visible outside GTM-IP
28 GPROFL_EIRQ_ EN	TIM_GPROFL_IRQ interrupt enable. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP 1 Enable error interrupt, error interrupt is visible outside GTM-IP
29 CNTOFL_EIRQ_ EN	TIM_CNTOFLx_IRQ interrupt enable. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP 1 Enable error interrupt, error interrupt is visible outside GTM-IP
30 ECNTOFL_ EIRQ_EN	TIM_ECNTOFLx_IRQ interrupt enable. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP 1 Enable error interrupt, error interrupt is visible outside GTM-IP
31 NEWVAL_EIRQ_ EN	TIM_NEWVALx_EIRQ error interrupt enable. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP 1 Enable error interrupt, error interrupt is visible outside GTM-IP

10.7.17 TIM0 Input Source Register (TIM_0_IN_SRC)

NOTE

Any read access to a VAL_[x], bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored.

NOTE

Any read access to a MODE_[x], bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored.

NOTE

The combination MODE_[n] = '11' , VAL_[n] = '11' is allowed. It will be used to route an additional input source TIM_AUX_IN to the desired TIM channel.

Address: 1000h base + 78h offset = 1078h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	MODE_7		VAL_7		MODE_6		VAL_6		MODE_5		VAL_5		MODE_4		VAL_4	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	MODE_3		VAL_3		MODE_2		VAL_2		MODE_1		VAL_1		MODE_0		VAL_0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIM_0_IN_SRC field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>0–1
MODE_7</td><td>Input source to Channel 7

Multicore encoding in use (MODE_x(1) defines the state of the signal)

Table 10-53. Function table
<table>
<tr><th>MODE_x(1)</th><th>VAL_x(1)</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>The input signal defined by bit field CICTRL of the TIM channel is used as input source.</td></tr>
<tr><td>0</td><td>1</td><td>The signal TIM_AUX_IN of the TIM channel is used as input source.</td></tr>
<tr><td>1</td><td>–</td><td>The state VAL_x(1) defines the input level for the TIM channel.</td></tr>
</table>
NOTE: Any read access to a MODE_x, bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored.

00 State is 0 (ignores write access).
01 Change state to 0.
10 Change state to 1.
11 State is 1 (ignores write access).</td></tr>
<tr><td>2–3
VAL_7</td><td>Value to be fed to Channel 7

Multicore encoding in use (VAL_x(1) defines the state of the signal).

Function depends on the combination of VAL_x(1) and MODE_x(1) see MODE_0 description.

NOTE: Any read access to a VAL_x field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored.

00 State is 0 (ignores write access).
01 Change state to 0.
10 Change state to 1.
11 State is 1 (ignores write access).</td></tr>
<tr><td>4–5
MODE_6</td><td>Input source to Channel 6</td></tr>
</table>

Table continues on the next page...

TIM_0_IN_SRC field descriptions (continued)

Field	Description
	Multicore encoding in use (MODE_x(1) defines the state of the signal) **Table 10-52. Function table** MODE_x(1) / VAL_x(1) / Description 0 / 0 / The input signal defined by bit field CICTRL of the TIM channel is used as input source. 0 / 1 / The signal TIM_AUX_IN of the TIM channel is used as input source. 1 / – / The state VAL_x(1) defines the input level for the TIM channel. **NOTE:** Any read access to a MODE_x, bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. 00 State is 0 (ignores write access). 01 Change state to 0. 10 Change state to 1. 11 State is 1 (ignores write access).
6–7 VAL_6	Value to be fed to Channel 6 Multicore encoding in use (VAL_x(1) defines the state of the signal). Function depends on the combination of VAL_x(1) and MODE_x(1) see MODE_0 description. **NOTE:** Any read access to a VAL_x field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. 00 State is 0 (ignores write access). 01 Change state to 0. 10 Change state to 1. 11 State is 1 (ignores write access).
8–9 MODE_5	Input source to Channel 5 Multicore encoding in use (MODE_x(1) defines the state of the signal) **Table 10-51. Function table** MODE_x(1) / VAL_x(1) / Description 0 / 0 / The input signal defined by bit field CICTRL of the TIM channel is used as input source. 0 / 1 / The signal TIM_AUX_IN of the TIM channel is used as input source. 1 / – / The state VAL_x(1) defines the input level for the TIM channel. **NOTE:** Any read access to a MODE_x, bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored.

Table continues on the next page...

TIM_0_IN_SRC field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>00 State is 0 (ignores write access).
01 Change state to 0.
10 Change state to 1.
11 State is 1 (ignores write access).</td></tr>
<tr><td>10–11
VAL_5</td><td>Value to be fed to Channel 5

Multicore encoding in use (VAL_x(1) defines the state of the signal).

Function depends on the combination of VAL_x(1) and MODE_x(1) see MODE_0 description.

NOTE: Any read access to a VAL_x field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored.

00 State is 0 (ignores write access).
01 Change state to 0.
10 IChange state to 1.
11 State is 1 (ignores write access).</td></tr>
<tr><td>12–13
MODE_4</td><td>Input source to Channel 4

Multicore encoding in use (MODE_x(1) defines the state of the signal)

Table 10-50. Function table
<table><tr><th>MODE_x(1)</th><th>VAL_x(1)</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>The input signal defined by bit field CICTRL of the TIM channel is used as input source.</td></tr><tr><td>0</td><td>1</td><td>The signal TIM_AUX_IN of the TIM channel is used as input source.</td></tr><tr><td>1</td><td>–</td><td>The state VAL_x(1) defines the input level for the TIM channel.</td></tr></table>
NOTE: Any read access to a MODE_x, bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored.

00 State is 0 (ignores write access).
01 Change state to 0.
10 Change state to 1.
11 State is 1 (ignores write access).</td></tr>
<tr><td>14–15
VAL_4</td><td>Value to be fed to Channel 4

Multicore encoding in use (VAL_x(1) defines the state of the signal).

Function depends on the combination of VAL_x(1) and MODE_x(1) see MODE_0 description.

NOTE: Any read access to a VAL_x field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored.

00 State is 0 (ignores write access).
01 Change state to 0.</td></tr>
</table>

Table continues on the next page...

TIM_0_IN_SRC field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>10 Change state to 1.
11 State is 1 (ignores write access).</td></tr>
<tr><td>16–17
MODE_3</td><td>Input source to Channel 3

Multicore encoding in use (MODE_x(1) defines the state of the signal)
Table 10-49. Function table<table><tr><th>MODE_x(1)</th><th>VAL_x(1)</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>The input signal defined by bit field CICTRL of the TIM channel is used as input source.</td></tr><tr><td>0</td><td>1</td><td>The signal TIM_AUX_IN of the TIM channel is used as input source.</td></tr><tr><td>1</td><td>–</td><td>The state VAL_x(1) defines the input level for the TIM channel.</td></tr></table>
NOTE: Any read access to a MODE_x, bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored.

00 State is 0 (ignores write access).
01 Change state to 0.
10 Change state to 1.
11 State is 1 (ignores write access).</td></tr>
<tr><td>18–19
VAL_3</td><td>Value to be fed to Channel 3

Multicore encoding in use (VAL_x(1) defines the state of the signal).

Function depends on the combination of VAL_x(1) and MODE_x(1) see MODE_0 description.

NOTE: Any read access to a VAL_x field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored.

00 State is 0 (ignores write access).
01 Input_signal is set to 0
10 Input signal is set to 1
11 State is 1 (ignores write access).</td></tr>
<tr><td>20–21
MODE_2</td><td>Input source to Channel 2

Multicore encoding in use (MODE_x(1) defines the state of the signal)
Table 10-48. Function table<table><tr><th>MODE_x(1)</th><th>VAL_x(1)</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>The input signal defined by bit field CICTRL of the TIM channel is used as input source.</td></tr><tr><td>0</td><td>1</td><td>The signal TIM_AUX_IN of the TIM channel is used as input source.</td></tr><tr><td>1</td><td>–</td><td>The state VAL_x(1) defines the input level for the TIM channel.</td></tr></table></td></tr>
</table>

Table continues on the next page...

TIM_0_IN_SRC field descriptions (continued)

Field	Description
	NOTE: Any read access to a MODE_x, bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. 00 State is 0 (ignores write access). 01 Change state to 0. 10 Change state to 1. 11 State is 1 (ignores write access).
22–23 VAL_2	Value to be fed to Channel 2 Multicore encoding in use (VAL_x(1) defines the state of the signal). Function depends on the combination of VAL_x(1) and MODE_x(1) see MODE_0 description. **NOTE:** Any read access to a VAL_x field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. 00 State is 0 (ignores write access). 01 Change state to 0. 10 Change state to 1. 11 State is 1 (ignores write access).
24–25 MODE_1	Input source to Channel 1 Multicore encoding in use (MODE_x(1) defines the state of the signal) **Table 10-47. Function table** MODE_x(1) / VAL_x(1) / Description: 0 / 0 / The input signal defined by bit field CICTRL of the TIM channel is used as input source. 0 / 1 / The signal TIM_AUX_IN of the TIM channel is used as input source. 1 / – / The state VAL_x(1) defines the input level for the TIM channel. **NOTE:** Any read access to a MODE_x, bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. 00 State is 0 (ignores write access). 01 Change state to 0. 10 Change state to 1. 11 State is 1 (ignores write access).
26–27 VAL_1	Value to be fed to Channel 1 Multicore encoding in use (VAL_x(1) defines the state of the signal). Function depends on the combination of VAL_x(1) and MODE_x(1) see MODE_0 description.

Table continues on the next page...

TIM_0_IN_SRC field descriptions (continued)

Field	Description
	NOTE: Any read access to a VAL_x field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. 00 State is 0 (ignores write access). 01 Change state to 0. 10 Change state to 1. 11 State is 1 (ignores write access).
28–29 MODE_0	Input source to Channel 0 Multicore encoding in use (MODE_x(1) defines the state of the signal)

Table 10-46. Function table

MODE_x(1)	VAL_x(1)	Description
0	0	The input signal defined by bit field CICTRL of the TIM channel is used as input source.
0	1	The signal TIM_AUX_IN of the TIM channel is used as input source.
1	–	The state VAL_x(1) defines the input level for the TIM channel.

NOTE: Any read access to a MODE_x, bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored.

00 State is 0 (ignores write access).
01 Change state to 0.
10 Change state to 1.
11 State is 1 (ignores write access).

30–31 VAL_0	Value to be fed to Channel 0 Multicore encoding in use (VAL_x(1) defines the state of the signal). Function depends on the combination of VAL_x(1) and MODE_x(1) see MODE_0 description. **NOTE:** Any read access to a VAL_x field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored. 00 State is 0 (ignores write access). 01 Change state to 0. 10 Change state to 1. 11 State is 1 (ignores write access).

10.7.18 TIM0 Channel Reset Register (TIM_0_RST)

Address: 1000h base + 7Ch offset = 107Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								RST_CH7	RST_CH6	RST_CH5	RST_CH4	RST_CH3	RST_CH2	RST_CH1	RST_CH0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TIM_0_RST field descriptions

Field	Description
0–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 RST_CH7	Software reset of channel 7. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. 0 No action. 1 Reset channel 7.
25 RST_CH6	Software reset of channel 6. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. 0 No action. 1 Reset channel 6.
26 RST_CH5	Software reset of channel 5. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. 0 No action. 1 Reset channel 5.
27 RST_CH4	Software reset of channel 4. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. 0 No action. 1 Reset channel 4.
28 RST_CH3	Software reset of channel 3. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately.

Table continues on the next page...

TIM_0_RST field descriptions (continued)

Field	Description
	0 No action. 1 Reset channel 3.
29 RST_CH2	Software reset of channel 2. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. 0 No action. 1 Reset channel 2.
30 RST_CH1	Software reset of channel 1. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. 0 No action. 1 Reset channel 1.
31 RST_CH0	Software reset of channel 0. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. 0 No action. 1 Reset channel 0.

Chapter 11
Timer Output Module (TOM)

11.1 TOM Overview

The Timer Output Module (TOM) provides up to 16 independent channels (index n) to generate simple PWM signals, which are output on the corresponding TOM[i]_CHn_OUT signals. Additionally, a pulse count modulated signal can be generated and output on the TOM[i]_CH15_OUT signal.

The architecture of the TOM submodule is shown in Figure 11-1.

Indices and their range as used inside this chapter are:

- y=0,1
- z=0..7

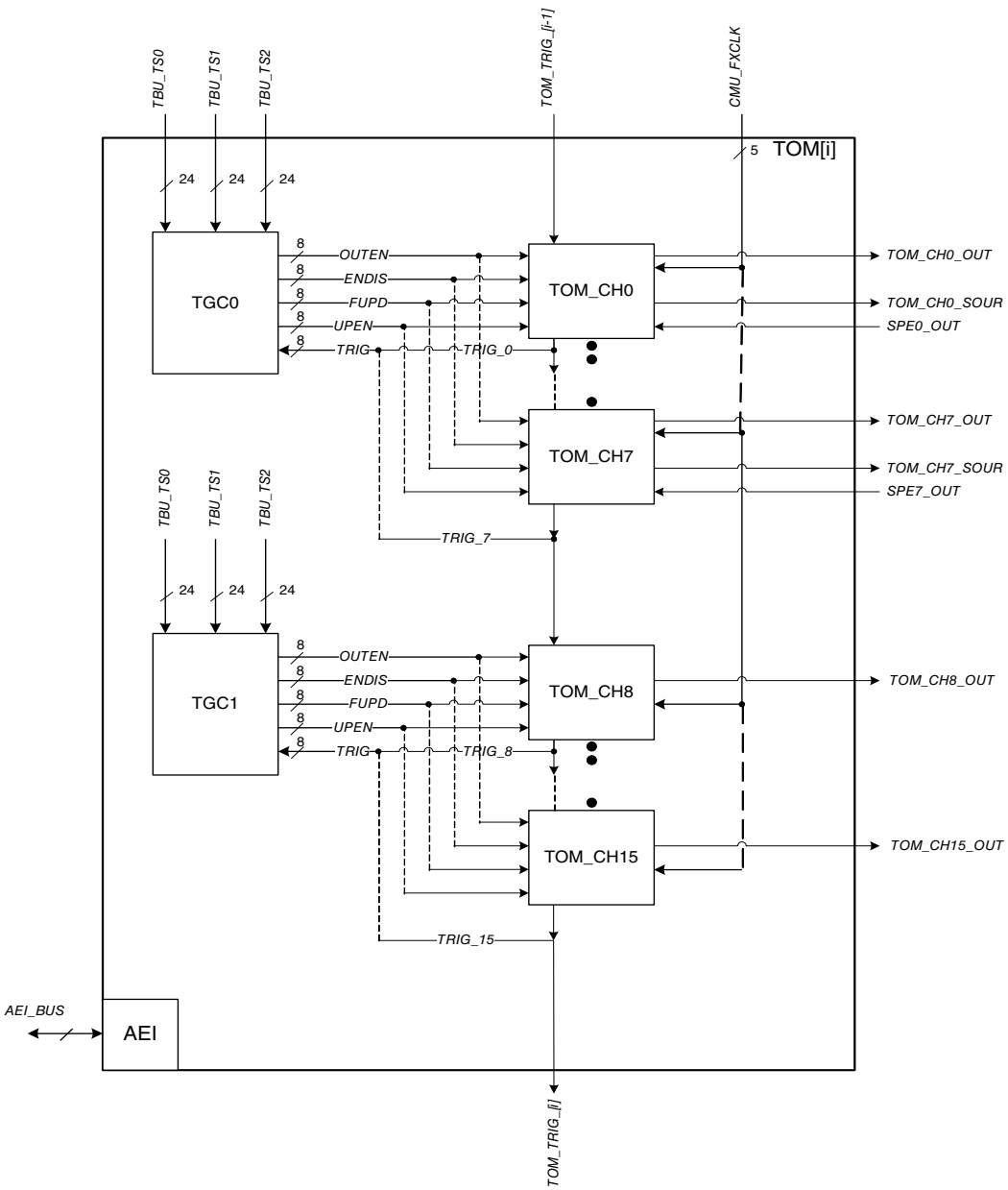


Figure 11-1. TOM Block Diagram

The TGC0 and TGC1 submodules are global channel control units that:

- control the enabling or disabling of corresponding channels and it's outputs, and
- update the period registers and duty cycle registers of corresponding channels.

The TOM module receives three time stamp values (TBU_TS0, TBU_TS1, TBU_TS2) in order to realize synchronized output behavior on behalf of a common time base.

Five dedicated *CMU_FXCLKn* inputs provide divided clock enables that can be selected to enable SYS_CLK to clock the registers that source the output signals.

11.2 TOM Global Channel Control (TGC0, TGC1)

11.2.1 TOM Global Channel Control Overview

The TGC0 and TGC1 global channel control units synchronize individual TOM channels to external or internal events. TGC0 controls TOM channels zero through seven and TGC1 controls TOM channels eight through 15.

The TOM submodule provides four mechanisms to control each channel:

- Global enable/disable mechanism – uses the **TOM[i]_TGCn_ENDIS_CTRL** control register and the **TOM[i]_TGCn_ENDIS_STAT** status register of each channel.
- Global output enable mechanism – uses the **TOM[i]_TGCn_OUTEN_CTRL** output enable control register and the **TOM[i]_TGCn_OUTEN_STAT** output enable status register of each channel.
- Global force update mechanism – uses the **TOM[i]_TGC[y]_FUPD_CTRL** force update control register of each channel.
- Update enables for the **CM0**, **CM1** and **CLK_SRC_STAT** registers – uses the **TOM[i]_TGC[y]_GLB_CTRL[UPEN_CTRLz]** bit of each channel.

11.2.2 TGC Subunit

Each of the first three individual mechanisms (enable/disable of the channel, output enable and force update) can be driven by three different trigger sources.

The three trigger sources are :

- the host CPU (bit **HOST_TRIG** of register **TOM[i]_TGC[y]_GLB_CTRL**)
- the TBU time stamp (signal TBU_TS0, TBU_TS1, TBU_TS2 if available)
- the internal trigger signal *TRIG* (bunch of trigger signals *TRIG_[x]*)

Note

The trigger signal is only active for one configured CMU clock period.

As a consequence, if the configured CMU clock period of the channel generating the trigger *TRIG_[X]* is smaller than the CMU clock period of triggered channel, the triggered channel may not recognize the trigger signal *TRIG_[X]*.

Thus, it is recommended to configure all channels connected via the trigger*TRIG_[x]* to the same CMU clock period.

The first way is to trigger the control mechanism by a direct register write access via host CPU (bit **HOST_TRIG** of register **TOM[i]_TGC[y]_GLB_CTRL**).

The second way is provided by a compare match trigger on behalf of a specified time base coming from the module TBU (selected by bits **TBU_SEL**) and the time stamp compare value defined in the bit field **ACT_TB** of register **TOM[i]_TGC[y]_ACT_TB**.

NOTE

A signed compare of **ACT_TB** and selected *TBU_TS*[*x*] with x=0,1,2 is performed.

The third possibility is the input *TRIG* (signals *TRIG_[x]*) coming from the TOM channels 0 to 7 / 8 to 15.

The corresponding trigger signal *TRIG_[x]* coming from channel [x] can be masked by the register **TOM[i]_TGC[y]_INT_TRIG**.

To enable or disable each individual TOM channel, the registers **TOM[i]_TGC[y]_ENDIS_CTRL** and/or **TOM[i]_TGC[y]_ENDIS_STAT** have to be used.

The register **TOM[i]_TGC[y]_ENDIS_STAT** controls directly the signal *ENDIS*. A write access to this register is possible.

The register **TOM[i]_TGC[y]_ENDIS_CTRL** is a shadow register that overwrites the value of register **TOM[i]_TGC[y]_ENDIS_STAT** if one of the three trigger conditions matches.

11.2.2.1 TOM Global channel control mechanism

The TOM global channel control mechanism is shown in Figure 11-2.

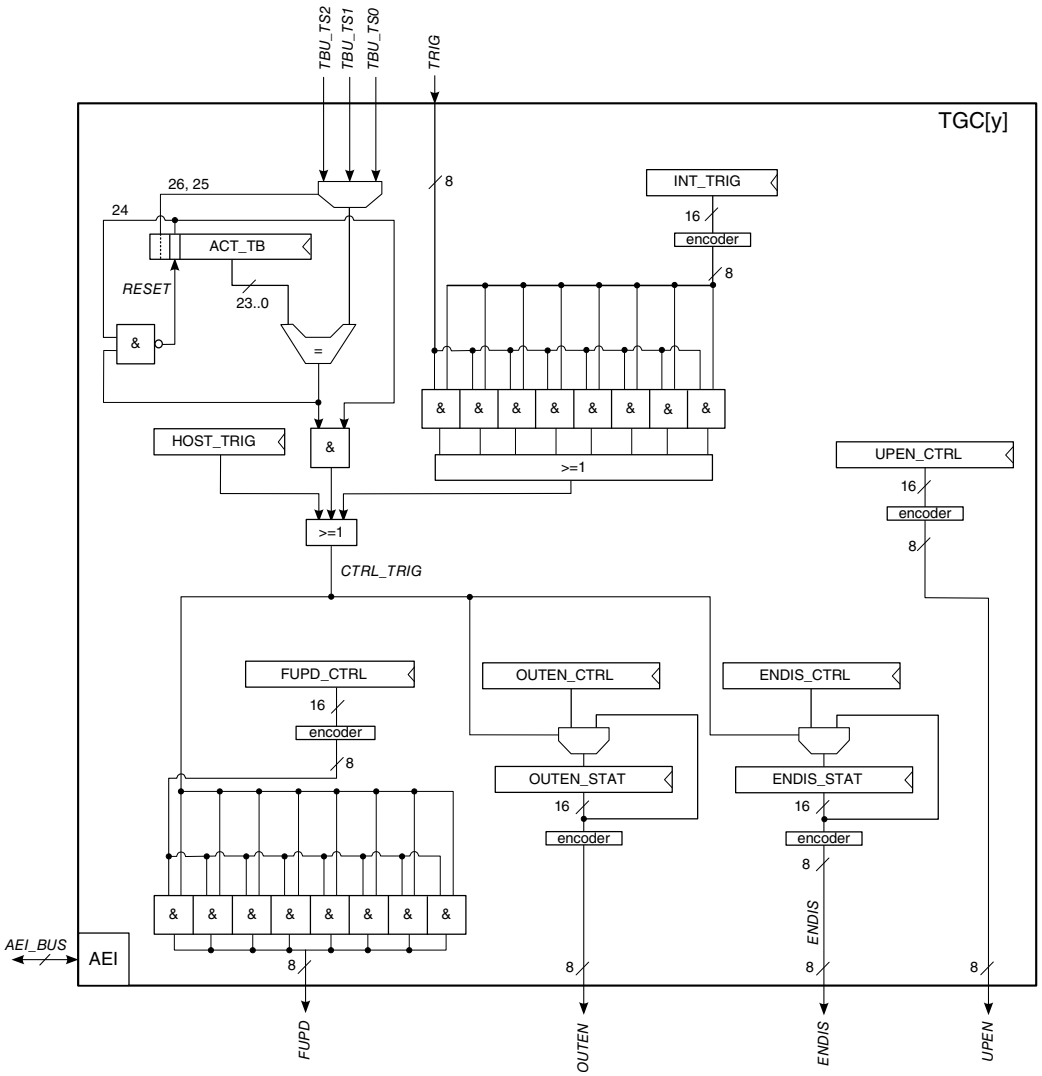


Figure 11-2. Channel control mechanism

The output of an individual TOM channel can be controlled using the TOM[i]_TGC[y]_OUTEN_CTRL register and the TOM[i]_TGC[y]_OUTEN_STAT register. The write-accessible TOM[i]_TGC[y]_OUTEN_STAT register directly controls the OUTEN signal for the corresponding channel.

The TOM[i]_TGC[y]_OUTEN_CTRL register is a shadow register that overwrites the value of the TOM[i]_TGC[y]_OUTEN_STAT register when a trigger from one of the three trigger sources occurs.

If a TOM channel is disabled by the **TOM[i]_TGCn_OUTEN_STAT** register, the actual value of the *TOM_CHn_OUT* signal is defined by the **TOM[i]_CHn_CTRL[SL]** bit.

If the output is enabled, the level of the *TOM_CHn_OUT* signal depends on the output level of the **SOUR** flip-flop (see Figure 11-3).

The **TOM[i]_TGC[y]_FUPD_CTRL** register defines which of the TOM channels receive a *FORCE_UPDATE* event when the *CTRL_TRIG* trigger signal is raised.

The **UPEN_CTRL[z]** register bits configure which TOM channels are enabled for the **CM0**, **CM1** and **CLK_SRC** working registers to be updated with the values in the corresponding **SR0**, **SR1** and **CLK_SRC_SR** shadow registers. If an update is enabled, the update will occur when the **CN0** counter register is reset (see Figure 11-3).

11.3 TOM Channel (TOM_CH[x])

Each individual TOM channel has:

- a Counter Compare Unit 0 (CCU0),
- a Counter Compare Unit 1 (CCU1), and
- a Signal Output Generation Unit (SOU).

The TOM architecture for channels zero through seven is shown in Figure 11-3 and in Figure 11-4 for channels eight through 15.

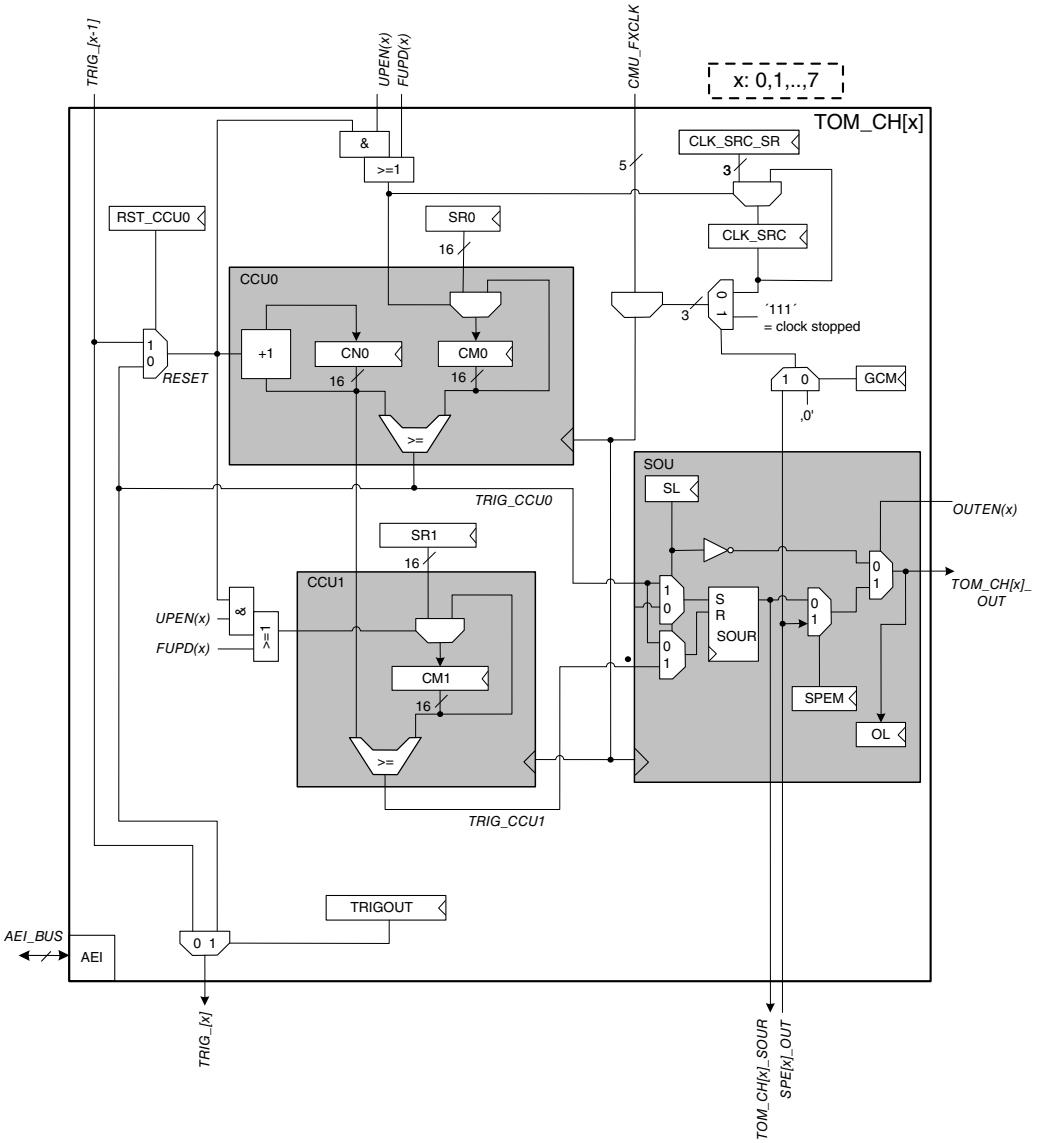


Figure 11-3. Architecture of TOM channels 0:7

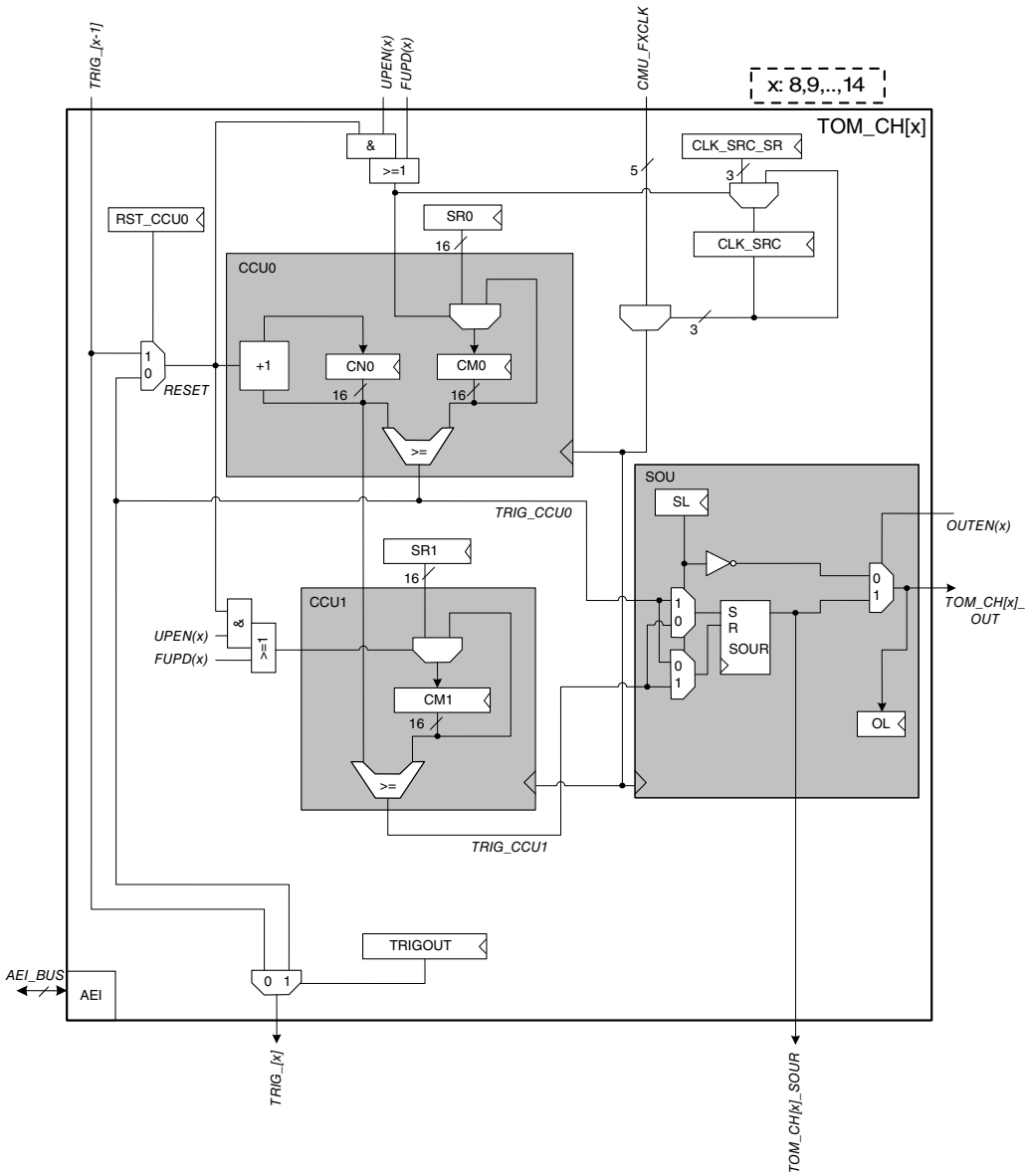


Figure 11-4. Architecture of TOM channels 8:14

11.3.1 TOM Channel 15 architecture for PCM generation

TOM channel 15, shown in Figure 11-5, can be used for PCM generation.

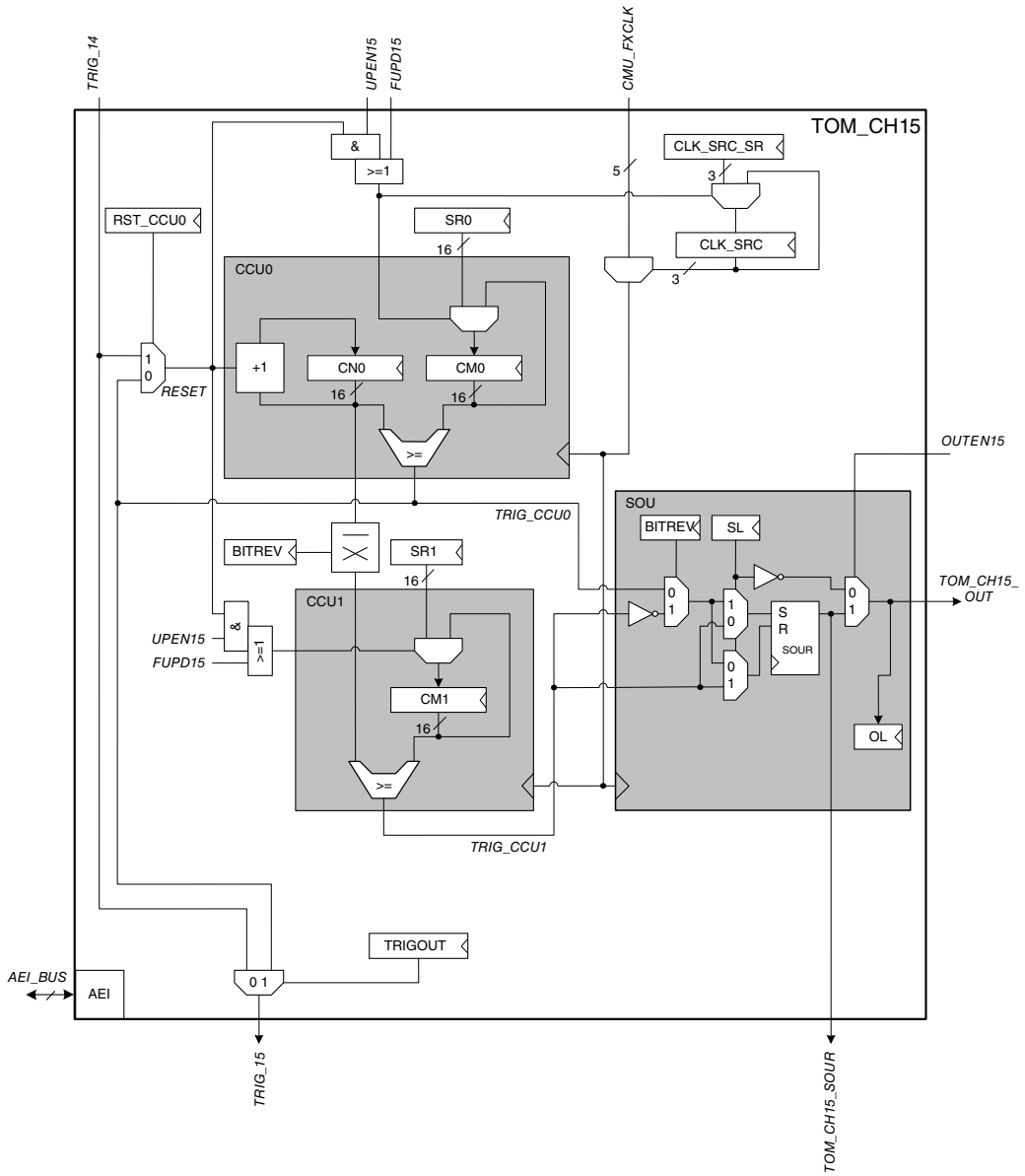


Figure 11-5. TOM channel 15 architecture

The CCU0 subunit contains a **CN0** counter register that is clocked with one of the *CMU_FXCLKn* input clocks.

Depending on the **TOM[i]_CHn_CTRL[RST_CCU0]** configuration bits, the **CN0** counter register can be reset:

- when the counter value is equal to the compare value of the **CM0** register, or
- by the *TRIG_[n−1]* trigger signal from TOM channel [n−1], or by the TRIG_15 trigger signal from the TOM[i−1] submodule.

The CCU0 subunit compares the value in the **CN0** counter register with the value in the **CM0** register. If **CN0** is ≥ **CM0**, the TRIG_CCU0 signal triggers the SOU subunit and the TOM submodule channel [n+1].

The CCU1 subunit compares the value in the **CN0** counter register with the value in the **CM1**register. If **CN0** is ≥ **CM1**, the TRIG_CCU1 signal triggers the SOU subunit.

The configuration of **CM1**=0 causes a 0% duty cycle at the output of the TOM channel. The configuration of **CM1** >= **CM0** causes a 100% duty cycle at the output of the TOM channel when **CM0**>1 or a 0% duty cycle when **CM0**=0 or **CM0**=1 independent of **CM1** (the counter **CN0** is constant 0). The configuration of **CM0**=0 and **CM1**=0 causes a 0% duty cycle at the output of the TOM channel.

The hardware ensures that for both 0% and 100% duty cycles, no glitch occurs at the output of the TOM channel.

The SOU subunit is responsible for output signal generation. When a *TRIG_CCU0* trigger from the CCU0 subunit or when a *TRIG_CCU1* trigger from CCU1 subunit is received by the SOU subunit, the SR flip-flop of the SOU subunit is either set or reset (depending on the configuration of the **TOM[i]_CHn_CTRL[SL]** bit). The initial signal output level for the channel is the inverse of the value of the **SL** bit.

Figure 11-8 shows PWM output behavior with respect to the value of the **SL** bit.

The level on the *TOM[i]_CHn_OUT* output is captured by the **TOM[i]_CHn_STAT[OL]** bit.

11.3.2 Duty cycle, period and selected counter clock frequency update mechanisms

The **CM0** and **CM1** action registers can be reloaded with the values in the **SR0** and **SR1** shadow registers. The **CLK_SRC** register (that determines the clock frequency of the **CN0** counter register) can be reloaded with the value in its **CLK_SRC_SR** shadow register (**TOM[i]_CHn_CTRL[CLK_SRC_SR]** bit field).

The update of the **CM0**, **CM1** and **CLK_SRC** registers with the values from its corresponding shadow registers, occurs at the end of a period and when the **CN0** counter register is reset by:

- the *RESET* signal, which is enabled if the value in the **CN0** counter register is ≥ the value in the **CM0** register, or
- the *TRIG_[n−1]* signal, when triggered by TOM channel n−1.

For channel 0 of TOM0, the TRIG_[0−1] signal is hard-coded to zero.

An update (if enabled) of the duty cycle, period and **CN0** counter register's clock frequency becomes effective synchronously with the start of a new period. To configure new values for the **SR0**, **SR1**, and **CLK_SRC_SR**register and then enable an update:

1. Disable updating of the **CM0** and **CM1** action registers by clearing the **TOM[i]_TGC[y]_GLB_CTRL[UPEN_CTRL[z]]** bit to zero.
2. Write new values to the **SR0**, **SR1**,and **CLK_SRC_SR** registers.
3. Enable updating of the **CM0** and **CM1** action registers by setting the **TOM[i]_TGC[y]_GLB_CTRL[UPEN_CTRL[z]]** bit to one.

11.3.2.1 Synchronous update of duty cycle only

Synchronous update of only the duty cycle is accomplished by writing a new value to the **SR1** register without first disabling the updating of the CM0 and CM1 action registers (as described in Duty cycle, period and selected counter clock frequency update mechanisms). The new duty cycle (shown in Figure 11-6) is then applied in the period following the period where the **SR1** register was updated.

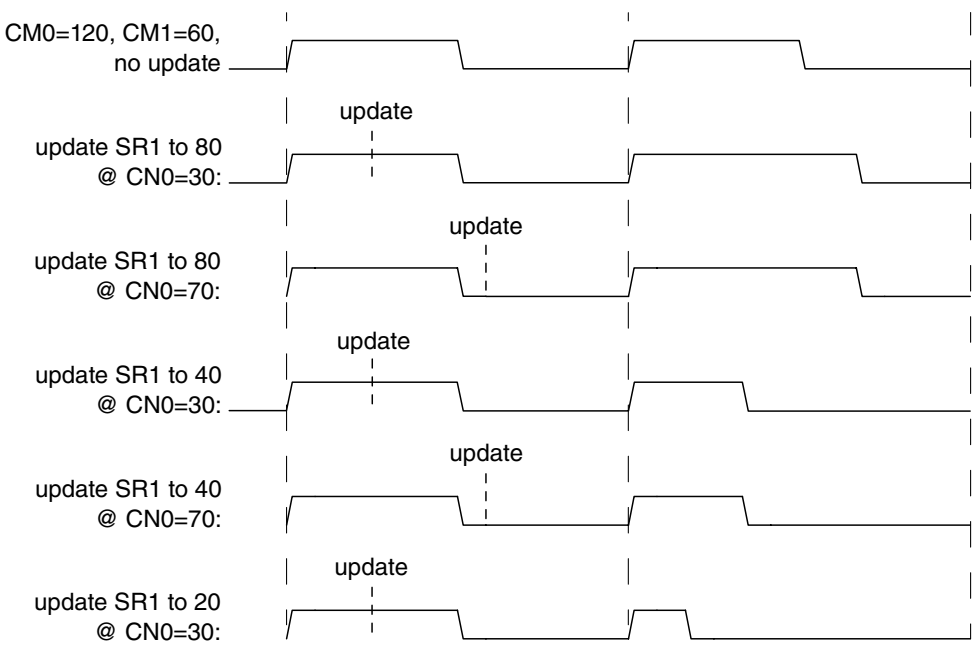


Figure 11-6. Duty cycle update examples

11.3.2.2 Asynchronous update of duty cycle only

If an update of only the duty cycle is to be accomplished independent of the start of a new period (asynchronous), the new value can be written directly to **CM1** register. It is recommended that:

- the synchronous update mechanism be disabled as a whole by clearing the **TOM[i]_TGX[y]_GLB_CTRL[UPEN_CTRL[z]]** bit to zero, or
- the **SR1** register be updated with the same value as the **CM1** register before writing to the **CM1** register.

Depending on the point in time that the **CM1** register is updated relative to the actual value of the **CN0** counter register and the **CM1** register, the new duty cycle (shown in Figure 11-7) is applied in either the current period or the following period. Either way, glitches are avoided. The new duty cycle may jitter from update to update for a maximum of one period (given by the **CM0** register). However, the period itself remains unchanged.

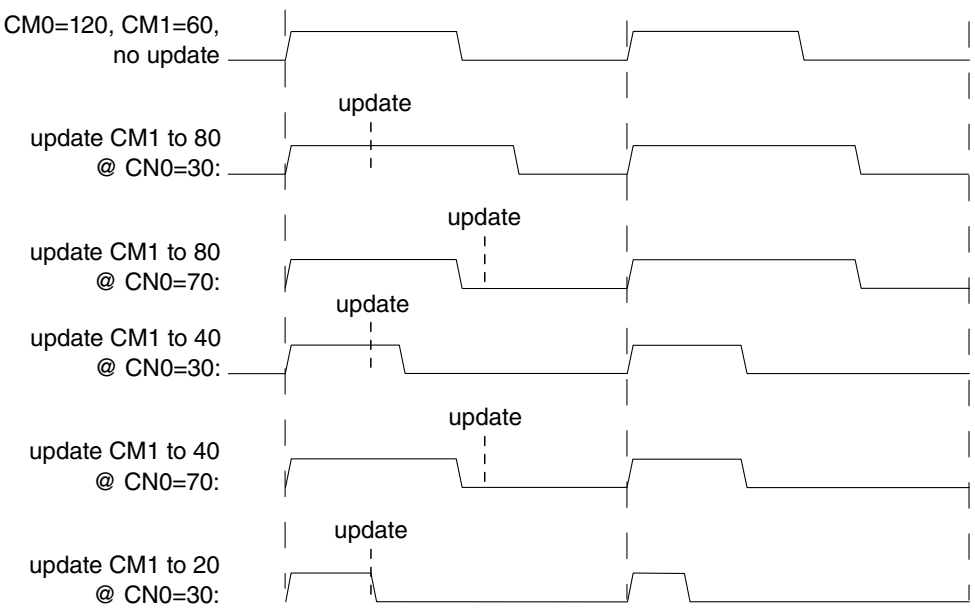


Figure 11-7. Asynchronous duty cycle update examples

11.3.3 TOM continuous mode

In continuous mode, the TOM channel starts incrementing the **CN0** counter register when it is enabled by the corresponding **TOM[i]_TGC[y]_ENDIS_STAT[ENDES_STATn]** bit.

The signal level of the generated output signal can be configured by writing to the TOM[i]_CHn_CTRL[SL] bit. The output signal level is the inverse of the SL bit.

When the **CN0** counter register is reset from the **CM0** register back to zero, the first edge of a period is generated on the *TOM[i]_CHn_OUT* signal.

The second edge of the period is generated when the value in the **CN0** counter register reaches the value in the **CM1** register. At this time, the **CN0** counter register is reset to zero and continues to increment (as shown in Figure 11-8).

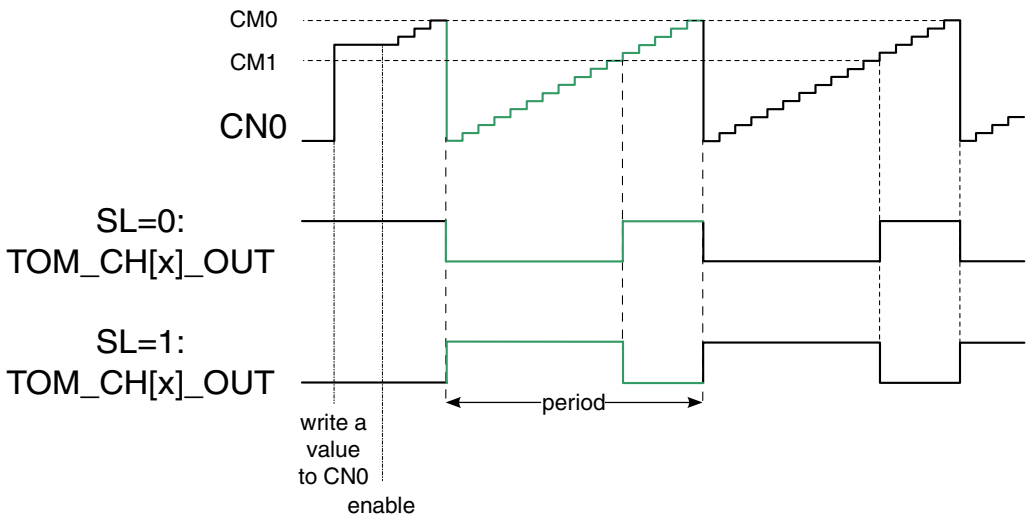


Figure 11-8. PWM output in continuous mode

11.3.4 TOM One shot mode

In One-shot mode, the TOM channel generates one pulse. The pulse has a signal level that is configured by writing to the the TOM_CHn_CTRL[SL] bit.

First, the channel has to be enabled by configuring the corresponding **TOM[i]_TGC[y]_ENDIS_STAT** value. Then, the one-shot mode has to be enabled by setting the TOM_CHn_CTRL[OSM] bit to one.

A write access to the **CN0** counter register triggers the incrementing of the **CN0** counter register.

If SPE mode (see SPE Overview) of TOM[i] channel 2 is enabled by setting the TOM_CHn_CTRL[SPEM] bit to one, the *SPE[i]_NIPD* trigger signal can reset the **CN0** counter register to zero.

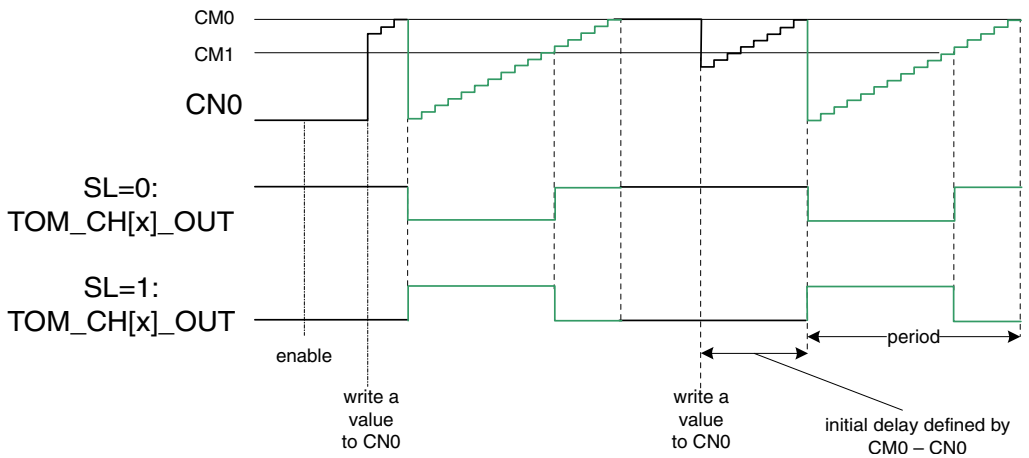
A new value in the **CN0** counter register determines the start delay of the first edge of the pulse. The delay time is given by (**CM0–CN0**) multiplied by the period that is defined by the current value of the**CLK_SRC** register.

When the **CN0** counter register is reset from the **CM0** register back to zero, the first edge of the pulse is generated on the *TOM[i]_CHn_OUT* output.

To avoid an update of the CMx register with content of the SRx register at this point in time, the automatic update should be disabled by setting UPEN_CTRLn = 00 (in register TOM[i]_CHn_CTRL).

The second edge of the pulse is generated on the *TOM[i]_CHn_OUT* output when the value of the **CN0** ≥ **CM1** after an update of the **CM1** register).

If the **CN0** counter register reaches the value of the **CM0** register a second time, the **CN0** counter register stops incrementing.



Additional output of single pulses can be started by a write to the **CN0** counter register.

- Writing the value

  $CN0_{new}$ < CM0 before CN0 reaches first time CM0 for the first time,

  leads to a shift of first edge (generated if CN0 reaches CM0 first time) by the time CM0 – $CN0_{new}$.
- Writing the value

  $CN0_{new}$ < CM1 while $CN0_{old}$ is below CM1,

  to the CN0 counter register while it is incrementing causes a lengthening of the pulse. The CN0 counter register stops incrementing when it's value reaches the value of the CM0 register.
- Writing the value

  $CN0_{new}$ > CM1 while $CN0_{old}$ is already greater than CM1,

  to the CN0 counter register while it is incrementing causes an immediate restart of the generation of a single pulse, which includes the initial delay defined by CM0 – $CN0_{new}$.

11.3.5 Pulse count modulation

A pulse count modulated (PCM) signal can be generated, instead of a simple PWM signal, on the *TOM[i]_CH15_OUT* output.

Figure 11-5 shows the architecture for pulse count modulation.

The PCM mode is enabled by setting the **TOM[i]_CH15_CTRL[BITREV**] bit to one.

As a result, a bit-reversing of the **CN0** counter register's output is configured, where the LSB and MSB bits are swapped, the LSB+1 and MSB-1 bits are swapped, the LSB+2 and MSB-2 bits are swapped, and so on.

The effect of bit-reversing the **CN0** counter register is shown in Figure 11-9.

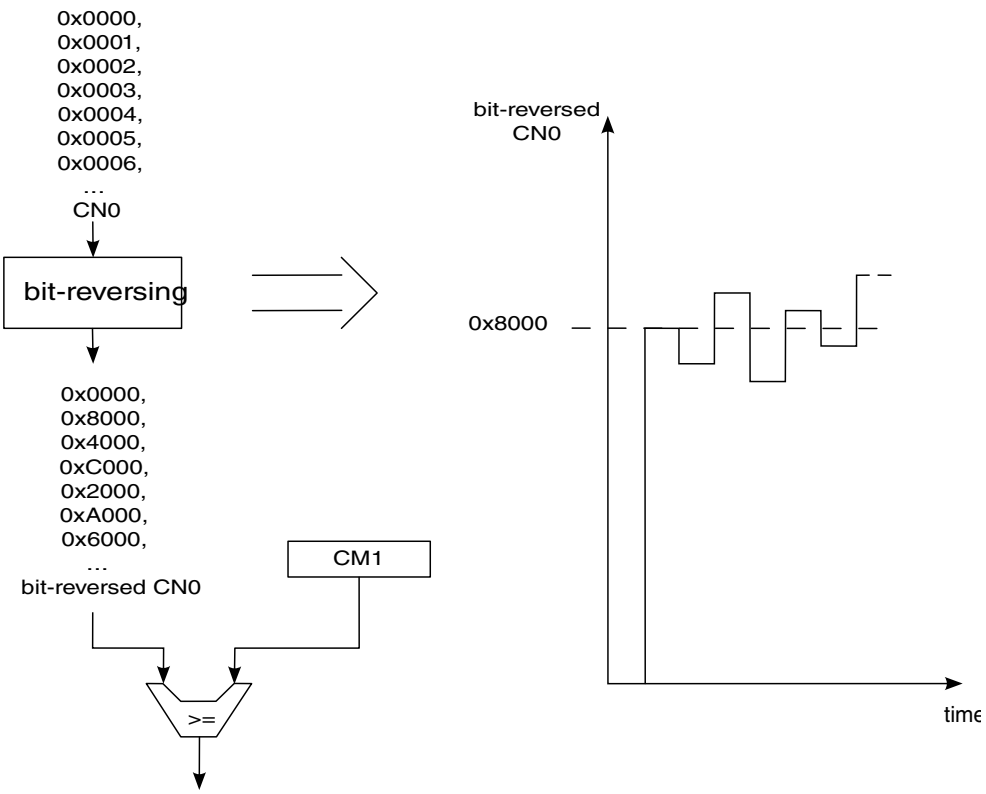


Figure 11-9. Bit-reversing example

In the PCM mode, the **CN0** counter register is incremented by every tick of the configured *CMU_FXCLKn* clock.

The output of the **CN0** counter register is first bit-reversed and then it is compared to the value in the **CM1**register.

If the bit-reversed value of the **CN0** counter register is > the value in the **CM1** register, the SR flip-flop in the SOU submodule is set depending on configuration of the **SL** bit, otherwise the SR flip-flop is reset. The changing value of the SR flip-flop generates the pulse count modulated signal on the *TOM[i]_CH15_OUT* output.

In PCM mode, the value of the **CM0** register must always be set to 0xFFFF (maximum).

11.4 TOM Brush-Less Direct Current (BLDC) Motor Support

The TOM submodule offers, in combination with the SPE submodule, support for a BLDC engine. TOM channels zero through seven can be used to drive a BLDC motor.

BLDC support can be configured by setting the **TOM[i]_CHn_CTRL[SPEM]** bit to one. When the SPEM bit is set, the TOM channel's output is controlled by the SPE_OUT(z) signal, which is sourced by the SPE submodule (see SPE to TOM Connections).

After a new input pattern is detected by the SPE (signaled by SPE[i]_NIPD), TOM[i]_CH2 can be used together with the SPE module to trigger a delayed update of the SPE_OUT_CTRL register. This feature is configured on TOM[i]_CH2 by setting bits SPEM = 1 and OSM = 1 in the TOM[i]_CHn_CTRL register. For details, see SPE Overview.

11.5 TOM Gated Counter Mode

Each TOM - SPE module combination can implementation a gated counter mode, where an *FSOI* input (see Figure 17-1) of a TIM module is used to gate the clock of a TOM CCU0 submodule. That is, the CN0 counter register of a TOM CCU0 submodule increments as long as the FSOI signal is zero.

To configure the SPE for gated counter mode:

- Enable the SPE by setting the SPE_CTRL_STAT[SPE_EN] bit to one.
- Disable all three TIM inputs to the SPE by clearing the SPE_CTRL_STAT[SIE2, SIE1, SIE0] bits to '000'.
- Enable the Fast Shut Off Mode (FSOM) mode by setting the **SPE_CTRL_STAT[FSOM]** bit to one.
- Select TOM channel n by setting the corresponding SPE_CTRL_STAT[FSOL] bit to one.
- Set the **SPE[i]_OUT_CTRL[SPE_OUT_CTRL]** bit field to 0x5555.

NOTE

The actions in the first four bullets above can be accomplished by a single write to the SPE_CTRL_STAT register.

To configure the TOM for gated counter mode:

- Disable SPE mode by clearing the TOM[i]_CH_CTRL[**SPEM]** bit to zero.
- Enable gated counter mode by setting the TOM[i]_CH_CTRL[**GCM]** bit to one.

NOTE

The actions in the two bullets above can be accomplished by a single write to the TOM[i]_CH_CTRL register.

11.6 TOM Interrupt signals

Table 11-1 shows TOM interrupt signals:

Table 11-1. TOM interrupt signals

Signal	Description
TOM_CCU0TCn_IRQ	CCU0 Trigger condition interrupt for channel n
TOM_CCU1TCn_IRQ	CCU1 Trigger condition interrupt for channel n

NOTE

For the TOM submodules, the interrupts are bundled within the ICM submodule a second time to reduce external interrupt lines. The interrupts are OR-ed in a manner that one GTM external interrupt line represents two adjacent TOM channel interrupts. For TOM[i] , the bundling is shown in Table 18-1.

11.7 Memory Map and Registers

The Timer Output Module (TOM0) registers are described as follows:

TOM_0 memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	TOM0 Channel n Control Register, n[0:14] (TOM_0_CH0_CTRL)	32	R/W	See section	11.7.1/248
4	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH0_SR0)	32	R/W	0000_0000h	11.7.2/250
8	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH0_SR1)	32	R/W	0000_0000h	11.7.3/250

Table continues on the next page...

TOM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
C	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH0_CM0)	32	R/W	0000_0000h	11.7.4/251
10	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH0_CM1)	32	R/W	0000_0000h	11.7.5/251
14	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH0_CN0)	32	R/W	0000_0000h	11.7.6/252
18	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH0_STAT)	32	R	0000_0001h	11.7.7/252
1C	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH0_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253
20	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH0_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254
24	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH0_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255
28	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH0_IRQ_MODE)	32	R/W	See section	11.7.11/256
30	TOM0 TGC0 Global Control Register (TOM_0_TGC0_GLB_CTRL)	32	R/W	0000_0000h	11.7.12/257
34	TOM0 TGC0 Action Time Base Register (TOM_0_TGC0_ACT_TB)	32	R/W	0000_0000h	11.7.13/261
38	TOM0 TGC0 Force Update Control Register (TOM_0_TGC0_FUPD_CTRL)	32	R/W	0000_0000h	11.7.14/262
3C	TOM0 TGC0 Interrupt Trigger Register (TOM_0_TGC0_INT_TRIG)	32	R/W	0000_0000h	11.7.15/266
40	TOM0 Channel n Control Register, n[0:14] (TOM_0_CH1_CTRL)	32	R/W	See section	11.7.1/248
44	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH1_SR0)	32	R/W	0000_0000h	11.7.2/250
48	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH1_SR1)	32	R/W	0000_0000h	11.7.3/250
4C	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH1_CM0)	32	R/W	0000_0000h	11.7.4/251
50	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH1_CM1)	32	R/W	0000_0000h	11.7.5/251
54	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH1_CN0)	32	R/W	0000_0000h	11.7.6/252
58	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH1_STAT)	32	R	0000_0001h	11.7.7/252
5C	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH1_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253
60	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH1_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254
64	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH1_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255
68	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH1_IRQ_MODE)	32	R/W	See section	11.7.11/256

Table continues on the next page...

TOM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
70	TOM0 TGC0 Enable/Disable Control Register (TOM_0_TGC0_ENDIS_CTRL)	32	R/W	0000_0000h	11.7.16/269
74	TOM0 TGC0 Enable/Disable Status Register (TOM_0_TGC0_ENDIS_STAT)	32	R/W	0000_0000h	11.7.17/271
78	TOM0 TGC0 Out Enable Control Register (TOM_0_TGC0_OUTEN_CTRL)	32	R/W	0000_0000h	11.7.18/274
7C	TOM0 TGC0 Out Enable Control/Status Register (TOM_0_TGC0_OUTEN_STAT)	32	R/W	0000_0000h	11.7.19/277
80	TOM0 Channel n Control Register, n[0:14] (TOM_0_CH2_CTRL)	32	R/W	See section	11.7.1/248
84	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH2_SR0)	32	R/W	0000_0000h	11.7.2/250
88	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH2_SR1)	32	R/W	0000_0000h	11.7.3/250
8C	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH2_CM0)	32	R/W	0000_0000h	11.7.4/251
90	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH2_CM1)	32	R/W	0000_0000h	11.7.5/251
94	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH2_CN0)	32	R/W	0000_0000h	11.7.6/252
98	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH2_STAT)	32	R	0000_0001h	11.7.7/252
9C	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH2_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253
A0	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH2_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254
A4	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH2_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255
A8	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH2_IRQ_MODE)	32	R/W	See section	11.7.11/256
C0	TOM0 Channel n Control Register, n[0:14] (TOM_0_CH3_CTRL)	32	R/W	See section	11.7.1/248
C4	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH3_SR0)	32	R/W	0000_0000h	11.7.2/250
C8	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH3_SR1)	32	R/W	0000_0000h	11.7.3/250
CC	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH3_CM0)	32	R/W	0000_0000h	11.7.4/251
D0	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH3_CM1)	32	R/W	0000_0000h	11.7.5/251
D4	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH3_CN0)	32	R/W	0000_0000h	11.7.6/252
D8	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH3_STAT)	32	R	0000_0001h	11.7.7/252
DC	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH3_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253

Table continues on the next page...

TOM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
E0	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH3_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254
E4	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH3_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255
E8	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH3_IRQ_MODE)	32	R/W	See section	11.7.11/256
100	TOM0 Channel n Control Register, n[0:14] (TOM_0_CH4_CTRL)	32	R/W	See section	11.7.1/248
104	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH4_SR0)	32	R/W	0000_0000h	11.7.2/250
108	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH4_SR1)	32	R/W	0000_0000h	11.7.3/250
10C	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH4_CM0)	32	R/W	0000_0000h	11.7.4/251
110	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH4_CM1)	32	R/W	0000_0000h	11.7.5/251
114	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH4_CN0)	32	R/W	0000_0000h	11.7.6/252
118	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH4_STAT)	32	R	0000_0001h	11.7.7/252
11C	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH4_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253
120	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH4_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254
124	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH4_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255
128	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH4_IRQ_MODE)	32	R/W	See section	11.7.11/256
140	TOM0 Channel n Control Register, n[0:14] (TOM_0_CH5_CTRL)	32	R/W	See section	11.7.1/248
144	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH5_SR0)	32	R/W	0000_0000h	11.7.2/250
148	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH5_SR1)	32	R/W	0000_0000h	11.7.3/250
14C	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH5_CM0)	32	R/W	0000_0000h	11.7.4/251
150	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH5_CM1)	32	R/W	0000_0000h	11.7.5/251
154	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH5_CN0)	32	R/W	0000_0000h	11.7.6/252
158	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH5_STAT)	32	R	0000_0001h	11.7.7/252
15C	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH5_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253
160	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH5_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254

Table continues on the next page...

TOM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
164	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH5_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255
168	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH5_IRQ_MODE)	32	R/W	See section	11.7.11/256
180	TOM0 Channel n Control Register, n[0:14] (TOM_0_CH6_CTRL)	32	R/W	See section	11.7.1/248
184	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH6_SR0)	32	R/W	0000_0000h	11.7.2/250
188	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH6_SR1)	32	R/W	0000_0000h	11.7.3/250
18C	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH6_CM0)	32	R/W	0000_0000h	11.7.4/251
190	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH6_CM1)	32	R/W	0000_0000h	11.7.5/251
194	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH6_CN0)	32	R/W	0000_0000h	11.7.6/252
198	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH6_STAT)	32	R	0000_0001h	11.7.7/252
19C	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH6_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253
1A0	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH6_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254
1A4	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH6_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255
1A8	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH6_IRQ_MODE)	32	R/W	See section	11.7.11/256
1C0	TOM0 Channel n Control Register, n[0:14] (TOM_0_CH7_CTRL)	32	R/W	See section	11.7.1/248
1C4	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH7_SR0)	32	R/W	0000_0000h	11.7.2/250
1C8	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH7_SR1)	32	R/W	0000_0000h	11.7.3/250
1CC	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH7_CM0)	32	R/W	0000_0000h	11.7.4/251
1D0	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH7_CM1)	32	R/W	0000_0000h	11.7.5/251
1D4	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH7_CN0)	32	R/W	0000_0000h	11.7.6/252
1D8	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH7_STAT)	32	R	0000_0001h	11.7.7/252
1DC	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH7_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253
1E0	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH7_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254
1E4	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH7_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255

Table continues on the next page...

TOM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E8	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH7_IRQ_MODE)	32	R/W	See section	11.7.11/256
200	TOM0 Channel n Control Register, n[0:14] (TOM_0_CH8_CTRL)	32	R/W	See section	11.7.1/248
204	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH8_SR0)	32	R/W	0000_0000h	11.7.2/250
208	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH8_SR1)	32	R/W	0000_0000h	11.7.3/250
20C	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH8_CM0)	32	R/W	0000_0000h	11.7.4/251
210	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH8_CM1)	32	R/W	0000_0000h	11.7.5/251
214	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH8_CN0)	32	R/W	0000_0000h	11.7.6/252
218	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH8_STAT)	32	R	0000_0001h	11.7.7/252
21C	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH8_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253
220	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH8_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254
224	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH8_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255
228	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH8_IRQ_MODE)	32	R/W	See section	11.7.11/256
230	TOM0 TGC1 Global Control Register (TOM_0_TGC1_GLB_CTRL)	32	R/W	0000_0000h	11.7.20/279
234	TOM0 TGC1 Action Time Base Register (TOM_0_TGC1_ACT_TB)	32	R/W	0000_0000h	11.7.21/283
238	TOM0 TGC1 Force Update Control Register (TOM_0_TGC1_FUPD_CTRL)	32	R/W	0000_0000h	11.7.22/284
23C	TOM0 TGC1 Interrupt Trigger Register (TOM_0_TGC1_INT_TRIG)	32	R/W	0000_0000h	11.7.23/289
240	TOM0 Channel n Control Register, n[0:14] (TOM_0_CH9_CTRL)	32	R/W	See section	11.7.1/248
244	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH9_SR0)	32	R/W	0000_0000h	11.7.2/250
248	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH9_SR1)	32	R/W	0000_0000h	11.7.3/250
24C	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH9_CM0)	32	R/W	0000_0000h	11.7.4/251
250	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH9_CM1)	32	R/W	0000_0000h	11.7.5/251
254	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH9_CN0)	32	R/W	0000_0000h	11.7.6/252
258	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH9_STAT)	32	R	0000_0001h	11.7.7/252

Table continues on the next page...

TOM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
25C	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH9_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253
260	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH9_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254
264	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH9_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255
268	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH9_IRQ_MODE)	32	R/W	See section	11.7.11/256
270	TOM0 TGC1 Enable/Disable Control Register (TOM_0_TGC1_ENDIS_CTRL)	32	R/W	0000_0000h	11.7.24/291
274	TOM0 TGC1 Enable/Disable Status Register (TOM_0_TGC1_ENDIS_STAT)	32	R/W	0000_0000h	11.7.25/294
278	TOM0 TGC1 Out Enable Control Register (TOM_0_TGC1_OUTEN_CTRL)	32	R/W	0000_0000h	11.7.26/297
27C	TOM0 TGC1 Out Enable Control/Status Register (TOM_0_TGC1_OUTEN_STAT)	32	R/W	0000_0000h	11.7.27/299
280	TOM0 Channel n Control Register, n[0:14] (TOM_0_CH10_CTRL)	32	R/W	See section	11.7.1/248
284	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH10_SR0)	32	R/W	0000_0000h	11.7.2/250
288	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH10_SR1)	32	R/W	0000_0000h	11.7.3/250
28C	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH10_CM0)	32	R/W	0000_0000h	11.7.4/251
290	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH10_CM1)	32	R/W	0000_0000h	11.7.5/251
294	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH10_CN0)	32	R/W	0000_0000h	11.7.6/252
298	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH10_STAT)	32	R	0000_0001h	11.7.7/252
29C	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH10_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253
2A0	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH10_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254
2A4	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH10_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255
2A8	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH10_IRQ_MODE)	32	R/W	See section	11.7.11/256
2C0	TOM0 Channel n Control Register, n[0:14] (TOM_0_CH11_CTRL)	32	R/W	See section	11.7.1/248
2C4	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH11_SR0)	32	R/W	0000_0000h	11.7.2/250
2C8	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH11_SR1)	32	R/W	0000_0000h	11.7.3/250

Table continues on the next page...

TOM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
2CC	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH11_CM0)	32	R/W	0000_0000h	11.7.4/251
2D0	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH11_CM1)	32	R/W	0000_0000h	11.7.5/251
2D4	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH11_CN0)	32	R/W	0000_0000h	11.7.6/252
2D8	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH11_STAT)	32	R	0000_0001h	11.7.7/252
2DC	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH11_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253
2E0	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH11_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254
2E4	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH11_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255
2E8	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH11_IRQ_MODE)	32	R/W	See section	11.7.11/256
300	TOM0 Channel n Control Register, n[0:14] (TOM_0_CH12_CTRL)	32	R/W	See section	11.7.1/248
304	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH12_SR0)	32	R/W	0000_0000h	11.7.2/250
308	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH12_SR1)	32	R/W	0000_0000h	11.7.3/250
30C	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH12_CM0)	32	R/W	0000_0000h	11.7.4/251
310	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH12_CM1)	32	R/W	0000_0000h	11.7.5/251
314	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH12_CN0)	32	R/W	0000_0000h	11.7.6/252
318	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH12_STAT)	32	R	0000_0001h	11.7.7/252
31C	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH12_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253
320	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH12_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254
324	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH12_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255
328	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH12_IRQ_MODE)	32	R/W	See section	11.7.11/256
340	TOM0 Channel n Control Register, n[0:14] (TOM_0_CH13_CTRL)	32	R/W	See section	11.7.1/248
344	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH13_SR0)	32	R/W	0000_0000h	11.7.2/250
348	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH13_SR1)	32	R/W	0000_0000h	11.7.3/250

Table continues on the next page...

TOM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
34C	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH13_CM0)	32	R/W	0000_0000h	11.7.4/251
350	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH13_CM1)	32	R/W	0000_0000h	11.7.5/251
354	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH13_CN0)	32	R/W	0000_0000h	11.7.6/252
358	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH13_STAT)	32	R	0000_0001h	11.7.7/252
35C	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH13_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253
360	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH13_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254
364	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH13_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255
368	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH13_IRQ_MODE)	32	R/W	See section	11.7.11/256
380	TOM0 Channel n Control Register, n[0:14] (TOM_0_CH14_CTRL)	32	R/W	See section	11.7.1/248
384	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH14_SR0)	32	R/W	0000_0000h	11.7.2/250
388	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH14_SR1)	32	R/W	0000_0000h	11.7.3/250
38C	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH14_CM0)	32	R/W	0000_0000h	11.7.4/251
390	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH14_CM1)	32	R/W	0000_0000h	11.7.5/251
394	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH14_CN0)	32	R/W	0000_0000h	11.7.6/252
398	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH14_STAT)	32	R	0000_0001h	11.7.7/252
39C	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH14_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253
3A0	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH14_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254
3A4	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH14_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255
3A8	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH14_IRQ_MODE)	32	R/W	See section	11.7.11/256
3C0	TOM0 Channel 15 Control register (TOM_0_CH15_CTRL)	32	R/W	See section	11.7.28/302
3C4	TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH15_SR0)	32	R/W	0000_0000h	11.7.2/250
3C8	TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH15_SR1)	32	R/W	0000_0000h	11.7.3/250
3CC	TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH15_CM0)	32	R/W	0000_0000h	11.7.4/251

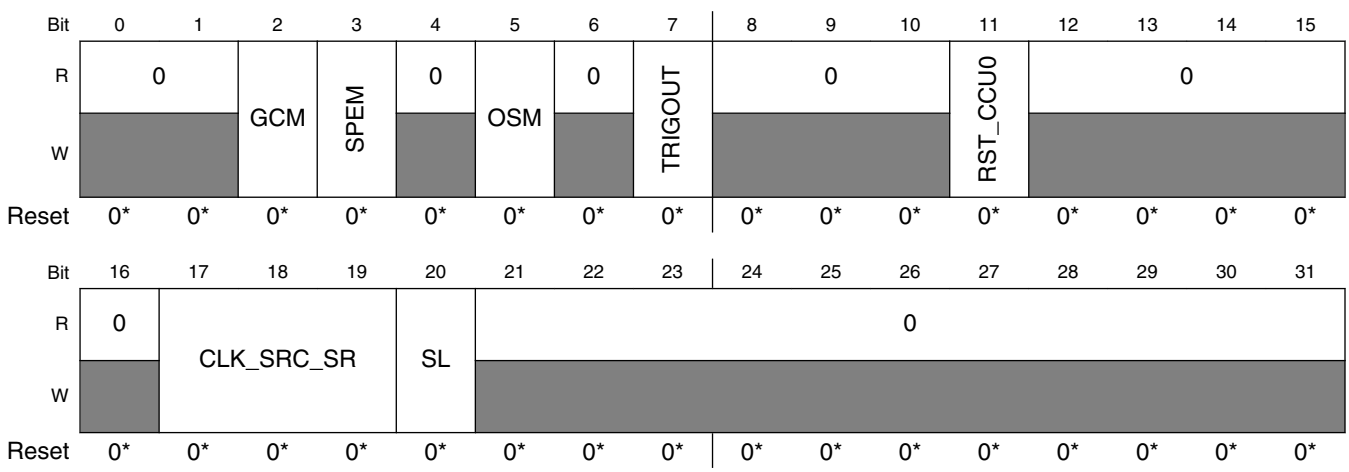
Table continues on the next page...

TOM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3D0	TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH15_CM1)	32	R/W	0000_0000h	11.7.5/251
3D4	TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH15_CN0)	32	R/W	0000_0000h	11.7.6/252
3D8	TOM0 Channel n Status Register, n[0:15] (TOM_0_CH15_STAT)	32	R	0000_0001h	11.7.7/252
3DC	TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH15_IRQ_NOTIFY)	32	R/W	0000_0000h	11.7.8/253
3E0	TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH15_IRQ_EN)	32	R/W	0000_0000h	11.7.9/254
3E4	TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH15_IRQ_FORCINT)	32	R/W	0000_0000h	11.7.10/255
3E8	TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH15_IRQ_MODE)	32	R/W	See section	11.7.11/256

11.7.1 TOM0 Channel n Control Register, n[0:14] (TOM_0_CH*n*_CTRL)

Address: 8000h base + 0h offset + (64d × i), where i=0d to 14d



* Notes:
- Bits 20-23 do not reset to a given value.

TOM_0_CH*n*_CTRL field descriptions

Field	Description
0–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 GCM	Gated Counter Mode enable.

Table continues on the next page...

TOM_0_CH*n*_CTRL field descriptions (continued)

Field	Description
	NOTE: The Gated Counter mode is only available for TOM instances connected to a SPE module and only for channels 0:7. 0 Gated Counter mode disabled. 1 Gated Counter mode enabled.
3 SPEM	SPE mode enable for channel. **NOTE:** The SPE mode is only implemented for TOM instances connected to a SPE module and only for channels 0:7. 0 SPE mode disabled. 1 SPE mode enabled.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 OSM	One-shot mode.. In this mode the counter CN0 counts for only one period. The length of period is defined by CM0. A write access to the register CM0 triggers the start of counting.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 TRIGOUT	Trigger output selection (output signal TRIG_[n]) of module TOM_CH[n]. 0 TRIG_[x] is TRIG_[x-1]. 1 TRIG_[x] is TRIG_CCU0.
8–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 RST_CCU0	Reset source of CCU0. **NOTE:** On TOM channel 2 SPEM=1 has special meaning. If SPEM = 1, the signal SPE_NIPD triggers the reset of CN0 independent of RST_CN0. 0 Reset counter register CN0 to 0 on matching comparison CM0. 1 Reset counter register CN0 to 0 on trigger TRIG_[x-1]
12–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–19 CLK_SRC_SR	Clock source select for channel. The register CLK_SRC is updated with the value of CLK_SRC_SR together with the update of register CM0 and CM1. The input of the FX clock divider depends on the value of FXCLK_SEL (see CMU) 000 CMU_FXCLK(0) selected: GTM system clock. 001 CMU_FXCLK(1) selected: GTM system clock / 2^4 . 010 CMU_FXCLK(2) selected: GTM system clock / 2^8 . 011 CMU_FXCLK(3) selected: GTM system clock / 2^{12} . 100 CMU_FXCLK(4) selected: GTM system clock / 2^{16} .

Table continues on the next page...

TOM_0_CH*n*_CTRL field descriptions (continued)

Field	Description
	101 No CMU_FXCLK selected, clock of channel stopped. 110 No CMU_FXCLK selected, clock of channel stopped. 111 No CMU_FXCLK selected, clock of channel stopped. **NOTE:** If clock of channel is stopped, the channel can only be restarted by resetting CLK_SRC_SR to a value of 000 to 100 and by forcing an update via the force update mechanism.
20 SL	Signal level for duty cycle. If the output is disabled, the output TOM_OUT[n] is set to inverse value of SL.
21–31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.7.2 TOM0 Channel n Shadow 0 Register, n[0:15] (TOM_0_CH*n*_SR0)

Address: 8000h base + 4h offset + (64d × i), where i=0d to 15d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																SR0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_CH*n*_SR0 field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–31 SR0	TOM channel n shadow register SR0 for update of compare register CM0.

11.7.3 TOM0 Channel n Shadow 1 Register, n[0:15] (TOM_0_CH*n*_SR1)

Address: 8000h base + 8h offset + (64d × i), where i=0d to 15d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																SR1															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_CH*n*_SR1 field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–31 SR1	TOM channel n shadow register SR1 for update of compare register CM1.

11.7.4 TOM0 Channel n Compare 0 Register, n[0:15] (TOM_0_CH*n*_CM0)

Address: 8000h base + Ch offset + (64d × i), where i=0d to 15d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																CM0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_CH*n*_CM0 field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–31 CM0	TOM CCU0 compare register. Setting CM0 less than CM1 configures a duty cycle of 100%.

11.7.5 TOM0 Channel n Compare 1 Register, n[0:15] (TOM_0_CH*n*_CM1)

Address: 8000h base + 10h offset + (64d × i), where i=0d to 15d

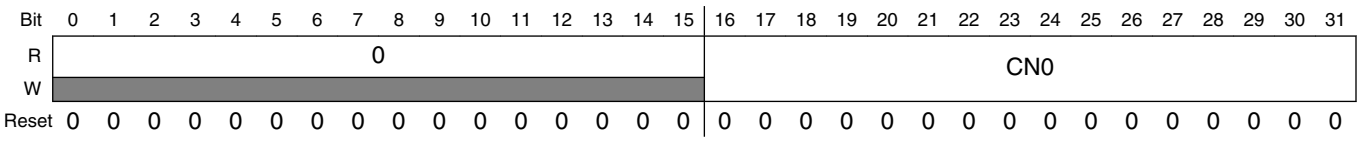
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																CM1															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_CH*n*_CM1 field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–31 CM1	TOM CCU1 compare register. Setting CM1 = 0 configures a duty cycle of 0% independent of the configured value of CM0.

11.7.6 TOM0 Channel n CN0 Register, n[0:15] (TOM_0_CH*n*_CN0)

Address: 8000h base + 14h offset + (64d × i), where i=0d to 15d

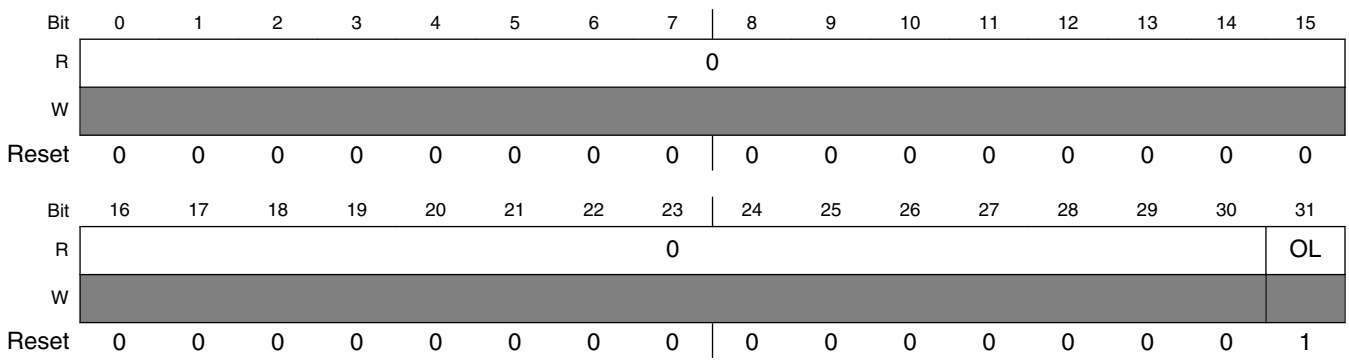


TOM_0_CH*n*_CN0 field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–31 CN0	TOM CCU0 counter register. **NOTE:** This counter CN0 is stopped if the TOM channel is disabled and not reset on an enable event of TOM channel.

11.7.7 TOM0 Channel n Status Register, n[0:15] (TOM_0_CH*n*_STAT)

Address: 8000h base + 18h offset + (64d × i), where i=0d to 15d

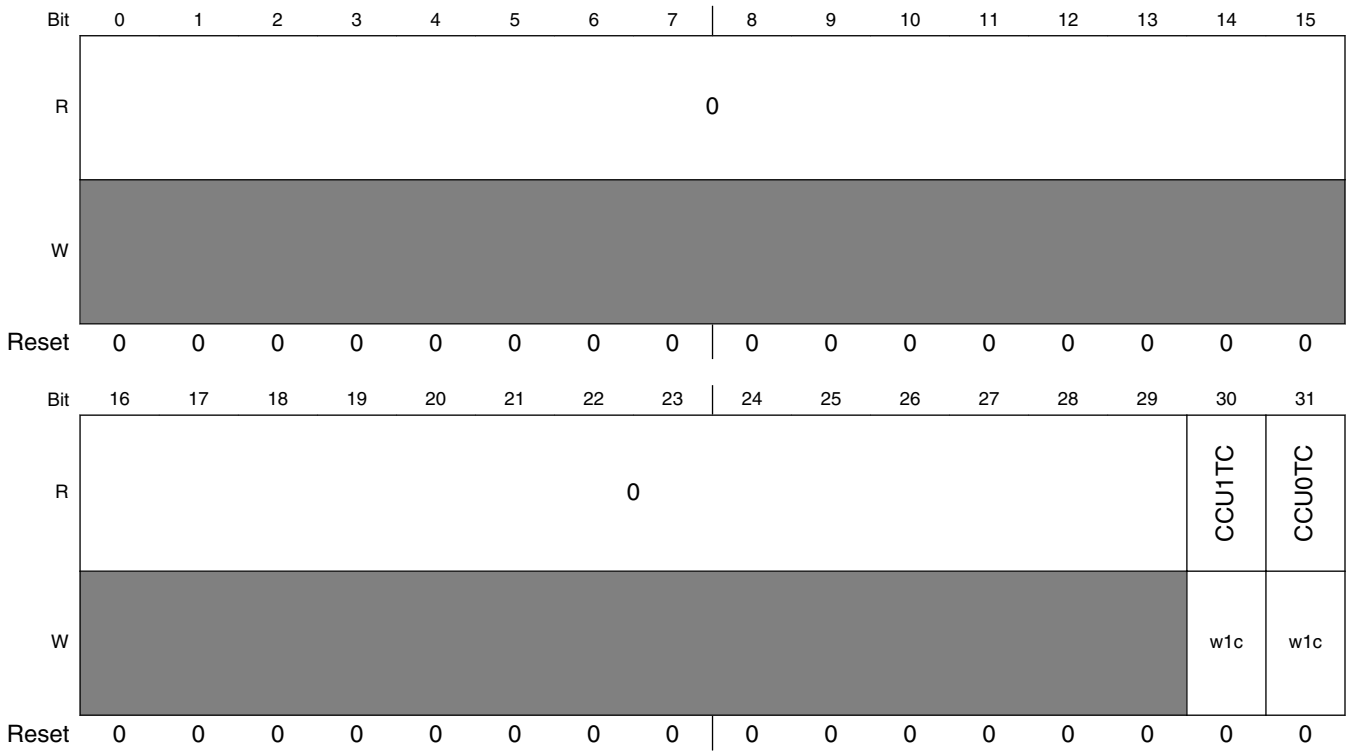


TOM_0_CH*n*_STAT field descriptions

Field	Description
0–30 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
31 OL	Output level of output TOM_OUT(n). **NOTE:** Reset value is the inverted value of SL bit .

11.7.8 TOM0 Channel n Interrupt Request Notification Register, n[0:15] (TOM_0_CH*n*_IRQ_NOTIFY)

Address: 8000h base + 1Ch offset + (64d × i), where i=0d to 15d

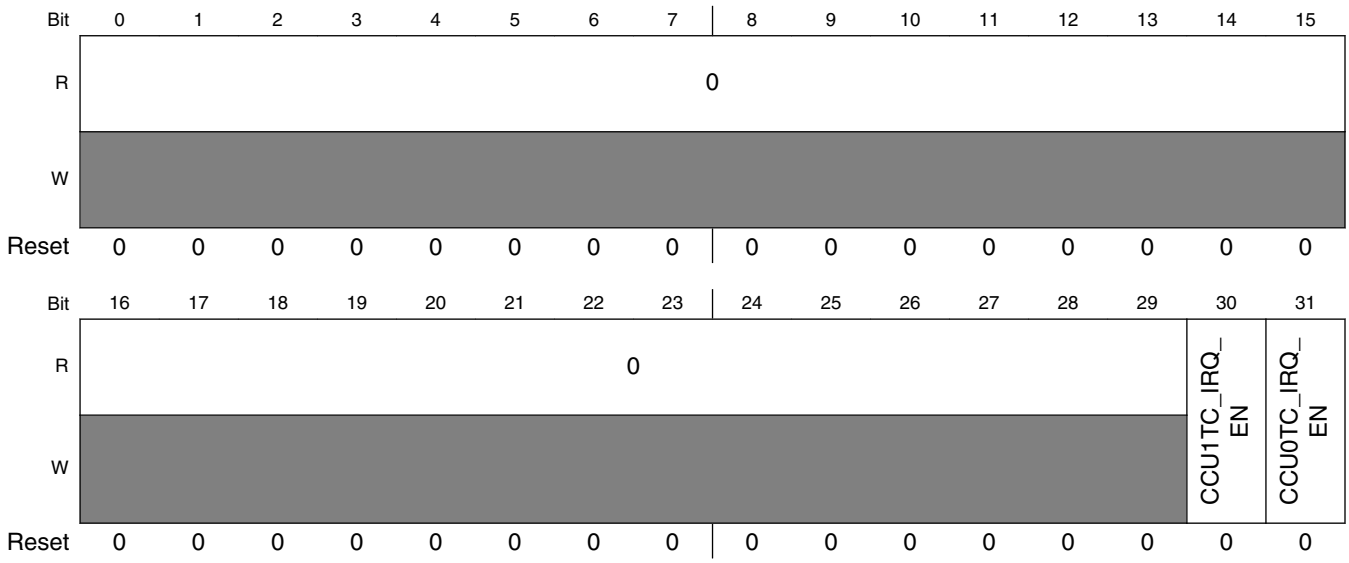


TOM_0_CH*n*_IRQ_NOTIFY field descriptions

Field	Description
0–29 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
30 CCU1TC	CCU1 Trigger condition interrupt for channel n. **NOTE:** The notification of the interrupt is only triggered one time after reaching the condition CN0 ? CM0. To re-trigger the notification, first the condition CN0 < CM0 has to occur. 0 No interrupt occurred. 1 The condition CN0 grater than or equal to CM1 was detected.
31 CCU0TC	CCU0 Trigger condition interrupt for channel n. **NOTE:** The notification of the interrupt is only triggered one time after reaching the condition CN0 ≥ CM0. To re-trigger the notification, first the condition CN0 < CM0 has to occur.

11.7.9 TOM0 Channel n Interrupt Request Enable Register, n[0:15] (TOM_0_CH*n*_IRQ_EN)

Address: 8000h base + 20h offset + (64d × i), where i=0d to 15d

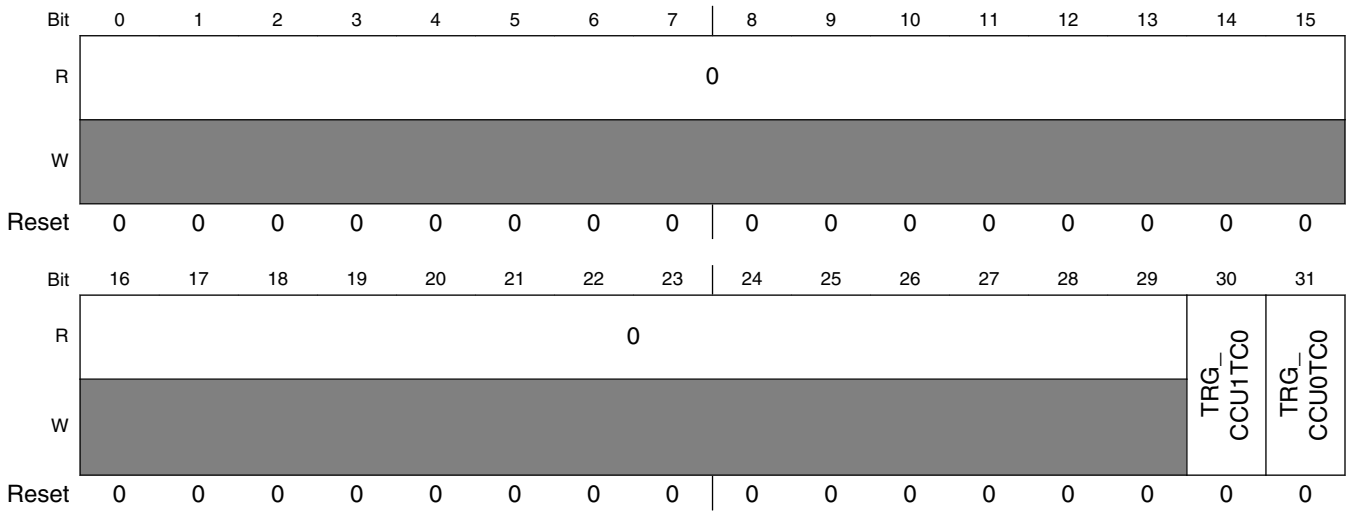


TOM_0_CH*n*_IRQ_EN field descriptions

Field	Description
0–29 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
30 CCU1TC_IRQ_EN	TOM_CCU0TC_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM-IP. 1 Enable interrupt, interrupt is visible outside GTM-IP.
31 CCU0TC_IRQ_EN	TOM_CCU0TC_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM-IP. 1 Enable interrupt, interrupt is visible outside GTM-IP.

11.7.10 TOM0 Channel n Force Interrupt Request Register, n[0:15] (TOM_0_CH*n*_IRQ_FORCINT)

Address: 8000h base + 24h offset + (64d × i), where i=0d to 15d



TOM_0_CH*n*_IRQ_FORCINT field descriptions

Field	Description
0–29 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
30 TRG_CCU1TC0	Trigger TOM_CCU1TC0_IRQ interrupt by software. This bit is cleared automatically after write. This bit is write protected by bit GTM_CT[RF_PROT]. 0 No interrupt triggering. 1 Assert CCU1TC0_IRQ interrupt for one clock cycle.
31 TRG_CCU0TC0	Trigger TOM_CCU0TC0_IRQ interrupt by software. This bit is cleared automatically after write. This bit is write protected by bit GTM_CT[RF_PROT]. 0 No interrupt triggering. 1 Assert CCU0TC0_IRQ interrupt for one clock cycle.

11.7.11 TOM0 Channel n Interrupt Request Mode Register, n[0:15] (TOM_0_CH*n*_IRQ_MODE)

Address: 8000h base + 28h offset + (64d × i), where i=0d to 15d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0														IRQ_MODE	
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- Bits 28-31 do not reset to a given value.

TOM_0_CH*n*_IRQ_MODE field descriptions

Field	Description
0–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–31 IRQ_MODE	IRQ mode selection. The interrupt modes are described in TOM Interrupt signals . 00 Level mode. 01 Pulse mode. 10 Pulse-Notify mode. 11 Single-Pulse mode.

11.7.12 TOM0 TGC0 Global Control Register (TOM_0_TGC0_GLB_CTRL)

Address: 8000h base + 30h offset = 8030h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	UPEN_CTRL7		UPEN_CTRL6		UPEN_CTRL5		UPEN_CTRL4		UPEN_CTRL3		UPEN_CTRL2		UPEN_CTRL1		UPEN_CTRL0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	RST_CH7	RST_CH6	RST_CH5	RST_CH4	RST_CH3	RST_CH2	RST_CH1	RST_CH0	0							HOST_TRIG
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_TGC0_GLB_CTRL field descriptions

Field	Description
0–1 UPEN_CTRL7	TOM channel 7 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
2–3 UPEN_CTRL6	TOM channel 6 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled

Table continues on the next page...

TOM_0_TGC0_GLB_CTRL field descriptions (continued)

Field	Description
	11 Channel enabled
4–5 UPEN_CTRL5	TOM channel 5 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
6–7 UPEN_CTRL4	TOM channel 4 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
8–9 UPEN_CTRL3	TOM channel 3 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
10–11 UPEN_CTRL2	TOM channel 2 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11

Table continues on the next page...

TOM_0_TGC0_GLB_CTRL field descriptions (continued)

Field	Description
	11 Don’t care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
12–13 UPEN_CTRL1	TOM channel 1 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Write of following double bit values is possible: 00 Don’t care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don’t care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
14–15 UPEN_CTRL0	TOM channel 0 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Write of following double bit values is possible: 00 Don’t care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don’t care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
16 RST_CH7	Software reset of channel 7. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'. 0 No action. 1 Reset channel.
17 RST_CH6	Software reset of channel 6. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'. 0 No action. 1 Reset channel.
18 RST_CH5	Software reset of channel 5. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'.

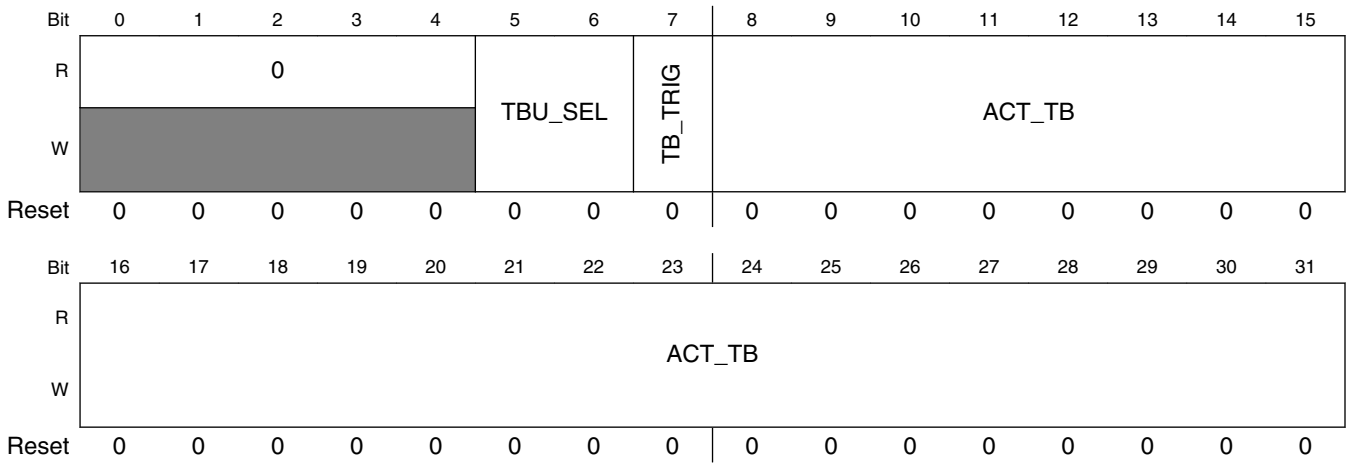
Table continues on the next page...

TOM_0_TGC0_GLB_CTRL field descriptions (continued)

Field	Description
	0 No action. 1 Reset channel.
19 RST_CH4	Software reset of channel 4. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'. 0 No action. 1 Reset channel.
20 RST_CH3	Software reset of channel 3. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'. 0 No action. 1 Reset channel.
21 RST_CH2	Software reset of channel 2. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'. 0 No action. 1 Reset channel.
22 RST_CH1	Software reset of channel 1. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'. 0 No action. 1 Reset channel.
23 RST_CH0	Software reset of channel 0. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'. 0 No action. 1 Reset channel.
24–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
31 HOST_TRIG	Trigger request signal (see TGC0, TGC1) to update the register ENDIS_STAT and OUTEN_STAT. This flag is reset automatically after triggering the update. 0 No trigger request. 1 Set trigger request.

11.7.13 TOM0 TGC0 Action Time Base Register (TOM_0_TGC0_ACT_TB)

Address: 8000h base + 34h offset = 8034h



TOM_0_TGC0_ACT_TB field descriptions

Field	Description
0–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–6 TBU_SEL	Selection of time base used for comparison. **NOTE:** The bit combination “10” is only applicable if the TBU of the device contains three time base channels. Otherwise, this bit combination is also reserved. Please refer to GTM Architecture block diagram to determine the number of channels for TBU of this device. 00 TBU_TS0 selected. 01 TBU_TS1 selected. 10 TBU_TS2 selected. 11 TBU_TS0 selected.
7 TB_TRIG	Set trigger request. This flag is reset automatically if the selected time base unit (TBU_TS0 or TBU_TS1 or TBU_TS2 if present) has reached the value ACT_TB and the update of the register was triggered. 0 No trigger request. 1 Set trigger request.
8–31 ACT_TB	Specifies the signed compare value with selected signal TBU_TS[n], x=0..2. If selected TBU_TS[n] value is in the interval [ACT_TB-007FFFFFh, ACT_TB], the event is in the past and the trigger is generated immediately. Otherwise the event is in the future and the trigger is generated if selected TBU_TS[n] is equal to ACT_TB.

11.7.14 TOM0 TGC0 Force Update Control Register (TOM_0_TGC0_FUPD_CTRL)

Address: 8000h base + 38h offset = 8038h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	RSTCN0_CH7		RSTCN0_CH6		RSTCN0_CH5		RSTCN0_CH4		RSTCN0_CH3		RSTCN0_CH2		RSTCN0_CH1		RSTCN0_CH0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	FUPD_CTRL7		FUPD_CTRL6		FUPD_CTRL5		FUPD_CTRL4		FUPD_CTRL3		FUPD_CTRL2		FUPD_CTRL1		FUPD_CTRL0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_TGC0_FUPD_CTRL field descriptions

Field	Description
0–1 RSTCN0_CH7	Reset CN0 of channel 7 on force update event. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
2–3 RSTCN0_CH6	Reset CN0 of channel 6 on force update event. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
4–5 RSTCN0_CH5	Reset CN0 of channel 5 on force update event. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11.

Table continues on the next page...

TOM_0_TGC0_FUPD_CTRL field descriptions (continued)

Field	Description
	11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
6–7 RSTCN0_CH4	Reset CN0 of channel 4 on force update event. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
8–9 RSTCN0_CH3	Reset CN0 of channel 3 on force update event. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
10–11 RSTCN0_CH2	Reset CN0 of channel 2 on force update event. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
12–13 RSTCN0_CH1	Reset CN0 of channel 1 on force update event. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11.

Table continues on the next page...

TOM_0_TGC0_FUPD_CTRL field descriptions (continued)

Field	Description
	11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
14–15 RSTCN0_CH0	Reset CN0 of channel 0 on force update event. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
16–17 FUPD_CTRL7	Force update of TOM channel 7 operation registers. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
18–19 FUPD_CTRL6	Force update of TOM channel 6 operation registers, Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
20–21 FUPD_CTRL5	Force update of TOM channel 5 operation registers. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11.

Table continues on the next page...

TOM_0_TGC0_FUPD_CTRL field descriptions (continued)

Field	Description
	11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
22–23 FUPD_CTRL4	Force update of TOM channel 4 operation registers. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
24–25 FUPD_CTRL3	Force update of TOM channel 3 operation registers. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
26–27 FUPD_CTRL2	ce update of TOM channel 2 operation registers. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
28–29 FUPD_CTRL1	Force update of TOM channel 1 operation registers. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11.

Table continues on the next page...

TOM_0_TGC0_FUPD_CTRL field descriptions (continued)

Field	Description
	11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
30–31 FUPD_CTRL0	Force update of TOM channel 0 operation registers. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.

11.7.15 TOM0 TGC0 Interrupt Trigger Register (TOM_0_TGC0_INT_TRIG)

Address: 8000h base + 3Ch offset = 803Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	INT_TRIG7		INT_TRIG6		INT_TRIG5		INT_TRIG4		INT_TRIG3		INT_TRIG2		INT_TRIG1		INT_TRIG0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_TGC0_INT_TRIG field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–17 INT_TRIG7	Select input signal TRIG_7 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel 7 (TRIG_7) not used, reads as 00. 10 Internal trigger from channel 7 (TRIG_7) used, reads as 11. 11 Don't care, bits 1:0 will not be changed.

Table continues on the next page...

TOM_0_TGC0_INT_TRIG field descriptions (continued)

Field	Description
	Read of following double values means: 00 Internal trigger from channel 7 (TRIG_7) not used. 11 Internal trigger from channel 7 (TRIG_7) used.
18–19 INT_TRIG6	Select input signal TRIG_6 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel 6 (TRIG_6) not used, reads as 00. 10 Internal trigger from channel 6 (TRIG_6) used, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Internal trigger from channel 6 (TRIG_6) not used. 11 Internal trigger from channel 6 (TRIG_6) used.
20–21 INT_TRIG5	Select input signal TRIG_5 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel 5 (TRIG_5) not used, reads as 00. 10 Internal trigger from channel 5 (TRIG_5) used, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Internal trigger from channel 5 (TRIG_5) not used. 11 Internal trigger from channel 5 (TRIG_5) used.
22–23 INT_TRIG4	Select input signal TRIG_4 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel 4 (TRIG_4) not used, reads as 00. 10 Internal trigger from channel 4 (TRIG_4) used, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Internal trigger from channel 4 (TRIG_4) not used. 11 Internal trigger from channel 4 (TRIG_4) used.
24–25 INT_TRIG3	Select input signal TRIG_3 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel 3 (TRIG_3) not used, reads as 00. 10 Internal trigger from channel 3 (TRIG_3) used, reads as 11. 11 Don't care, bits 1:0 will not be changed.

Table continues on the next page...

TOM_0_TGC0_INT_TRIG field descriptions (continued)

Field	Description
	Read of following double values means: 00 Internal trigger from channel 3 (TRIG_3) not used. 11 Internal trigger from channel 3 (TRIG_3) used.
26–27 INT_TRIG2	Select input signal TRIG_2 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel 2 (TRIG_2) not used, reads as 00. 10 Internal trigger from channel 2 (TRIG_2) used, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Internal trigger from channel 2 (TRIG_2) not used. 11 Internal trigger from channel 2 (TRIG_2) used.
28–29 INT_TRIG1	Select input signal TRIG_1 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel 1 (TRIG_1) not used, reads as 00. 10 Internal trigger from channel 1 (TRIG_1) used, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Internal trigger from channel 1 (TRIG_1) not used. 11 Internal trigger from channel 1 (TRIG_1) used.
30–31 INT_TRIG0	Select input signal TRIG_0 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel 0 (TRIG_0) not used, reads as 00. 10 Internal trigger from channel 0 (TRIG_0) used, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Internal trigger from channel 0 (TRIG_0) not used. 11 Internal trigger from channel 0 (TRIG_0) used.

11.7.16 TOM0 TGC0 Enable/Disable Control Register (TOM_0_TGC0_ENDIS_CTRL)

Address: 8000h base + 70h offset = 8070h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	ENDIS_CTRL7		ENDIS_CTRL6		ENDIS_CTRL5		ENDIS_CTRL4		ENDIS_CTRL3		ENDIS_CTRL2		ENDIS_CTRL1		ENDIS_CTRL0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_TGC0_ENDIS_CTRL field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–17 ENDIS_CTRL7	Channel 7 enable/disable update value. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[7]=0) or the output is disabled (OUTEN[7]=0), the TOM channel 7 output TOM_OUT[7] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 15:14 of this register.
18–19 ENDIS_CTRL6	Channel 6 enable/disable update value. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[6]=0) or the output is disabled (OUTEN[6]=0), the TOM channel 6 output TOM_OUT[6] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 13:12 of this register.
20–21 ENDIS_CTRL5	Channel 5 enable/disable update value. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value.

Table continues on the next page...

TOM_0_TGC0_ENDIS_CTRL field descriptions (continued)

Field	Description
	If the channel is disabled (ENDIS[5]=0) or the output is disabled (OUTEN[5]=0), the TOM channel 5 output TOM_OUT[5] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 11:10 of this register.
22–23 ENDIS_CTRL4	TOM channel 4 enable/disable update value. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[4]=0) or the output is disabled (OUTEN[4]=0), the TOM channel 4 output TOM_OUT[4] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 9:8 of this register.
24–25 ENDIS_CTRL3	TOM channel 3 enable/disable update value. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[3]=0) or the output is disabled (OUTEN[3]=0), the TOM channel 3 output TOM_OUT[3] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 7:6 of this register.
26–27 ENDIS_CTRL2	TOM channel 2 enable/disable update value. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[2]=0) or the output is disabled (OUTEN[2]=0), the TOM channel 2 output TOM_OUT[2] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 5:4 of this register.

Table continues on the next page...

TOM_0_TGC0_ENDIS_CTRL field descriptions (continued)

Field	Description
28–29 ENDIS_CTRL1	TOM channel 1 enable/disable update value. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[1]=0) or the output is disabled (OUTEN[1]=0), the TOM channel 1 output TOM_OUT[1] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 3:2 of this register.
30–31 ENDIS_CTRL0	TOM channel 0 enable/disable update value. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[0]=0) or the output is disabled (OUTEN[0]=0), the TOM channel 0 output TOM_OUT[0] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 1:0 of this register.

11.7.17 TOM0 TGC0 Enable/Disable Status Register (TOM_0_TGC0_ENDIS_STAT)

Address: 8000h base + 74h offset = 8074h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	ENDIS_ STAT7		ENDIS_ STAT6		ENDIS_ STAT5		ENDIS_ STAT4		ENDIS_ STAT3		ENDIS_ STAT2		ENDIS_ STAT1		ENDIS_ STAT0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_TGC0_ENDIS_STAT field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

TOM_0_TGC0_ENDIS_STAT field descriptions (continued)

Field	Description
16–17 ENDIS_STAT7	TOM channel 7 enable/disable. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
18–19 ENDIS_STAT6	TOM channel 6 enable/disable. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
20–21 ENDIS_STAT5	TOM channel 5 enable/disable. If a TOM channel is disabled, e counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
22–23 ENDIS_STAT4	TOM channel 4 enable/disable. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value.

Table continues on the next page...

TOM_0_TGC0_ENDIS_STAT field descriptions (continued)

Field	Description
	Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
24–25 ENDIS_STAT3	TOM channel 3 enable/disable. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
26–27 ENDIS_STAT2	TOM channel 2 enable/disable. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
28–29 ENDIS_STAT1	TOM channel 1 enable/disable. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00

Table continues on the next page...

TOM_0_TGC0_ENDIS_STAT field descriptions (continued)

Field	Description
	10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
30–31 ENDIS_STAT0	TOM channel 0 enable/disable. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled

11.7.18 TOM0 TGC0 Out Enable Control Register (TOM_0_TGC0_OUTEN_CTRL)

Address: 8000h base + 78h offset = 8078h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	OUTEN_CTRL7		OUTEN_CTRL6		OUTEN_CTRL5		OUTEN_CTRL4		OUTEN_CTRL3		OUTEN_CTRL2		OUTEN_CTRL1		OUTEN_CTRL0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_TGC0_OUTEN_CTRL field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–17 OUTEN_CTRL7	Output TOM_OUT(7) enable/disable update value.

Table continues on the next page...

TOM_0_TGC0_OUTEN_CTRL field descriptions (continued)

Field	Description
	NOTE: If the channel is disabled (ENDIS[7]=0) or the output is disabled (OUTEN[7]=0), the TOM channel 7 output TOM_OUT[7] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 1:0 of this register.
18–19 OUTEN_CTRL6	Output TOM_OUT(6) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[6]=0) or the output is disabled (OUTEN[6]=0), the TOM channel 6 output TOM_OUT[6] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 1:0 of this register.
20–21 OUTEN_CTRL5	Output TOM_OUT(5) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[5]=0) or the output is disabled (OUTEN[5]=0), the TOM channel 5 output TOM_OUT[5] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 1:0 of this register.
22–23 OUTEN_CTRL4	Output TOM_OUT(4) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[4]=0) or the output is disabled (OUTEN[4]=0), the TOM channel 4 output TOM_OUT[4] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 1:0 of this register.
24–25 OUTEN_CTRL3	Output TOM_OUT(3) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[3]=0) or the output is disabled (OUTEN[3]=0), the TOM channel 3 output TOM_OUT[3] is the inverted value of bit SL.

Table continues on the next page...

TOM_0_TGC0_OUTEN_CTRL field descriptions (continued)

Field	Description
	Write of following double bit values is possible: 00 Don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 1:0 of this register.
26–27 OUTEN_CTRL2	Output TOM_OUT(2) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[2]=0) or the output is disabled (OUTEN[2]=0), the TOM channel 2 output TOM_OUT[2] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 1:0 of this register.
28–29 OUTEN_CTRL1	Output TOM_OUT(1) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[1]=0) or the output is disabled (OUTEN[1]=0), the TOM channel 1 output TOM_OUT[1] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 1:0 of this register.
30–31 OUTEN_CTRL0	Output TOM_OUT(0) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[0]=0) or the output is disabled (OUTEN[0]=0), the TOM channel 0 output TOM_OUT[0] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 1:0 of this register.

11.7.19 TOM0 TGC0 Out Enable Control/Status Register (TOM_0_TGC0_OUTEN_STAT)

Address: 8000h base + 7Ch offset = 807Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	OUTEN_STAT7		OUTEN_STAT6		OUTEN_STAT5		OUTEN_STAT4		OUTEN_STAT3		OUTEN_STAT2		OUTEN_STAT1		OUTEN_STAT0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_TGC0_OUTEN_STAT field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–17 OUTEN_STAT7	Control/status of output TOM_OUT(7). Write of following double bit values is possible: 00 Don’t care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don’t care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
18–19 OUTEN_STAT6	Control/status of output TOM_OUT(6). Write of following double bit values is possible: 00 Don’t care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don’t care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
20–21 OUTEN_STAT5	Control/status of output TOM_OUT(5). Write of following double bit values is possible: 00 Don’t care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00.

Table continues on the next page...

TOM_0_TGC0_OUTEN_STAT field descriptions (continued)

Field	Description
	10 Channel enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
22–23 OUTEN_STAT4	Control/status of output TOM_OUT(4) Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
24–25 OUTEN_STAT3	Control/status of output TOM_OUT(3). Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
26–27 OUTEN_STAT2	Control/status of output TOM_OUT(2). Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
28–29 OUTEN_STAT1	Control/status of output TOM_OUT(1). Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00.

Table continues on the next page...

TOM_0_TGC0_OUTEN_STAT field descriptions (continued)

Field	Description
	10 Channel enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
30–31 OUTEN_STAT0	Control/status of output TOM_OUT(0). Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.

11.7.20 TOM0 TGC1 Global Control Register (TOM_0_TGC1_GLB_CTRL)

Address: 8000h base + 230h offset = 8230h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	UPEN_CTRL15		UPEN_CTRL14		UPEN_CTRL13		UPEN_CTRL12		UPEN_CTRL11		UPEN_CTRL10		UPEN_CTRL9		UPEN_CTRL8	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	RST_CH15	RST_CH14	RST_CH13	RST_CH12	RST_CH11	RST_CH10	RST_CH9	RST_CH8	0							HOST_TRIG
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_TGC1_GLB_CTRL field descriptions

Field	Description
0–1 UPEN_CTRL15	TOM channel 15 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC.

Table continues on the next page...

TOM_0_TGC1_GLB_CTRL field descriptions (continued)

Field	Description
	Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
2–3 UPEN_CTRL14	TOM channel 14 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
4–5 UPEN_CTRL13	TOM channel 13 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
6–7 UPEN_CTRL12	TOM channel 12 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled

Table continues on the next page...

TOM_0_TGC1_GLB_CTRL field descriptions (continued)

Field	Description
	11 Channel enabled
8–9 UPEN_CTRL11	TOM channel 11 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
10–11 UPEN_CTRL10	TOM channel 10 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
12–13 UPEN_CTRL9	TOM channel 9 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
14–15 UPEN_CTRL8	TOM channel 8 enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11

Table continues on the next page...

TOM_0_TGC1_GLB_CTRL field descriptions (continued)

Field	Description
	11 Don’t care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
16 RST_CH15	Software reset of channel 15. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'. 0 No action. 1 Reset channel.
17 RST_CH14	Software reset of channel 14. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'. 0 No action. 1 Reset channel.
18 RST_CH13	Software reset of channel 13. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'. 0 No action. 1 Reset channel.
19 RST_CH12	Software reset of channel 12. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'. 0 No action. 1 Reset channel.
20 RST_CH11	Software reset of channel 11. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'. 0 No action. 1 Reset channel.
21 RST_CH10	Software reset of channel 10. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'. 0 No action. 1 Reset channel.
22 RST_CH9	Software reset of channel 9. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'.

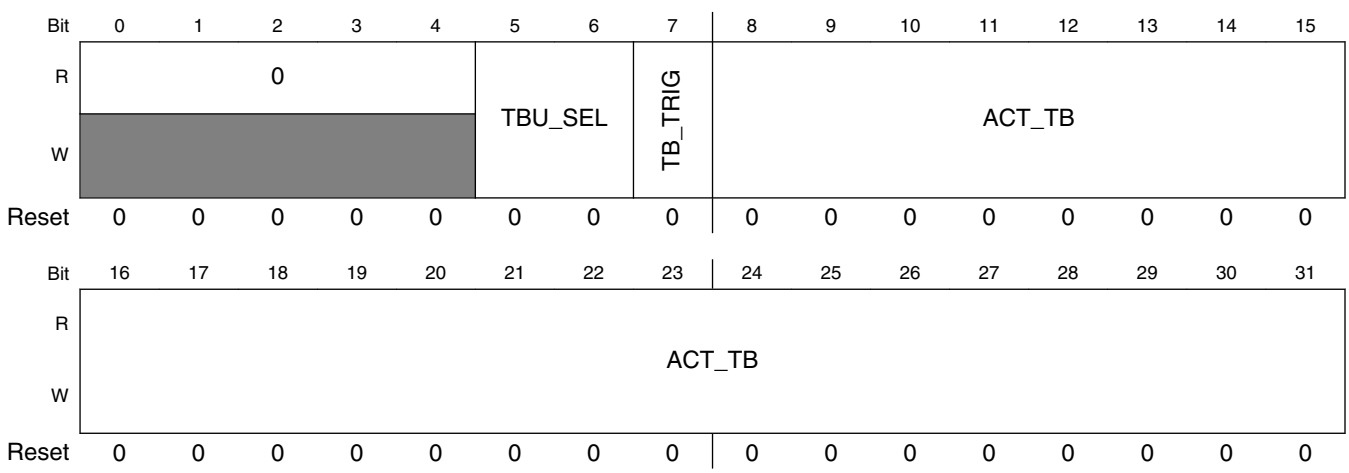
Table continues on the next page...

TOM_0_TGC1_GLB_CTRL field descriptions (continued)

Field	Description
	0 No action. 1 Reset channel.
23 RST_CH8	Software reset of channel 8. This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is set to '1'. 0 No action. 1 Reset channel.
24–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
31 HOST_TRIG	Trigger request signal (see TGC0, TGC1) to update the register ENDIS_STAT and OUTEN_STAT. This flag is reset automatically after triggering the update. 0 No trigger request. 1 Set trigger request.

11.7.21 TOM0 TGC1 Action Time Base Register (TOM_0_TGC1_ACT_TB)

Address: 8000h base + 234h offset = 8234h



TOM_0_TGC1_ACT_TB field descriptions

Field	Description
0–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–6 TBU_SEL	Selection of time base used for comparison.

Table continues on the next page...

TOM_0_TGC1_ACT_TB field descriptions (continued)

Field	Description
	NOTE: The bit combination “10” is only applicable if the TBU of the device contains three time base channels. Otherwise, this bit combination is also reserved. Please refer to GTM Architecture block diagram to determine the number of channels for TBU of this device. 00 TBU_TS0 selected. 01 TBU_TS1 selected. 10 TBU_TS2 selected. 11 TBU_TS0 selected.
7 TB_TRIG	Set trigger request. This flag is reset automatically if the selected time base unit (TBU_TS0 or TBU_TS1 or TBU_TS2 if present) has reached the value ACT_TB and the update of the register was triggered. 0 No trigger request. 1 Set trigger request.
8–31 ACT_TB	Specifies the signed compare value with selected signal TBU_TS[n], x=0..2. If selected TBU_TS[n] value is in the interval [ACT_TB-007FFFFFh, ACT_TB], the event is in the past and the trigger is generated immediately. Otherwise the event is in the future and the trigger is generated if selected TBU_TS[n] is equal to ACT_TB.

11.7.22 TOM0 TGC1 Force Update Control Register (TOM_0_TGC1_FUPD_CTRL)

Address: 8000h base + 238h offset = 8238h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	RSTCN0_CH15		RSTCN0_CH14		RSTCN0_CH13		RSTCN0_CH12		RSTCN0_CH11		RSTCN0_CH10		RSTCN0_CH9		RSTCN0_CH8	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	FUPD_CTRL15		FUPD_CTRL14		FUPD_CTRL13		FUPD_CTRL12		FUPD_CTRL11		FUPD_CTRL10		FUPD_CTRL9		FUPD_CTRL8	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_TGC1_FUPD_CTRL field descriptions

Field	Description
0–1 RSTCN0_CH15	Reset CN0 of channel 15 on force update event. Write of following double bit values is possible: 00 Don’t care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don’t care, bits 1:0 will not be changed. Read of following double values means:

Table continues on the next page...

TOM_0_TGC1_FUPD_CTRL field descriptions (continued)

Field	Description
	00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
2–3 RSTCN0_CH14	Reset CN0 of channel 14 on force update event. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
4–5 RSTCN0_CH13	Reset CN0 of channel 13 on force update event. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
6–7 RSTCN0_CH12	Reset CN0 of channel 12 on force update event. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
8–9 RSTCN0_CH11	Reset CN0 of channel 11 on force update event. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means:

Table continues on the next page...

TOM_0_TGC1_FUPD_CTRL field descriptions (continued)

Field	Description
	00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
10–11 RSTCN0_CH10	Reset CN0 of channel 10 on force update event. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
12–13 RSTCN0_CH9	Reset CN0 of channel 9 on force update event. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
14–15 RSTCN0_CH8	Reset CN0 of channel 8 on force update event. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
16–17 FUPD_CTRL15	Force update of TOM channel 15 operation registers. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means:

Table continues on the next page...

TOM_0_TGC1_FUPD_CTRL field descriptions (continued)

Field	Description
	00 Channel disabled. 11 Channel enabled.
18–19 FUPD_CTRL14	Force update of TOM channel 14 operation registers, Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
20–21 FUPD_CTRL13	Force update of TOM channel 13 operation registers. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
22–23 FUPD_CTRL12	Force update of TOM channel 12 operation registers. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
24–25 FUPD_CTRL11	Force update of TOM channel 11 operation registers. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means:

Table continues on the next page...

TOM_0_TGC1_FUPD_CTRL field descriptions (continued)

Field	Description
	00 Channel disabled. 11 Channel enabled.
26–27 FUPD_CTRL10	ce update of TOM channel 10 operation registers. Write of following double bit values is possible: 00 Don’t care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don’t care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
28–29 FUPD_CTRL9	Force update of TOM channel 9 operation registers. Write of following double bit values is possible: 00 Don’t care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don’t care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
30–31 FUPD_CTRL8	Force update of TOM channel 8 operation registers. Write of following double bit values is possible: 00 Don’t care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don’t care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.

11.7.23 TOM0 TGC1 Interrupt Trigger Register (TOM_0_TGC1_INT_TRIG)

Address: 8000h base + 23Ch offset = 823Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	INT_TRIG15		INT_TRIG14		INT_TRIG13		INT_TRIG12		INT_TRIG11		INT_TRIG10		INT_TRIG9		INT_TRIG8	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_TGC1_INT_TRIG field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–17 INT_TRIG15	Select input signal TRIG_15 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel not used, reads as 00. 10 Internal trigger from channel used, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Internal trigger from channel not used. 11 Internal trigger from channel used.
18–19 INT_TRIG14	Select input signal TRIG_14 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel not used, reads as 00. 10 Internal trigger from channel used, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Internal trigger from channel not used. 11 Internal trigger from channel used.
20–21 INT_TRIG13	Select input signal TRIG_13as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel not used, reads as 00.

Table continues on the next page...

TOM_0_TGC1_INT_TRIG field descriptions (continued)

Field	Description
	10 Internal trigger from channel used, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Internal trigger from channel not used. 11 Internal trigger from channel used.
22–23 INT_TRIG12	Select input signal TRIG_12 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel not used, reads as 00. 10 Internal trigger from channel used, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Internal trigger from channel not used. 11 Internal trigger from channel used.
24–25 INT_TRIG11	Select input signal TRIG_1 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel not used, reads as 00. 10 Internal trigger from channel used, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Internal trigger from channel not used. 11 Internal trigger from channel used.
26–27 INT_TRIG10	Select input signal TRIG_10 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel not used, reads as 00. 10 Internal trigger from channel used, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Internal trigger from channel not used. 11 Internal trigger from channel used.
28–29 INT_TRIG9	Select input signal TRIG_9 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel not used, reads as 00.

Table continues on the next page...

TOM_0_TGC1_INT_TRIG field descriptions (continued)

Field	Description
	10 Internal trigger from channel used, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Internal trigger from channel not used. 11 Internal trigger from channel used.
30–31 INT_TRIG8	Select input signal TRIG_8 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Internal trigger from channel not used, reads as 00. 10 Internal trigger from channel used, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Internal trigger from channel not used. 11 Internal trigger from channel used.

11.7.24 TOM0 TGC1 Enable/Disable Control Register (TOM_0_TGC1_ENDIS_CTRL)

Address: 8000h base + 270h offset = 8270h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	ENDIS_ CTRL15		ENDIS_ CTRL14		ENDIS_ CTRL13		ENDIS_ CTRL12		ENDIS_ CTRL11		ENDIS_ CTRL10		ENDIS_ CTRL9		ENDIS_ CTRL8	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_TGC1_ENDIS_CTRL field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–17 ENDIS_CTRL15	Channel 15 enable/disable update value. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[15]=0) or the output is disabled (OUTEN[15]=0), the TOM channel 15 output TOM_OUT[15] is the inverted value of bit SL.

Table continues on the next page...

TOM_0_TGC1_ENDIS_CTRL field descriptions (continued)

Field	Description
	Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 15:14 of this register.
18–19 ENDIS_CTRL14	Channel 14 enable/disable update value. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[14]=0) or the output is disabled (OUTEN[14]=0), the TOM channel 14 output TOM_OUT[14] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 13:12 of this register.
20–21 ENDIS_CTRL13	Channel 13 enable/disable update value. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[13]=0) or the output is disabled (OUTEN[13]=0), the TOM channel 13 output TOM_OUT[13] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 11:10 of this register.
22–23 ENDIS_CTRL12	TOM channel 12 enable/disable update value. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[12]=0) or the output is disabled (OUTEN[12]=0), the TOM channel 12 output TOM_OUT[12] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 9:8 of this register.
24–25 ENDIS_CTRL11	TOM channel 11 enable/disable update value. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value.

Table continues on the next page...

TOM_0_TGC1_ENDIS_CTRL field descriptions (continued)

Field	Description
	If the channel is disabled (ENDIS[11]=0) or the output is disabled (OUTEN[11]=0), the TOM channel 11 output TOM_OUT[11] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 7:6 of this register.
26–27 ENDIS_CTRL10	TOM channel 10 enable/disable update value. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[10]=0) or the output is disabled (OUTEN[10]=0), the TOM channel 10 output TOM_OUT[10] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 5:4 of this register.
28–29 ENDIS_CTRL9	TOM channel 9 enable/disable update value. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[9]=0) or the output is disabled (OUTEN[9]=0), the TOM channel 9 output TOM_OUT[9] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 3:2 of this register.
30–31 ENDIS_CTRL8	TOM channel 8 enable/disable update value. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[8]=0) or the output is disabled (OUTEN[8]=0), the TOM channel 8 output TOM_OUT[8] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 1:0 of this register.

11.7.25 TOM0 TGC1 Enable/Disable Status Register (TOM_0_TGC1_ENDIS_STAT)

Address: 8000h base + 274h offset = 8274h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	ENDIS_ STAT15		ENDIS_ STAT14		ENDIS_ STAT13		ENDIS_ STAT12		ENDIS_ STAT11		ENDIS_ STAT10		ENDIS_ STAT9		ENDIS_ STAT8	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_TGC1_ENDIS_STAT field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–17 ENDIS_STAT15	TOM channel 15 enable/disable. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don’t care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don’t care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
18–19 ENDIS_STAT14	TOM channel 14 enable/disable. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don’t care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don’t care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled

Table continues on the next page...

TOM_0_TGC1_ENDIS_STAT field descriptions (continued)

Field	Description
20–21 ENDIS_STAT13	TOM channel 13 enable/disable. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
22–23 ENDIS_STAT12	TOM channel 12 enable/disable. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
24–25 ENDIS_STAT11	TOM channel 11 enable/disable. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
26–27 ENDIS_STAT10	TOM channel 10 enable/disable. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value.

Table continues on the next page...

TOM_0_TGC1_ENDIS_STAT field descriptions (continued)

Field	Description
	Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
28–29 ENDIS_STAT9	TOM channel 9 enable/disable. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
30–31 ENDIS_STAT8	TOM channel 8 enable/disable. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 1:0 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled

11.7.26 TOM0 TGC1 Out Enable Control Register (TOM_0_TGC1_OUTEN_CTRL)

Address: 8000h base + 278h offset = 8278h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	OUTEN_CTRL15		OUTEN_CTRL14		OUTEN_CTRL13		OUTEN_CTRL12		OUTEN_CTRL11		OUTEN_CTRL10		OUTEN_CTRL9		OUTEN_CTRL8	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_TGC1_OUTEN_CTRL field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–17 OUTEN_CTRL15	Output TOM_OUT(15) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[15]=0) or the output is disabled (OUTEN[15]=0), the TOM channel15 output TOM_OUT[15] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don’t care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don’t change bits 1:0 of this register.
18–19 OUTEN_CTRL14	Output TOM_OUT(14) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[14]=0) or the output is disabled (OUTEN[14]=0), the TOM channel 14 output TOM_OUT[14] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don’t care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don’t change bits 1:0 of this register.
20–21 OUTEN_CTRL13	Output TOM_OUT(13) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[13]=0) or the output is disabled (OUTEN[5]=13), the TOM channel 13 output TOM_OUT[13] is the inverted value of bit SL. Write of following double bit values is possible:

Table continues on the next page...

TOM_0_TGC1_OUTEN_CTRL field descriptions (continued)

Field	Description
	00 Don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 1:0 of this register.
22–23 OUTEN_CTRL12	Output TOM_OUT(12) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[12]=0) or the output is disabled (OUTEN[12]=0), the TOM channel 12 output TOM_OUT[12] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 1:0 of this register.
24–25 OUTEN_CTRL11	Output TOM_OUT(11) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[11]=0) or the output is disabled (OUTEN[11]=0), the TOM channel 11 output TOM_OUT[11] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 1:0 of this register.
26–27 OUTEN_CTRL10	Output TOM_OUT(10) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[10]=0) or the output is disabled (OUTEN[10]=0), the TOM channel 10 output TOM_OUT[10] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 1:0 of this register.
28–29 OUTEN_CTRL9	Output TOM_OUT(9) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[9]=0) or the output is disabled (OUTEN[9]=0), the TOM channel 9 output TOM_OUT[9] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger.

Table continues on the next page...

TOM_0_TGC1_OUTEN_CTRL field descriptions (continued)

Field	Description
	10 Enable channel output on an update trigger. 11 Don't change bits 1:0 of this register.
30–31 OUTEN_CTRL8	Output TOM_OUT(8) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[8]=0) or the output is disabled (OUTEN[8]=0), the TOM channel 8 output TOM_OUT[8] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 1:0 of this register.

11.7.27 TOM0 TGC1 Out Enable Control/Status Register (TOM_0_TGC1_OUTEN_STAT)

Address: 8000h base + 27Ch offset = 827Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	OUTEN_STAT15		OUTEN_STAT14		OUTEN_STAT13		OUTEN_STAT12		OUTEN_STAT11		OUTEN_STAT10		OUTEN_STAT9		OUTEN_STAT8	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOM_0_TGC1_OUTEN_STAT field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–17 OUTEN_STAT15	Control/status of output TOM_OUT(15). Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable.

Table continues on the next page...

TOM_0_TGC1_OUTEN_STAT field descriptions (continued)

Field	Description
	11 Channel enable.
18–19 OUTEN_STAT14	Control/status of output TOM_OUT(14). Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
20–21 OUTEN_STAT13	Control/status of output TOM_OUT(13). Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
22–23 OUTEN_STAT12	Control/status of output TOM_OUT(12) Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
24–25 OUTEN_STAT11	Control/status of output TOM_OUT(11). Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable.

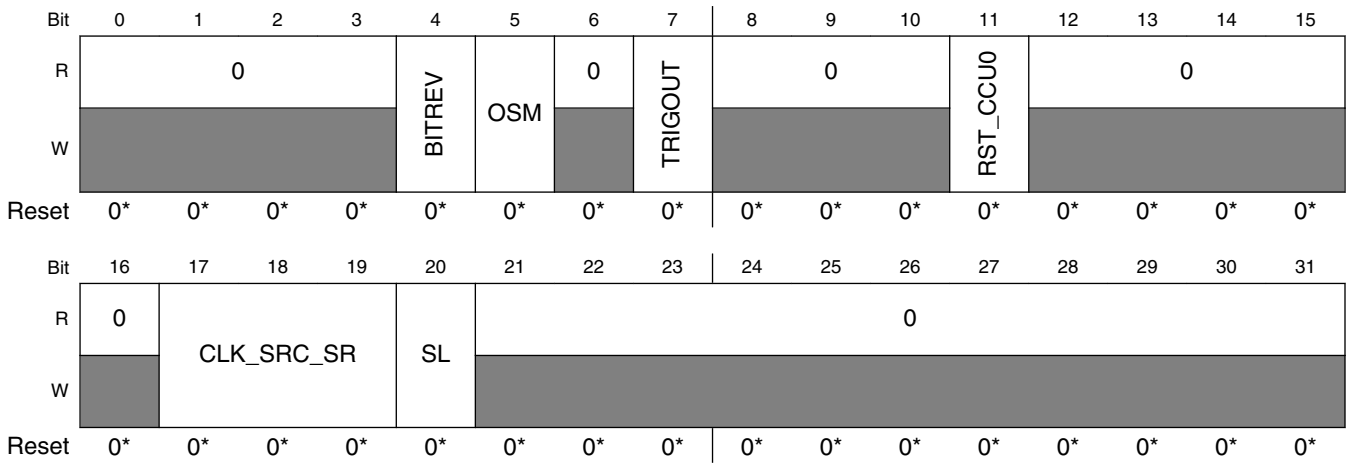
Table continues on the next page...

TOM_0_TGC1_OUTEN_STAT field descriptions (continued)

Field	Description
	11 Channel enable.
26–27 OUTEN_STAT10	Control/status of output TOM_OUT(10). Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
28–29 OUTEN_STAT9	Control/status of output TOM_OUT(9). Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
30–31 OUTEN_STAT8	Control/status of output TOM_OUT(8). Write of following double bit values is possible: 00 Don't care, bits 1:0 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 1:0 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.

11.7.28 TOM0 Channel 15 Control register (TOM_0_CH15_CTRL)

Address: 8000h base + 3C0h offset = 83C0h



* Notes:
- Bits 20-23 do not reset to a given value.

TOM_0_CH15_CTRL field descriptions

Field	Description
0–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 BITREV	Bit-reversing of output of counter register CN0.. This bit enables the PCM mode of channel 15.
5 OSM	One-shot mode. In this mode the counter CN0 counts for only one period. The length of period is defined by CM0. A write access to the register CM0 triggers the start of counting. 0 One-shot mode disabled. 1 One-shot mode enabled.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 TRIGOUT	Trigger output selection (output signal TRIG_15) of module TOM_CH15. 0 TRIG_15 is TRIG_14. 1 TRIG_15 is TRIG_CCU0.
8–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 RST_CCU0	Reset source of CCU0. 0 Reset counter register CN0 to 0 on matching comparison CM0. 1 Reset counter register CN0 to 0 on trigger TRIG_14.

Table continues on the next page...

TOM_0_CH15_CTRL field descriptions (continued)

Field	Description
12–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–19 CLK_SRC_SR	Clock source select for channel. The register CLK_SRC is updated with the value of CLK_SRC_SR together with the update of register CM0 and CM1. The input of the FX clock divider depends on the value of FXCLK_SEL (see CMU) 000 CMU_FXCLK(0) selected: GTM system clock. 001 CMU_FXCLK(1) selected: GTM system clock / 2^4 . 010 CMU_FXCLK(2) selected: GTM system clock / 2^8 . 011 CMU_FXCLK(3) selected: GTM system clock / 2^{12} . 100 CMU_FXCLK(4) selected: GTM system clock / 2^{16} . 101 No CMU_FXCLK selected, clock of channel stopped. 110 No CMU_FXCLK selected, clock of channel stopped. 111 No CMU_FXCLK selected, clock of channel stopped. **NOTE:** If clock of channel is stopped, the channel can only be restarted by resetting CLK_SRC_SR to a value of 000 to 100 and by forcing an update via the force update mechanism.
20 SL	Signal level for duty cycle. If the output is disabled, the output ATOM_OUT[n] is set to inverse value of SL. 0 Low signal level. 1 High signal level.
21–31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Chapter 12
ARU-connected Timer Output Module (ATOM)

12.1 ATOM Overview

The ARU-connected Timer Output Module (ATOM) is able to generate complex output signals without CPU interaction. Typically, output signal characteristics are provided through the ARU to destinations such as the MCS, DPLL or PSM submodules. Each ATOM has eight ATOM_CHn_OUT output channels, which can operate independently of each other. Figure 12-1 shows a block diagram of the ATOM.

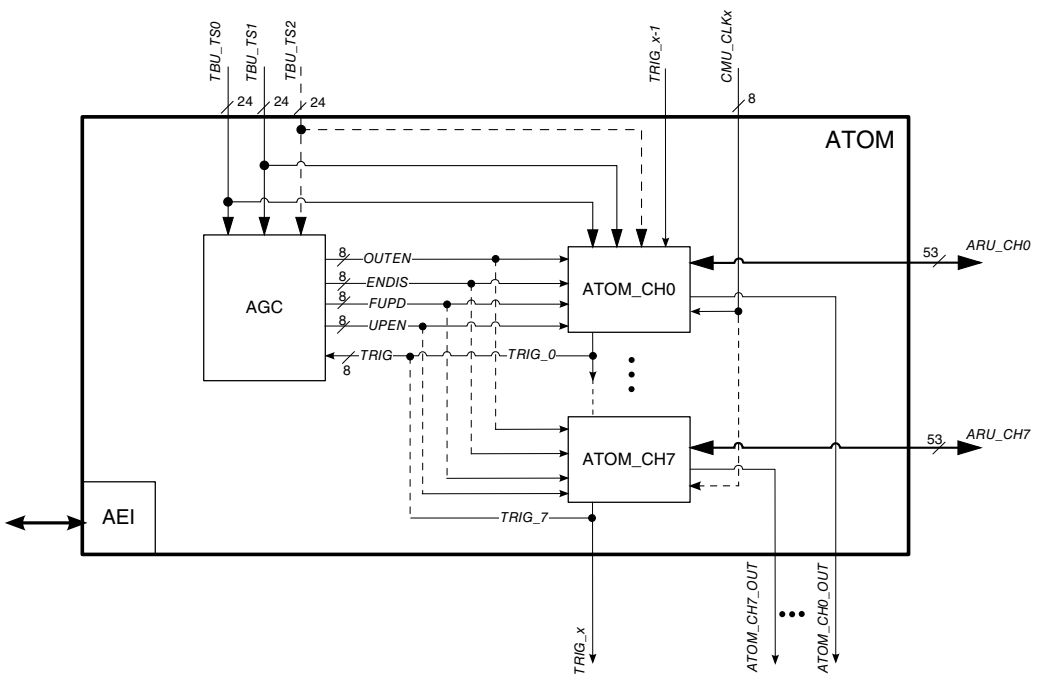


Figure 12-1. ATOM block diagram

The architecture of the ATOM is similar to that of the TOM submodule, but there are some differences:

- The ATOM integrates only eight output channels. Hence, there is only one ATOM Global Control subunit (AGC) for the eight ATOM channels.

- The ATOM is connected to the ARU and it can issue individual read requests from the ARU and write requests to the ARU.
- The ATOM channels are able to generate signals on behalf of time stamps.
- The ATOM channels are able to generate a serial output signal on behalf of an internal shift register.

Each ATOM channel can operate in one of four modes (see ATOM Channel modes):

- ATOM Signal Output Mode Immediate (SOMI).
- ATOM Signal Output Mode Compare (SOMC).
- ATOM Signal Output Mode PWM (SOMP).
- ATOM Signal Output Mode Serial (SOMS).

The operation registers (e.g. CN0 counter register, CM0 and CM1 compare registers) in the ATOM channels are 24 bits wide. The *CMU_CLKn* input clock for an ATOM channel is sourced by the CMU submodule, and is configured by writing to the ATOM[i]_CH_CTRL[CLK_SRC/CLK_SRC_SR] bit field.

Each ATOM channel is able to generate a serial bit stream, which is shifted out at the ATOM[i]_CH[x]_OUT output. When configured in this serial shift mode (SOMS), the selected CMU_CLKn clock defines the shift frequency.

Each ATOM channel has:

- a CN0 counter register,
- CM0 and CM1 compare registers, and
- SR0 and SR1 shadow registers.

The SR0 and SR1 shadow registers can be loaded with new parameters from the CPU and/or the ARU while the CN0, CN1 and CM0 registers operate with current parameters. When in SOMP mode, and when an update via the ARU is selected by setting the **ATOM[i]_CHn_CTRL[ARU_EN]** bit to one, updating of both shadow registers or only one shadow register can be configured.

For an ATOM channel, it is possible to simultaneously:

- load the contents of the SR0 and SR1 shadow registers into the corresponding CM0 and CM1 compare registers, and
- change the CMU_CLKn input clock selection for the CN0 counter register.

The ARU connection to the ATOM can be enabled or disabled by writing to the ATOM[i]_CH_CTRL[ARU_EN] bit. When ARU_EN = 0, the behavior of the CCU0 and CCU1 compare units, and the output signal, is controlled by the ACB bit field inside the **ATOM[i]_CHn_CTRL[ACB]** bit field. register. When ARU_EN = 1, the behavior is controlled via the ARU through the **ATOM[i]_CHn_STAT[ACBI]** bit field.

The SR0 and SR1 shadow registers of an ATOM channel can be reloaded via the ARU, or by the CPU over the ATOM's AEI interface. When loaded via the ARU, the shadow registers act as buffers between the ARU and the channel's operation registers. For example, a new parameter set for a PWM signal can be reloaded via the ARU into the shadow registers, while the operation registers work on the actual parameter set.

12.1.1 ATOM Global control (AGC)

The AGC subunit provides the capability to synchronously start or stop one or more ATOM output channels. The AGC has the same functionality as that of the TGC subunit in the TOM submodule. For a description of the AGC subunit's functionality, please see TGC Subunit.

12.2 ATOM Channel architecture

Each ATOM channel can generate an output signal according to four operation modes. The general architecture of an ATOM channel is shown in Figure 12-2.

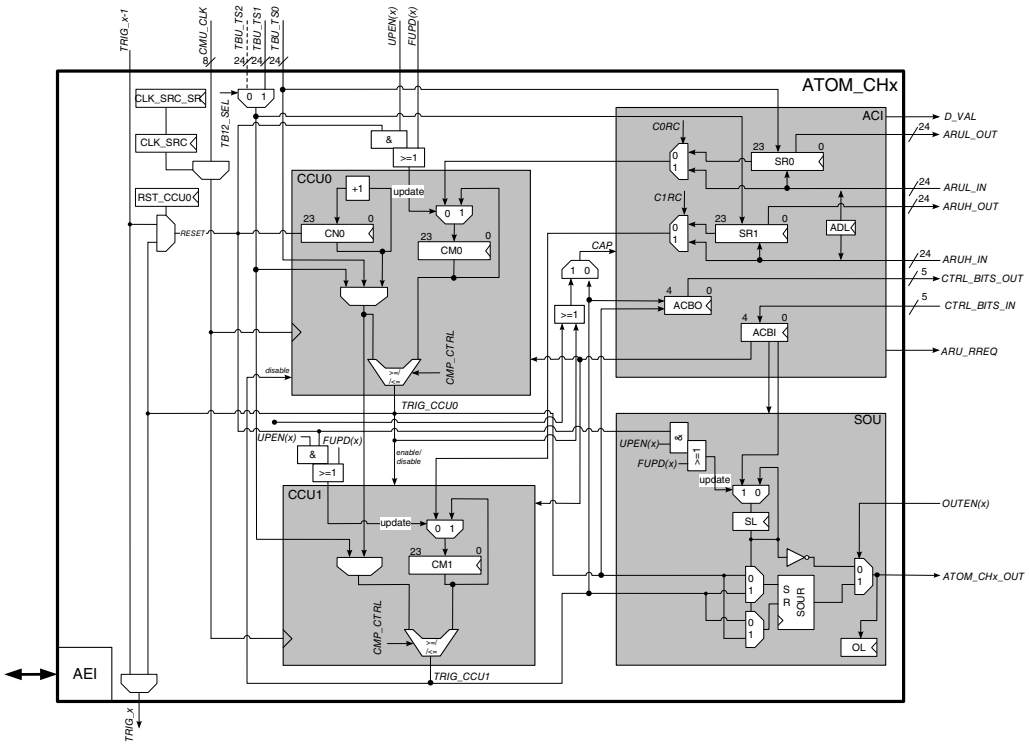


Figure 12-2. ATOM channel architecture

The ATOM channels have 24-bit wide **CN0**, **CM0** and **CM1** operation registers and 24-bit wide **SR0** and **SR1** shadow registers. The comparators inside CCU0 and CCU1 provide a selectable signed ≥ or ≤ comparison against the *TBU_TS0* and *TBU_TS1* time bases that are received from the TBU submodule. If a *TBU_TS2* time base is implemented, it can be selected inside the ATOM channel by setting the **ATOM[i]_CHn_CTRL[TB12_SEL]** bit to one (see TBU Overview and subsequent sections for further details). For an overview of the implemented TBU submodule version please refer to GTM Architecture.

The CCU0 and CCU1 units perform different tasks depending on the configured channel mode. The selectable signed ≥ or ≤ comparisons are used to detect time base overflows. They can also be used to guarantee that a compare match event can be set up for the future even if the time base overflows before reaching the match value.

NOTE

To insure correct behavior of this signed comparison, the new compare value must not be specified larger or smaller than half of the range of the total time base value (0x7FFFFF).

In SOMC mode, the two CCU0 and CCU1 compare units can be used in combination with each other where the TRIG_CCU0 and TRIG_CCU1 trigger signals can be used to enable or disable the other compare unit on a match event (see ATOM Signal Output Mode Compare (SOMC) for further details.

The Signal Output Unit (SOU) generates the output signal for each ATOM channel. The level of the output signal depends on the channel's configured operating mode and on the level of the **ATOM[i]_CHn_CTRL[SL]**bit in combination with the ACB1 and ACB0 control bits. These two control bits, bits zero and one of the 5-bit ACB field, can be received:

- via the CPU into the **ATOM[i]_CHn_CTRL[ACB]** bit field, or
- via the ARU into the the **ATOM[i]_CHn_STAT[ACBI]** bit field.

In all operating modes, the **ATOM[i]_CHn_CTRL[SL]** bit defines the operational behavior of the ATOM channel.

When the channel and its output is disabled, the output signal level of the channel is the inverse of the SL bit.

In the SOMI and SOMC modes, the output signal level depends on the SL, ACB0 and ACB1 bits. In SOMP mode, the output signal level depends on the *TRIG_CCU0* and *TRIG_CCU1* trigger signals (that define the PWM timing characteristics) and the SL bit (that defines the level of the duty cycle). In SOMS mode, the output signal level is defined by the CM1 register bit pattern that is shifted out of the ATOM[i]_CHn_OUT output.

The ARU Communication Interface (ACI) subunit is responsible for requesting data that is routed through ARU to the ATOM channel in SOMI, SOMP and SOMS modes, and for providing data to the ARU in SOMC mode (where the SR0 and SR1 shadow registers are used as output buffer registers).

12.2.1 ARU Communication Interface

Each ATOM channel has an ARU Communication Interface (ACI) subunit, as shown in Figure 12-3 . The ACI uses the SR0 and SR1 registers with the **ATOM[i]_CHn_STAT[ACBI, ACBO]** bit fields to implement data exchange to and from the ARU module.

If the ARU interface is enabled by setting the **ATOM[i]_CHn_CTRL[ARU_EN]** to one and the channel is enabled by writing to the corresponding **ATOM[i]_AGC_ENDIS_STAT[ENDIS_CTRLn]** bit field (and the CPU has not written data that is not equal to zero into the **CM0**, **CM1, SR0, SR1** registers), the ATOM channel will request data from the ARU before signal generation starts in either the SOMP, SOMS or SOMC mode.

Note

When in SOMP mode, and data inside the **CM0** or **SR0** registers is not equal to zero, the **CN0** counter register will immediately start incrementing regardless of whether or not the channel has received ARU data.

Note

When in SOMS mode, and data inside the **CM0** or **SR0** registers is not equal to zero, the channel will immediately start shifting regardless of whether or not the channel has received ARU data.

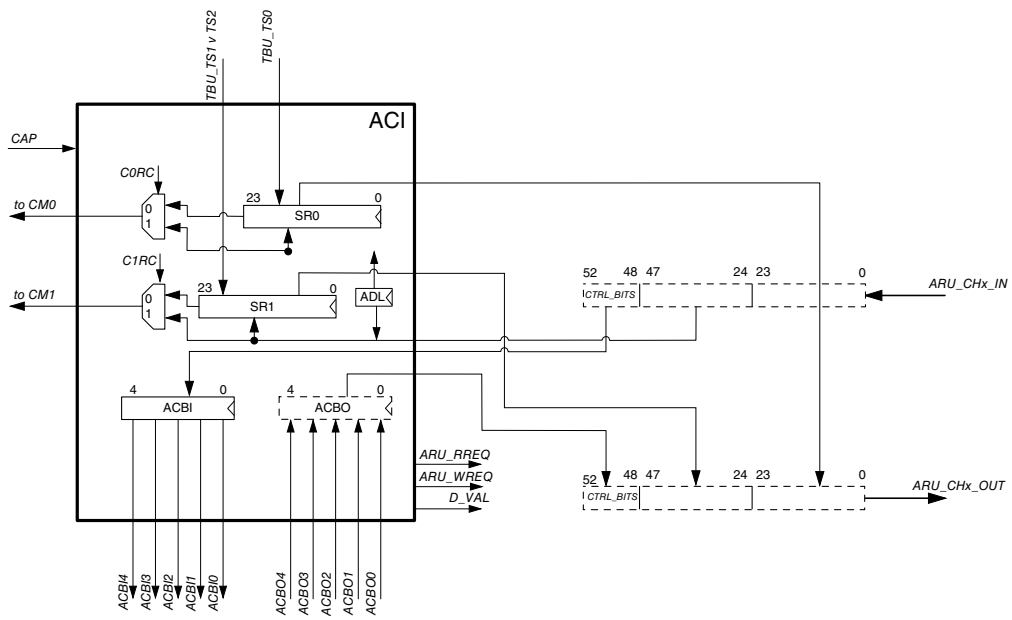


Figure 12-3. ACI architecture

An incoming *ARU_CHn_IN* data word is 53 bits wide. When in either the SOMI, SOMP or SOMS mode, the *ARU_CHn_IN* data word is split three ways:

- the lower 24 bits (23:0) are stored in the **SR0** register,
- the upper 24 bits (47:24) are stored in the **SR1** register, and
- control bits 52 through 48 are stored in the **ATOM[i]_CHn_STAT[ACBI]** bit field.

If the CM0 and CM1 registers are to be updated from the SR0 and SR1 registers, but the ACI receives a data word from the ARU at the same time, the ATOM channel must ensure that either the old data in both shadow registers is transferred into the CM0 and CM1 registers or both new values from the ARU are transferred into the CM0 and CM1 registers.

When in the SOMC mode, the *ARU_CHn_IN* data word is split three ways:

- the lower 24 bits (23:0) are stored in the **CM0** compare register,

- the upper 24 bits (47:24) are stored in the **CM1** compare register, and
- bits 52 through 48 are stored in the **ATOM[i]_CHn_STAT[ACBI]** bit field.

When in the SOMC mode, and when a 'true' compare event in the CCU0 and/or CCU1 subunits is indicated by the *CAP* signal, the **SR0** and **SR1** registers serve as capture registers for the two time stamps that are received from the TBU submodule. If the ARU is enabled, the two time stamps in the **SR0** and **SR1** registers and the actual ATOM channel status information from the **ATOM[i]_CHn_STAT[ACBO]** bit field is provided to the ARU at the dedicated ARU write address for the ATOM channel.

The encoding of the ARU control bits for the ATOM operation modes is described further in subsequent sections of this chapter.

12.3 ATOM Channel modes

ATOM channels can operate independently from each other. Each channel can be configured to operate in one of four dedicated output modes by writing to the ATOM[i]_CH_CTRL[MODE] bit field:

- ATOM Signal Output Mode Immediate (SOMI), see ATOM Signal Output Mode Immediate (SOMI).
- ATOM Signal Output Mode Compare (SOMC), see ATOM Signal Output Mode Compare (SOMC).
- ATOM Signal Output Mode PWM (SOMP), see ATOM Signal Output Mode PWM (SOMP).
- ATOM Signal Output Mode Serial (SOMS), see ATOM Signal Output Mode Serial (SOMS).

The Signal Output Mode PWM (SOMP) is the same as the output mode for the TOM submodule except that the bit reverse mode is not included in the ATOM. It is possible to reload the shadow registers via the ARU without CPU interaction.

The SOMI, SOMC, and SOMS modes provide additional functionality for output signal control.

NOTE

When a channel becomes enabled in any output mode, the **CN0** counter register increments as long as **CM0 >= CN0.** CN0 stops incrementing when it's value reaches the value in the **CM0** register. Additionally, when a channel becomes enabled in SOMP or SOMS mode, one shot mode becomes disabled

(ATOM[i]_CH_CTRL[OSM] bit is cleared to zero). To avoid unintended counting of the **CN0** counter register after enabling a channel, it is recommended that the channel be reset (or at least **CN0** and **CM0** be reset to zero) before writing to any of the ATOM[i]_CH_CTRL[MODE, ARU_EN, OSM] bits.

12.3.1 ATOM Signal Output Mode Immediate (SOMI)

In ATOM Signal Output Mode Immediate (SOMI), an ATOM channel generates an output signal on the *ATOM[i]_CHn_OUT* output immediately after an update of either bit zero of the **ATOM[i]_CHn_STAT[ACBI]** 5-bit field or bit zero of the **ATOM[i]_CHn_CTRL[ACB]** 5-bit field.

If ARU access is enabled (**ATOM[i]_CHn_CTRL[ARU_EN]** bit = 1), the update of the *ATOM[i]_CHn_OUT* output level (see Table 12-1) depends on bit zero of the **ATOM[i]_CHn_STAT[ACBI]** 5-bit field (that is received by the ACI subunit) and the **ATOM[i]_CHn_CTRL[SL]** bit. The remaining 48 ARU bits (47:0) (that were received by the ACI subunit) have no meaning in this mode.

If ARU access is disabled (**ATOM[i]_CHn_CTRL[ARU_EN]** bit = 0), the update of the *ATOM[i]_CHn_OUT* output level depends on bit zero of the ATOM[i]_CHn_CTRL[ACB] 5-bit field and the ATOM[i]_CHn_CTRL[SL] bit.

When the **ATOM[i]_AGC_OUTEN_CTRL[OUTEN_CTRLn]** bit field = 01 (disabled), the initial *ATOM[i]_CHn_OUT*output signal level is specified by the ATOM[i]_CHn_CTRL[SL] bit (see Figure 12-2).

The output level depends on the **ATOM[i]_CHn_CTRL[SL]** bit and either bit zero of the **ATOM[i]_CHn_STAT[ACBI]** 5-bit field or bit zero of the **ATOM[i]_CHn_CTRL[ACB]** 5-bit field.

Table 12-1. ATOM[i] Channel Output Level

SL Bit	ACBI(0) or ACB(0)	Output Level
0	0	Output level = high (inverse of SL bit)
0	1	Output level = low (SL bit)
1	0	Output level = low (inverse of SL bit)
1	1	Output level = high (SL bit)

Immediately after the data is received by the ACI, bit zero of the **ATOM[i]_CHn_STAT[ACBI]** 5-bit field is transferred to the SOU subunit and then routed to the ATOM[i]_CHn_OUT output, with the level specified in Table 12-1. This can introduce a jitter on the output signal since the ARU channels are served in a time multiplexed fashion.

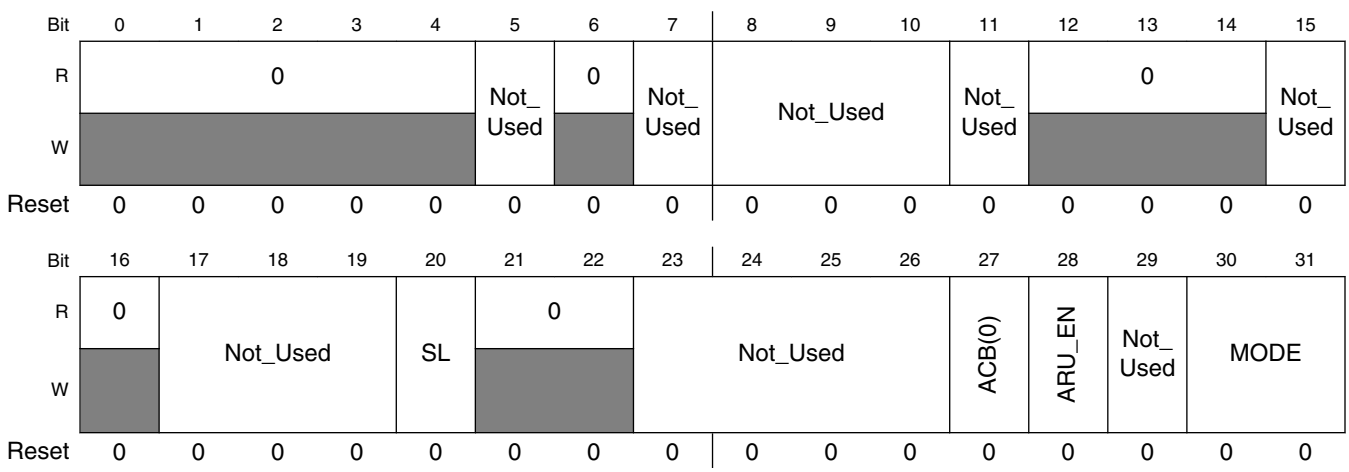
12.3.1.1 Memory Map and Registers

memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
D004	ATOM Channel Control Register in SOMI mode (ATOM[i]_CHn_CTRL_in_SOMI)	32	R/W	See section	12.3.1.1.1/ 313

12.3.1.1.1 ATOM Channel Control Register in SOMI mode (ATOM[i]_CHn_CTRL_in_SOMI)

Address: 0h base + D004h offset = D004h



ATOM[i]_CHn_CTRL_in_SOMI field descriptions

Field	Description
0–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 Not_Used	Not used in SOMI.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0. 0 Capture SRn time stamps after CCU0 match event not provided to ARU. 1 Capture SRx time stamps after CCU0 match event provided to ARU.

Table continues on the next page...

ATOM[i]_CHn_CTRL_in_SOMI field descriptions (continued)

Field	Description
7 Not_Used	Not used in SOMI.
8–10 Not_Used	Not used in this SOMI mode.
11 Not_Used	Not used in SOMI.
12–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 Not_Used	Not used in SOMI.
16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–19 Not_Used	Not used in SOMI.
20 SL	Initial signal level after channel is enabled. After reset and if channel is disabled, the register SOUR is set to the inverse reset value of bit SL (i.e. '1'). If the channel is disabled or the output is disabled, the output ATOM_OUT[x] is set to inverse value of SL. 0 Low signal level. 1 High signal level.
21–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–26 Not_Used	Not used in SOMI.
27 ACB(0)	ACB bit 0. 0 Set output to inverse of SL bit. 1 Set output to SL bit.
28 ARU_EN	ARU Input stream enable. 0 ARU Input stream disabled. 1 ARU Input stream enabled.
29 Not_Used	Not used in SOMI.
30–31 MODE	ATOM channel mode select. (SOMI). 00 ATOM Signal Output Mode Serial (SOMI).

12.3.2 ATOM Signal Output Mode Compare (SOMC)

12.3.2.1 SOMC Overview

In ATOM Signal Output Mode Compare (SOMC), the output action depends on the comparison between the input values located in the **CM0** and/or **CM1** registers and the TBU_TSx time base values, which are provided by the TBU (see TBU Overview). Which of the two (three) time bases is to be compared with one or both of the **CM0** and **CM1** register values is configurable by writing to the ATOM[i]_AGC_ACT_TB[TBU_SEL] bit field.

The behavior of the CCU0 and CCU1 compare units is controlled by:

- bits four through two of the **ATOM[i]_CHn_CTRL[ACB]** 5-bit field (when the ARU connection is disabled by clearing the **ATOM[i]_CHn_CTRL[ARU_EN]** to zero), or
- the **ATOM[i]_CHn_STAT[ACBI]** bit field (when the ARU connection is enabled by setting the **ATOM[i]_CHn_CTRL[ARU_EN]** to one).

When the ARU connection is enabled, the **ATOM[i]_CHn_STAT[ACBI]** 5-bit field is updated with control bits 52 through 48 from the received 52-bit ARU word.

The TRIG_CCU0 and TRIG_CCU1 trigger signals create edges on the ATOM[i]_CHn_OUT output. These edges are dependent on the predefined signal level of the **ATOM[i]_CHn_CTRL[SL]** bit in combination with control bits that are specified by:

- the **ATOM[i]_CHn_CTRL** via the CPU, or
- the **ATOM[i]_CHn_STAT** register via the ARU.

The channel is always disabled (ATOM[i]_AGC_ENDIS_STAT[ENDIS_STATn] = 01) after the specified compare match event has occurred. The SR0 and SR1 shadow registers are used to store the two TBU time stamp values when the match occurs. Afterward, the channel can be enabled again by:

- reading the SR0 and SR1 shadow registers, either by the CPU or the ARU, and
- providing new data to the CM0 and CM1 registers from either by the CPU or the ARU.

For a detailed description, see SOMC Mode under CPU control and SOMC Mode under ARU control.

If a third time base (TBU_TS2) has been implemented in the TBU submodule, either the TBU_TS1 or TBU_TS2 time base can be configured in the ATOM by writing to the **ATOM[i]_CHn_CTRL[TB12_SEL]** bit..

The CCU0 or CCU1 unit can perform a ≥ or ≤ comparison against the TBU_TS1 or TBU_TS2 time base, where the ≥ or ≤ comparison can be configured by writing to the ATOM[i]_CHn_CTRL[CMP_CTRL] bit. The CMP_CTRL control bit has no effect when the CCU0 or CCU1 unit is compared against the TBU_TS0 time base, which always performs a ≥ comparison.

Before the **ATOM[i]_CHn_OUT** output is enabled by setting ATOM[i]_AGC_ENDIS_CTRL[ENDIS_CTRLn] = 0b10, the output's signal level must be initialized. The initial signal level is configured by writing to the **ATOM[i]_CHn_CTRL[SL]** bit. If the output is disabled, the **ATOM[i]_CHn_OUT** output signal level is set to the inverse level of the SL bit.

If a channel is disabled, it's SOUR signal level is set to the level of the SL bit.

When a compare match event is 'true', the **SR0** and **SR1** shadow registers capture the TBU time stamp values. The **SR0** register captures the TBU_TS0 time stamp value and the **SR1** register captures either the TBU_TS1 or TBU_TS2 time stamp value, which is configured by writing to the ATOM[i]_CHn_CTRL[TB12_SEL] bit.

NOTE

When a channel is disabled and new values are written to the CM0 and CM1 compare registers, the channel (when enabled) starts a comparison that is based on these new values.

12.3.2.2 SOMC Mode under CPU control

An ATOM channel can be controlled by either the CPU or the ARU module. When a channel is to be controlled by the CPU, the **ATOM[i]_CHn_CTRL[**ARU_EN] bit must be cleared to zero.

The level of an ATOM[i]_CHn_OUT output is set on a compare match event, depending on the the **ATOM[i]_CHn_CTRL[ABC10, SL]** bit field and bit, respectively. The effect of these bits on the ATOM[i]_CHn_OUT output level is shown in Table 12-4.

Table 12-4. SOMC output levels

SL bit	ACB10, bit 26	ACB10, bit 27	Output level
0	0	0	No level change at output (exception in the table in ATOM CCU0/CCU1 Serve first definition ACB42 = 001 mode ACB42 = 001).
0	0	1	Set output level to 1
0	1	0	Set output level to 0
0	1	1	Toggle output level (exception in the table in ATOM CCU0/CCU1 Serve first definition ACB42 = 001 mode ACB42 = 001).

Table continues on the next page...

Table 12-4. SOMC output levels (continued)

SL bit	ACB10, bit 26	ACB10, bit 27	Output level
1	0	0	No level change at output (exception in the table in ATOM CCU0/CCU1 Serve first definition ACB42 = 001 mode ACB42 = 001).
1	0	1	Set output level to 0
1	1	0	Set output level to 1
1	1	1	Toggle output level (exception in the table in ATOM CCU0/CCU1 Serve first definition ACB42 = 001 mode ACB42 = 001)

The capture/compare strategy of the CCU0 and CCU1 units can be controlled by the **ATOM[i]_CHn_CTRL[ACB42]** bit field. The effect of these bits on CCU0 and CCU1 control is shown in Table 12-5.

Table 12-5. SOMC CCU0 and CCU1 control

ACB42, bit 23	ACB42, bit 24	ACB42, bit 25	CCU0 and CCU1 control
0	0	0	Serve First: Compare in CCU0 using *TBU_TS0* and in parallel in CCU1 using *TBU_TS1* or *TBU_TS2*. Disable other CCUn on compare match. Output level on the compare match of the matching CCUn unit is defined by combination of SL, ACB10(26) and ACB10(27). For more detail, see the table in ATOM CCU0/CCU1 Serve first definition ACB42 = 001.
0	0	1	Serve First: Compare in CCU0 using *TBU_TS0* and in parallel in CCU1 using *TBU_TS1* or *TBU_TS2*. Disable other CCUn on compare match. Output level on the compare match of the matching CCUn unit is defined by combination of SL, ACB10(26) and ACB10(27). For more detail, seethe table in ATOM CCU0/CCU1 Serve first definition ACB42 = 001.
0	1	0	Compare in CCU0 only, use time base *TBU_TS0*. Output level is defined by combination of SL, ACB10(26) and ACB10(27) bits.
0	1	1	Compare in CCU1 only, use time base *TBU_TS1* or *TBU_TS2*. Output level is defined by combination of SL, ACB10(26) and ACB10(27) bits.
1	0	0	Serve Last: Compare in CCU0 and then in CCU1 using *TBU_TS0*. Output level when CCU0 matches is defined by combination of SL, ACB10(26) and ACB10(27). On the CCU1 match the output level is toggled.
1	0	1	Serve Last: Compare in CCU0 and then in CCU1 using *TBU_TS1* or *TBU_TS2*. Output level when CCU0 matches is defined by combination of SL, ACB10(26) and ACB10(27). On the CCU1 match the output level is toggled.

Table continues on the next page...

Table 12-5. SOMC CCU0 and CCU1 control (continued)

ACB42, bit 23	ACB42, bit 24	ACB42, bit 25	CCU0 and CCU1 control
1	1	0	Serve Last: Compare in CCU0 using *TBU_TS0* and then in CCU1 using *TBU_TS1* or *TBU_TS2*. Output level when CCU1 matches is defined by combination of SL, ACB10(26) and ACB10(27).
1	1	1	Not used when ARU disabled.

The effect of ACB42 bit values, 000 and 001, is described in more detail in the tables in ATOM CCUx Serve first definition ACB42 = 000 and ATOM CCU0/CCU1 Serve first definition ACB42 = 001.

12.3.2.2.1 ATOM CCUx Serve first definition ACB42 = 000

Table 12-6 shows the effect of control bits and match conditions on the level of the ATOM[i]_CHn_OUT output.

Table 12-6. ACBn, SL, and CCUn effect on output level

ACB4 bit	ACB3 bit	ACB2 bit	ACB1 bit	ACB0 bit	SL bit	CCU0 match	CCU1 match	New output level
0	0	0	0	0	0	0	1	Hold
0	0	0	0	0	0	1	0	Hold
0	0	0	0	0	0	1	1	Hold
0	0	0	0	1	0	0	1	1
0	0	0	0	1	0	1	0	1
0	0	0	0	1	0	1	1	1
0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	1	1	0
0	0	0	1	1	0	0	1	Toggle
0	0	0	1	1	0	1	0	Toggle
0	0	0	1	1	0	1	1	Toggle
0	0	0	0	0	1	0	1	Hold
0	0	0	0	0	1	1	0	Hold
0	0	0	0	0	1	1	1	Hold
0	0	0	0	1	1	0	1	0
0	0	0	0	1	1	1	0	0
0	0	0	0	1	1	1	1	0
0	0	0	1	0	1	0	1	1
0	0	0	1	0	1	1	0	1

Table continues on the next page...

Table 12-6. ACBn, SL, and CCUn effect on output level (continued)

ACB4 bit	ACB3 bit	ACB2 bit	ACB1 bit	ACB0 bit	SL bit	CCU0 match	CCU1 match	New output level
0	0	0	1	0	1	1	1	1
0	0	0	1	1	1	0	1	Toggle
0	0	0	1	1	1	1	0	Toggle
0	0	0	1	1	1	1	1	Toggle

12.3.2.2.2 ATOM CCU0/CCU1 Serve first definition ACB42 = 001

Table 12-7 shows the effect of control bits and match conditions on the level of the ATOM[i]_CHn_OUT output.

Table 12-7. ACBn, SL, and CCUn effect on output level

ACB4 bit	ACB3 bit	ACB2 bit	ACB1 bit	ACB0 bit	SL bit	CCU0 match	CCU1 match	New output level
0	0	0	0	0	0	0	1	Hold
0	0	0	0	0	0	1	0	Toggle
0	0	0	0	0	0	1	1	Hold
0	0	0	0	1	0	0	1	0
0	0	0	0	1	0	1	0	1
0	0	0	0	1	0	1	1	0
0	0	0	0	1	0	0	1	1
0	0	0	0	1	0	1	0	0
0	0	0	0	1	0	1	1	1
0	0	0	1	1	0	0	1	Toggle
0	0	0	1	1	0	1	0	Hold
0	0	0	1	1	0	1	1	Toggle
0	0	0	0	0	1	0	1	Hold
0	0	0	0	0	1	1	0	Toggle
0	0	0	0	0	1	1	1	Hold
0	0	0	0	1	1	0	1	1
0	0	0	0	1	1	1	0	0
0	0	0	0	1	1	1	1	1
0	0	0	1	0	1	0	1	0
0	0	0	1	0	1	1	0	1
0	0	0	1	0	1	1	1	0
0	0	0	1	1	1	0	1	Toggle

Table continues on the next page...

Table 12-7. ACBn, SL, and CCUn effect on output level (continued)

ACB4 bit	ACB3 bit	ACB2 bit	ACB1 bit	ACB0 bit	SL bit	CCU0 match	CCU1 match	New output level
0	0	0	1	1	1	1	0	Hold
0	0	0	1	1	1	1	1	Toggle

If an ATOM channel is enabled, the **CM0** and/or **CM1** registers and the **ATOM[i]_CHn_CTRL[ACB42]** bit field can be updated by the CPU as long as:

- the first match event occurs during serve last compare strategy, or
- the overall match event occurs during other compare strategies.

After a compare match event that causes an update of the **SR0/SR1** shadow registers, and before reading the **SR0** and/or **SR1** register via the ARU, an update of the **CM0** and/or **CM1** registers is possible although it has no effect.

To set up a new compare action, first the **SR0** and/or **SR1** registers that contain captured values must be read, and then new compare values must be written to the **CM0** and/or **CM1** registers.

The **CMn** register(s) that have to be updated depends on the compare strategy that is defined in the **ATOM[i]_CHn_CTRL[ACB42]** bit field. Since the channel immediately starts a comparison after the **CMn** register(s) are written, the compare strategy has to be updated before the **CMn** register(s) are written.

For the serve last compare strategies, if the registers CM0 and CM1 are updated, it is possible that one or both compare values are old. To avoid a deadlock, the ATOM channel will wait until both compare values are written before it starts new comparisons.

At any time, the CPU can read the **ATOM[i]_CHn_STAT[DV]** bit to determine if an ATOM channel's CM0 and CM1 compare registers contain valid data. DV = 1 indicates that the CM0 and CM1 compare registers contain valid data, and that the compare event has started.

NOTE

For serve last compare strategies, if the DV bit is currently not set, writing to CM0 or CM1 immediately sets the DV bit although the compare is only started if both values are written.

When in SOMC mode and 'serve last' CCUn control mode, and if the CCU0 compare match event has occurred, the update of the **CM0/CM1** registers by the CPU is blocked until the CCU1 compare match event has occurred.

When in the serve last mode and when ACB42 = "100" or "101", very small pulses can be generated on the ATOM[i]_CHn_OUT output when the **CM0** and **CM1** compare registers are updated with time stamp values that are close in magnitude. When the first match event is 'true' in CCU0, the ATOM[i]_CHn_OUT output will be set or reset dependent on the SL bit and the specified ACB42(26) and ACB42(27) bits in the **ATOM[i]_CHn_CTRL[SL, ACB42]** bit and bit field, respectively. When the second match event is 'true' in CCU1, the ATOM[i]_CHn_OUT output will toggle.

NOTE

The time stamp has to be loaded into the CM1 register, since CCU0 will enable CCU1 when the CCU0 match event becomes 'true'. The order of the comparison depends on the defined ≥ or ≤ comparison in the CCUn units.

In addition to storing the captured time stamps in the SR0 and SR1 shadow registers, the ATOM channel loads the result of a 'true' compare match event into the ACBO(4) and ACBO(3) bits of the **ATOM[i]_CHn_STAT[ACBO]** bit field. The meaning of these bits is shown in Table 12-8.

Table 12-8. Compare match status

ACBO(4)	ACBO(3)	Indication
0	1	CCU0 compare match occurred
1	0	CCU1 compare match occurred

NOTE

In the case of the 'serve last' compare strategy and when the ATOM[i]_CHn_CTRL[SLA] bit = 0, the ACBO(4) bit is always set to one and the ACBO(3) bit is always reset to zero after the compare match event occurred.

When the ATOM[i]_CHn_STAT[DV] bit is set to one, the ACBO bit field is reset to 0b00000.

Depending on which CCUn had the occurrence of a 'true' match event , the *CCU0TCn_IRQ* or *CCU1TCn_IRQ* interrupt is asserted.

The behavior of an ATOM channel in SOMC mode and under CPU control is visualized in SOMC state diagram for channel under CPU control.

12.3.2.2.3 SOMC state diagram for channel under CPU control

Figure 12-5 shows a visualization of how a channel operates when in SOMC mode and under CPU control.

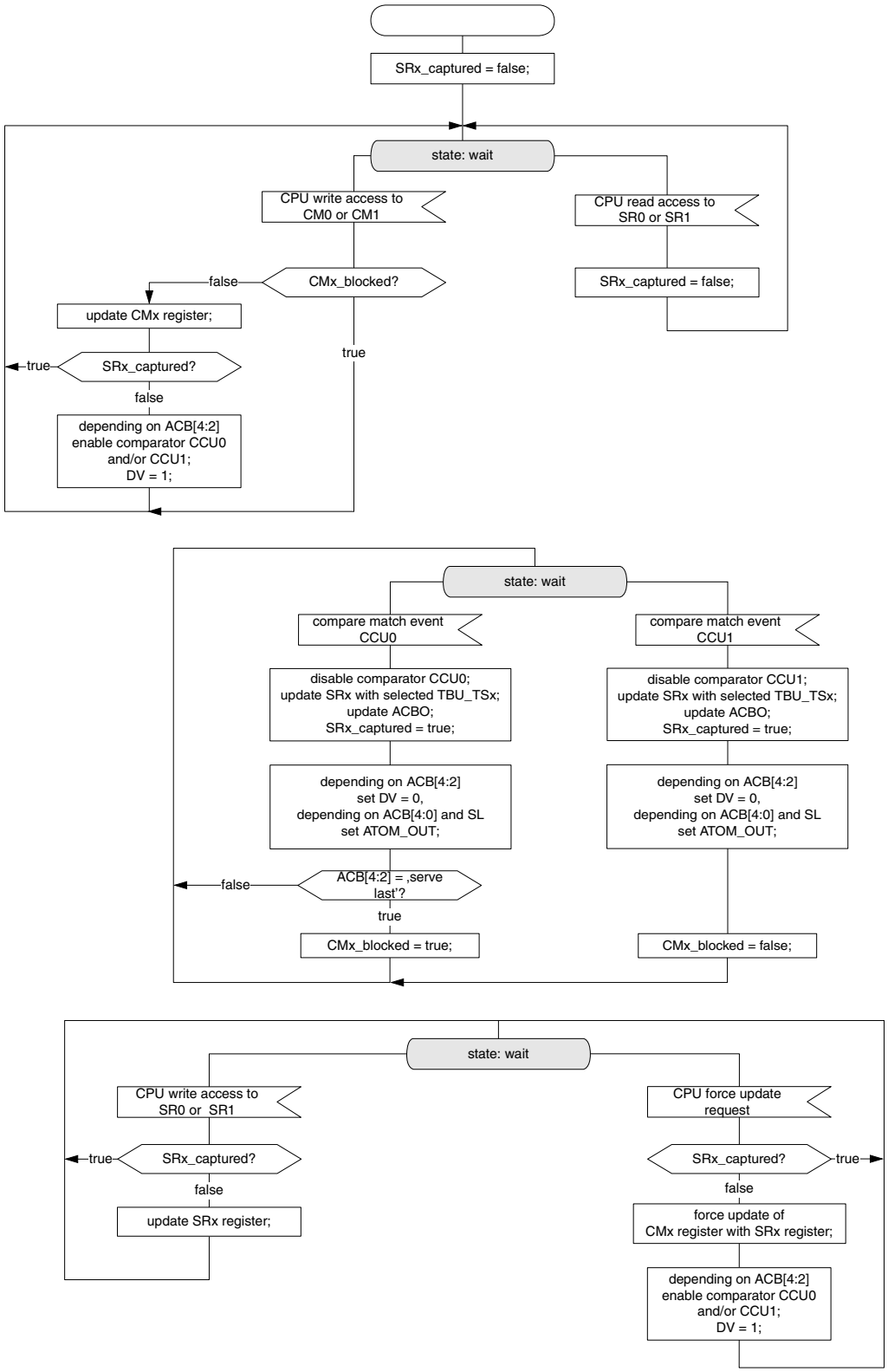


Figure 12-5. SOMC state diagram for channel under CPU control

12.3.2.3 SOMC Mode under ARU control

An ATOM[i] channel is enabled for ARU control by setting the **ATOM[i]_CHn_CTRL[ARU_EN]** bit to one.

The content for the **CM0** and **CM1** compare registers as well as the update of the compare strategy can be loaded via the 53 bit ARU word, where:

- ARU word bits 23 through zero are loaded into the **CM0** register,
- ARU word bits 47 through 24 are loaded into the **CM1** register, and
- ARU control bits 52 through 48 are loaded into the 5-bit **ATOM[i]_CHn_STAT[ACBI]** bit field.

ARU control bits 49 and 48 are loaded into ACBI(1) and ACBI(0) where they specify a channel's compare strategy. The ACBI(1) and ACBI(0) bits along with the ATOM[i]_CHn_CTR[SL] bit specify the ATOM[i]_CHn_OUT output behavior when a 'true' compare match event occurs, as shown in Table 12-9.

Table 12-9. SL, ACBI(1), and ACBI(2) effect on output behavior

SL	ACBI(1)	ACBI(0)	Output behavior
0	0	0	No signal level change at output (exception in ATOM CCUx Serve first definition ACB42 = 000 and ATOM CCU0/CCU1 Serve first definition ACB42 = 001 mode ACB42 = 001).
0	0	1	Set output signal level to 1
0	1	0	Set output signal level to 0
0	1	1	Toggle output signal level (exception in ATOM CCUx Serve first definition ACB42 = 000 and ATOM CCU0/CCU1 Serve first definition ACB42 = 001 mode ACB42=001)
1	0	0	No signal level change at output (exception in ATOM CCUx Serve first definition ACB42 = 000 and ATOM CCU0/CCU1 Serve first definition ACB42 = 001 mode ACB42=001).
1	0	1	Set output signal level to 0
1	1	0	Set output signal level to 1
1	1	1	Toggle output signal level (exception in ATOM CCUx Serve first definition ACB42 = 000 and ATOM CCU0/CCU1 Serve first definition ACB42 = 001 mode ACB42=001)

ARU control bits 52 through 50 are loaded into ACBI42, where they specify the operation of the CCU0 and CCU1 capture/compare units as shown in Table 12-10.

Table 12-10. ACBI(4:2) effect on CCUn operation

ACBI(4)	ACBI(3)	ACBI(2)	CCUn operation
0	0	0	Serve First: Compare in CCU0 using *TBU_TS0* and in parallel in CCU1 using *TBU_TS1* or *TBU_TS2*. Disable other CCUx on compare

Table continues on the next page...

Table 12-10. ACBI(4:2) effect on CCUn operation (continued)

ACBI(4)	ACBI(3)	ACBI(2)	CCUn operation
			match. Output signal level on the compare match of the matching CCUx unit is defined by combination of SL, ACBI(1) and ACBI(0). Details see ATOM CCU0/CCU1 Serve first definition ACB42 = 001.
0	0	1	Serve First: Compare in CCU0 using *TBU_TS0* and in parallel in CCU1 using *TBU_TS1* or *TBU_TS2*. Disable other CCUx on compare match. Output signal level on the compare match of the matching CCUx unit is defined by combination of SL, ACBI(1) and ACBI(0). Details see ATOM CCUx Serve first definition ACB42 = 000.
0	1	0	Compare in CCU0 only, use time base *TBU_TS0*. Output signal level is defined by combination of SL, ACBI(1) and ACBI(0) bits.
0	1	1	Compare in CCU1 only, use time base *TBU_TS1* or *TBU_TS2*. Output signal level is defined by combination of SL, ACBI(1) and ACBI(0) bits.
1	0	0	Serve Last: Compare in CCU0 and then in CCU1 using *TBU_TS0*. Output signal level when CCU0 matches is defined by combination of SL, ACBI(1) and ACBI(0). On the CCU1 match the output level is toggled.
1	0	1	Serve Last: Compare in CCU0 and then in CCU1 using *TBU_TS1* or *TBU_TS2*. Output signal level when CCU0 matches is defined by combination of SL, ACBI(1) and ACBI(0). On the CCU1 match the output level is toggled.
1	1	0	Serve Last: Compare in CCU0 using *TBU_TS0* and then in CCU1 using *TBU_TS1* or *TBU_TS2*. Output signal level when CCU1 matches is defined by combination of SL, ACBI(1) and ACBI(0).
1	1	1	Change ARU read address to ATOM_RDADDR1 DV flag is not set. Neither ACBI(1) nor ACBI(0) is evaluated.

NOTE

An ACBI(4), ACBI(3), ACBI(2) value of '111' forces a channel to request new compare values from a different destination read address, which is configured by writing to the **ATOM[i]_CHn_RDADDR[ATOM_RDADDR1]** bit field. After read data from the new destination read address is successfully received and a compare event occurs, the channel

switches back to the read address specified by the **ATOM[i]_CHn_RDADDR[ATOM_RDADDR0]** bit field to receive the next read data.

After the specified compare match event occurs, the captured time stamps are stored in **SR0** and **SR1** registers and the 'true' compare result is stored in bits four and three of the **ATOM[i]_CHn_STAT[ACBO]** bit field. The meaning of the ACBO(4) and ACBO(3) bits is shown in Table 12-11.

Table 12-11. ACBO(4:3) effect on value returned to ARU

ACBO(4)	ACBO(3)	Return value to ARU
0	1	CCU0 compare match occurred
1	0	CCU1 compare match occurred

NOTE

In the case of the 'serve last' compare strategy, when the ATOM[i]_CHn_CTRL[SLA] bit is not set to one, the ACBO(4) bit is always set to one and the ACBO(3) bit is always reset to zero after a 'true' compare match event occurred.

When the **ATOM[i]_CHn_STAT[DV]** bit bit is set to one, the ACBO bit field is reset to 0b00000.

Depending on which CCUn had a 'true' compare match event occur, the *CCU0TCn_IRQ* or *CCU1TCn_IRQ* interrupt is asserted.

When CCU0 and CCU1 are both used for comparison, it is possible to generate very small pulses on the ATOM[i]_CHn_OUT output by loading the **CM0** and **CM1** registers with values that are close in magnitude. For the first 'true' match event, the ATOM[i]_CHn_OUT output will be set or reset dependent on the ATOM[i]_CHn_CTRL[SL] bit and the ACBI(0) and ACBI(1) bits in the **ATOM[i]_CH[x]_STAT[ACBI]** bit field. For the second 'true' match event, the ATOM[i]_CHn_OUT output will toggle.

NOTE

The bigger (smaller) value must be loaded into the CM1 register, since CCU0 will enable CCU1 once CMU0 reaches a 'true' compare match event. The order of the comparisons depend on the defined ≥ or ≤ comparison thresholds of the CCUn units.

For the 'serve last' compare strategy, the CCU0 and CCU1 compare match may occur sequently. During different phases of compare match, the CPU access rights to register CM0 and CM1 as well as to WR_REQ bit are different. These access rights are depicted in the following figure.

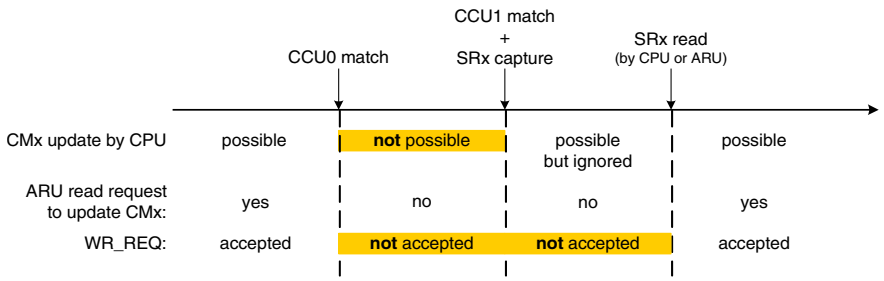


Figure 12-6. CPU access rights for the 'serve last' compare strategy

12.3.2.3.1 ARU Non-Blocking mode

When the CM0 and CM1 compare registers are updated via ARU, the update behavior of a channel is configurable by writing to the **ATOM[i]_CHn_CTRL[ABM]** bit. When the ABM bit is reset to zero, the ATOM channel is in ARU non-blocking mode.

When in ARU non-blocking mode, data received via ARU is continuously transferred to the **CM0** and **CM1** registers and to the **ATOM[i]_CHn_STAT[ACBI]** bit field as long as no specified 'true' compare match events occur.

After a 'true' compare match event causes an update of the **SR0/SR1** shadow registers, and before reading the **SR0/SR1** registers by the CPU or via the ARU, the update of the **CM0/CM1** registers by the CPU or via the ARU is possible although it has no effect.

To set up a new compare action, the **SR0/SR1** registers containing captured values must be read, and then new compare values must be written into the **CM0/CM1** registers. This can be accomplished by the CPU or via the ARU.

When the CPU accesses the registers, only one of the shadow registers must be read. Dependent on the compare strategy, the CPU must write to one or both of the CM0/CM1 compare registers.

When in SOMC mode and CCUx control mode 'serve last', and if the CCU0 'true' compare match event has occurred, the update of the **CM0/CM1** registers by the CPU or via the ARU is blocked until the CCU1 'true' compare match event occurs.

At any time, the CPU can read the **ATOM[i]_CHn_STAT[DV]** bit to determine if the ATOM channel has received valid data from the ARU or not. When DV = 1, received data is valid and CCUn starts a compare match event.

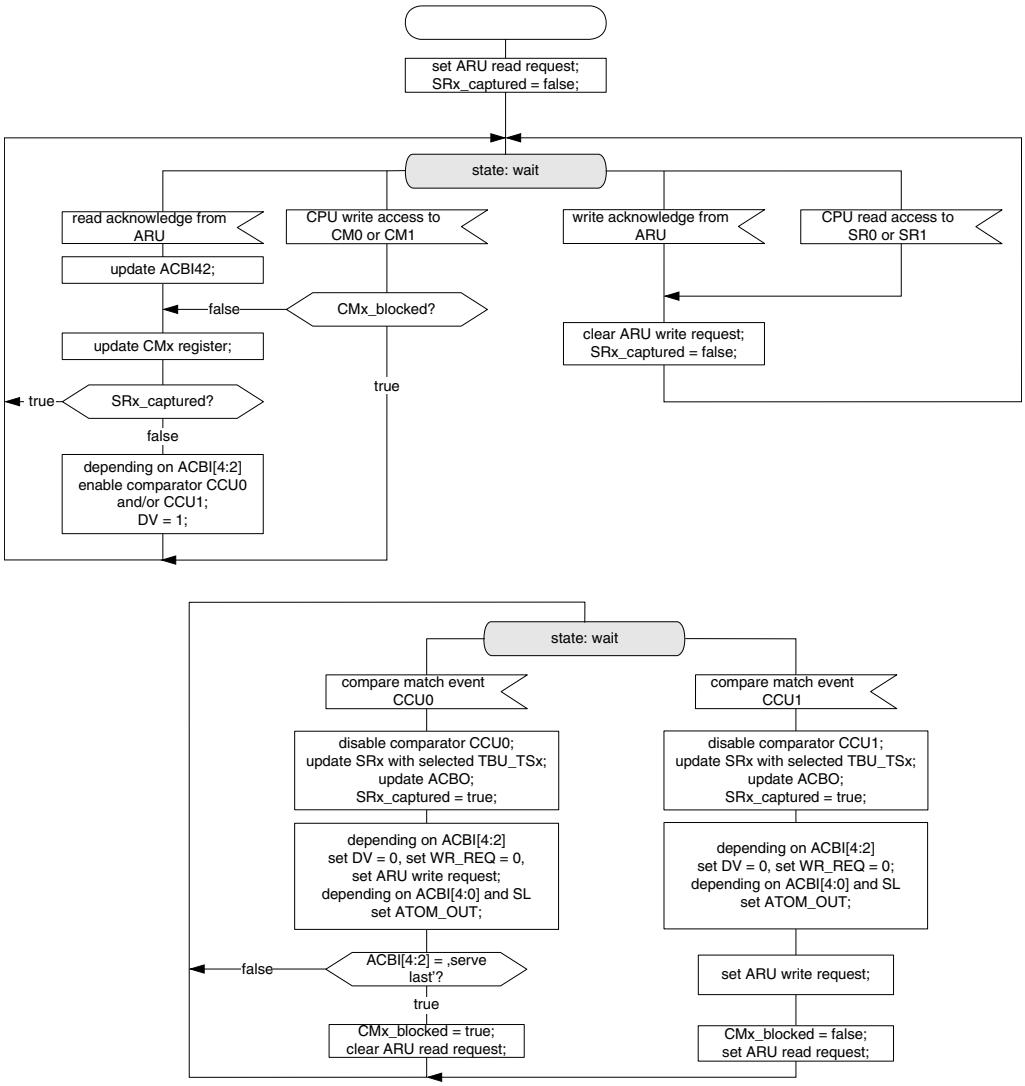


Figure 12-7. State diagram for SOMC mode with ARU enabled and ARU blocking mode disabled

12.3.2.3.2 ARU Blocking mode

When the CM0 and CM1 compare registers are updated via the ARU, an ATOM channel can be configured to receive ARU data in a blocking manner by setting the **ATOM[i]_CHn_CTRL[ABM]** bit to one.

When the ABM and ARU_EN bits are both set to one, the CM0/CM1 compare register values can be loaded by the CPU or via the ARU. When the **CM0** and/or **CM1** compare registers are/is updated, the ATOM channel waits for a 'true' compare match event to occur. During that time, no additional data is requested from the ARU.

When the specified 'true' compare match event occurs, the **SR0** and **SR1** shadow registers are updated along with the **ATOM[i]_CHn_STAT[ACBO]** bit field. The data in the shadow registers is marked as valid for the ARU and the **ATOM[i]_CHn_STAT[DV]** bit is reset to zero.

When the **SR0** and **SR1** registers that are holding the captured TBU_TSn input time stamp values are read by the CPU or via the ARU, the next write access by the CPU or via the ARU to update of the **CM0** or **CM1** registers starts a new compare match event.

At least one of the SR0 or SR1 registers must be read before new data is requested from the ARU.

At any time, the COU can read the **ATOM[i]_CHn_STAT[DV]** bit to determine if the ATOM channel has received valid data from the ARU. DV = 1 indicates that data is valid and a new compare match event is started.

The behavior of an ATOM channel in SOMC mode with the ARU enabled and ARU blocking mode enabled is shown in Figure 12-8.

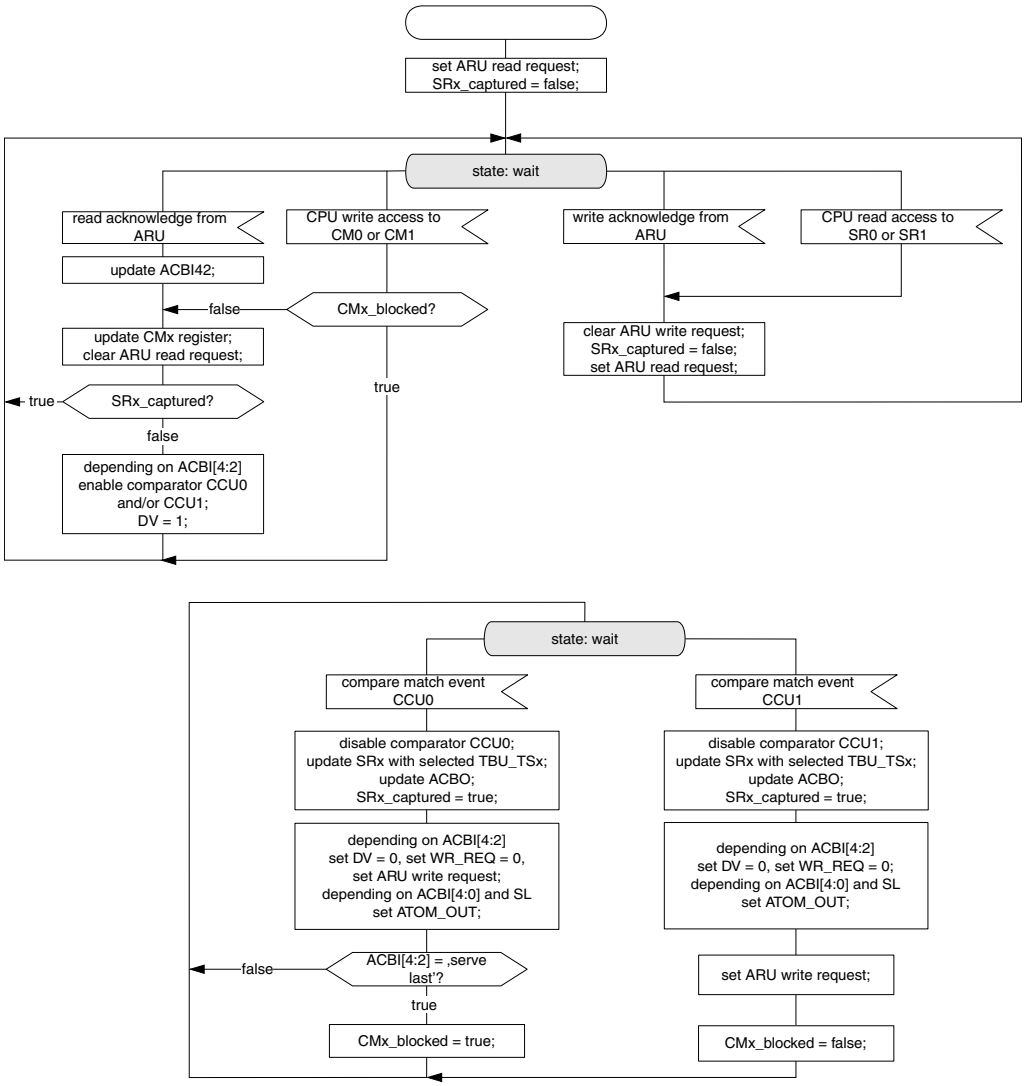


Figure 12-8. State diagram for SOMC mode with ARU and ABM enabled

12.3.2.2.3.3 ATOM SOMC late update mechanism

Although the ATOM channel may be controlled by data received via the ARU or the CPU, at any time, is able to request a late update of the CM0 and CM1 compare registers. This late update can be initiated by setting the **ATOM[i]_CHn_CTRL[WR_REQ]** bit to one. When the WR_REQ bit = 1, the ATOM channel will not request further data from the ARU (if ARU access is enabled by setting the **ATOM[i]_CHn_CTRL[ARU_EN]** bit to one). The channel will, in any case, continue to compare against the values stored inside the CM0 and CM1 compare registers (if the **ATOM[i]_CHn_STAT[DV]** bit = 1). The CPU can enable a late update of the CM0 and CM1 compare registers by setting the ATOM[i]_AGC_FUPD_CTRL[FUPD_CTRLn] bit to one. The late update will be effective if it writes data to the SR0 and SR1 shadow registers any time before the start of

the compare match event. If the late update occurs after the start of the compare match event, the compare will be performed on the data that was in the CM0 and CM1 registers prior to the late update.

When WR_REQ bit = 1 and a compare match event starts, any further access to the SR0 and SR1 shadow registers is blocked and the late update itself is blocked. Additionally, the **ATOM[i]_CHn_STAT[WRF]** bit is set to one. The CPU can read the WRF bit to determine if the late update failed.

If a compare match event has ended, the WR_REQ bit cannot be set to one until the channel is unlocked for a new compare match event by reading the SR0 and SR1 shadow registers. However, the CPU can set the WRF bit to one during that time period.

During the time period:

- when the WR_REQ bit is correctly set,
- a shadow register is correctly written, and
- before a force update is requested by the AGC,

a compare match event that occurs uses the old CM0 and CM1 compare values, and the WRF bit will be set to one.

The WRF bit will be set to one when the CPU tries to write to a blocked shadow register.

The WR_REQ bit and the DV bit will be reset to zero on a 'true' compare match event.

A blocked force update will be unblocked after a read access to the **SR0** or **SR1** register by the CPU or via the ARU.

The "SOMC late update from the CPU" state diagram is shown in Figure 12-9.

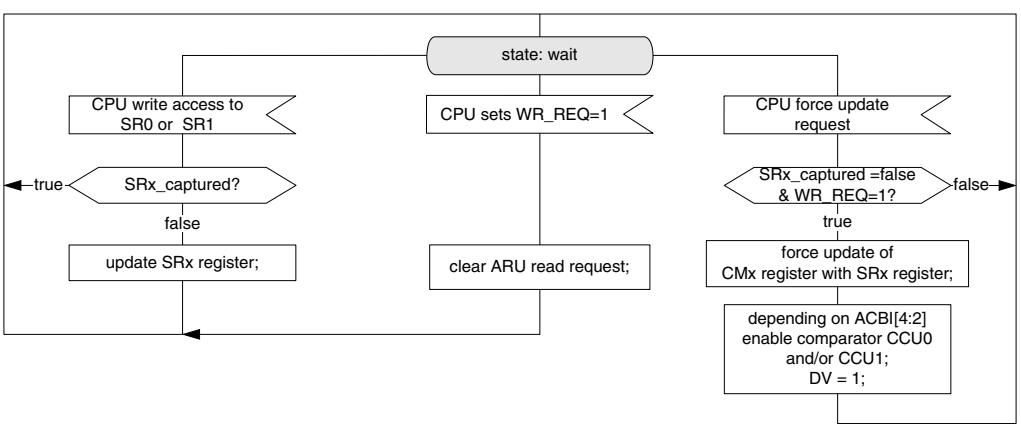


Figure 12-9. SOMC state diagram for late update requests by the CPU

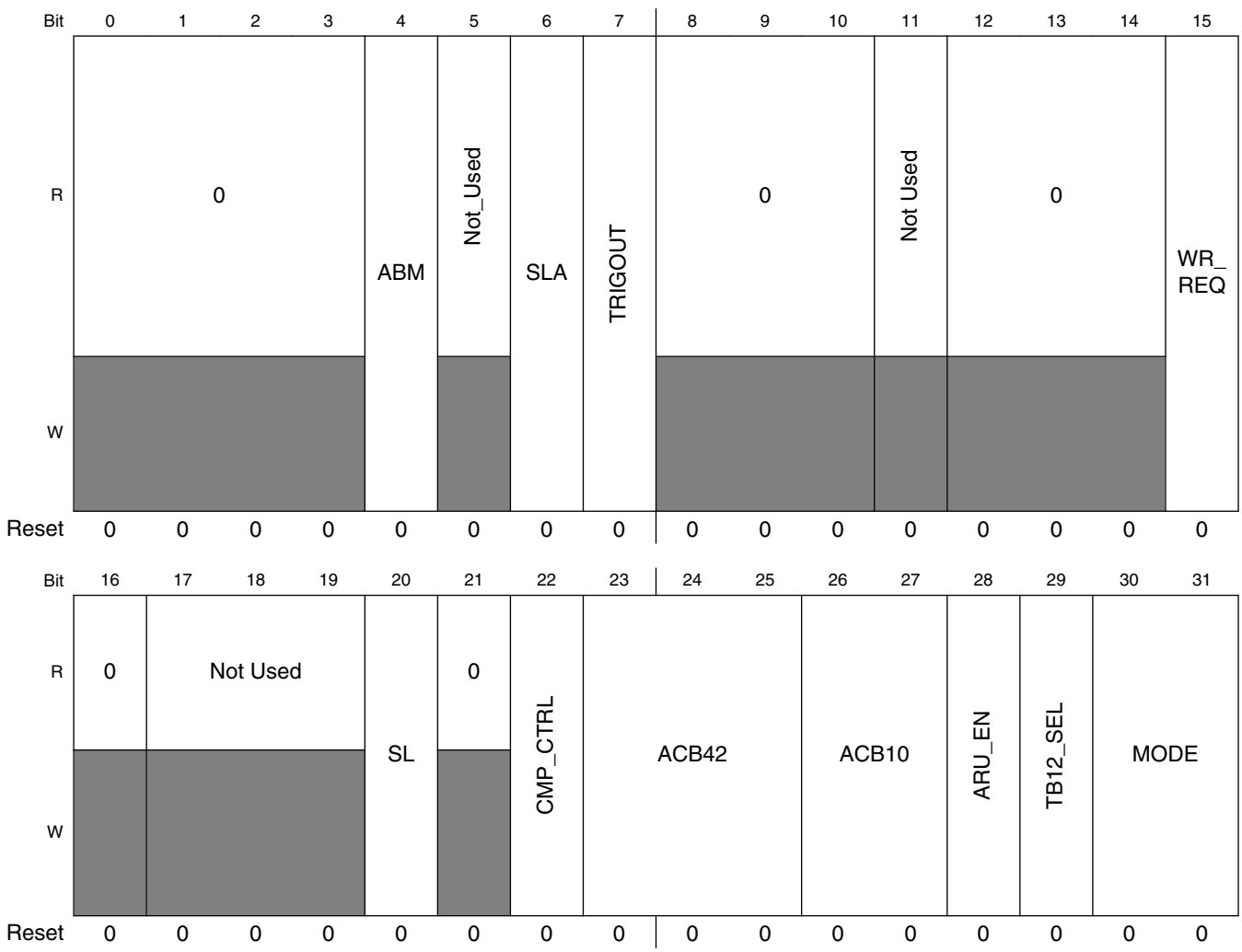
12.3.2.3.3.1 Memory Map and Registers

memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
D004	ATOM[i]_CHn_CTRL_in_SOMC	32	R/W	See section	12.3.2.3.3.1. 1/331

12.3.2.3.3.1.1 ATOM[i]_CHn_CTRL_in_SOMC

Address: 0h base + D004h offset = D004h



ATOM[i]_CHn_CTRL_in_SOMC field descriptions

Field	Description
0–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ATOM[i]_CHn_CTRL_in_SOMC field descriptions (continued)

Field	Description
4 ABM	ARU blocking mode. 0 ARU blocking mode disabled: ATOM reads continuously from ARU and updates CM0, CM1 independent of pending compare match event. 1 ARU blocking mode enabled: after updating CM0,CM1 via ARU, no new data is read from ARU until compare match event occurred and SR0 and/or SR1 are read.
5 Not_Used	Not used in SOMC.
6 SLA	Serve last ARU communication strategy. setting of this bit has only effect, when ACBI(4:2) is configured for serve last compare strategy ("100", "101", or "110"). When this bit is not set, the captured time stamps in the shadow registers SRx are only provided after the CCU1 match occurred. The ACBO(4:3) bits always return "10" in that case. By setting this bit, the ATOM channel also provides the captured time stamps after the CCU0 match event to the ARU. The ACBO(4:3) bits are set to “01” in that case. After the CCU1 match event, the time stamps are captured again in the SRx registers and provided to the ARU. The ACBO(4:3) bits are set to “10”. When the data in the shadow registers after the CCU0 match was not consumed by an ARU destination and the CCU1 match occurs, the data in the shadow registers is overwritten by the new captured time stamps. The ATOM channel does not request new data from the ARU when the CCU0 match values are read from an ARU destination. 0 Capture SRn time stamps after CCU0 match event not provided to ARU. 1 Capture SRx time stamps after CCU0 match event provided to ARU.
7 TRIGOUT	Trigger output selection (output signal TRIG_CHn) of module ATOM_CHn. 0 TRIG_[x] is TRIG_[x-1]. 1 TRIG_[x] is TRIG_CCU0.
8–10 Reserved	Not used in this mode. This read-only field is reserved and always has the value 0.
11 Not Used	Not used in SOMC.
12–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 WR_REQ	CPU write request bit. The CPU can disable subsequent ARU read requests by the channel and can update the shadow registers with new compare values, while the compare units operate on old compare values received by former ARU accesses, if occurred. On a compare match event, the WR_REQ bit will be reset by hardware. At the point of the force update only the shadow registers SR0 and SR1 are transferred into the CM0, CM1 registers. The output action is still defined by the ACBI bit field described by the ARU together with the old compare values for CM0/CM1. 0 No late update requested by the CPU. 1 Late update requested by the CPU.
16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ATOM[i]_CHn_CTRL_in_SOMC field descriptions (continued)

Field	Description
17–19 Not Used	Not used in SOMC.
20 SL	Initial signal level after channel enable. If the output is disabled, the output ATOM_OUTn is set to inverse value of SL. If the channel and output are disabled in MODE=01 (SOMC mode), the output register of SOU unit is set to value of SL. If the output is enabled afterwards, the output ATOM_OUT[x] is equal to the value of SL. 0 Low signal level. 1 High signal level.
21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 CMP_CTRL	CCUx compare strategy select. The compare unit CCU0 or CCU1 that compares against TBU_TS0 (depending on CCUx control mode defined by ACBI(4:2) or ACB42) always performs a greater/equal comparison, independant on CMP_CTRL bit. 0 Greater/equal compare against TBU_TS1/2 time base values greater than or equal to CM0/1). 1 Less/equal compare against TBU time base values (TBU_TS1/2 less than or equal to CM0/1).
23–25 ACB42	ATOM control bits ACB(4), ACB(3), ACB(2). These bits are only applicable if ARU_EN = '0'. 000 Compare in CCU0 and CCU1 in parallel, disable the CCUx on a compare match on either compare units. Use TBU_TS0 in CCU0 and TBU_TS1 or TBU_TS2 in CCU1. 001 Compare in CCU0 and CCU1 in parallel, disable the CCUx on a compare match on either compare units. Use TBU_TS0 in CCU0 and TBU_TS1 or TBU_TS2 in CCU1. 010 Compare in CCU0 only against TBU_TS0. 011 Compare in CCU1 only against TBU_TS1 or TBU_TS2. 100 Compare first in CCU0 and then in CCU1. Use TBU_TS0. 101 Compare first in CCU0 and then in CCU1. Use TBU_TS1 or TBU_TS2. 110 Compare first in CCU0 and then in CCU1. Use TBU_TS0 in CCU0 and TBU_TS1 or TBU_TS2 in CCU1. 111 Reserved.
26–27 ACB10	Signal level control bits. These bits are only applicable if ARU_EN = '0'. 00 No signal level change at output (see exceptions in ATOM CCUx Serve first definition ACB42 = 000 and ATOM CCU0/CCU1 Serve first definition ACB42 = 001). 01 Set output signal level to 1 when SL bit = 0 else output signal level to 0. 10 Set output signal level to 0 when SL bit = 0 else output signal level to 1. 11 Toggle output signal level (see exceptions in see exceptions in ATOM CCUx Serve first definition ACB42 = 000 and ATOM CCU0/CCU1 Serve first definition ACB42 = 001).
28 ARU_EN	ARU Input stream enable. 0 ARU Input stream disabled. 1 ARU Input stream enabled.
29 TB12_SEL	Select time base value TBU_TS1 or TBU_TS2.

Table continues on the next page...

ATOM[i]_CHn_CTRL_in_SOMC field descriptions (continued)

Field	Description
	0 TBU_TS1 selected for comparison. 1 TBU_TS2 selected for comparison.
30–31 MODE	ATOM channel mode select. 01 ATOM Signal Output Mode Serial (SOMC).

12.3.3 ATOM Signal Output Mode PWM (SOMP)

When in ATOM Signal Output Mode PWM (SOMP), a channel is able to generate complex PWM signals with different duty cycles and periods. Duty cycles and periods can be changed synchronously and asynchronously. Synchronous change of the duty cycle and/or period means that the duty cycle or period duration changes after the end of the preceding period. An asynchronous change of period and/or duty cycle means that the duration of the period changes during the actual running PWM period.

The signal level of the pulse that is generated inside the period is configured by writing to the **ATOM[i]_CHn_CTRL[SL]** bit. The initial ATOM[i]_CHn_OUTn signal output level for the channel is the inverse of the level of the SL bit (see PWM Output behavior with respect to the SL bit).

Depending on the ATOM[i]_CH[x]_CTRL[RST_CCU0] configuration bit, the CN0 counter register can be reset either when:

- the counter value is equal to the compare value CM0, or
- signaled by ATOM[i] trigger signal TRIG_[x-1] of the preceding channel.

In this case, if UPEN_CTRL[x] = 1, the CM0, CM1 and CLK_SRC working registers are also updated.

NOTE

As an exception, the TRIG_[0] input of instance ATOM0 is triggered by its own last channel cCATO via signal TRIG_[cCATO]. Please refer to device specific Appendix B for value cCATO of ATOM0.

12.3.3.1 PWM Output behavior with respect to the SL bit

Figure 12-11 shows output behavior with respect to the ATOM[i]_CHn_CTRL[SL] bit.

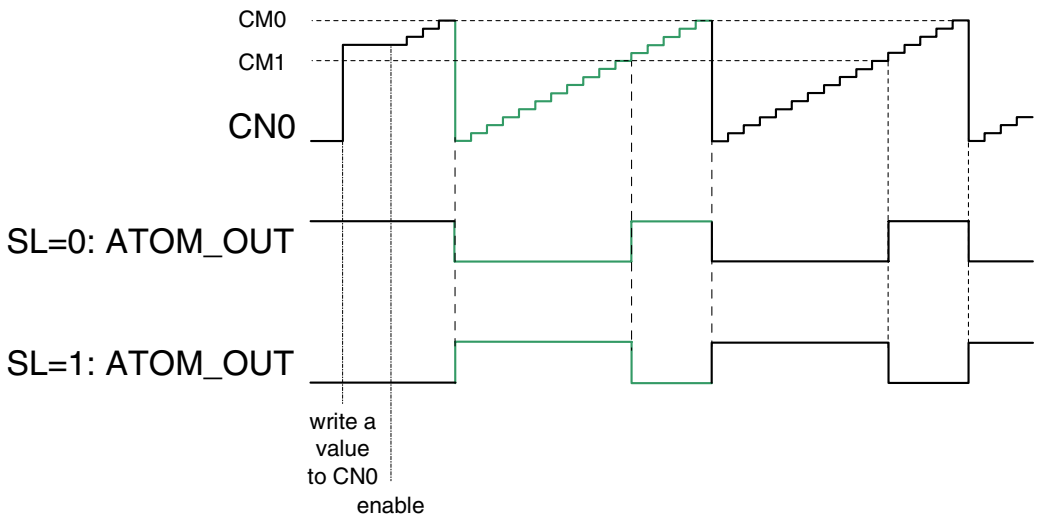


Figure 12-11. PWM output behavior depending on SL bit

When an asynchronous update occurs, it is guaranteed that no spike occurs at the ATOM[i]_CHn_OUT output due to a (too) late update of the CN0, CM0, and CM2 operation registers. The behavior of the ATOM[i]_CHn_OUT output, due to the different possibilities of an asynchronous update during a PWM period, is shown in PWM Output behavior in case of an asynchronous update of the duty cycle.

12.3.3.2 PWM Output behavior in case of an asynchronous update of the duty cycle

Figure 12-12 shows how the duty cycle of a PWM output is effected by an asynchronous update.

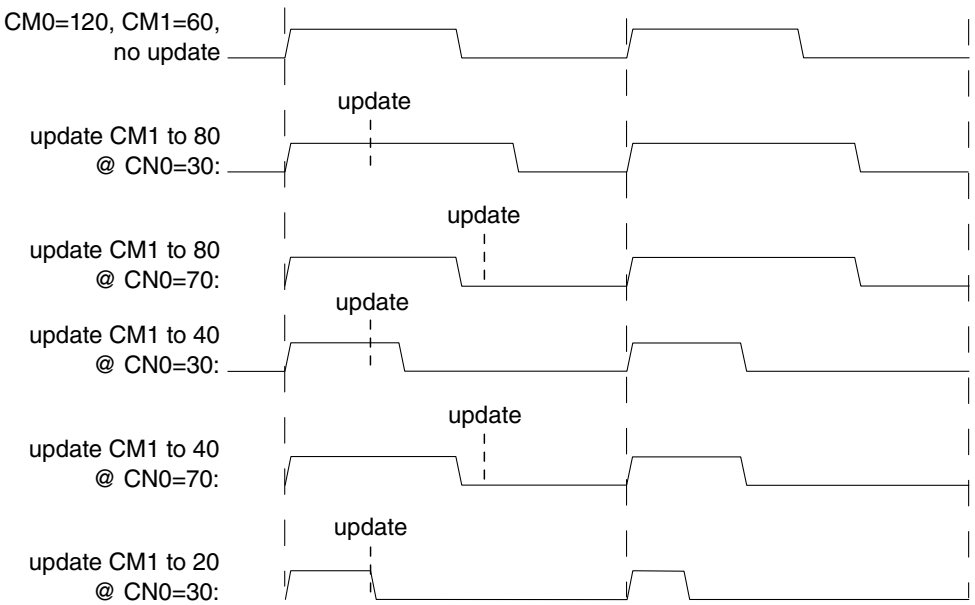


Figure 12-12. PWM output when duty cycle gets an asynchronous update

The duration of a pulse's high or low time and period is measured with the CN0 counter register (that is in CCU0 subunit). One of the eight CMU_CLKn clocks signals (that are received from the CMU submodule) is configured (by writing to the to the ATOM[i]_CHn_CTRL[CLK_SRC] bit field) to trigger the CN0 counter. The **CM0** register holds a value that specifies the duration of the period (in number of CMU_CLKn ticks) and the **CM1** register holds a value that specifies the duration of the duty cycle (in number of CMU_CLKn ticks).

If counter register CN0 is reset by its own CCU0 unit (i.e. the compare match of CN0>=CM0 configured by RST_CCU0=0), following statements are valid:

- the configuration of CM1=0 represents 0% duty cycle at the output
- the configuration of CM1 >= CM0 represents 100% duty cycle
- if both registers are configured to 0 (CM0=CM1=0), the output is 0% duty cycle - if CM0=0, 0% duty cycle is generated independent of CM1.

If the counter register CN0 is reset by the trigger signal coming from another channel or the assigned TIM module (configured by RST_CCU0=1), following statements are valid:

- CM0 defines the edge to SL value, CM1 defines the edge to !SL value
- if CM0=CM1, the output is 100% SL (CM0 has higher priority)
- if CM0=0, the output stays at its last value (CN0 stops counting)

In case the counter value CN0 reaches the compare value in register CM0 or the channel receives an external update trigger via the FUPD(x) signal, a synchronous update is performed. A synchronous update means that the registers CM0 and CM1 are updated with the content of the shadow registers SR0 and SR1 and the CLK_SRC register is updated with the value of the CLK_SRC_SR register.

The clock source for the CN0 counter can be synchronously changed at the end of a period.

If ARU accesses are disabled (ATOM[i]_CHn_CTRL[ARU_EN] bit = 0), the change of the clock source is accomplished by using the **ATOM[i]_CHn_CTRL[CLK_SRC_SR]** bit field as a shadow register that specifies the new CMU clock source that is used to update the ATOM[i]_CHn_CTRL[CLK_SRC] bit field.

If ARU accesses are enabled (ATOM[i]_CHn_CTRL[ARU_EN] bit = 1), the ACBI(4), ACBI(3) and ACBI(2) bits of the received ARU control bit field are stored in the **ATOM_[i]_CHn_STAT[ACBI]** bit field and are used as a shadow register that specifies the new CMU clock source that is used to update the ATOM[i]_CHn_CTRL[CLK_SRC] bit field.

For a synchronous update, the generation of a complex PWM output waveform is possible (without CPU interaction) by reloading the **SR0** and **SR1** shadow registers and the ACBI bit field from the ARU (see Figure 12-3) while the channel continues to operate using the **CM0** and **CM1** registers.

When the current PWM period ends:

- the SR0 and SR1 shadow registers are loaded into the CM0 and CM1 compare registers,
- the CN0 counter register is reset to zero,
- the new clock source, according to the CLK_SRC_SR bit field or the ACBI(4), ACBI(3) and ACBI(2) bits is selected, and
- the new PWM generation starts.

In parallel, the channel issues a read request to the ARU to reload the shadow registers with new values while the channel continues to operate on the current CM0 and CM1 registers. To guarantee that the reloading is accomplished in a timely manor, the PWM period must not be smaller than the worst case ARU round trip time. Otherwise, the old PWM characteristics are loaded again from the shadow registers.

When receiving updates via the ARU:

- the new period duration must be specified in the lower part of the ARU data word (bits 23:0),

- the duty cycle duration must be specified in the upper part of the ARU data word (bits 47:24) ARU data word, and
- the new clock source must be specified in the control bit field (bits 52:48) of the ARU data word.

See ARU Data input stream pipeline structure for SOMP mode.

12.3.3.3 ARU Data input stream pipeline structure for SOMP mode

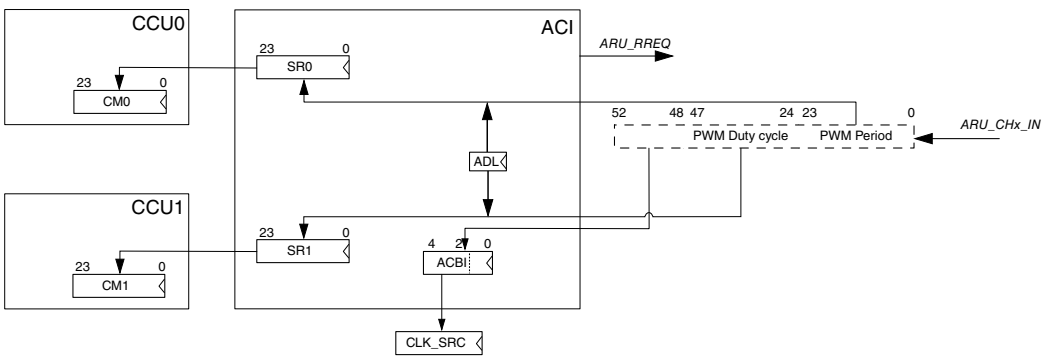


Figure 12-13. SOMP mode input data stream from ARU

When an ARU transfer is in progress (*ARU_RREQ* is served by the ARU), the ACI locks the update mechanism of the **CM0**, **CM1** and CLK_SRC registers until the ARU read request has completed. During the time period that the update mechanism is locked, the CCU0 and CCU1 units operate on old values.

The **SR0** and **SR1** shadow registers can be updated, over the AEI bus interface, by the CPU. During this update, the **CM0** and **CM1** register's update mechanism must be locked by setting the **ATOM[i]_AGC_GLB_CTRL[UPENn]** bit field to '01'. To select a new clock source (in this case), the CPU must write a new value to the **ATOM[i]_CHn_CTRL[**CLK_SRC_SR] bit field.

For an asynchronous update of the duty cycle and/or period, new values can be written directly into the **CM0** and/or **CM1** compare registers while the **CN0** counter register continues to count.

This asynchronous update can be accomplished by:

- the CPU (over the AEI bus) writing new values directly into the **CM0** and **CM1** registers, or
- the CPU (over the AEI bus) setting the ATOM[i]_AGC_FUPD_CTRL[FUPD_CTRLn] bit field to '10', which causes the *FUPDn* signal to trigger loading of the SR0 and SR1 shadow registers into the CM0 and CM1 compare registers.

Values received via the ARU are never loaded asynchronously into the **CM0** and **CM1** compare registers.

The ATOM channel can generate a PWM signal on it's *ATOM[i]_CHn_OUT* output (based on the content of the **CM0** and **CM1** registers) while the channel receives new PWM values via the ARU, and then loads those values into the SR0 and SR1 shadow registers.

On a 'true' compare match of the **CN0** counter register and the **CM0** or **CM1** compare register, the level of the *ATOM[i]_CHn_OUT* output toggles to the level of the **ATOM[i]_CHn_CTRL[SL]** bit.

So, the duty cycle output level can be changed during runtime by writing a new duty cycle level into the **ATOM[i]_CHn_CTRL[SL]** bit.. The new signal level becomes active for the next *CCU_TRIGn* trigger (since the bit SL bit was written to).

Because the *ATOM[i]_CHn_OUT* output level is defined as the reverse duty cycle output level when the ATOM channel is enabled (by setting the ATOM[i]_AGC_ENDIS_STAT[ENDIS_STATLn] bit field to '10'), a PWM period can be shifted earlier by writing an initial offset value to the **CN0** counter register. The ATOM channel allows the **CN0** counter register to count until it reaches the value of the **CM0** compare register, and then the channel toggles the level of the *ATOM[i]_CHn_OUT* output.

12.3.3.4 SOMP One-shot mode

The ATOM channel can operate in one-shot mode by setting the ATOM[i]_CHn_CTRL[**OSM**] bit to one. When in one-shot mode, a single pulse (with the level of the ATOM[i]_CHn_CTRL[**SL**] bit) is generated on the ATOM[i]_CHn_OUT output .

First the channel has to be enabled by setting the corresponding **ATOM[i]_CHn_STAT[ENDIS_STAT**] bit.

In one-shot mode, the **CN0** counter register will not start incrementing when the channel becomes enabled.

A write access to the **CN0** counter register triggers the start of pulse generation (i.e. incrementation of the **CN0** counter register).

When the **CN0** counter register is loaded with zeroes from the **CM0** register, the first edge is generated at the *ATOM[i]_CHn_OUT* output.

To avoid an update of the CMx register with content of the SRx register at this point in time, the automatic update should be disabled by setting UPEN_CTRLn = 00 (in register ATOM[i]_CHn_CTRL).

The second edge is generated when the value in the incrementing **CN0** counter register becomes ≥ to the value in the **CM1** register. The CM1 register can be updated while the CN0 counter register is incrementing. If the updated value in the CM1 register is ≤ the value in the incrementing **CN0** counter register, the second edge is generated.

If the value in the **CN0** counter register reaches the value in the **CM0** register a second time, the counter stops incrementing.

After the generation of the second edge, loading a value into the **CN0** counter register determines the start delay of another first edge. The start delay time of the first edge is calculated by:

(**CM0 – CN0**) x (period of **CLK_SRC**).

Figure 12-14 shows pulse generation in SOMP one-shot mode.

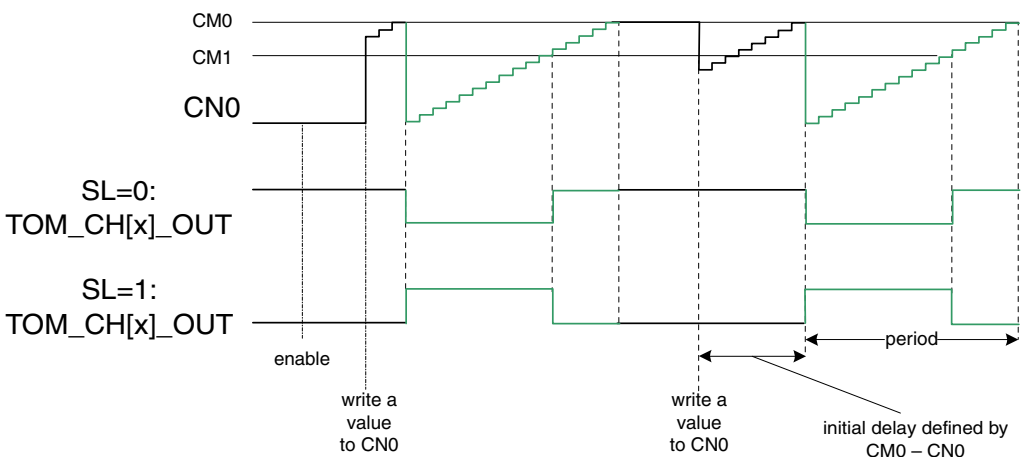


Figure 12-14. PWM output vs SL bit in SOMP one-shot mode

Further output of single pulses can be started by a write access to register CN0.

12.3.3.5 Memory Map and Registers

memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
D004	ATOM Channel Control Register in SOMP mode (ATOM[i]_CHn_CTRL_in_SOMP)	32	R/W	See section	12.3.3.5.1/ 340

12.3.3.5.1 ATOM Channel Control Register in SOMP mode (ATOM[i]_CHn_CTRL_in_SOMP)

Address: 0h base + D004h offset = D004h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0				Not_ Used	OSM	0	TRIGOUT	0			RST_CCU0	0			Not_ Used
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0	CLK_SRC_SR			SL	0		Not_Used			ADL		ARU_EN	Not_ Used	MODE	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATOM[i]_CHn_CTRL_in_SOMP field descriptions

Field	Description
0–3 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
4 Not_Used	Not used in SOMP mode.
5 OSM	One-shot mode. 0 Continuous PWM generation after channel enable. 1 A single pulse is generated.
6 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0. 0 Capture SRx time stamps after CCU0 match event not provided to ARU. 1 Capture SRx time stamps after CCU0 match event provided to ARU.
7 TRIGOUT	Trigger output selection (output signal TRIG_CHn) of module ATOM_CHn. 0 TRIG_n is TRIG_[n-1]. 1 TRIG_n is TRIG_CCU0.
8–10 Reserved	Not used in this mode. This read-only field is reserved and always has the value 0.
11 RST_CCU0	Reset source of CCU0.

Table continues on the next page...

ATOM[i]_CHn_CTRL_in_SOMP field descriptions (continued)

Field	Description
	NOTE: If RST_CCU0 =1 and UPEN_CTRLx = 1 are set, TRIG_[x-1] also triggers the update of the CM0, CM1 and CLK_SRC working registers. 0 Reset counter register CN0 to 0 on matching comparison with CM0. 1 Reset counter register CN0 to 0 on trigger TRIG_[n-1].
12–14 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
15 Not_Used	Not used in SOMP mode.
16 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
17–19 CLK_SRC_SR	Shadow register for CMU clock source register CLK_SRC. This register is a shadow register for the CMU_CLKx select. Thus, if the CMU_CLK source for PWM generation should be changed during operation, the old CMU_CLK has to operate until the update of the ATOM channels internal CLK_SRC register by the CLK_SRC_SR content is done either by an end of a period or a FORCE_UPDATE. After (channel) reset, the selected CLK_SRC value is the SYS_CLK (input of Global Clock Divider). To use one of the CMU_CLKx, it is required to perform a forced update of CLK_SRC with the value of CLK_SRC_SR value before/with enabling the channel. The following bits are applicable only if ARU_EN = 0. 000 CMU_CLK0 selected. 001 CMU_CLK1 selected. 010 CMU_CLK2 selected. 011 CMU_CLK3 selected. 100 CMU_CLK4 selected. 101 CMU_CLK5 selected. 110 CMU_CLK6 selected. 111 CMU_CLK7 selected
20 SL	Defines signal level when channel and output are disabled. If the output is disabled, the output ATOM_OUTn is set to inverse value of SL. 0 Low signal level. 1 High signal level.
21–22 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
23–25 Not_Used	Not used in SOMP mode.
26–27 ADL	ARU data select for SOMP. This bit field is only used in SOMP mode to select the ARU data source. 00 Load both ARU words into shadow registers.

Table continues on the next page...

ATOM[i]_CHn_CTRL_in_SOMP field descriptions (continued)

Field	Description
	01 Load ARU low word (Bits 23..0) into shadow register SR0. 10 Load ARU low word (Bits 23..0) into shadow register SR1. 11 Reserved.
28 ARU_EN	ARU Input stream enable. 0 ARU Input stream disabled. 1 ARU Input stream enabled.
29 Not_Used	Not used in SOMP mode.
30–31 MODE	ATOM channel mode select. 00 Not used in SOMP mode. 01 Not used in SOMP mode.. 10 ATOM Signal Output Mode Serial (SOMP) 11 Not used in SOMP mode.

12.3.4 ATOM Signal Output Mode Serial (SOMS)

In ATOM Signal Output Mode Serial (SOMS), the ATOM channel acts as a serial output shift register where the content of the **CM1** register is shifted to the **ATOM[i]_CHn_OUT** output whenever the CM1 register is triggered by the configured (**ATOM[i]_CHn_CTRL[CLK_SRC]**) *CMU_CLKn* input clock signal. The shift direction is configurable by writing to bit zero of:

- the **ATOM[i]_CHn_CTRL[ACB]** bit field when the ARU connection is disabled (**ATOM[i]_CHn_CTRL[ARU_EN]** bit = 0), or
- the **ATOM[i]_CHn_STAT[ACBI]** bit field when the ARU connection is enabled (**ATOM[i]_CHn_CTRL[ARU_EN]** bit = 1).

The data inside the **CM1** register has to be aligned according to the configured shift direction. When a right shift is configured, that the data word has to be aligned to bit zero of the **CM1** register; when a left shift is configured, the data has to be aligned to bit 23 of the **CM1** register.

12.3.4.1 SOMS Mode output generation

Figure 12-16 shows output generation if SOMS mode is configured (ATOM[i]_CHn_CTRL[MODE] bit field = 11).

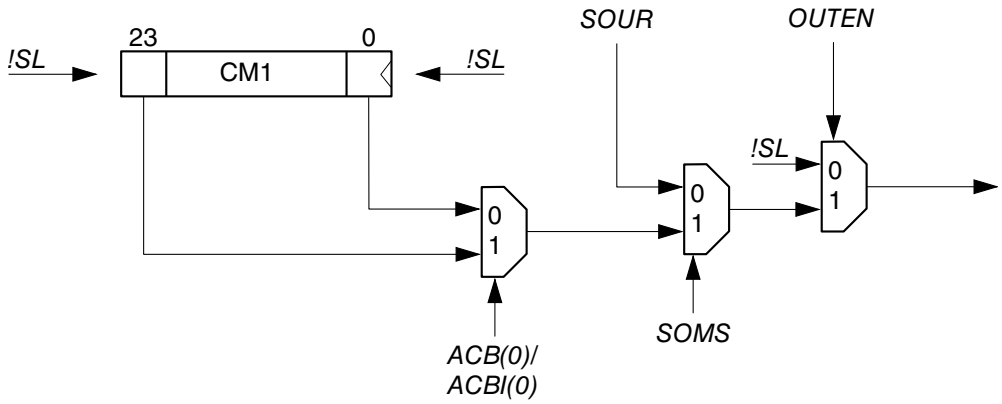


Figure 12-16. SOMS mode output generation

In SOMS mode, CCU0 runs in counter/compare mode and counts the number of bits that have been shifted. The total number of bits that are to be shifted is defined by the value in the **CM0** register. The total number of bits that are shifted via the ATOM_CHn_OUT output is **CM0 + 1**.

When the ATOM_CHn_OUT output is disabled (ATOM[i]_AGC_OUTEN_STAT[OUTEN_STATn] bit field = 01), the output is set to the inverse level of the ATOM[i]_CHn_CTRL[SL] bit.

When the content of the **CM1** register is shifted out, the inverse signal level is shifted into the **CM1** register.

When the output becomes enabled (ATOM[i]_AGC_OUTEN_STAT[OUTEN_STATn] bit field = 10) while the **ATOM[i]_AGC_GLB_CTRL[UPEN_CTRLn]** bit field = 01 (disabled), the level of the ATOM_OUT output is the same as the level of bit zero or 23 of the **CM1** register, dependent on the configured shift direction.

Figure 12-17 shows the startup behavior for a right shift. To show startup behavior for a left shift, bit zero in the CM1 register has to be replaced by bit 23 in the **CM1** register.

12.3.4.2 SOMS Output signal level at startup, UPEN_CTRLn disabled

Figure 12-17 shows SOMS mode output behavior at startup and when the channel is disabled (ATOM[i]_AGC_GLB_CTRL[UPEN_CTRLn] = 01).

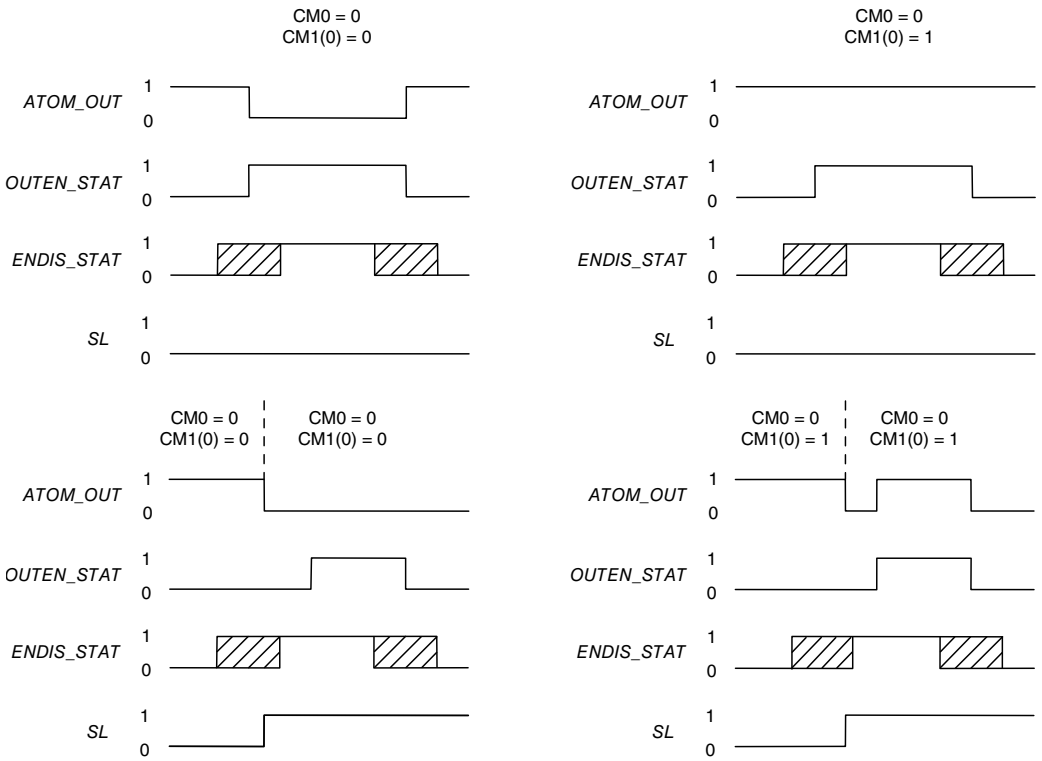


Figure 12-17. Output level at startup, ATOM[i]_AGC_GLB_CTRL[UPEN_CTRLn] = 01 (disabled)

If **ATOM[i]_AGC_GLB_CTRL[UPEN_CTRLn]** is set to '10' (enabled) and the channel is enabled (ATOM[i]_AGC_OUTEN_STAT[OUTEN_STATn] = 10), the level of the ATOM[i]_CHn_Out output is the same as bit zero or 23 of the **CM1** register, depending on the shift direction (see SOMS Output signal level at startup, UPEN_CTRLn enabled).

12.3.4.3 SOMS Output signal level at startup, UPEN_CTRLn enabled

Figure 12-18 shows the the level of the ATOM_CHn_OUT output when in SOMS mode with ATOM[i]_AGC_GLB_CTRL[UPEN_CTRLn] = 10 (enabled).

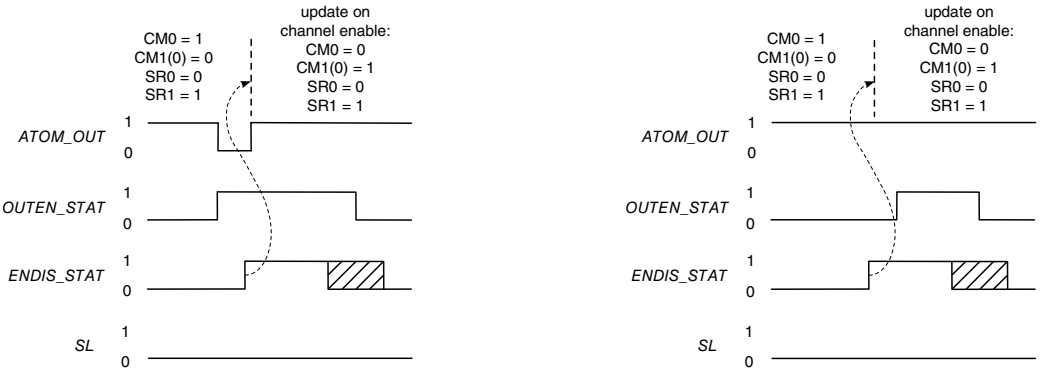


Figure 12-18. SOMS output level at startup with UPEN_CTRLn enabled

When the serial data that is to be shifted is provided via the ARU, the number of bits that are to be shifted must be defined in the lower 24 bits of the ARU word (bits 23:0) and the data that is to be shifted must be defined in the upper 24 bits of the ARU word (bits 47:24) and aligned according to the shift direction. The shift direction (SL0 bit) must be defined in bit 48 of the ARU word.

If the **ATOM[i]_AGC_GLB_CTRL[UPEN_CTRLn]** bit field is set to '10' after an update of the **CM0/CM1** register (with the content of the **SR0/SR1** register), a new ARU read request is set up.

If the **UPEN_CTRLn** bit field is not set to '10', a new ARU read request is not set up and the channel will stop shifting (with no automatic restart) after the value in the **CN0** counter register has reached the value in the **CM0** register.

If a channel becomes enabled (ATOM[i]_AGC_ENDIS_STAT[ENDIS_STATn] bit field = 10) while in the **ARU_EN** = 1 (enabled) and MODE = 11 (SOMS) configuration (ATOM[i]_CHn_CTRL[ARU_EN, MODE]), the first received values from the ARU are stored in the **SR0** and **SR1** registers. If the values in the **CN0** counter register and the **CM0** register are zeroes (i.e. **CN0** is not counting) and the the **ATOM[i]_AGC_GLB_CTRL[UPEN_CTRLn]** bit field =10 (enabled), an immediate update of the **CM0** and **CM1** registers (from the SR0 and SR1 registers) takes place. This update triggers the start of shifting.

When the ATOM[i]_CHn_CTRL[**ARU_EN]** bit is set to '0' (disabled), it is recommended that the ATOM channel be configured in one-shot mode (ATOM[i]_CHn_CTRL[OSM=1, MODE=10]), allowing the channel to reload new values from the SR0 and SR1shadow registers when the value in the **CN0** counter register reaches the value in the **CM0** register.

12.3.4.4 SOMS mode with ARU_EN = 1, OSM = 0, UPEN_CTRLn = 1

When ATOM[i]_CHn_CTRL[OSM=0, ARL_EN=1,MODE=11] and **ATOM[i]_AGC_GLB_CTRL[UPEN_CTRLn=1]**, the channel is running in the SOMS continuous mode. When the value in the **CM0** register equals the value in the **CN0** counter register and if new values are available in the SR0 and SR1 register (ATOM[i]_CHn_STAT[DV=1])):

- the **CM0** and **CM1** registers are reloaded with values from the **SR0** and **SR1** registers, and
- new values are requested from the ARU.

If the update of the SR0 and SR1 shadow registers does not happen before the value in the **CN0** counter register reaches the value in the **CM0** register, the old values in the **SR0** and **SR1** registers are used to reload the CM0 and CM1 registers.

Unlike when the CPU controls the channel via the AEI bus, the shift direction (defined by bit 48 of the ARU word) is effective only after the **CM0** and **CM1** registers are reloaded with values from the **SR0** and **SR1** registers.

12.3.4.5 SOMS mode with ARU_EN = 1 and OSM = 1, UPEN_CTRLn = 1

When ATOM[i]_CHn_CTRL[OSM=1, ARL_EN=1,MODE=11] and **ATOM[i]_AGC_GLB_CTRL[UPEN_CTRLn=1]**, the channel is running in the SOMS one-shot mode. When the value in the **CM0** register equals the value in the **CN0** counter register and if new values are available in the SR0 and SR1 register (ATOM[i]_CHn_STAT[DV=1])):

- the **CM0** and **CM1** registers are reloaded with values from the **SR0** and **SR1**registers, and
- new values are requested from the ARU.

If new values are not available in **SR0** and **SR1** registers:

- the **CM0** and **CM1** registers will not be updated,
- the **CN0** counter will stop incrementing, and
- the channel continues to request new data from ARU.

Later, when new data from the ARU is received and loaded into the **SR0** and **SR1** registers, the **CM0** and **CM1** registers are reloaded with new values from the **SR0** and **SR1**registers, and incrementing of the **CN0** counter is restarted.

12.3.4.6 SOMS mode with ARU_EN = 0 and OSM = 0, UPEN_CTRLn = 1

When ATOM[i]_CHn_CTRL[OSM=0, ARL_EN=0,MODE=11] and **ATOM[i]_AGC_GLB_CTRL[UPEN_CTRLn=1]**, the channel updates the CM0 and CM1 registers with the values in the SR0 and SR1 registers and immediately restarts shifting . The first bit of of the new CM1 register value will be routed to the ATOM[i]_CHn_OUT output without any gap between it and the end of the last bit of the previous CM1 register value.

12.3.4.7 SOMS mode with ARU_EN = 0, OSM = 1, UPEN_CTRLn = 1

When ATOM[i]_CHn_CTRL[OSM=1, ARL_EN=0,MODE=11] and **ATOM[i]_AGC_GLB_CTRL[UPEN_CTRLn=1]**, the ATOM channel stops shifting when **CN0** reaches **CM0** and no update of **CM0** and **CM1** is performed.

Then, the shifting of the channel can be restarted again by writing a zero (0) to the **CN0** register again. Please note, that the **CN0** register should be written with a zero since the **CN0** register counts the number of bits shifted out by the ATOM channel.

The writing of a zero to **CN0** causes also an immediate update of **CM0/CM1** register with the content of **SR0/SR1** register.

12.3.4.8 Interrupts in SOMS mode

When in Signal Output Mode Serial and when the value in the **CN0** counter register is >= the value in the **CM0** register, the **ATOM[i]_CHn_IRQ_NOTIFY**[CCU0TC] interrupt is asserted. The **ATOM[i]_CHn_IRQ_NOTIFY**[CCU1TC] interrupt has no meaning and it is not asserted.

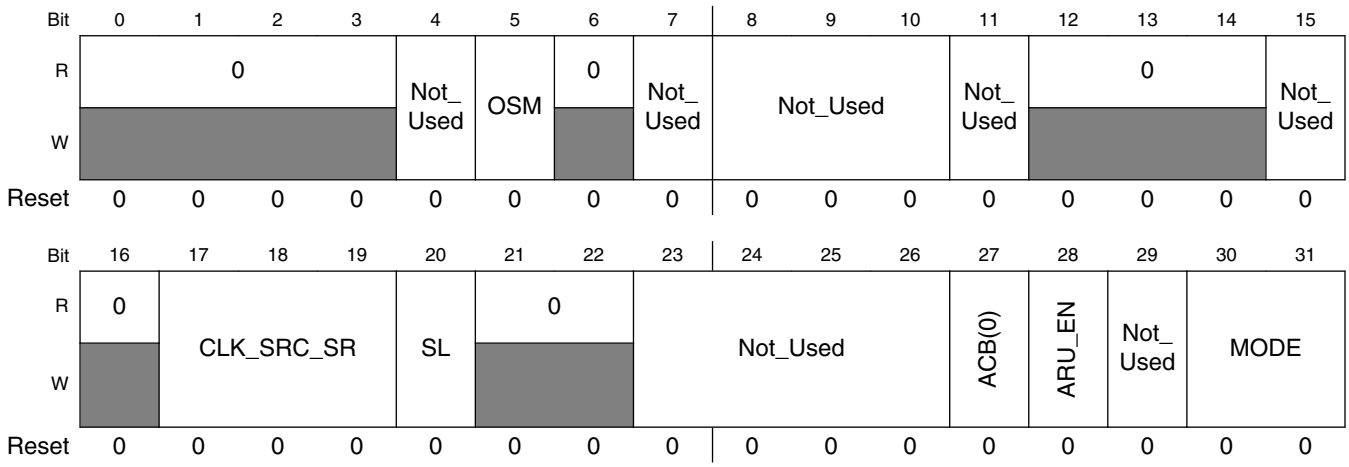
12.3.4.9 Memory Map and Registers

memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
D004	ATOM Channel Control Register in SOMS mode (ATOM[i]_CHn_CTRL_in_SOMS)	32	R/W	0000_0000h	12.3.4.9.1/ 348

12.3.4.9.1 ATOM Channel Control Register in SOMS mode (ATOM[i]_CHn_CTRL_in_SOMS)

Address: 0h base + D004h offset = D004h



ATOM[i]_CHn_CTRL_in_SOMS field descriptions

Field	Description
0–3 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
4 Not_Used	Not used in SOMS mode.
5 OSM	One-shot mode. 0 Continuous shifting is enabled. 1 Channel stops, after number of bits defined in CM0 is shifted out.
6 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
7 Not_Used	Not used in SOMS mode.
8–10 Not_Used	Not used in SOMS mode.
11 Not_Used	Not used in SOMS mode.
12–14 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
15 Not_Used	Not used in SOMS mode.

Table continues on the next page...

ATOM[i]_CHn_CTRL_in_SOMS field descriptions (continued)

Field	Description
16 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
17–19 CLK_SRC_SR	Shift frequency select for channel. This register is a shadow register for the CMU_CLKx select. Thus, if the channel should operate on another CMU_CLK then CMU_CKL0 at the beginning, the different CMU_CLK has to be specified inside this register and the CMU_CLK has to be configured with a FORCE_UPDATE in that case before the channel operation would start. 000 CMU_CLK0 selected. 001 CMU_CLK1 selected. 010 CMU_CLK2 selected. 011 CMU_CLK3 selected. 100 CMU_CLK4 selected. 101 CMU_CLK5 selected. 110 CMU_CLK6 selected. 111 CMU_CLK7 selected
20 SL	Defines signal level when channel and output are disabled. If the output is disabled, the output ATOM_OUTn is set to inverse value of SL. If the output is enabled, the output ATOM_OUT[x] is set to bit 0 or 23 of CM1 register. The inverse value of SL is shifted into the CM1 register. An enable or disable of channel n has no effect on ATOM_OUTn. 0 Low signal level. 1 High signal level.
21–22 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
23–26 Not_Used	Not used in SOMS mode.
27 ACB(0)	Shift direction for CM1 register. The data that has to be shifted out has to be aligned inside the CM1 register according to the defined shift direction. This bit is applicable only if ARU_EN = '0'. If the direction (ACB0) is changed the output ATOM_OUTn switches immediately to the other 'first' bit of CM1 (bit 0 if ACB(0) = 0, bit 23 if ACB(0) = 1). 0 Right shift of data is started from bit 0 of CM1. 1 Left shift of data is started from bit 23 of CM1.
28 ARU_EN	ARU Input stream enable. 0 ARU Input stream disabled. 1 ARU Input stream enabled.

Table continues on the next page...

ATOM[i]_CHn_CTRL_in_SOMS field descriptions (continued)

Field	Description
29 Not_Used	Not used in SOMS mode.
30–31 MODE	ATOM channel mode select. 11 ATOM Signal Output Mode Serial (SOMS).

12.4 ATOM Interrupt signals

Table 12-18 show the ATOM's interrupt signals.

Table 12-18. ATOM interrupt signals

Signal	Description
CCU0TCn_IRQ	CCU0 Trigger condition interrupt for channel n
CCU1TCn_IRQ	CCU1 Trigger condition interrupt for channel n

NOTE

For the ATOM submodules, the interrupts are bundled within the ICM submodule a second time to reduce external interrupt lines. The interrupts are OR-ed in a manner that one GTM external interrupt line represents two adjacent ATOM channel interrupts. For ATOM[i] , the bundling is shown in Table 18-1.

12.5 Memory Map and Registers

The ARU-connected Timer Output Module (ATOM0) registers are described as follows:

ATOM_0 memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	ATOM0 Channel n Read Address Register, n=0:7 (ATOM_0_CH0_RDADDR)	32	R/W	01FE_01FEh	12.5.1/356
4	ATOM0 Channel n Control Register, n=0:7 (ATOM_0_CH0_CTRL)	32	R/W	0000_0000h	12.5.2/357
8	ATOM0 Channel n Shadow 0 Register, n=0:7 (ATOM_0_CH0_SR0)	32	R/W	0000_0000h	12.5.3/360
C	ATOM0 Channel n Shadow 1 Register, n=0:7 (ATOM_0_CH0_SR1)	32	R/W	0000_0000h	12.5.4/360

Table continues on the next page...

ATOM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
10	ATOM0 Channel n Compare Match 0 Register, n=0:7 (ATOM_0_CH0_CM0)	32	R/W	0000_0000h	12.5.5/361
14	ATOM0 Channel n Compare Match 1 Register, n=0:7 (ATOM_0_CH0_CM1)	32	R/W	0000_0000h	12.5.6/362
18	ATOM0 Channel n Counter 0 Register, n=0:7 (ATOM_0_CH0_CN0)	32	R/W	0000_0000h	12.5.7/362
1C	ATOM0 Channel n Status Register, n=0:7 (ATOM_0_CH0_STAT)	32	R/W	0000_0001h	12.5.8/363
20	ATOM0 Channel n Interrupt Request Notification Register, n=0:7 (ATOM_0_CH0_IRQ_NOTIFY)	32	R/W	0000_0000h	12.5.9/364
24	ATOM0 Channel n Interrupt Request Enable Register, n=0:7 (ATOM_0_CH0_IRQ_EN)	32	R/W	0000_0000h	12.5.10/365
28	ATOM0 Channel n Force Interrupt Request Register, n=0:7 (ATOM_0_CH0_IRQ_FORCINT)	32	R/W	0000_0000h	12.5.11/366
2C	ATOM0 Channel n Interrupt Request Mode Register, n=0:7 (ATOM_0_CH0_IRQ_MODE)	32	R/W	See section	12.5.12/367
40	ATOM0 AGC Global Control Register (ATOM_0_AGC_GLB_CTRL)	32	R/W	0000_0000h	12.5.13/368
44	ATOM0 TGC0 Enable/Disable Control Register (ATOM_0_AGC_ENDIS_CTRL)	32	R/W	0000_0000h	12.5.14/371
48	ATOM0 TGC0 Enable/Disable Status Register (ATOM_0_AGC_ENDIS_STAT)	32	R/W	0000_0000h	12.5.15/374
4C	ATOM0 TGC0 Action Time Base Register (ATOM_0_AGC_ACT_TB)	32	R/W	0000_0000h	12.5.16/377
50	ATOM0 TGC0 Out Enable Control Register (ATOM_0_AGC_OUTEN_CTRL)	32	R/W	0000_0000h	12.5.17/378
54	ATOM0 TGC0 Out Enable Control/Status Register (ATOM_0_AGC_OUTEN_STAT)	32	R/W	0000_0000h	12.5.18/380
58	ATOM0 TGC0 Force Update Control Register (ATOM_0_AGC_FUPD_CTRL)	32	R/W	0000_0000h	12.5.19/383
5C	ATOM0 TGC0 Interrupt Trigger Register (ATOM_0_AGC_INT_TRIG)	32	R/W	0000_0000h	12.5.20/387
80	ATOM0 Channel n Read Address Register, n=0:7 (ATOM_0_CH1_RDADDR)	32	R/W	01FE_01FEh	12.5.1/356
84	ATOM0 Channel n Control Register, n=0:7 (ATOM_0_CH1_CTRL)	32	R/W	0000_0000h	12.5.2/357
88	ATOM0 Channel n Shadow 0 Register, n=0:7 (ATOM_0_CH1_SR0)	32	R/W	0000_0000h	12.5.3/360
8C	ATOM0 Channel n Shadow 1 Register, n=0:7 (ATOM_0_CH1_SR1)	32	R/W	0000_0000h	12.5.4/360
90	ATOM0 Channel n Compare Match 0 Register, n=0:7 (ATOM_0_CH1_CM0)	32	R/W	0000_0000h	12.5.5/361
94	ATOM0 Channel n Compare Match 1 Register, n=0:7 (ATOM_0_CH1_CM1)	32	R/W	0000_0000h	12.5.6/362

Table continues on the next page...

ATOM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
98	ATOM0 Channel n Counter 0 Register, n=0:7 (ATOM_0_CH1_CN0)	32	R/W	0000_0000h	12.5.7/362
9C	ATOM0 Channel n Status Register, n=0:7 (ATOM_0_CH1_STAT)	32	R/W	0000_0001h	12.5.8/363
A0	ATOM0 Channel n Interrupt Request Notification Register, n=0:7 (ATOM_0_CH1_IRQ_NOTIFY)	32	R/W	0000_0000h	12.5.9/364
A4	ATOM0 Channel n Interrupt Request Enable Register, n=0:7 (ATOM_0_CH1_IRQ_EN)	32	R/W	0000_0000h	12.5.10/365
A8	ATOM0 Channel n Force Interrupt Request Register, n=0:7 (ATOM_0_CH1_IRQ_FORCINT)	32	R/W	0000_0000h	12.5.11/366
AC	ATOM0 Channel n Interrupt Request Mode Register, n=0:7 (ATOM_0_CH1_IRQ_MODE)	32	R/W	See section	12.5.12/367
100	ATOM0 Channel n Read Address Register, n=0:7 (ATOM_0_CH2_RDADDR)	32	R/W	01FE_01FEh	12.5.1/356
104	ATOM0 Channel n Control Register, n=0:7 (ATOM_0_CH2_CTRL)	32	R/W	0000_0000h	12.5.2/357
108	ATOM0 Channel n Shadow 0 Register, n=0:7 (ATOM_0_CH2_SR0)	32	R/W	0000_0000h	12.5.3/360
10C	ATOM0 Channel n Shadow 1 Register, n=0:7 (ATOM_0_CH2_SR1)	32	R/W	0000_0000h	12.5.4/360
110	ATOM0 Channel n Compare Match 0 Register, n=0:7 (ATOM_0_CH2_CM0)	32	R/W	0000_0000h	12.5.5/361
114	ATOM0 Channel n Compare Match 1 Register, n=0:7 (ATOM_0_CH2_CM1)	32	R/W	0000_0000h	12.5.6/362
118	ATOM0 Channel n Counter 0 Register, n=0:7 (ATOM_0_CH2_CN0)	32	R/W	0000_0000h	12.5.7/362
11C	ATOM0 Channel n Status Register, n=0:7 (ATOM_0_CH2_STAT)	32	R/W	0000_0001h	12.5.8/363
120	ATOM0 Channel n Interrupt Request Notification Register, n=0:7 (ATOM_0_CH2_IRQ_NOTIFY)	32	R/W	0000_0000h	12.5.9/364
124	ATOM0 Channel n Interrupt Request Enable Register, n=0:7 (ATOM_0_CH2_IRQ_EN)	32	R/W	0000_0000h	12.5.10/365
128	ATOM0 Channel n Force Interrupt Request Register, n=0:7 (ATOM_0_CH2_IRQ_FORCINT)	32	R/W	0000_0000h	12.5.11/366
12C	ATOM0 Channel n Interrupt Request Mode Register, n=0:7 (ATOM_0_CH2_IRQ_MODE)	32	R/W	See section	12.5.12/367
180	ATOM0 Channel n Read Address Register, n=0:7 (ATOM_0_CH3_RDADDR)	32	R/W	01FE_01FEh	12.5.1/356
184	ATOM0 Channel n Control Register, n=0:7 (ATOM_0_CH3_CTRL)	32	R/W	0000_0000h	12.5.2/357
188	ATOM0 Channel n Shadow 0 Register, n=0:7 (ATOM_0_CH3_SR0)	32	R/W	0000_0000h	12.5.3/360
18C	ATOM0 Channel n Shadow 1 Register, n=0:7 (ATOM_0_CH3_SR1)	32	R/W	0000_0000h	12.5.4/360

Table continues on the next page...

ATOM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
190	ATOM0 Channel n Compare Match 0 Register, n=0:7 (ATOM_0_CH3_CM0)	32	R/W	0000_0000h	12.5.5/361
194	ATOM0 Channel n Compare Match 1 Register, n=0:7 (ATOM_0_CH3_CM1)	32	R/W	0000_0000h	12.5.6/362
198	ATOM0 Channel n Counter 0 Register, n=0:7 (ATOM_0_CH3_CN0)	32	R/W	0000_0000h	12.5.7/362
19C	ATOM0 Channel n Status Register, n=0:7 (ATOM_0_CH3_STAT)	32	R/W	0000_0001h	12.5.8/363
1A0	ATOM0 Channel n Interrupt Request Notification Register, n=0:7 (ATOM_0_CH3_IRQ_NOTIFY)	32	R/W	0000_0000h	12.5.9/364
1A4	ATOM0 Channel n Interrupt Request Enable Register, n=0:7 (ATOM_0_CH3_IRQ_EN)	32	R/W	0000_0000h	12.5.10/365
1A8	ATOM0 Channel n Force Interrupt Request Register, n=0:7 (ATOM_0_CH3_IRQ_FORCINT)	32	R/W	0000_0000h	12.5.11/366
1AC	ATOM0 Channel n Interrupt Request Mode Register, n=0:7 (ATOM_0_CH3_IRQ_MODE)	32	R/W	See section	12.5.12/367
200	ATOM0 Channel n Read Address Register, n=0:7 (ATOM_0_CH4_RDADDR)	32	R/W	01FE_01FEh	12.5.1/356
204	ATOM0 Channel n Control Register, n=0:7 (ATOM_0_CH4_CTRL)	32	R/W	0000_0000h	12.5.2/357
208	ATOM0 Channel n Shadow 0 Register, n=0:7 (ATOM_0_CH4_SR0)	32	R/W	0000_0000h	12.5.3/360
20C	ATOM0 Channel n Shadow 1 Register, n=0:7 (ATOM_0_CH4_SR1)	32	R/W	0000_0000h	12.5.4/360
210	ATOM0 Channel n Compare Match 0 Register, n=0:7 (ATOM_0_CH4_CM0)	32	R/W	0000_0000h	12.5.5/361
214	ATOM0 Channel n Compare Match 1 Register, n=0:7 (ATOM_0_CH4_CM1)	32	R/W	0000_0000h	12.5.6/362
218	ATOM0 Channel n Counter 0 Register, n=0:7 (ATOM_0_CH4_CN0)	32	R/W	0000_0000h	12.5.7/362
21C	ATOM0 Channel n Status Register, n=0:7 (ATOM_0_CH4_STAT)	32	R/W	0000_0001h	12.5.8/363
220	ATOM0 Channel n Interrupt Request Notification Register, n=0:7 (ATOM_0_CH4_IRQ_NOTIFY)	32	R/W	0000_0000h	12.5.9/364
224	ATOM0 Channel n Interrupt Request Enable Register, n=0:7 (ATOM_0_CH4_IRQ_EN)	32	R/W	0000_0000h	12.5.10/365
228	ATOM0 Channel n Force Interrupt Request Register, n=0:7 (ATOM_0_CH4_IRQ_FORCINT)	32	R/W	0000_0000h	12.5.11/366
22C	ATOM0 Channel n Interrupt Request Mode Register, n=0:7 (ATOM_0_CH4_IRQ_MODE)	32	R/W	See section	12.5.12/367
280	ATOM0 Channel n Read Address Register, n=0:7 (ATOM_0_CH5_RDADDR)	32	R/W	01FE_01FEh	12.5.1/356
284	ATOM0 Channel n Control Register, n=0:7 (ATOM_0_CH5_CTRL)	32	R/W	0000_0000h	12.5.2/357

Table continues on the next page...

ATOM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
288	ATOM0 Channel n Shadow 0 Register, n=0:7 (ATOM_0_CH5_SR0)	32	R/W	0000_0000h	12.5.3/360
28C	ATOM0 Channel n Shadow 1 Register, n=0:7 (ATOM_0_CH5_SR1)	32	R/W	0000_0000h	12.5.4/360
290	ATOM0 Channel n Compare Match 0 Register, n=0:7 (ATOM_0_CH5_CM0)	32	R/W	0000_0000h	12.5.5/361
294	ATOM0 Channel n Compare Match 1 Register, n=0:7 (ATOM_0_CH5_CM1)	32	R/W	0000_0000h	12.5.6/362
298	ATOM0 Channel n Counter 0 Register, n=0:7 (ATOM_0_CH5_CN0)	32	R/W	0000_0000h	12.5.7/362
29C	ATOM0 Channel n Status Register, n=0:7 (ATOM_0_CH5_STAT)	32	R/W	0000_0001h	12.5.8/363
2A0	ATOM0 Channel n Interrupt Request Notification Register, n=0:7 (ATOM_0_CH5_IRQ_NOTIFY)	32	R/W	0000_0000h	12.5.9/364
2A4	ATOM0 Channel n Interrupt Request Enable Register, n=0:7 (ATOM_0_CH5_IRQ_EN)	32	R/W	0000_0000h	12.5.10/365
2A8	ATOM0 Channel n Force Interrupt Request Register, n=0:7 (ATOM_0_CH5_IRQ_FORCINT)	32	R/W	0000_0000h	12.5.11/366
2AC	ATOM0 Channel n Interrupt Request Mode Register, n=0:7 (ATOM_0_CH5_IRQ_MODE)	32	R/W	See section	12.5.12/367
300	ATOM0 Channel n Read Address Register, n=0:7 (ATOM_0_CH6_RDADDR)	32	R/W	01FE_01FEh	12.5.1/356
304	ATOM0 Channel n Control Register, n=0:7 (ATOM_0_CH6_CTRL)	32	R/W	0000_0000h	12.5.2/357
308	ATOM0 Channel n Shadow 0 Register, n=0:7 (ATOM_0_CH6_SR0)	32	R/W	0000_0000h	12.5.3/360
30C	ATOM0 Channel n Shadow 1 Register, n=0:7 (ATOM_0_CH6_SR1)	32	R/W	0000_0000h	12.5.4/360
310	ATOM0 Channel n Compare Match 0 Register, n=0:7 (ATOM_0_CH6_CM0)	32	R/W	0000_0000h	12.5.5/361
314	ATOM0 Channel n Compare Match 1 Register, n=0:7 (ATOM_0_CH6_CM1)	32	R/W	0000_0000h	12.5.6/362
318	ATOM0 Channel n Counter 0 Register, n=0:7 (ATOM_0_CH6_CN0)	32	R/W	0000_0000h	12.5.7/362
31C	ATOM0 Channel n Status Register, n=0:7 (ATOM_0_CH6_STAT)	32	R/W	0000_0001h	12.5.8/363
320	ATOM0 Channel n Interrupt Request Notification Register, n=0:7 (ATOM_0_CH6_IRQ_NOTIFY)	32	R/W	0000_0000h	12.5.9/364
324	ATOM0 Channel n Interrupt Request Enable Register, n=0:7 (ATOM_0_CH6_IRQ_EN)	32	R/W	0000_0000h	12.5.10/365
328	ATOM0 Channel n Force Interrupt Request Register, n=0:7 (ATOM_0_CH6_IRQ_FORCINT)	32	R/W	0000_0000h	12.5.11/366
32C	ATOM0 Channel n Interrupt Request Mode Register, n=0:7 (ATOM_0_CH6_IRQ_MODE)	32	R/W	See section	12.5.12/367

Table continues on the next page...

ATOM_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
380	ATOM0 Channel n Read Address Register, n=0:7 (ATOM_0_CH7_RDADDR)	32	R/W	01FE_01FEh	12.5.1/356
384	ATOM0 Channel n Control Register, n=0:7 (ATOM_0_CH7_CTRL)	32	R/W	0000_0000h	12.5.2/357
388	ATOM0 Channel n Shadow 0 Register, n=0:7 (ATOM_0_CH7_SR0)	32	R/W	0000_0000h	12.5.3/360
38C	ATOM0 Channel n Shadow 1 Register, n=0:7 (ATOM_0_CH7_SR1)	32	R/W	0000_0000h	12.5.4/360
390	ATOM0 Channel n Compare Match 0 Register, n=0:7 (ATOM_0_CH7_CM0)	32	R/W	0000_0000h	12.5.5/361
394	ATOM0 Channel n Compare Match 1 Register, n=0:7 (ATOM_0_CH7_CM1)	32	R/W	0000_0000h	12.5.6/362
398	ATOM0 Channel n Counter 0 Register, n=0:7 (ATOM_0_CH7_CN0)	32	R/W	0000_0000h	12.5.7/362
39C	ATOM0 Channel n Status Register, n=0:7 (ATOM_0_CH7_STAT)	32	R/W	0000_0001h	12.5.8/363
3A0	ATOM0 Channel n Interrupt Request Notification Register, n=0:7 (ATOM_0_CH7_IRQ_NOTIFY)	32	R/W	0000_0000h	12.5.9/364
3A4	ATOM0 Channel n Interrupt Request Enable Register, n=0:7 (ATOM_0_CH7_IRQ_EN)	32	R/W	0000_0000h	12.5.10/365
3A8	ATOM0 Channel n Force Interrupt Request Register, n=0:7 (ATOM_0_CH7_IRQ_FORCINT)	32	R/W	0000_0000h	12.5.11/366
3AC	ATOM0 Channel n Interrupt Request Mode Register, n=0:7 (ATOM_0_CH7_IRQ_MODE)	32	R/W	See section	12.5.12/367

12.5.1 ATOM0 Channel n Read Address Register, n=0:7 (ATOM_0_CH*n*_RDADDR)

Address: D000h base + 0h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0							RDADDR1									0							RDADDR0								
W																																
Reset	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0

ATOM_0_CH*n*_RDADDR field descriptions

Field	Description
0–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–15 RDADDR1	ARU Read address 1. **NOTE:** This read address is only applicable in SOMC mode.

Table continues on the next page...

ATOM_0_CH*n*_RDADDR field descriptions (continued)

Field	Description
	The ATOM channel switches to this read address, when requested in ARU control bit[52:48] with the pattern "111--". The channel switches back to the RDADDR0 after one ARU data package was received on RDADDR1. **NOTE:** This bitfield is only writeable if channel is disabled.
16–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–31 RDADDR0	ARU Read address 0. This read address is used by the ATOM channel to receive data from ARU immediately after the channel and ARU access is enabled (see ATOM[i]_CH[x]_CTRL register for details). **NOTE:** This bitfield is only writeable if channel is disabled.

12.5.2 ATOM0 Channel n Control Register, n=0:7 (ATOM_0_CH*n*_CTRL)

Address: D000h base + 4h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0				ABM	OSM	SLA	TRIGOUT	0			RST_CCU0	0			WR_REQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0	CLK_SRC_SR			SL	0	CMP_CTRL	ACB					ARU_EN	TB12_SEL	MODE	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATOM_0_CH*n*_CTRL field descriptions

Field	Description
0–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 ABM	ARU blocking mode. **NOTE:** This bit is only applicable in SOMC mode.

Table continues on the next page...

ATOM_0_CH*n*_CTRL field descriptions (continued)

Field	Description
	0 ARU blocking mode disabled: ATOM reads continuously from ARU and updates CM0, CM1 independent of pending compare match event. 1 ARU blocking mode enabled: After updating CM0,CM1 via ARU, no new data is read from ARU until compare match event occurred and SR0 and/or SR1 are read.
5 OSM	One-shot mode. **NOTE:** This bit is only applicable in SOMP and SOMS modes. 0 Continuous PWM generation after channel enable. 1 A single pulse is generated.
6 SLA	Serve last ARU communication strategy This bit is only applicable in SOMC mode. Setting this bit is only effective when ACBI(4:2) is configured for serve last compare strategy (“100”, “101”, or “110”). When this bit is not set, the captured time stamps in the shadow registers SRx are only provided after the CCU1 match occurred. The ACBO(4:3) bits always return "10" in that case. By setting this bit, the ATOM channel also provides the captured time stamps after the CCU0 match event to the ARU. The ACBO(4:3) bits are set to “01” in that case. After the CCU1 match event, the time stamps are captured again in the SRx registers and provided to the ARU. The ACBO(4:3) bits are set to “10”. When the data in the shadow registers after the CCU0 match was not consumed by an ARU destination and the CCU1 match occurs, the data in the shadow registers is overwritten by the new captured time stamps. The ATOM channel does not request new data from the ARU when the CCU0 match values are read from an ARU destination. 0 Capture SRx time stamps after CCU0 match event not provided to ARU. 1 Capture SRx time stamps after CCU0 match event provided to ARU.
7 TRIGOUT	NTrigger output selection (output signal TRIG_CHn) of module ATOM_CHn. **NOTE:** This bit is only applicable in SOMC mode. 0 TRIG_[x[is TRIG_[x-1]. 1 TRIG_[x] is TRIG_CCU0.
8–10 Reserved	Reads as zero, should be written as zero. This field is reserved. This read-only field is reserved and always has the value 0.
11 RST_CCU0	Reset source of CCU0. **NOTE:** This bit is only applicable in SOMP mode. 0 Reset counter register CN0 to 0 on matching comparison with CM0. 1 Reset counter register CN0 to 0 on trigger TRIG_[x-1]. **NOTE:** If RST_CCU0=1 and UPEN_CTRLx=1, TRIG_[x-1] also triggers the update of work registers (CM0, CM1 and CLK_SRC).
12–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ATOM_0_CH*n*_CTRL field descriptions (continued)

Field	Description
15 WR_REQ	CPU Write request bit for late compare register update. **NOTE:** This bit is only applicable in SOMC mode. 0 No late update requested by CPU. 1 Late update requested by CPU.
16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–19 CLK_SRC_SR	Actual CMU clock source (SOMS) / shadow register for CMU clock source (SOMP). **NOTE:** This register is a shadow register for the CMU_CLKn select. Thus, if the CMU_CLK source for PWM generation should be changed during operation, the old CMU_CLK has to operate until the update of the ATOM channels internal CLK_SRC register by the CLK_SRC_SR content is done either by an end of a period or a FORCE_UPDATE. **NOTE:** After (channel) reset, the selected CLK_SRC value is the SYS_CLK (input of Global Clock Divider). To use in SOMP mode, one of the CMU_CLKx, it is required to perform a forced update of CLK_SRC with the value of CLK_SRC_SR value before/with enabling the channel. 000 CMU_CLK0 selected. 001 CMU_CLK1 selected. 010 CMU_CLK2 selected. 011 CMU_CLK3 selected. 100 CMU_CLK4 selected. 101 CMU_CLK5 selected. 110 CMU_CLK6 selected. 111 CMU_CLK7 selected.
20 SL	Initial signal level after channel enable. **NOTE:** If the channel is disabled or the output is disabled, the output ATOM_OUT[n] is set to inverse value of SL. **NOTE:** In SOMS mode, this bit is only applicable when the channel and its output is disabled. 0 Low signal level. 1 High signal level.
21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 CMP_CTRL	CCUn compare strategy select.. **NOTE:** This bit is only applicable in SOMC mode. 0 Greater/equal compare against TBU time base values (TBU_TS1/2 greater than or equal to CM0/1). 1 Less/equal compare against TBU time base values (TBU_TS1/2 less than or equal to CM0/1).
23–27 ACB	ATOM Mode control bits. **NOTE:** These bits have different meaning in the different ATOM channel modes. Please refer to the mode description sections. **NOTE:** These bits are only applicable when ARU_EN = '0'.

Table continues on the next page...

ATOM_0_CH*n*_CTRL field descriptions (continued)

Field	Description
28 ARU_EN	ARU Input stream enable. 0 ARU Input stream disabled. 1 ARU Input stream enabled.
29 TB12_SEL	Select time base value TBU_TS1 or TBU_TS2. **NOTE:** This bit is only applicable in SOMC mode. 0 TBU_TS1 selected for comparison. 1 TBU_TS2 selected for comparison.
30–31 MODE	ATOM channel mode select. 00 ATOM Signal Output Mode Immediate (SOMI). 01 ATOM Signal Output Mode Compare (SOMC). 10 ATOM Signal Output Mode PWM (SOMP) 11 ATOM Signal Output Mode Serial (SOMS)

12.5.3 ATOM0 Channel n Shadow 0 Register, n=0:7 (ATOM_0_CH*n*_SR0)

Address: D000h base + 8h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								SR0																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATOM_0_CH*n*_SR0 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 SR0	ATOM channel n shadow register SR0. The SR0 register is used as shadow register for CM0 in SOMP and SOMS modes and is used as capture register for time base TBU_TS0 in SOMC mode.

12.5.4 ATOM0 Channel n Shadow 1 Register, n=0:7 (ATOM_0_CH*n*_SR1)

Address: D000h base + Ch offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								SR1																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATOM_0_CH*n*_SR1 field descriptions

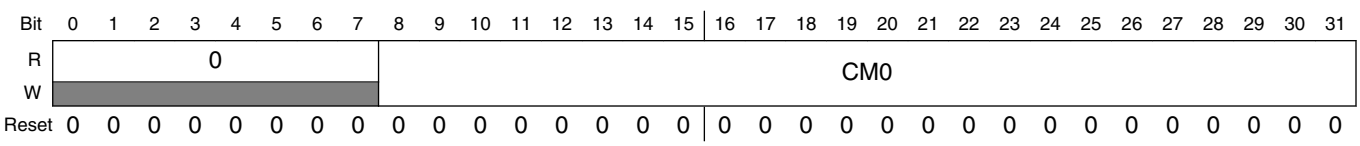
Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 SR1	ATOM channel n shadow register SR1. In SOMC mode, the SR1 register is used as shadow register for CM1 in SOMP and SOMS modes and is used as capture register for time base TBU_TS1 or TBU_TS2 (when selected in ATOM[i]_CH[x]_CTRL register).

12.5.5 ATOM0 Channel n Compare Match 0 Register, n=0:7 (ATOM_0_CH*n*_CM0)

NOTE

This register is write protected in SOMC mode and returns AEI_STATUS='10' on write access, when in serve last compare strategy the first match of CCU0 occurred.

Address: D000h base + 10h offset + (128d × i), where i=0d to 7d



ATOM_0_CH*n*_CM0 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 CM0	ATOM CCU0 compare register.

12.5.6 ATOM0 Channel n Compare Match 1 Register, n=0:7 (ATOM_0_CH*n*_CM1)

NOTE

This register is write protected in SOMC mode and returns AEI_STATUS='10' on write access, when in serve last compare strategy the first match of CCU0 occurred.

Address: D000h base + 14h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								CM1																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATOM_0_CH*n*_CM1 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 CM1	ATOM CCU1 compare register.

12.5.7 ATOM0 Channel n Counter 0 Register, n=0:7 (ATOM_0_CH*n*_CN0)

Address: D000h base + 18h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								CN0																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATOM_0_CH*n*_CN0 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 CN0	ATOM CCU0 counter register.

12.5.8 ATOM0 Channel n Status Register, n=0:7 (ATOM_0_CH*n*_STAT)

Address: D000h base + 1Ch offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0			ACBO					0	WRF	DV	ACBI				
W										w1c						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0															OL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

ATOM_0_CH*n*_STAT field descriptions

Field	Description
0–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–7 ACBO	ATOM Internal status bits. **NOTE:** These bits are only set in SOMC mode. ACBO is reset to 0b00000 on an update of register CM0 or CM1 (via ARU or CPU). This field reflects the internal status of the ATOM channel. In SOMC mode, these bits are send as ARU control bits 52 .. 48. 01000 CCU0 Compare match occurred. 10000 CCU1 Compare match occurred.
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 WRF	Write request of CPU failed for late update. **NOTE:** This bit is only applicable in SOMC mode. The bit WRF can be reset by writing a 1 to it. 0 Late update was successful, CCUx units wait for comparison. ATOM_1401. 1 Late update failed.
10 DV	Valid ARU Data stored in compare registers. **NOTE:** This bit is only applicable in SOMC mode. The CPU can determine the status of the ARU transfers with this bit. After the compare event occurred, the bit is reset by hardware. 0 No valid data was received by ARU. 1 Valid data received by ARU and stored in CM0 and/or CM1.

Table continues on the next page...

ATOM_0_CH*n*_STAT field descriptions (continued)

Field	Description
11–15 ACBI	ATOM Mode control bits received through ARU. **NOTE:** This field serves as a mirror for the five ARU control bits received through the ARU interface. The bits are valid, when the DV bit is set.
16–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
31 OL	Actual output signal level of ATOM_CHn_OUT. **NOTE:** Reset value is the inverted value of bit SL 0 Actual output signal level is low. 1 Actual output signal level is high.

12.5.9 ATOM0 Channel n Interrupt Request Notification Register, n=0:7 (ATOM_0_CH*n*_IRQ_NOTIFY)

Address: D000h base + 20h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

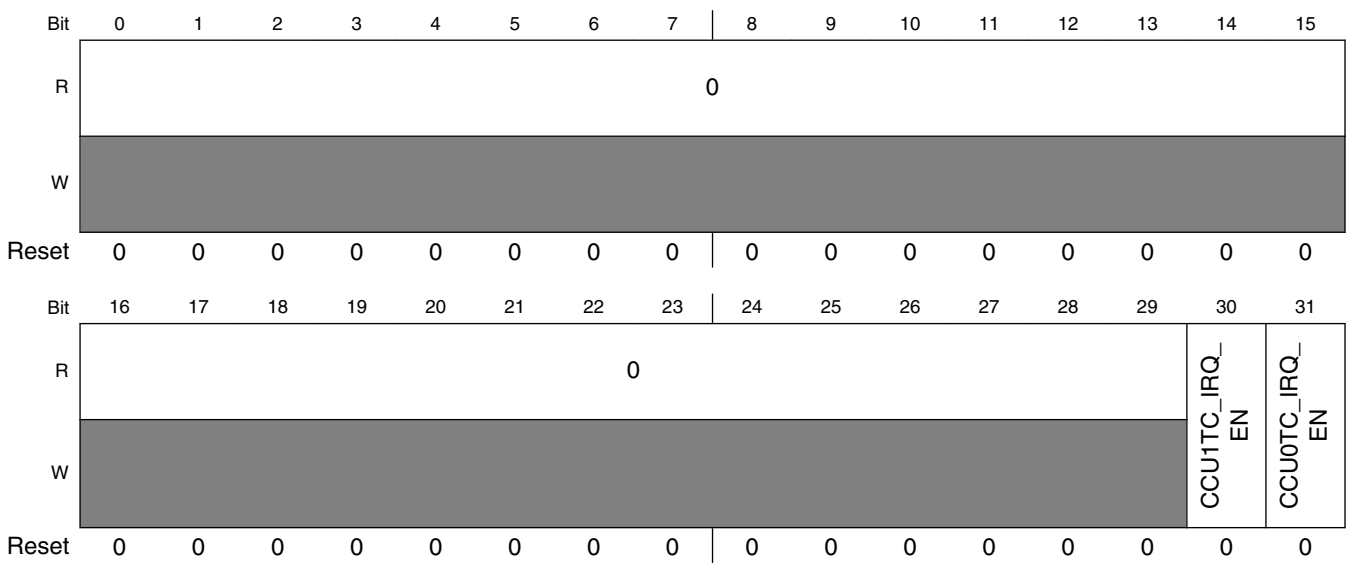
Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0														CCU1TC	CCU0TC
W															w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATOM_0_CH*n*_IRQ_NOTIFY field descriptions

Field	Description
0–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 CCU1TC	CCU1 Trigger condition interrupt for channel n. **NOTE:** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged. 0 No interrupt occurred.. 1 CCU1 trigger condition interrupt was raised by ATOM channel x.
31 CCU0TC	CCU0 Trigger condition interrupt for channel n. **NOTE:** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged. 0 No interrupt occurred. 1 CCU0 trigger condition interrupt was raised by ATOM channel x.

12.5.10 ATOM0 Channel n Interrupt Request Enable Register, n=0:7 (ATOM_0_CH*n*_IRQ_EN)

Address: D000h base + 24h offset + (128d × i), where i=0d to 7d



ATOM_0_CH*n*_IRQ_EN field descriptions

Field	Description
0–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

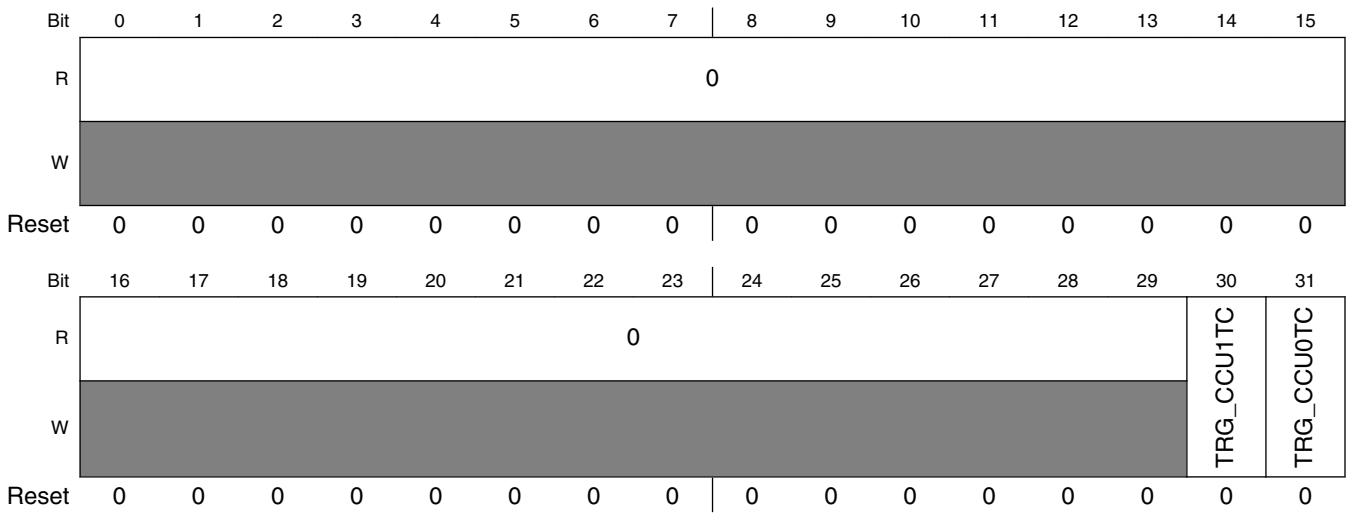
Table continues on the next page...

ATOM_0_CH*n*_IRQ_EN field descriptions (continued)

Field	Description
30 CCU1TC_IRQ_EN	ATOM_CCU1TC_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
31 CCU0TC_IRQ_EN	ATOM_CCU0TC_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.

12.5.11 ATOM0 Channel n Force Interrupt Request Register, n=0:7 (ATOM_0_CH*n*_IRQ_FORCINT)

Address: D000h base + 28h offset + (128d × i), where i=0d to 7d



ATOM_0_CH*n*_IRQ_FORCINT field descriptions

Field	Description
0–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 TRG_CCU1TC	Trigger ATOM_CCU1TC_IRQ interrupt by software. **NOTE:** This bit is cleared automatically after write. **NOTE:** This bit is write protected by bit GTM_CTRL[RF_PROT]. 0 No interrupt triggering. 1 Assert CCU1TC_IRQ interrupt for one clock cycle.
31 TRG_CCU0TC	Trigger ATOM_CCU0TC_IRQ interrupt by software. **NOTE:** This bit is cleared automatically after write.

Table continues on the next page...

ATOM_0_CH*n*_IRQ_FORCINT field descriptions (continued)

Field	Description
	NOTE: This bit is write protected by bit GTM_CTRL[RF_PROT]. 0 No interrupt triggering. 1 Assert CCU0TC_IRQ interrupt for one clock cycle.

12.5.12 ATOM0 Channel n Interrupt Request Mode Register, n=0:7 (ATOM_0_CH*n*_IRQ_MODE)

Address: D000h base + 2Ch offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0														IRQ_MODE	
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- Bits 28-31 do not reset to a given value.

ATOM_0_CH*n*_IRQ_MODE field descriptions

Field	Description
0–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–31 IRQ_MODE	IRQ mode selection. 00 Level mode. 01 Pulse mode. 10 Pulse-Notify mode. 11 Single-Pulse mode.

12.5.13 ATOM0 AGC Global Control Register (ATOM_0_AGC_GLB_CTRL)

Address: D000h base + 40h offset = D040h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	UPEN_CTRL7		UPEN_CTRL6		UPEN_CTRL5		UPEN_CTRL4		UPEN_CTRL3		UPEN_CTRL2		UPEN_CTRL1		UPEN_CTRL0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	RST_CH7	RST_CH6	RST_CH5	RST_CH4	RST_CH3	RST_CH2	RST_CH1	RST_CH0	0							HOST_TRIG
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATOM_0_AGC_GLB_CTRL field descriptions

Field	Description
0–1 UPEN_CTRL7	For channel 7, ATOM channel enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Per channel write of following double bit values is possible: 00 Don't care, bits 0:1 will not change. 01 Update disabled: reads as 00. 10 Update enabled: reads as 11. 11 Don't care, bits 0:1 will not change. Per channel read of following double values means: 00 Channel disabled. 11 Channel enabled.
2–3 UPEN_CTRL6	For channel 6, ATOM channel enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Per channel write of following double bit values is possible: 00 Don't care, bits 2:3 will not change. 01 Update disabled: reads as 00. 10 Update enabled: reads as 11. 11 Don't care, bits 2:3 will not change. Per channel read of following double values means: 00 Channel disabled. 11 Channel enabled.

Table continues on the next page...

ATOM_0_AGC_GLB_CTRL field descriptions (continued)

Field	Description
4–5 UPEN_CTRL5	For channel 5, ATOM channel enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Per channel write of following double bit values is possible: 00 Don't care, bits 4:5 will not change. 01 Update disabled: reads as 00. 10 Update enabled: reads as 11. 11 Don't care, bits 4:5 will not change. Per channel read of following double values means: 00 Channel disabled. 11 Channel enabled.
6–7 UPEN_CTRL4	For channel 4, ATOM channel enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Per channel write of following double bit values is possible: 00 Don't care, bits 6:7 will not change. 01 Update disabled: reads as 00. 10 Update enabled: reads as 11. 11 Don't care, bits 6:7 will not change. Per channel read of following double values means: 00 Channel disabled. 11 Channel enabled.
8–9 UPEN_CTRL3	For channel 3, ATOM channel enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Per channel write of following double bit values is possible: 00 Don't care, bits 8:9 will not change. 01 Update disabled: reads as 00. 10 Update enabled: reads as 11. 11 Don't care, bits 8:9 will not change. Per channel read of following double values means: 00 Channel disabled. 11 Channel enabled.
10–11 UPEN_CTRL2	For channel 2, ATOM channel enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Per channel write of following double bit values is possible: 00 Don't care, bits 10:11 will not change. 01 Update disabled: reads as 00. 10 Update enabled: reads as 11. 11 Don't care, bits 10:11 will not change. Per channel read of following double values means:

Table continues on the next page...

ATOM_0_AGC_GLB_CTRL field descriptions (continued)

Field	Description
	00 Channel disabled. 11 Channel enabled.
12–13 UPEN_CTRL1	For channel 1, ATOM channel enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. Per channel write of following double bit values is possible: 00 Don't care, bits 12:13 will not change. 01 Update disabled: reads as 00. 10 Update enabled: reads as 11. 11 Don't care, bits 12:13 will not change. Per channel read of following double values means: 00 Channel disabled. 11 Channel enabled.
14–15 UPEN_CTRL0	For channel 0, ATOM channel enable update of register CM0, CM1 and CLK_SRC_STAT from SR0, SR1 and CLK_SRC. >Per channel write of following double bit values is possible: 00 Don't care, bits 14:15 will not change. 01 Update disabled: reads as 00. 10 Update enabled: reads as 11. 11 Don't care, bits 14:15 will not change. Per channel read of following double values means: 00 Channel disabled. 11 Channel enabled.
16 RST_CH7	See RST_CH0
17 RST_CH6	See RST_CH0
18 RST_CH5	See RST_CH0
19 RST_CH4	See RST_CH0
20 RST_CH3	See RST_CH0
21 RST_CH2	See RST_CH0
22 RST_CH1	See RST_CH0
23 RST_CH0	Software reset of corresponding channel 0. **NOTE:** Bit is automatically cleared after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The S-R FlipFlop SOUR is reset to '1'.

Table continues on the next page...

ATOM_0_AGC_GLB_CTRL field descriptions (continued)

Field	Description
	0 No action. 1 Reset the corresponding channel.
24–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
31 HOST_TRIG	Trigger request signal (see AGC) to update the register ENDIS_STAT and OUTEN_STAT. **NOTE:** This flag is reset automatically after triggering the update. 0 No trigger request. 1 Set trigger request.

12.5.14 ATOM0 TGC0 Enable/Disable Control Register (ATOM_0_AGC_ENDIS_CTRL)

Address: D000h base + 44h offset = D044h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	ENDIS_CTRL7		ENDIS_CTRL6		ENDIS_CTRL5		ENDIS_CTRL4		ENDIS_CTRL3		ENDIS_CTRL2		ENDIS_CTRL1		ENDIS_CTRL0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATOM_0_AGC_ENDIS_CTRL field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–17 ENDIS_CTRL7	Channel 7 enable/disable update value. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[7]=0) or the output is disabled (OUTEN[7]=0), the TOM channel 7 output TOM_OUT[7] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 16:17 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 16:17 of this register.
18–19 ENDIS_CTRL6	Channel 6 enable/disable update value.

Table continues on the next page...

ATOM_0_AGC_ENDIS_CTRL field descriptions (continued)

Field	Description
	If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[6]=0) or the output is disabled (OUTEN[6]=0), the TOM channel 6 output TOM_OUT[6] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 18:19 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 18:19 of this register.
20–21 ENDIS_CTRL5	Channel 5 enable/disable update value. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[5]=0) or the output is disabled (OUTEN[5]=0), the TOM channel 5 output TOM_OUT[5] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 20:21 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 20:21 of this register.
22–23 ENDIS_CTRL4	TOM channel 4 enable/disable update value. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[4]=0) or the output is disabled (OUTEN[4]=0), the TOM channel 4 output TOM_OUT[4] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 22:23 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don't change bits 22:23 of this register.
24–25 ENDIS_CTRL3	TOM channel 3 enable/disable update value. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[3]=0) or the output is disabled (OUTEN[3]=0), the TOM channel 3 output TOM_OUT[3] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 24:25 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger.

Table continues on the next page...

ATOM_0_AGC_ENDIS_CTRL field descriptions (continued)

Field	Description
	11 Don’t change bits 24:25 of this register.
26–27 ENDIS_CTRL2	TOM channel 2 enable/disable update value. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[2]=0) or the output is disabled (OUTEN[2]=0), the TOM channel 2 output TOM_OUT[2] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don’t care, bits 26:27 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don’t change bits 26:27 of this register.
28–29 ENDIS_CTRL1	TOM channel 1 enable/disable update value. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[1]=0) or the output is disabled (OUTEN[1]=0), the TOM channel 1 output TOM_OUT[1] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don’t care, bits 28:29 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don’t change bits 28:29 of this register.
30–31 ENDIS_CTRL0	TOM channel 0 enable/disable update value. If a TOM channel is disabled, the counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If the channel is disabled (ENDIS[0]=0) or the output is disabled (OUTEN[0]=0), the TOM channel 0 output TOM_OUT[0] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don’t care, bits 30:31 of register ENDIS_STAT will not be changed on an update trigger. 01 Disable channel on an update trigger. 10 Enable channel on an update trigger. 11 Don’t change bits 30:31 of this register.

12.5.15 ATOM0 TGC0 Enable/Disable Status Register (ATOM_0_AGC_ENDIS_STAT)

Address: D000h base + 48h offset = D048h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	ENDIS_ STAT7		ENDIS_ STAT6		ENDIS_ STAT5		ENDIS_ STAT4		ENDIS_ STAT3		ENDIS_ STAT2		ENDIS_ STAT1		ENDIS_ STAT0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATOM_0_AGC_ENDIS_STAT field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–17 ENDIS_STAT7	TOM channel 7 enable/disable. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don’t care, bits 16:17 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don’t care, bits 16:17 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
18–19 ENDIS_STAT6	TOM channel 6 enable/disable. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don’t care, bits 18:19 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don’t care, bits 18:19 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled

Table continues on the next page...

ATOM_0_AGC_ENDIS_STAT field descriptions (continued)

Field	Description
20–21 ENDIS_STAT5	TOM channel 5 enable/disable. If a TOM channel is disabled, e counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 20:21 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 20:21 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
22–23 ENDIS_STAT4	TOM channel 4 enable/disable. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 22:23 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 22:23 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
24–25 ENDIS_STAT3	TOM channel 3 enable/disable. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 24:25 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 24:26 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
26–27 ENDIS_STAT2	TOM channel 2 enable/disable. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value.

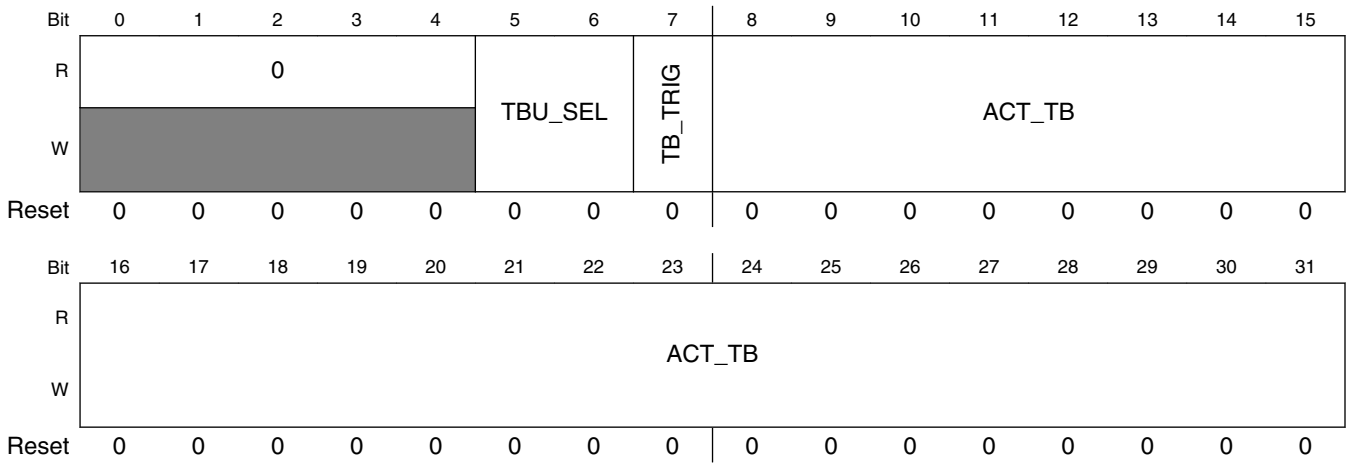
Table continues on the next page...

ATOM_0_AGC_ENDIS_STAT field descriptions (continued)

Field	Description
	Write of following double bit values is possible: 00 Don't care, bits 26:27 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 26:27 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
28–29 ENDIS_STAT1	TOM channel 1 enable/disable. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 28:29 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 28:29 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled
30–31 ENDIS_STAT0	TOM channel 0 enable/disable. If a TOM channel is disabled, counter CN0 is stopped and the FlipFlop SOUR is set to the inverse value of control bit SL. On an enable event, counter CN0 starts counting from its current value. Write of following double bit values is possible: 00 Don't care, bits 30:31 will not be changed 01 Update disabled, reads as 00 10 Update enabled, reads as 11 11 Don't care, bits 30:31 will not be changed Read of following double values means: 00 Channel disabled 11 Channel enabled

12.5.16 ATOM0 TGC0 Action Time Base Register (ATOM_0_AGC_ACT_TB)

Address: D000h base + 4Ch offset = D04Ch



ATOM_0_AGC_ACT_TB field descriptions

Field	Description
0–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–6 TBU_SEL	Selection of time base used for comparison. **NOTE:** The bit combination “10” is only applicable if the TBU of the device contains three time base channels. Otherwise, this bit combination is also reserved. Please refer to GTM Architecture block diagram to determine the number of channels for TBU of this device. 00 TBU_TS0 selected. 01 TBU_TS1 selected. 10 TBU_TS2 selected. 11 TBU_TS0 selected.
7 TB_TRIG	Set trigger request. This flag is reset automatically if the selected time base unit (TBU_TS0 or TBU_TS1 or TBU_TS2 if present) has reached the value ACT_TB and the update of the register was triggered. 0 No trigger request. 1 Set trigger request.
8–31 ACT_TB	Specifies the signed compare value with selected signal TBU_TS[n], x=0..2. If selected TBU_TS[n] value is in the interval [ACT_TB-007FFFFFh, ACT_TB], the event is in the past and the trigger is generated immediately. Otherwise the event is in the future and the trigger is generated if selected TBU_TS[n] is equal to ACT_TB.

12.5.17 ATOM0 TGC0 Out Enable Control Register (ATOM_0_AGC_OUTEN_CTRL)

Address: D000h base + 50h offset = D050h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	OUTEN_CTRL7		OUTEN_CTRL6		OUTEN_CTRL5		OUTEN_CTRL4		OUTEN_CTRL3		OUTEN_CTRL2		OUTEN_CTRL1		OUTEN_CTRL0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATOM_0_AGC_OUTEN_CTRL field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–17 OUTEN_CTRL7	Output TOM_OUT(7) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[7]=0) or the output is disabled (OUTEN[7]=0), the TOM channel 7 output TOM_OUT[7] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 16:17 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 16:17 of this register.
18–19 OUTEN_CTRL6	Output TOM_OUT(6) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[6]=0) or the output is disabled (OUTEN[6]=0), the TOM channel 6 output TOM_OUT[6] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 18:19 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 18:19 of this register.
20–21 OUTEN_CTRL5	Output TOM_OUT(5) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[5]=0) or the output is disabled (OUTEN[5]=0), the TOM channel 5 output TOM_OUT[5] is the inverted value of bit SL. Write of following double bit values is possible:

Table continues on the next page...

ATOM_0_AGC_OUTEN_CTRL field descriptions (continued)

Field	Description
	00 Don't care, bits 20:21 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 20:21 of this register.
22–23 OUTEN_CTRL4	Output TOM_OUT(4) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[4]=0) or the output is disabled (OUTEN[4]=0), the TOM channel 4 output TOM_OUT[4] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 22:23 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 22:23 of this register.
24–25 OUTEN_CTRL3	Output TOM_OUT(3) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[3]=0) or the output is disabled (OUTEN[3]=0), the TOM channel 3 output TOM_OUT[3] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 24:25 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 24:25 of this register.
26–27 OUTEN_CTRL2	Output TOM_OUT(2) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[2]=0) or the output is disabled (OUTEN[2]=0), the TOM channel 2 output TOM_OUT[2] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 26:27 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 26:27 of this register.
28–29 OUTEN_CTRL1	Output TOM_OUT(1) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[1]=0) or the output is disabled (OUTEN[1]=0), the TOM channel 1 output TOM_OUT[1] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 28:29 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger.

Table continues on the next page...

ATOM_0_AGC_OUTEN_CTRL field descriptions (continued)

Field	Description
	10 Enable channel output on an update trigger. 11 Don't change bits 28:29 of this register.
30–31 OUTEN_CTRL0	Output TOM_OUT(0) enable/disable update value. **NOTE:** If the channel is disabled (ENDIS[0]=0) or the output is disabled (OUTEN[0]=0), the TOM channel 0 output TOM_OUT[0] is the inverted value of bit SL. Write of following double bit values is possible: 00 Don't care, bits 30:31 of register OUTEN_STAT will not be changed on an update trigger. 01 Disable channel output on an update trigger. 10 Enable channel output on an update trigger. 11 Don't change bits 30:31 of this register.

12.5.18 ATOM0 TGC0 Out Enable Control/Status Register (ATOM_0_AGC_OUTEN_STAT)

Address: D000h base + 54h offset = D054h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	OUTEN_STAT7		OUTEN_STAT6		OUTEN_STAT5		OUTEN_STAT4		OUTEN_STAT3		OUTEN_STAT2		OUTEN_STAT1		OUTEN_STAT0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATOM_0_AGC_OUTEN_STAT field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–17 OUTEN_STAT7	Control/status of output TOM_OUT(7). Write of following double bit values is possible: 00 Don't care, bits 16:17 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 16:17 will not be changed. Read of following double values means: 00 Channel disable.

Table continues on the next page...

ATOM_0_AGC_OUTEN_STAT field descriptions (continued)

Field	Description
	11 Channel enable.
18–19 OUTEN_STAT6	Control/status of output TOM_OUT(6). Write of following double bit values is possible: 00 Don't care, bits 18:19 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 18:19 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
20–21 OUTEN_STAT5	Control/status of output TOM_OUT(5). Write of following double bit values is possible: 00 Don't care, bits 20:21 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 20:21 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
22–23 OUTEN_STAT4	Control/status of output TOM_OUT(4) Write of following double bit values is possible: 00 Don't care, bits 22:23 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 22:23 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
24–25 OUTEN_STAT3	Control/status of output TOM_OUT(3). Write of following double bit values is possible: 00 Don't care, bits 24:25 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 24:25 will not be changed. Read of following double values means: 00 Channel disable.

Table continues on the next page...

ATOM_0_AGC_OUTEN_STAT field descriptions (continued)

Field	Description
	11 Channel enable.
26–27 OUTEN_STAT2	Control/status of output TOM_OUT(2). Write of following double bit values is possible: 00 Don't care, bits 26:27 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 26:27 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
28–29 OUTEN_STAT1	Control/status of output TOM_OUT(1). Write of following double bit values is possible: 00 Don't care, bits 28:29 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 28:29 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.
30–31 OUTEN_STAT0	Control/status of output TOM_OUT(0). Write of following double bit values is possible: 00 Don't care, bits 30:31 will not be changed. 01 Channel disabled, reads as 00. 10 Channel enabled, reads as 11. 11 Don't care, bits 30:31 will not be changed. Read of following double values means: 00 Channel disable. 11 Channel enable.

12.5.19 ATOM0 TGC0 Force Update Control Register (ATOM_0_AGC_FUPD_CTRL)

Address: D000h base + 58h offset = D058h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	RSTCN0_CH7		RSTCN0_CH6		RSTCN0_CH5		RSTCN0_CH4		RSTCN0_CH3		RSTCN0_CH2		RSTCN0_CH1		RSTCN0_CH0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	FUPD_CTRL7		FUPD_CTRL6		FUPD_CTRL5		FUPD_CTRL4		FUPD_CTRL3		FUPD_CTRL2		FUPD_CTRL1		FUPD_CTRL0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATOM_0_AGC_FUPD_CTRL field descriptions

Field	Description
0–1 RSTCN0_CH7	Reset CN0 of channel 7 on force update event. Write of following double bit values is possible: 00 Don't care, bits 0:1 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 0:1 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
2–3 RSTCN0_CH6	Reset CN0 of channel 6 on force update event. Write of following double bit values is possible: 00 Don't care, bits 2:3 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 2:3 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
4–5 RSTCN0_CH5	Reset CN0 of channel 5 on force update event. Write of following double bit values is possible: 00 Don't care, bits 4:5 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11.

Table continues on the next page...

ATOM_0_AGC_FUPD_CTRL field descriptions (continued)

Field	Description
	11 Don't care, bits 4:5 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
6–7 RSTCN0_CH4	Reset CN0 of channel 4 on force update event. Write of following double bit values is possible: 00 Don't care, bits 6:7 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 6:7 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
8–9 RSTCN0_CH3	Reset CN0 of channel 3 on force update event. Write of following double bit values is possible: 00 Don't care, bits 8:9 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 8:9 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
10–11 RSTCN0_CH2	Reset CN0 of channel 2 on force update event. Write of following double bit values is possible: 00 Don't care, bits 10:11 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 10:11 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
12–13 RSTCN0_CH1	Reset CN0 of channel 1 on force update event. Write of following double bit values is possible: 00 Don't care, bits 12:13 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11.

Table continues on the next page...

ATOM_0_AGC_FUPD_CTRL field descriptions (continued)

Field	Description
	11 Don't care, bits 12:13 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
14–15 RSTCN0_CH0	Reset CN0 of channel 0 on force update event. Write of following double bit values is possible: 00 Don't care, bits 14:15 will not be changed. 01 CN0 is not reset on forced update, reads as 00. 10 CN0 is reset on forced update, reads as 11. 11 Don't care, bits 14:15 will not be changed. Read of following double values means: 00 CN0 is not reset on forced update. 11 CN0 is reset on forced update.
16–17 FUPD_CTRL7	Force update of TOM channel 7 operation registers. Write of following double bit values is possible: 00 Don't care, bits 16:17 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don't care, bits 16:17 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
18–19 FUPD_CTRL6	Force update of TOM channel 6 operation registers, Write of following double bit values is possible: 00 Don't care, bits 18:19 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don't care, bits 18:19 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
20–21 FUPD_CTRL5	Force update of TOM channel 5 operation registers. Write of following double bit values is possible: 00 Don't care, bits 20:21 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11.

Table continues on the next page...

ATOM_0_AGC_FUPD_CTRL field descriptions (continued)

Field	Description
	11 Don’t care, bits 20:21 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
22–23 FUPD_CTRL4	Force update of TOM channel 4 operation registers. Write of following double bit values is possible: 00 Don’t care, bits 22:23 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don’t care, bits 22:23 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
24–25 FUPD_CTRL3	Force update of TOM channel 3 operation registers. Write of following double bit values is possible: 00 Don’t care, bits 24:25 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don’t care, bits 24:25 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
26–27 FUPD_CTRL2	ce update of TOM channel 2 operation registers. Write of following double bit values is possible: 00 Don’t care, bits 26:27 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don’t care, bits 26:27 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
28–29 FUPD_CTRL1	Force update of TOM channel 1 operation registers. Write of following double bit values is possible: 00 Don’t care, bits 28:29 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11.

Table continues on the next page...

ATOM_0_AGC_FUPD_CTRL field descriptions (continued)

Field	Description
	11 Don't care, bits 28:29 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.
30–31 FUPD_CTRL0	Force update of TOM channel 0 operation registers. Write of following double bit values is possible: 00 Don't care, bits 30:31 will not be changed. 01 Channel disabled, reads as 00. 10 Channel Enabled, reads as 11. 11 Don't care, bits 30:31 will not be changed. Read of following double values means: 00 Channel disabled. 11 Channel enabled.

12.5.20 ATOM0 TGC0 Interrupt Trigger Register (ATOM_0_AGC_INT_TRIG)

Address: D000h base + 5Ch offset = D05Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	INT_TRIG7		INT_TRIG6		INT_TRIG5		INT_TRIG4		INT_TRIG3		INT_TRIG2		INT_TRIG1		INT_TRIG0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ATOM_0_AGC_INT_TRIG field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–17 INT_TRIG7	Select input signal TRIG_7 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 16:17 will not be changed. 01 Internal trigger from channel 7 (TRIG_7) not used, reads as 00. 10 Internal trigger from channel 7 (TRIG_7) used, reads as 11. 11 Don't care, bits 16:17 will not be changed.

Table continues on the next page...

ATOM_0_AGC_INT_TRIG field descriptions (continued)

Field	Description
	Read of following double values means: 00 Internal trigger from channel 7 (TRIG_7) not used. 11 Internal trigger from channel 7 (TRIG_7) used.
18–19 INT_TRIG6	Select input signal TRIG_6 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 18:19 will not be changed. 01 Internal trigger from channel 6 (TRIG_6) not used, reads as 00. 10 Internal trigger from channel 6 (TRIG_6) used, reads as 11. 11 Don't care, bits 18:19 will not be changed. Read of following double values means: 00 Internal trigger from channel 6 (TRIG_6) not used. 11 Internal trigger from channel 6 (TRIG_6) used.
20–21 INT_TRIG5	Select input signal TRIG_5 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 20:21 will not be changed. 01 Internal trigger from channel 5 (TRIG_5) not used, reads as 00. 10 Internal trigger from channel 5 (TRIG_5) used, reads as 11. 11 Don't care, bits 20:21 will not be changed. Read of following double values means: 00 Internal trigger from channel 5 (TRIG_5) not used. 11 Internal trigger from channel 5 (TRIG_5) used.
22–23 INT_TRIG4	Select input signal TRIG_4 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 22:23 will not be changed. 01 Internal trigger from channel 4 (TRIG_4) not used, reads as 00. 10 Internal trigger from channel 4 (TRIG_4) used, reads as 11. 11 Don't care, bits 22:23 will not be changed. Read of following double values means: 00 Internal trigger from channel 4 (TRIG_4) not used. 11 Internal trigger from channel 4 (TRIG_4) used.
24–25 INT_TRIG3	Select input signal TRIG_3 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 24:25 will not be changed. 01 Internal trigger from channel 3 (TRIG_3) not used, reads as 00. 10 Internal trigger from channel 3 (TRIG_3) used, reads as 11. 11 Don't care, bits 24:25 will not be changed.

Table continues on the next page...

ATOM_0_AGC_INT_TRIG field descriptions (continued)

Field	Description
	Read of following double values means: 00 Internal trigger from channel 3 (TRIG_3) not used. 11 Internal trigger from channel 3 (TRIG_3) used.
26–27 INT_TRIG2	Select input signal TRIG_2 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 26:27 will not be changed. 01 Internal trigger from channel 2 (TRIG_2) not used, reads as 00. 10 Internal trigger from channel 2 (TRIG_2) used, reads as 11. 11 Don't care, bits 26:27 will not be changed. Read of following double values means: 00 Internal trigger from channel 2 (TRIG_2) not used. 11 Internal trigger from channel 2 (TRIG_2) used.
28–29 INT_TRIG1	Select input signal TRIG_1 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 28:29 will not be changed. 01 Internal trigger from channel 1 (TRIG_1) not used, reads as 00. 10 Internal trigger from channel 1 (TRIG_1) used, reads as 11. 11 Don't care, bits 28:29 will not be changed. Read of following double values means: 00 Internal trigger from channel 1 (TRIG_1) not used. 11 Internal trigger from channel 1 (TRIG_1) used.
30–31 INT_TRIG0	Select input signal TRIG_0 as a trigger source. Write of following double bit values is possible: 00 Don't care, bits 30:31 will not be changed. 01 Internal trigger from channel 0 (TRIG_0) not used, reads as 00. 10 Internal trigger from channel 0 (TRIG_1) used, reads as 11. 11 Don't care, bits 30:31 will not be changed. Read of following double values means: 00 Internal trigger from channel 0 (TRIG_0) not used. 11 Internal trigger from channel 0 (TRIG_0) used.

Chapter 13
Multi Channel Sequencer (MCS)

13.1 MCS Overview

The Multi Channel Sequencer (MCS) sub-module is a custom 24-bit RISC core for handling data processing in the GTM and is connected the ARU sub-module. The MCS is a fine grain temporal mutli-threaded core with a Von Neumann architecture. Each of the 8 channels (threads) have 8 dedicated hardware registers. It has a custom fixed width 32-bit instruction set that is optimized for 24-bit data operations and channel flow control.

Some examples of how the MCS can be used are to:

- Calculate complex output sequences that are processed in combination with time base values from the TBU submodule and the ATOM submodule. The resulting output sequences are sent to the ATOM submodule's outputs.
- Perform extended data processing of input data from the TIM submodule and send the results to the CPU.
- Process data provided by the CPU, and send the results to the ATOM submodule's outputs.

13.1.1 Architecture

Figure 13-1 gives an overview of the MCS architecture.

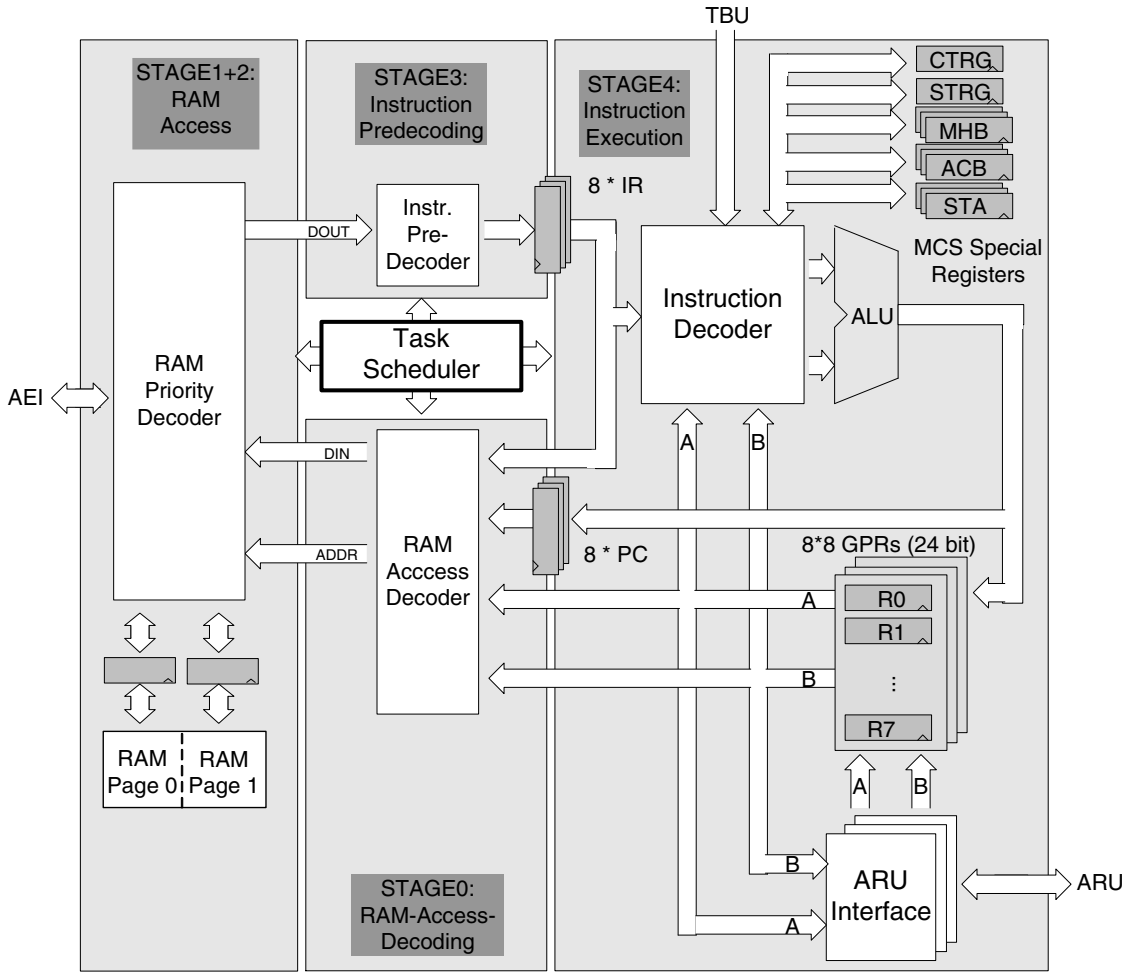


Figure 13-1. MCS architecture

The MCS submodule embeds a single data path with five pipeline stages consisting of:

- A simple 24-bit Arithmetic Logic Unit (ALU)
- Several decoders
- Two RAM pages (located outside of the MCS submodule).

The data path of the MCS is shared by eight MCS channels, where each MCS channel executes a dedicated microprogram that is stored inside the RAM pages.

Both RAM pages can contain arbitrary sized code and data sections that are accessible by:

- all of the MCS channels, and
- the CPU via the AEI bus.

The MCS handles a memory layout of up to 2^{12} memory locations, each 32 bits wide. This leads to an address range from 0 to $2^{14} – 1$ bytes. This address space is divided into two seamless memory pages.

Memory page 0 begins from address 0 and ranges to address MP0 – 4 and memory page 1 ranges from MP0 to MP1– 4.

The parameters MP0 and MP1 are defined externally by the memory configuration submodule MCFG (see MCFG Overview).

Depending on the silicon vendor configuration, the connected RAM pages are initialized with zeros when an MCS module reset is asserted.

If an ECC Error occurs while an MCS channel reads data from a memory page, the channel is disabled and the STA[ERR] bit is set to one.

A MCS channel can also be considered as an individual processor task that is scheduled at a specific point in time.

A detailed description of the task scheduling can be found in Scheduling

Each MCS channel has:

- a dedicated ARU interface for communication with other ARU connected modules,
- an Instruction Register (IR),
- a Program Counter Register (PC),
- a Status Register (STA),
- an ARU Control Bit Register (ACB),
- a Memory High Byte Register (MHB) and
- a Register Bank with eight 24 bit general purpose data registers (R0, R1, ….R7).

The registers (mentioned above) are only visible within its dedicated MCS channel. The MCS channels cannot exchange data using registers.

The only exception is the common 16-bit wide STRG and CTRG trigger registers that can be accessed by all MCS submodules in order to trigger other MCS channels located in the same submodule. A trigger bit is set by accessing the STRG register and a trigger bit is cleared by accessing to the CTRG register.

Whenever data has to be exchanged between different MCS channels, the connected RAM pages, which are accessible by all MCS channels, can be used.

Since the data registers are writable by the CPU, an MCS channel can also use its data registers to exchange data with the CPU.

The CPU can access the STRG and CTRG trigger registers to enable or disable, respectively, triggering of the MCS channels.

The main actions of the different pipeline stages are:

- Pipeline stage zero performs a setup of address, input data, and control signals for the next RAM access of a specific MCS channel.

- The actual RAM access of a specific MCS channel is executed in pipeline stages one and two, assuming an external connection of a synchronous RAM with a latency of one clock cycle.
- The RAM priority decoder arbitrates RAM accesses that are requested by the CPU via the AEI bus and by the active MCS channel. If the CPU and an MCS channel request an access to the same memory page at the same time, the MCS channel is prioritized.
- Pipeline stage three performs pre-decoding of instruction and data received from the RAM.
- Pipeline stage four executes the current instruction.

Since the internal registers of the MCS can be updated by different sources (MCS write access, AEI write access, ARU read access, and software reset), a write conflict to the registers occurs if more than one source wants to write to the same register during the same clock cycle. If no special hints are given in this specification, the following prioritizing is applied (highest priority first):

1. Software Reset
2. ARU read access
3. AEI write access
4. MCS write access

Software should setup its application in a way that avoids these possible conflicts.

13.1.2 Scheduling

The MCS submodule provides two different scheduling schemes:

- *round-robin schedule,* and
- *accelerated schedule*.

The scheduling scheme can be configured by writing to the **MCS[i]_CTRL[SCHED]** bit field.

Round-robin order scheduling assigns all MCS channels an equal amount of time slices.

In addition, the scheduler also assigns one time slice to the CPU, in order to guarantee at least one memory access by the CPU within each round-trip cycle.

Figure 13-2 shows round-robin scheduling with eight MCS channels (C_0 to C_7) that are scheduled together with a single CPU access.

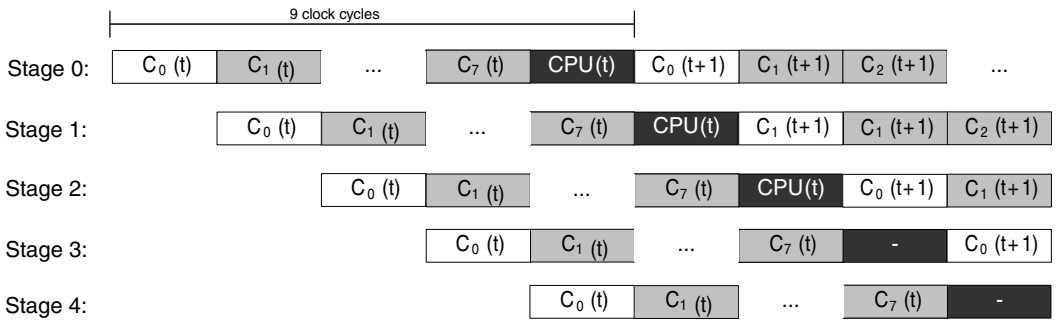


Figure 13-2. MCS round-robin scheduling

The figure also shows which MCS channel is activated in which pipeline stage at a specific point in time.

The execution time of an MCS channel, in a specific pipeline stage, is always one clock cycle.

The index, t ,marks all instruction parts of the corresponding MCS channels that belong to the same round-trip cycle.

Therefore, a single cycle instruction in an MCS channel requires an effective execution time of nine clock cycles, ignoring the five clock cycles of pipeline latency.

To improve memory bandwidth between the CPU and the MCS memory, the time slices of any suspended MCS channel are granted to the CPU.

An MCS channel can be suspended due to these reasons:

- An MCS channel is executing a blocking read or write request to an ARU connected submodule (instruction ARD, AWR, ARDI, AWRI).
- An MCS channel waits on a register match event (instruction WURM), in order to wait on a desired register value (e.g. trigger event from another MCS channel.
- An MCS channel is disabled.

Round-robin scheduling leads to a deterministic round trip time for the entire submodule. However, it can waste clock cycles by scheduling MCS channels that are not able to run at a specific point in time (assuming that there is no high CPU bandwidth required).

The *accelerated scheduling mode* improves the computational performance of the MCS by skipping suspended MCS channels (and channels that are currently in stage zero, one, two, or three) during scheduling.

Whenever the scheduler detects an MCS channel that is suspended or not in the last pipeline stage, it selects another non-suspended MCS channel.

Considering the timing shown in Figure 13-2 in conjunction with the accelerated scheduling scheme, a single cycle instruction of an MCS channel requires an effective execution time between five and nine clock cycles, depending on the number of suspended MCS channels.

In summary, the *round-robin scheduling* mode grants time slices of suspended MCS channels to the CPU and the *accelerated scheduling* mode grants time slices of suspended MCS channels to non-suspended MCS channels.

13.2 Instruction Set

This section describes the entire instruction set of the MCS submodule.

After the introduction of the different instruction formats in Instruction Format, the individual instructions are described starting with MOVL Instruction.

Each instruction is 32 bits wide, but the duration of each instruction varies by several *instruction cycles*. An instruction cycle is defined as the time (in system clock cycles) duration between two consecutive instructions of a channel.

As shown in the Scheduling section, the number of required clock cycles for a single instruction cycle can vary from five to nine clock cycles (depending on the number of suspended MCS channels) when the *accelerated scheduling* scheme is configured by writing to the **MCS[i]_CTRL** register.

Using the *round robin scheduling* scheme, each single cycle instruction takes exactly nine clock cycles.

Before the instruction formats and the actual instructions are described, some commonly used terms, abbreviations and expressions are:

- **OREG**: The operation register set, where OREG includes all MCS accessible internal registers (R0, R1, R2, …, R7, STA, ACB, CTRG, STRG, TBU_TS0, TBU_TS1, TBU_TS2, MHB) as well as the global time bases, TBU_TS0, TBU_TS1, and TBU_TS2 (that are provided by the TBU submodule).
- **AREG**: The ARU register set, where AREG includes all registers (R0, R1, R2, …, R7, ZERO) that can be written by incoming ARU transfers (ARD, ARDI, NARD, and NARDI instructions). These registers include all eight general purpose data registers. The dummy register, ZERO, can be used to discard an incoming 24-bit ARU word.
- **LIT4**: The LIT4 set, where LIT4 = {0,1, …, 15} is an unsigned 4-bit literal.
- **LIT16**: The LIT16 set, where LIT16 = {0,1, …, 2^{16}-1} is an unsigned 16-bit literal.
- **LIT24**: The LIT24 set, where LIT24 = {0,1, …, 2^{24}-1} is an unsigned 24-bit literal.

- **BIT SELECTION**: The VAR[i] expression represents the i-th bit of a VAR variable.
- **BIT RANGE SELECTION**: The VAR[m:n] expression represents the bit slice of a VARvariable that is ranging from bit n to bit m.
- **MEMORY ADRESSING**: The MEM(X) expression represents the 32-bit value at address X of the memory.

  Address X ranges between 0 and 2^{14} — 4, where X must be an integral multiple of four.

  The MEM(X)[m:n] expression represents the bit slice ranging from bit n to m of the 32-bit word at memory location X.

- **ARU ADRESSING**: When performing an ARU read, the ARU(X) expression represents the 53-bit ARU word of the ARU channel at address X.

  The read address, X, ranges between 0 and 2^{9} — 1.

  When performing an ARU write, the ARU(X) expression represents a 53-bit ARU word that is written to an ARU channel that is indexed by X.

  The X index selects a single ARU write channel from the pool of the MCS submodule's that are allocated ARU write channels.

  A single MCS submodule has 24 ARU write channels, indexed by values 0:23.

  The ARU(X)[m:n] expression represents the bit slice ranging from bit n to m of the 53-bit ARU word.

13.2.1 Instruction Format

13.2.1.1 Literal Instruction Format

The literal instruction format (shown in Figure 13-3) embeds:

- a primary 4-bit opcode, OPC0,
- a 24-bit literal value, C ∈ LIT24, and
- a 4-bit value, A, which may be an element of the OREG set or the AREG depending on the actual instruction.

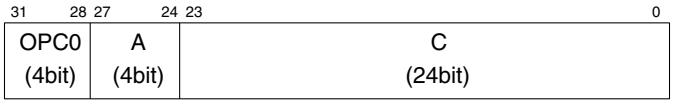


Figure 13-3. Literal instruction format

The literal instruction format is primarily used for instructions that are accessing both a 24-bit literal and a single 24-bit register as operands.

The binary codes of a 24-bit instruction are defined as "xxxxaaaaccccccccccccccccccccccc", where the 'x' digits encode the OPC0 field, the 'a' digits encode the A operand field, and the 'c' digits encode the 24-bit C literal field.

If a literal instruction is to ignore certain bits, the ignored bits are defined as '-'.

13.2.1.2 Double Operand Instruction format

Double operand instruction format (shown in Figure 13-4) embeds:

- a 4-bit primary OPC0 opcode ,
- a 4-bit OPC1 secondary opcode ,
- a 16-bit literal, C ∈ LIT16, and
- two 4-bit values, A and B, which can be elements of the OREG, AREG, or LIT4 sets, depending on the actual instruction.

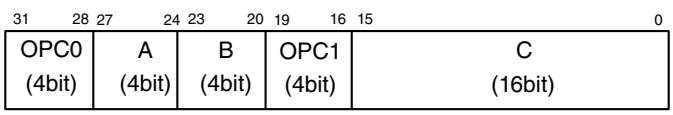


Figure 13-4. Double operand instruction format

This format is primarily used for instructions that are accessing two operands that are stored in the 24-bit registers.

The binary codes of a 16-bit literal instruction are defined as "xxxxaaaabbbbyyyyccccccccccccccc", whereas the 'x' digits encode the OPC0 bit field , the 'y' digits encode the OPC1 bit field, the 'a' digits encode the A operand field, the 'b' digits encode the B operand field , and the 'c' digits encode the 16-bit literal C field.

If an instruction is to ignore certain bits, the ignored bits are defined as '-'.

If the instruction decoder detects an invalid combination of the OPC0 and OPC1 opcode fields, the corresponding MCS channel is disabled and the STA[ERR] bit is set.

13.2.2 Data Transfer Instructions

13.2.2.1 MOVL Instruction

Syntax: MOVL A, C

Operation: A ← C

Status: Z

Duration: 1 instruction cycle

Code: 0001aaaaccccccccccccccccccccccc

Description: Transfer literal value in the C operand (C € LIT24) to the register specified by the A operand (A € OREG).

The STA[Z] zero bit is set if the transferred value is zero. Otherwise, the Z bit is cleared.

The PC program counter is incremented by 4.

13.2.2.2 MOV Instruction

Syntax: MOV A, B

Operation: A ← B

Status: Z

Duration: 1 instruction cycle

Code: 1010aaaabbbb0000----------

Description: Transfer the register specified by the B operand (B € OREG) to the register specified by the A operand (A € OREG).

The STA[Z] zero bit is set if the transferred value is zero. Otherwise, the Z bit is cleared.

The PC program counter is incremented by 4.

13.2.2.3 MRD Instruction

Syntax: MRD A, C

Operation: A ← MEM(C[13:0])[23:0];

MHB ← MEM(C[13:0])[31:24]

Status: Z

Duration: 2 instruction cycles

Code: 1010aaaa----0001--cccccccccccccc

Description: Transfer the lower 24 bits of memory content specified by the address in the C operand (C € LIT16) to the register specified by the A operand (A € OREG). The upper eight bits of the memory content at the C address are transferred to the MHB register.

The STA[Z] zero bit is set if the lower 24 bits of the transferred value is zero. Otherwise, the Z bit is cleared.

If the MHB register is specified by the A operand (A € OREG), bits 0:7 of the referenced memory location are transferred to the MHB register .

The program counter PC is incremented by the value 4.

13.2.2.4 MWR Instruction

Syntax: MWR A, C

Operation: MEM(C[13:0])[23:0] ← A;

MEM(C[13:0])[31:24] ← MHB

Status: -

Duration: 2 instruction cycles

Code: 1010aaaa----0010--cccccccccccccc

Description: Transfer the 24-bit value of the register specified by the A operand (A € OREG) together with the MHB register to the memory location address that is specified by the C operand (C € LIT16).

The 24-bit value of the register specified by A is stored in bits 0:23 of the referenced memory location; the MHB register is stored in bits 24:31 of the referenced memory location.

The PC program counter is incremented by 4.

13.2.2.5 MWR24 Instruction

Syntax: MWR24 A, C

Operation: MEM(C[13:0])[23:0] ← A

Status: -

Duration: 3 instruction cycles

Code: 1010aaaa----0111--ccccccccccccc

Description: Transfer the 24-bit value of the register specified by the A operand (A € OREG) to the memory location specified by the address in the C operand C (C € LIT16).

The 24-bit value of A is stored in bits 0:23 of the memory location and bits 24:31 are left unchanged.

The PC program counter is incremented by 4.

NOTE

MWR24 is not an atomic instruction.

13.2.2.6 MRDI Instruction

Syntax: MRDI A, B

Operation: A € MEM(B[13:0])[23:0]

MHB € MEM(B[13:0])[31:24]

Status: Z

Duration: 2 instruction cycles

Code: 1010aaaabbbb0011----------

Description: Using indirect addressing, transfer the lower 24 bits of a memory location to the register specified by the A operand (A € OREG). The upper 8 bits of the memory location are transferred to the MHB register.

The memory location is addressed by bits 0:13 of the register specified by the B operand (B € OREG).

The 24-bit read value is received from bits 0:23 of the memory location.

The STA[Z] zero bit is set if the lower 24 bits of the transferred value are zero. Otherwise, the Z bit is cleared.

If the MHB register is selected as a destination by the A operand (A € OREG), bits 0:7 of the referenced memory location are transferred to the MHB register.

The PC program counter is incremented by 4.

13.2.2.7 MWRI Instruction

Syntax: MWRI A, B

Operation: MEM(B[13:0])[23:0] € A;

MEM(B[13:0])[31:24] € MHB

Status: -

Duration: 2 instruction cycles

Code: 1010aaaabbbb0100----------

Description: Using indirect addressing, transfer the 24-bit value of the register specified by the A operand (A € OREG) together with the MHB register to a memory location.

The memory location is addressed by bits 0:13 of the register specified by the B operand (B € OREG).

The 24-bit value is stored in bits 0:23 of the memory location; the MHB register is stored in bits 24:31 of the memory location.

The PC program counter is incremented by 4.

13.2.2.8 MWRI24 Instruction

Syntax: MWRI24 A, B

Operation: MEM(B[13:0])[23:0] ← A;

Status: -

Duration: 3 instruction cycles

Code: 1010aaaabbbb1000----------

Description: Using indirect addressing, transfer the 24-bit value of the register specified by the A operand (A € OREG) to a memory location.

The memory location is addressed by bits 0:15 of the register specified by the B operand (B € OREG).

The 24-bit value is stored in bits 0:23 of the memory location and bits 24:31 are left unchanged.

The PC program counter is incremented by 4.

NOTE

MWRI24 is not an atomic instruction.

13.2.2.9 POP Instruction

Syntax: POP A

Operation: A ← MEM(R7[13:0])[23:0];

MHB ← MEM(R7[13:0])[31:24];

R7 ← R7 - 4;

SP_CNT ← SP_CNT - 1

Status: Z, EN

Duration: 2 instruction cycles

Code: 1010aaaa----0101----------

Description: Transfer bits 0:23 from the top of the stack to the register specified by the A operand (A € OREG). Then, decrement the stack pointer R7 register by 4. Transfer bits 24:31 from the top of the stack to the MHB register.

If the MHB register is selected as the destination by the A operand (A € OREG), bits 0:7 from the top of the stack are transferred to the MHB register.

The memory location of the top of the stack is addressed by bits 0:13 of the stack pointer R7 register.

The STA[Z] zero bit is set if the lower 24 bits of the transferred value are zero. Otherwise, the Z bit is cleared.

The program counter PC is incremented by 4.

The **MCS[i]_CHn_CTRL[SP_CNT]** bit field is decremented:

- If an underflow on the **SP_CNT** bit field occurs, the *STK_ERR[i]_IRQ* interrupt is asserted.
- If an underflow on the **SP_CNT** bit field occurs, the *STK_ERR[i]_IRQ* interrupt is asserted.
- If an underflow on the **SP_CNT** bit field occurs and the **MCS[i]_CTRL[HLT_SP_OFL]** bit is set, the current MCS channel is disabled by clearing the **STA[EN]** bit.

13.2.2.10 PUSH Instruction

Syntax: PUSH A

Operation: R7 ← R7 + 4;

SP_CNT ← SP_CNT + 1;

MEM(R7[13:0])[23:0] ← A;

MEM(R7[13:0])[31:24] ← MHB

Status: EN

Duration: 2 instruction cycles

Code: 1010aaaa----0110----------

Description: Increment the stack pointer R7 register by 4. Then, transfer the 24-bit value from the register specified by the A operand (A € OREG) together with the MHB register to the new top of the stack. The 24-bit value of A is stored in bits 0:23 of the memory location; the content of the MHB register stored in bits 24:31 of the memory location.

The memory location for the top of the stack is addressed by bits 0:13 of the stack pointer R7 register.

The program counter PC is incremented by 4.

The **MCS[i]_CHn_CTRL[SP_CNT]** bit field is incremented:

- If an overflow on the **SP_CNT** bit field occurs, the *STK_ERR[i]_IRQ* interrupt is asserted.
- If an overflow on the **SP_CNT** bit field occurs and the **MCS[i]_CTRL[HLT_SP_OFL]** bit is set, the current MCS channel is disabled by clearing the **STA[EN]** bit.
- If an overflow on the **SP_CNT** bit field occurs and the **MCS[i]_CTRL[HLT_SP_OFL]** bit is set, the memory write operations for the register specified by the A bit field, and the MHB register, are discarded (not performed).

13.2.3 ARU Instructions

13.2.3.1 ARD Instruction

Syntax: ARD A, B, C

Operation: A ← ARU(C[8:0])[23:0];

B ← ARU(C[8:0])[47:24];

ACB[4:0] ← ARU(C[8:0])[52:48]

Status: CAT

Duration: Suspends current MCS channel

Code: 1011aaaabbbb0000-------ccccccccc

Description: Perform a blocking read access to the ARU and transfer both 24-bit ARU values to the registers specified by the A operand (A € AREG) and the B operand (B € AREG), where the lower 24-bit ARU word goes to A and the upper 24-bit ARU word goes to B.

If A and B refer to the same register, only the upper 24-bit ARU word is stored in the register. The lower 24-bit ARU word is discarded.

If any transferred 24-bit value from the ARU is not to be stored in a register, the dummy ZERO register (€ AREG) can be selected by A or B. This action discards the corresponding ARU data. All address values of A and B that exceed the range 0:7 result in the discarding of the corresponding ARU data.

The received ARU control bits are stored in the ACB register.

The lower significant bits of the literal C operand (C € LIT16) specifies the ARU read address.

At the beginning of instruction execution, the STA[CAT] bit is always cleared. After execution of the instruction, the CAT bit is updated to show if the instruction finished successfully (CAT = 0) or if it was cancelled by the CPU (CAT = 1).

The PC program counter is incremented by 4.

13.2.3.2 ARDI Instruction

Syntax: ARDI A, B

Operation: A ← ARU(R6[8:0])[23:0];

B ← ARU(R6[8:0])[47:24];

ACB[4:0] ← ARU(R6[8:0])[52:48]

Status: CAT

Duration: Suspends current MCS channel

Code: 1011aaaabbbb0100----------

Description: Performs a blocking read access to the ARU and transfers both 24-bit ARU values to the registers specified by the A and B operands (A € AREG, B € AREG), where A stores the lower 24-bit ARU word and B stores the upper 24-bit ARU word.

If A and B refer to the same register, only the upper 24-bit ARU word is stored and the lower 24-bit ARU word is discarded.

If any transferred 24-bit ARU value is not to be stored in a register, the dummy ZERO register (€ AREG) can be selected by A or B to discard the corresponding ARU data. All address values specified by A and B that exceed the range 0:7 result in discarding of the corresponding ARU data.

The received ARU control bits are stored in the ACB register.

The ARU read address is specified by bits 0:8 of the channel's R6 register.

At the beginning of instruction execution, the STA[CAT] bit is always cleared. After execution of the instruction, the CAT bit is updated to show if the instruction finished successfully (CAT = 0) or if it was cancelled by the CPU (CAT = 1).

The PC program counter is incremented by 4.

13.2.3.3 AWR Instruction

Syntax: AWR A, B, C

Operation: ARU(C[4:0])[23:0] ← A;

ARU(C[4:0])[47:24] ← B;

ARU(C[4:0])[52:48] ← ACB[4:0];

Status: CAT

Duration: suspends current MCS-channel

Code: 1011aaaabbbb0001----------ccccc

Description: Perform a blocking write access to the ARU and transfer two 24 bit values to the ARU port using the registers A and B (A € OREG, B € OREG), whereas A holds the lower 24 bit ARU word and B holds the upper 24 bit ARU word.

The ARU control bits to be sent are taken from the register ACB.

The lower significant bits (bit 0 to bit 4) of the literal C (C € LIT16) define an index into the pool of ARU write address that is used for writing data.

Each MCS submodule has a pool of several write addresses that can be shared between all MCS-channels arbitrarily.

At the beginning of the instruction execution the CAT bit in the register STA is always cleared. After the execution of the instruction the CAT bit is updated in order to show if the instruction finished successfully (CAT = 0) or it was cancelled by the CPU (CAT = 1).

The program counter PC is incremented by the value 4.

13.2.3.4 AWRI Instruction

Syntax: AWRI A, B

Operation: ARU(R6[4:0])[23:0] ← A;

ARU(R6[4:0])[47:24] ← B;

ARU(R6[4:0])[52:48] ← ACB[4:0];

Status: CAT

Duration: Suspends current MCS channel

Code: 1011aaaabbbb0101----------

Description: Perform a blocking write access to the ARU and transfer two 24-bit ARU values from the registers specified by the A and B operands (A ← OREG, B ← OREG), where A stores the lower 24-bit ARU word and B stores the upper 24-bit ARU word.

The ARU control bits that are to be transferred are taken from the ACB register.

Bits 0:4 of the R6 register define an index into the pool of ARU write addresses that are used for writing data to the ARU.

Each MCS submodule has a pool of several write addresses that can be arbitrarily shared between all MCS channels.

At the beginning of instruction execution, the STA[CAT] bit is always cleared. After execution of the instruction, the CAT bit is updated to show if the instruction finished successfully (CAT = 0) or if it was cancelled by the CPU (CAT = 1).

The PC program counter is incremented by 4.

13.2.3.5 NARD Instruction

Syntax: NARD A, B, C

Operation: A ← ARU(C[8:0])[23:0];

B ← ARU(C[8:0])[47:24];

ACB[4:0] ← ARU(C[8:0])[52:48]

Status: SAT

Duration: Suspends current MCS channel for a maximum of one ARU round trip cycle

Code: 1011aaaabbbb0010-------ccccccccc

Description: Perform a non-blocking read access to the ARU to transfer both 24-bit ARU to the registers specified by the A and B operands (A € AREG, B € AREG), where A stores the lower 24-bit ARU word, B stores the upper 24-bit ARU word, and the ACB register stores the ARU control bits.

The lower significant bits of the literal C operand (C € LIT16) specify the ARU read address.

If the transfer finished successfully, the STA[SAT] bit is set and the transferred values are stored in the A, B, and ACB registers.

If the transfer failed due to missing data at the requested source, the SAT bit is cleared and the A, B, and ACB registers are not changed.

If A and B refer to the same register, only the upper 24-bit ARU word is stored and the lower 24-bit ARU word is discarded.

If any transferred 24-bit value from the ARU is not to be stored in a register, the dummy ZERO register (€ AREG) can be specified in A or B to discard the corresponding ARU data. All A and B address values that exceed the range 0:7 result in the discard of the corresponding ARU data.

The program counter PC is incremented by the value 4.

13.2.3.6 NARDI Instruction

Syntax: NARDI A, B

Operation: A ← ARU(R6[8:0])[23:0];

B ← ARU(R6[8:0])[47:24];

ACB[4:0] ← ARU(R6[8:0])[52:48]

Status: SAT

Duration: Suspends current MCS channel for a maximum of one ARU round trip cycle

Code: 1011aaaabbbb0011----------

Description: Perform a non-blocking read access to the ARU to transfer both 24-bit ARU values to the registers specified by the A and B operands (A € AREG, B € AREG), where A stores the lower 24-bit ARU word and B stores the upper 24-bit ARU word. The ARU control bits are stored in the ACB register.

The ARU read address is specified by bits 0:8 of the channel's R6 register.

If the transfer finished successfully, the STA[SAT] bit is set and the transferred values are stored in the A, B, and ACB registers.

If the transfer failed due to missing data at the requested source, the SAT bit is cleared and the A, B, and ACB registers are not changed.

If A and B refer to the same register, only the upper 24-bit ARU word is stored and the lower 24-bit ARU word is discarded.

If any transferred 24-bit value from the ARU is not to be stored in a register, the dummy ZERO register (€ AREG) can be specified in A or B to discard the corresponding ARU data. All address values in A and B that exceed the range 0:7 result in the discard the corresponding ARU data.

The PC program counter is incremented by 4.

13.2.4 Arithmetic Logic Instructions

13.2.4.1 ADDL Instruction

Syntax: ADDL A, C

Operation: A ← A + C

Status: Z, CY, N, V

Duration: 1 instruction cycle

Code: 0010aaaaccccccccccccccccccccccc

Description: Add the value in the register specified by the A operand (A € OREG) with the 24-bit literal value in the C operand (C € LIT24), and store the result in A.

The STA[Z] zero bit is set if the calculated value is zero. Otherwise the Z bit is cleared.

The STA[CY] carry bit is set if an unsigned overflow/underflow occurred during the addition. Otherwise, the CY bit is cleared. An unsigned overflow has occurred when the result of the addition cannot be represented in the range [0; 2^{24}-1], assuming that the A and C operands are unsigned values within the range [0; 2^{24}-1].

The STA[V] overflow bit is set, if a signed overflow/underflow occurred during the addition. Otherwise, the V bit is cleared. A signed overflow/underflow has occurred when the result of the addition cannot be represented in the range [-2^{23}; 2^{23}-1], assuming that the A and C operands are signed values within the range [-2^{23}; 2^{23}-1].

The STA[N] negative bit equals the most significant bit of the results of the addition. The N bit can be used to determine if a calculated signed result is negative (N=1) or positive (N=0), assuming that no overflow/underflow occurred.

The PC program counter is incremented by 4.

13.2.4.2 ADD Instruction

Syntax: ADD A, B

Operation: A ← A + B

Status: Z, CY, N, V

Duration: 1 instruction cycle

Code: 1100aaaabbbb0000----------

Description: Add the value in the register specified by the A operand (A € OREG) to the value in the register specified by the B operand (B € OREG) and store the result in A.

The STA[Z] zero bit is set if the calculated value is zero. Otherwise the Z bit is cleared.

The STA[CY] carry bit is set if an unsigned overflow occurred during the addition. Otherwise, the CY bit is cleared. An unsigned overflow has occurred when the result of the addition cannot be represented in the range [0; 2^{24}-1], assuming that A and B are unsigned values within the range [0; 2^{24}-1].

The STA[V] overflow bit is set if a signed overflow/underflow occurred during the addition. Otherwise, the V bit is cleared. A signed overflow/underflow has occurred when the result of the addition cannot be represented in the range [-2^{23}; 2^{23}-1], assuming that A and B are signed values within the range [-2^{23}; 2^{23}-1].

Th STA[N] negative bit equals the most significant bit of the result of the addition. The N bit is used to determine if a calculated signed result is negative (N=1) or positive (N=0), assuming that no overflow/underflow occurred.

The PC program counter is incremented by 4.

13.2.4.3 SUBL Instruction

Syntax: SUBL A, C

Operation: A ← A - C

Status: Z, CY, N, V

Duration: 1 instruction cycle

Code: 0011aaaaccccccccccccccccccccccc

Description: Subtract the 24-bit literal value in the C operand (C € LIT24) from the value in the register specified by the A operand (A € OREG) and store the result in A.

The STA[Z] zero bit is set if the calculated value is zero. Otherwise, the Z bit is cleared.

The STA[CY] carry bit is set if an unsigned underflow occurred during the subtraction. Otherwise, the CY bit is cleared. An unsigned underflow has occurred when the result of the subtraction cannot be represented in the range [0; 2^{24}-1], assuming that A and C are unsigned values within the range [0; 2^{24}-1].

The STA[V] overflow bit is set if a signed overflow/underflow occurs during the subtraction. Otherwise, the V bit is cleared. A signed overflow/underflow has occurred when the result of the subtraction cannot be represented in the range [-2^{23}; 2^{23}-1], assuming that A and C are signed values in the range [-2^{23}; 2^{23}-1].

The STA[N] negative bit equals the most significant bit of the result of the subtraction. The N bit is used to determine if a calculated signed result is negative (N=1) or positive (N=0), assuming that no overflow/underflow occurred.

The PC program counter is incremented by 4.

13.2.4.4 SUB Instruction

Syntax: SUB A, B

Operation: A ← A - B

Status: Z, CY, N, V

Duration: 1 instruction cycle

Code: 1100aaaabbbb0001----------

Description: Subtract the value in the register specified by the B operand (B € OREG) from the value in the register specified by the A operand (A € OREG) and store the result in A.

The STA[Z] zero bit is set if the calculated value is zero. Otherwise the Z bit is cleared.

The STA[CY] carry bit is set if an unsigned underflow occurs during subtraction. Otherwise, the CY bit is cleared. An unsigned underflow has occurred when the result of the subtraction cannot be represented in the range [0; 2^{24}-1], assuming that A and B are unsigned values in the range [0; 2^{24}-1].

The STA[V] overflow bit is set if a signed overflow/underflow occurs during subtraction. Otherwise, the V bit is cleared. A signed overflow/underflow has occurred when the result of the subtraction cannot be represented in the range [-2^{23}; 2^{23}-1], assuming that A and B are signed values in the range [-2^{23}; 2^{23}-1].

The STA[N] negative bit equals the most significant bit of the result of the subtraction. The N bit is used to determine if a calculated signed result is negative (N=1) or positive (N=0), assuming that no overflow/underflow occurred.

The PC program counter is incremented by 4.

13.2.4.5 NEG Instruction

Syntax: NEG A, B

Operation: A ← - B

Status: Z, N, V

Duration: 1 instruction cycle

Code: 1100aaaabbbb0010----------

Description: Negate (2's Complement) the value in the register specified by the B operand (B € OREG) and store the result in the register specified by the A operand (A € OREG).

The STA[Z] zero bit is set if the negated value is zero. Otherwise, the Z bit is cleared.

The STA[V] overflow bit is set if a signed overflow/underflow occurs during the negation. Otherwise the V bit is cleared. A signed overflow/underflow has occurred when the result of the negation cannot be represented in the range [-2^{23}; 2^{23}-1], assuming that A and B are signed values in the range [-2^{23}; 2^{23}-1].

The STA[N] negative bit equals the most significant bit of the result of the negation. The N bit is used to determine if a calculated signed result is negative (N=1) or positive (N=0), assuming that no overflow/underflow occurred.

The PC program counter is incremented by 4.

13.2.4.6 ANDL Instruction

Syntax: ANDL A, C

Operation: A ← A AND C

Status: Z

Duration: 1 instruction cycle

Code: 0100aaaaccccccccccccccccccccccc

Description: Perform a bitwise AND conjunction of the value in the register specified by the A operand (A € OREG) with the 24-bit literal value in the C operand (C € LIT24) and store the result in A.

The STA[Z] zero bit is set if the calculated value is zero. Otherwise, the Z bit is cleared.

The PC program counter is incremented by 4.

13.2.4.7 AND Instruction

Syntax: AND A, B

Operation: A ← A AND B

Status: Z

Duration: 1 instruction cycle

Code: 1100aaaabbbb0011----------

Description: Perform a bitwise AND conjunction of the value in the register specified by the A operand (A € OREG) with the value in the register specified by the Boperand (B € OREG). Store the result in A.

The STA[Z] zero bit is set if the conjuncted value is zero. Otherwise, the Z bit is cleared.

The PC program counter is incremented by 4.

13.2.4.8 ORL Instruction

Syntax: ORL A, C

Operation: A ← A OR C

Status: Z

Duration: 1 instruction cycle

Code: 0101aaaaccccccccccccccccccccccc

Description: Perform a bitwise OR conjunction of the value in the register specified by the A operand (A € OREG) with the 24-bit literal value in the C operand (C € LIT24) and store the result in A.

The STA[Z] zero bit is set if the calculated value is zero. Otherwise, the Z bit is cleared.

The PC program counter is incremented by 4.

13.2.4.9 OR Instruction

Syntax: OR A, B

Operation: A ← A OR B

Status: Z

Duration: 1 instruction cycle

Code: 1100aaaabbbb0100----------

Description: Perform a bitwise OR conjunction of the value in the register specified by the A operand (A € OREG) with the value in the register specified by the B operand (B € OREG) and store the result in A.

The STA[Z] zero bit is set if the calculated value is zero. Otherwise, the Z bit is cleared.

The PC program counter is incremented by 4.

13.2.4.10 XORL Instruction

Syntax: XORL A, C

Operation: A ← A XOR C

Status: Z

Duration: 1 instruction cycle

Code: 0110aaaaccccccccccccccccccccccc

Description: Perform a bitwise XOR conjunction of the value in the register specified by the A operand (A € OREG) with the 24-bit literal value in the C operand (C € LIT24) and store the result in A.

The STA[Z] zero bit is set if the calculated value is zero. Otherwise, the Z bit is cleared.

The PC program counter is incremented by 4.

13.2.4.11 XOR Instruction

Syntax: XOR A, B

Operation: A ← A XOR B

Status: Z

Duration: 1 instruction cycle

Code: 1100aaaabbbb0101----------

Description: Perform a bitwise XOR conjunction of the value in the register specified by the A operand (A € OREG) with the value in the register specified by the B operand (B € OREG) and store the result in A.

The STA[Z] zero bit is set if the calculated value is zero. Otherwise, the Z bit is cleared.

The PC program counter is incremented by 4.

13.2.4.12 SHR Instruction

Syntax: SHR A, C

Operation: A ← A >> C

Status: Z, CY

Duration: 1 instruction cycle

Code: 1100aaaa----0110----------ccccc

Description: Perform a right shift operation on the value in the register specified by the A operand (A € OREG). The shift count is specified by the 16-bit literal value in the C operand(C € LIT16), where C must be in the range [0:24]. Zeros are shifted into the most significant bit of A.

The STA[Z] zero bit is set if the resulting value is zero. Otherwise, the Zero bit is cleared.

The STA[CY] carry bit is updated to equal the last bit (LSB) that is shifted out of A.

The PC program counter is incremented by 4.

13.2.4.13 SHL Instruction

Syntax: SHL A, C

Operation: A ← A << C

Status: Z, CY

Duration: 1 instruction cycle

Code: 1100aaaa----0111----------ccccc

Description: Perform a left shift on the value in the register specified by the A operand (A € OREG). The shift count is specified by the value in the C operand (C € LIT16), where C must be in the range 0:24. Zeros are shifted into the LSB of A.

The STA[Z] zero bit is set if the shifted value is zero. Otherwise the Z bit is cleared.

The STA[CY] carry bit is equal to the last MSB that is shifted out of A.

The PC program counter is incremented by 4.

13.2.5 Test Instructions

13.2.5.1 ATUL Instruction

Syntax: ATUL A, C

Operation: A – C

Status: Z, CY

Duration: 1 instruction cycle

Code: 0111aaaaccccccccccccccccccccccc

Description: Perform an arithmetic test of the unsigned value in the register specified by the A operand (A € OREG) versus the unsigned 24-bit literal value in the C operand (C € LIT24).

The STA[CY] carry bit is set if unsigned A is < unsigned C. The CY bit is cleared if unsigned A is ≥ unsigned C.

The STA[Z] zero bit is set if A = C. The Z bit is cleared if A ≠ C.

The PC program counter is incremented by 4.

13.2.5.2 ATU Instruction

Syntax: ATU A, B

Operation: A – B

Status: Z, CY

Duration: 1 instruction cycle

Code: 1101aaaabbbb0000----------

Description: Perform an arithmetic test on the unsigned value in the register specified by the A operand (A € OREG) versus the unsigned value in the register specified by the B operand (B € OREG).

The STA[CY] carry bit is set if unsigned A is < unsigned B. The CY bit is cleared if unsigned A is ≥ unsigned B.

The STA[Z] zero bit is set if unsigned A = unsigned B.

The Z bit is cleared if unsigned A is ≠ unsigned B.

The PC program counter is incremented by 4.

13.2.5.3 ATSL Instruction

Syntax: ATSL A, C

Operation: A – C

Status: Z, CY

Duration: 1 instruction cycle

Code: 1000aaaaccccccccccccccccccccccc

Description: Perform an arithmetic test on the signed value in the register specified by the A operand (A € OREG) versus the signed 24-bit literal value in the C operand (C € LIT24).

The STA[CY] carry bit is set if signed A is < signed literal C. The CY bit is cleared if signed A is ≥ signed C.

The STA[Z] zero bit is set if signed A = signed C.

The Z bit is cleared if signed A is ≠ signed C.

The PC program counter is incremented by 4.

13.2.5.4 ATS Instruction

Syntax: ATS A, B

Operation: A – B

Status: Z, CY

Duration: 1 instruction cycle

Code: 1101aaaabbbb0001----------

Description: Perform an arithmetic test on the signed value in the register specified by the A operand (A € OREG) versus the signed value in the register specified by the B operand (B € OREG).

The STA[CY] carry bit is set if signed A is < signed B. The CY bit is cleared if signed A is ≥ signed B.

The STA[Z] zero bit is set if signed A = signed B.

The Z bit is cleared if signed A is ≠ to signed B.

The PC program counter is incremented by 4.

13.2.5.5 BTL Instruction

Syntax: BTL A, C

Operation: A AND C

Status: Z

Duration: 1 instruction cycle

Code: 1001aaaaccccccccccccccccccccccc

Description: Perform a bit test on the value in the register specified by the A operand (A € OREG) versus the 24-bit literal bit mask in the C operand (C € LIT24).

The bit test is performed by applying a bitwise logical AND operation on A and C without storing the result.

The STA[Z] zero bit is set if the resulting value is zero. Otherwise, the Z bit is cleared.

The PC program counter is incremented by 4.

13.2.5.6 BT Instruction

Syntax: BT A, B

Operation: A AND B

Status: Z

Duration: 1 instruction cycle

Code: 1101aaaabbbb0010----------

Description: Perform a test on the value in the register specified by the A operand (A € OREG) versus the value in the register specified by the B operand (B € OREG), where the value in one of the registers is usually a bit mask.

The bit test is performed by a bitwise logical AND operation on A and B without storing the result.

The STA[Z] zero bit is set, if the resulting value is zero. Otherwise, the Z bit is cleared.

The PC program counter is incremented by 4.

13.2.6 Control Flow Instructions

13.2.6.1 JMP Instruction

Syntax: JMP C

Operation: PC ← C[13:0]

Status: -

Duration: 1 instruction cycle

Code: 1110--------0000--ccccccccccccc

Description: Execute an unconditional jump to the memory location specified by the 16-bit literal value in the C operand (C € LIT16).

The PC program counter is loaded with the 16-bit literal value in the C operand.

13.2.6.2 JBS Instruction

Syntax: JBS A, B, C

Operation: PC ← C[13:0] if A[B] is set

Status : -

Duration : 1 instruction cycle

Code : 1110aaaabbbb0001--ccccccccccccc

Description: Execute a conditional jump to the memory location specified by the 16-bit literal value in the C operand (C € LIT16).

The PC program counter is loaded with the value in C if the bit at the position (specified in the 4-bit literal B operand (B € LIT4)) in the register (specified by the A operand (A € OREG)) is set. If the bit is cleared, the PC program counter PC is incremented by 4.

13.2.6.3 JBC Instruction

Syntax: JBC A, B, C

Operation: PC ← C[13:0] if A[B] is cleared

Status : -

Duration : 1 instruction cycle

Code : 1110aaaabbbb0010--ccccccccccccc

Description: Execute a conditional jump to the memory location specifed by the 16-bit literal value in the C operand (C € LIT16).

The PC program counter is loaded with C if the bit at the position (specified by the 4-bit literal value in the B operand (B € LIT4)) in the register (specified by the A operand (A € OREG)) is cleared. If the bit is set, the PC program counter is incremented by 4.

13.2.6.4 CALL Instruction

Syntax: CALL C

Operation: R7 ← R7 + 4;

MEM(R7[15:0])[15:0] ← PC + 4;

PC ← C[13:0];

SP_CNT ← SP_CNT + 1

Status: EN

Duration: 2 instruction cycles

Code: 1110--------0011--ccccccccccccc

Description: Call the subprogram at the memory location specified by the 16-bit literal value in the C operand (C € LIT16).

The stack pointer R7 register is incremented by 4.

The memory location for the top of the stack is identified by bits 0:15 of the stack pointer R7 register.

After the stack pointer is incremented, the incremented value of the PC is written to the top of the stack.

The PC program counter is loaded the 16-bit literal value in the C.

The **MCS[i]_CHn_CTR[SP_CNTL]** bit field is incremented.

- If an overflow of the **SP_CNT** bit field occurs, the *STK_ERR[i]_IRQ* interrupt is asserted.
- If an overflow of the **SP_CNT** bit field occurs and the **MCS[i]_CTRL[HLT_SP_OFL]** bit is set, the current MCS channel is disabled by clearing the **STA[EN]** bit.
- If an overflow on the **SP_CNT** bit field occurs and the **MCS[i]_CTRL[HLT_SP_OFL]** bit is set, the operation to write the incremented PC value to the top of the stack is discarded.

13.2.6.5 RET Instruction

Syntax: RET

Operation: PC ← MEM(R7[15:0])[15:0];

R7 ← R7 – 4;

SP_CNT ← SP_CNT – 1

Status: EN

Duration: 2 instruction cycles

Code: 1110--------0100----------

Description: Return from subprogram.

The PC program counter is loaded with current value of the top of the stack.

The stack pointer R7 register is decremented by 4.

The memory location for the top of the stack is specified by bits 0:15 of the stack pointer R7 register.

The **MCS[i]_CHn_CTRL[SP_CNT]** bit field is decremented.

- If an underflow of the **SP_CNT** bit field occurs, the *STK_ERR[i]_IRQ* interrupt is asserted.
- If an underflow of the **SP_CNT** bit field occurs and the **MCS[i]_CTRL[HLT_SP_OFL]** bit is set, the current MCS channel is disabled by clearing the **STA[EN]** bit.

13.2.7 Other Instructions

13.2.7.1 WURM Instruction

Syntax: WURM A B C

Operation: Wait until register match

Status: CWT

Duration: Suspends the current MCS channel

Code: 1111aaaabbbb0000ccccccccccccccc

Description: Suspend the current MCS channel until the following target condition occurs:

A = (B AND MASK)

where a bitwise AND operation is performed on the value in the register specified by the B operand (B € OREG) with the MASK value. Bits 16:23 of the MASK are set to ones and bits 0:15 of the MASK are specified by the 16-bit literal value in the C operand (C € LIT16). The target condition is written to the register specified by the A operand (A € OREG).

If the target condition is true at the beginning of the WURM instruction's execution, the instruction does not suspend the channel and the PC program counter is incremented by 4.

This WURM instruction can be used to wait for one or more trigger events that are generated by other MCS channels or by the CPU, where:

- B points to the STRG trigger register,
- A points to a general purpose register that contains the target condition, and
- C is the bit mask that specifies the trigger bits that are to be included in the 'match'.

The trigger bits can be set either by other MCS channels with a write access (e.g. using a MOVL instruction) to the **STRG** register or by the CPU with a write access to the **MCS[i]_STRG** register.

The trigger bits are not automatically cleared by hardware after the resumption of MCS channel activity. Rather, they have to be explicitly cleared with either an MCS channel write access to the **CTRG** register or a CPU write access to the **MCS[i]_CTRG** register.

NOTE

More than one suspended MCS channel can wait for the same trigger bit(s) to cause resumption of operation.

The WURM instruction can also be used to wait for a specific time/angle event (that is provided by the TBU submodule) to occur, where:

- B points to the TBU_TS0, TBU_TS1, or TBU_TS2 register,
- A points to a general purpose register that shows the target trigger condition, and
- the 16-bit literal value in C should be set to 0xFFFF.

At the beginning of the WURM instruction's execution, the STA[CWT] bit is always cleared. After execution of the instruction, the CWT bit is updated to show if the instruction finished successfully (CWT = 0) or if it was cancelled by the CPU (CWT = 1).

If the CWT bit is set simultaneously with the occurrence of the register match condition, the register match condition has the higher priority, which results in a cleared CWT bit.

When the target condition is reached, the PC program counter is incremented by 4 and the channel resumes operation.

13.2.7.2 NOP Instruction

Syntax: NOP

Operation: -

Status: -

Duration: 1 instruction cycle

Code: 0000----------

Description: No operation is performed.

The PC program counter is incremented by 4.

13.3 MCS Internal Registers

This section describes MCS internal registers which are partly accessible only by the corresponding MCS-channel itself and partly global to all channels. Please see Table 13.3.1 for clarification.

These registers can be directly accessed with the entire MCS instruction set, e.g. using the ORL instruction to set a specific bit.

13.3.1 Memory Map and Registers

memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
8	MCS[i] Status Register (STA)	32	R/W	0000_0000h	13.3.1.1/424
C	MCS[i] TBU Timestamp Register 0 (TBU_TS0)	32	R	See section	13.3.1.2/ 428
D	MCS[i] TBU Timestamp Register 1 (TBU_TS1)	32	R	See section	13.3.1.3/ 429

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
E	MCS[i] TBU Timestamp Register 2 (TBU_TS2)	32	R	See section	13.3.1.4/ 429
F	MCS[i] Memory High Byte Register (MHB)	32	R/W	0000_0000h	13.3.1.5/ 430
3_0000	MCS[i] General Purpose Register (GP_R0)	32	R/W	0000_0000h	13.3.1.6/ 430
3_0004	MCS[i] General Purpose Register (GP_R1)	32	R/W	0000_0000h	13.3.1.6/ 430
3_0008	MCS[i] General Purpose Register (GP_R2)	32	R/W	0000_0000h	13.3.1.6/ 430
3_000C	MCS[i] General Purpose Register (GP_R3)	32	R/W	0000_0000h	13.3.1.6/ 430
3_0010	MCS[i] General Purpose Register (GP_R4)	32	R/W	0000_0000h	13.3.1.6/ 430
3_0014	MCS[i] General Purpose Register (GP_R5)	32	R/W	0000_0000h	13.3.1.6/ 430
3_0018	MCS[i] General Purpose Register (GP_R6)	32	R/W	0000_0000h	13.3.1.6/ 430
3_001C	MCS[i] General Purpose Register (GP_R7)	32	R/W	0000_0000h	13.3.1.6/ 430
3_0024	MCS[i] ARU Control Bit Register (ACB)	32	R/W	0001_FE00h	13.3.1.7/ 431
3_0028	MCS[i] Clear Trigger Bits Register (CTRG)	32	R/W	0000_0000h	13.3.1.8/ 432
3_002C	MCS[i] Set Trigger Bits Register (STRG)	32	R/W	0000_0000h	13.3.1.9/ 433

13.3.1.1 MCS[i] Status Register (STA)

NOTE

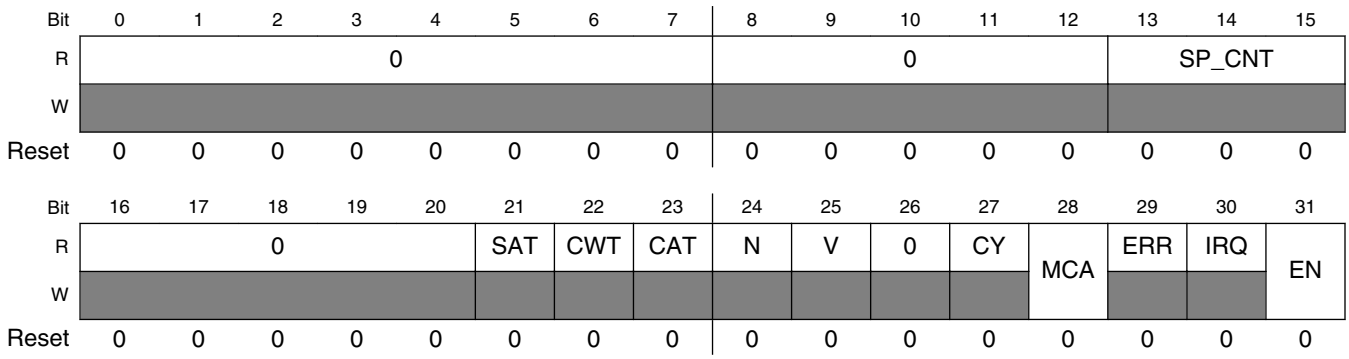
If both, the CPU and the MCS-channel are writing to the same general purpose register at the clock cycle, the value of the CPU is written to the register and the value of the MCS-channel is discarded.

NOTE

Writing to bits of the register STA with instructions that do implicitely a read-modify-write operation (e.g. "ANDL STA, 0xFFFFFE" or "OR STA, R0") is dangerous, since writing back the possibly modified content of the read access (which reflects status information) may cause undesireable results. A secure way for writing to bits of the register STA is to use instructions

that do not read the content of STA (e.g. "MOVL STA, 0x0 or MOV STA, R1").

Address: 0h base + 8h offset = 8h



STA field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–15 SP_CNT	Stack pointer counter value. Actual stack depth of channel. The bit field is incremented on behalf of a CALL or PUSH instruction and decremented on behalf of a RET or POP instruction. The MCS channel STK_ERR_IRQ is raised, when an overflow or underflow is detected on this bit field.
16–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 SAT	Successful ARU transfer bit. 0 Non-blocking ARU transfer failed due to missing data. 1 Non-blocking ARU transfer finished successfully.
22 CWT	Cancel WURM instruction bit. This bit is updated after each WURM instruction, and it should be evaluated immediately after the WURM instruction. Otherwise, the CPU could set the bit leading to bad status information in the MCS program. 0 Last WURM instruction was not cancelled. 1 CPU cancelled last WURM instruction of channel.
23 CAT	Cancel ARU transfer bit. This bit is updated after each ARU transfer, and it should be evaluated immediately after the ARU instruction. Otherwise, the CPU could set the bit leading bad status information in the MCS program. 0 Last ARU transfer was not cancelled. 1 CPU cancelled last ARU transfer.
24 N	Negative bit. The negative bit is updated by arithmetic instructions in order to indicate a negative result.
25 V	Overflow bit.

Table continues on the next page...

STA field descriptions (continued)

Field	Description
	The overflow bit is updated by arithmetic instructions in order to indicate a signed under/overflow.
26 Reserved	Zero bit. This read-only field is reserved and always has the value 0.
27 CY	Carry bit. The carry bit is updated by several arithmetic and logic instructions. In arithmetic operations, the carry bit indicates an unsigned under/overflow.
28 MCA	MON Activity signalling for MCS channel. When this bit is set, the corresponding channel in the MON submodule register MON_ACTIVITY is set (see section 20.8.2). This bit is automatically cleared after writing to it by the MCS channel program. 0 No activity signalled to submodule MON. 1 Activity signalled to submodule MON.
29 ERR	Set Error Signal. The ERR signal of an MCS-channel reflects Error status that may be caused by one of the following conditions: • MCS-channel program sets the ERR signal by writing to this bit • ECC RAM Error occurred while accessing the connected RAM pages (also disables MCS-channel by clearing bit EN) • Decoding an instruction with an invalid opcode (also disables MCS-channel by clearing bit EN) If the GTM includes a MON submodule, the ERR signal is always catured by this module. An MCS-channel releases an error signal by writing value 1 to the ERR bit. Writing a value 0 to this bit does not cancel the error signal, and thus has no effect. An MCS-channel can read the ERR bit in order to determine the current state of the error signal evaluated by the module MON. The MCS-channel reads a value 1 if an ERR occurred previously, but not cleared by CPU. If an MCS-channel reads a value 0, no error was set or it has been cleared by CPU. The ERR bit can only be cleared by the CPU writing a 1 to the MCS[i]_ERR register (see section 13.4.15). 0 No Error occurred. 1 Error occurred.
30 IRQ	Release IRQ. An MCS-channel releases an IRQ by writing value 1 to bit IRQ. Writing a value 0 to this bit does not cancel the IRQ, and thus has no effect. An MCS-channel can read the IRQ bit in order to determine the current state of the IRQ handling. The MCS-channel reads a value 1 if an IRQ was released but not cleared by CPU. If an MCS-channel reads a value 0, no IRQ was released or it has been cleared by CPU. The IRQ bit can only be cleared by CPU, by writing a 1 to the corresponding MCS[i]_CH[x]_NOTIFY register (see section 13.4.7). 0 No triggered IRQ signal. 1 Trigger IRQ signal.
31 EN	Enable current MCS-channel. 0 Disable current MCS-channel. 1 Enable current MCS-channel.

13.3.1.2 MCS[i] TBU Timestamp Register 0 (TBU_TS0)

NOTE

Any write access to a time base register discards the written data. A write access to a time base register may be used to destroy an unused 24-bit data word of an ARU read transfer.

Reset value for bit[0:31] is undefined.

Address: 0h base + Ch offset = Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								TS0																							
W																																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- Reset value for bit[0:31] is undefined.

TBU_TS0 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 TS0	Current TBU time stamp 0.

13.3.1.3 MCS[i] TBU Timestamp Register 1 (TBU_TS1)

NOTE

Any write access to a time base register discards the written data. A write access to a time base register may be used to destroy an unused 24-bit data word of an ARU read transfer.

Reset value for bit[0:31] is undefined.

Address: 0h base + Dh offset = Dh

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								TS1																							
W																																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- Reset value for bit[0:31] is undefined.

TBU_TS1 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 TS1	Current TBU time stamp 1.

13.3.1.4 MCS[i] TBU Timestamp Register 2 (TBU_TS2)

NOTE

Reset value for bit[0:31] is undefined.

Address: 0h base + Eh offset = Eh

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								TS2																							
W																																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

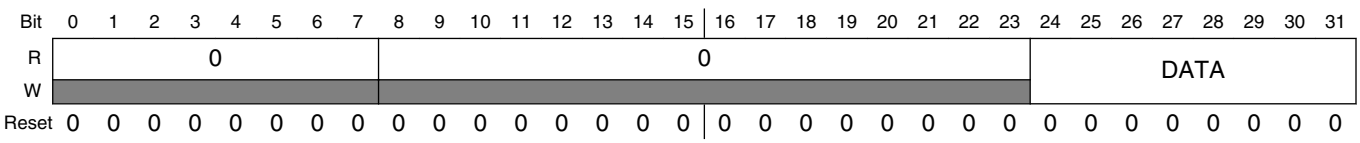
* Notes:
- Reset value for bit[0:31] is undefined.

TBU_TS2 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 TS2	Current TBU time stamp 2.

13.3.1.5 MCS[i] Memory High Byte Register (MHB)

Address: 0h base + Fh offset = Fh



MHB field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24–31 DATA	High Byte of a memory transfer.

13.3.1.6 MCS[i] General Purpose Register (GP_R*n*)

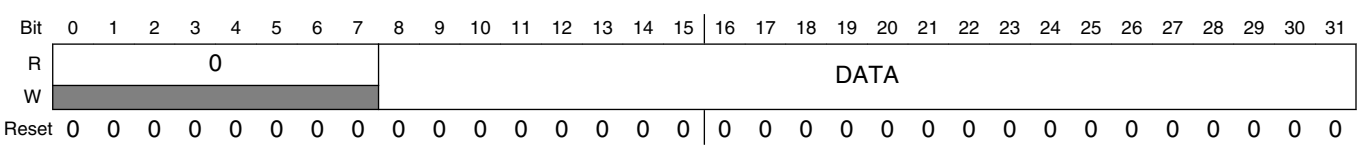
NOTE

Register R6 is also used as an index/address register for indirect ARU addressing instructions.

NOTE

Register R7 is also used as a stack pointer register, if stack operations are used in the MCS micro program.

Address: 0h base + 3_0000h offset + (4d × i), where i=0d to 7d



GP_R*n* field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 DATA	Data field of general purpose register. **NOTE:** Register R7 is also used as stack pointer register, if stack operations are used in the MCS micro program. **NOTE:** Register R6 is also used as index/address register for indirect ARU addressing instructions.

13.3.1.7 MCS[i] ARU Control Bit Register (ACB)

Address: 0h base + 3_0024h offset = 3_0024h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0								0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0											ACB4	ACB3	ACB2	ACB1	ACB0
W																
Reset	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0

ACB field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 ACB4	ARU Control bit 4. This bit is updated by each ARU read access and its value is sent to ARU by each ARU write access on bit[52] of the ARU word.
28 ACB3	ARU Control bit 3. This bit is updated by each ARU read access and its value is sent to ARU by each ARU write access on bit[51] of the ARU word.
29 ACB2	ARU Control bit 2. This bit is updated by each ARU read access and its value is sent to ARU by each ARU write access on bit[50] of the ARU word.

Table continues on the next page...

ACB field descriptions (continued)

Field	Description
30 ACB1	ARU Control bit 1. This bit is updated by each ARU read access and its value is sent to ARU by each ARU write access on bit[49] of the ARU word.
31 ACB0	ARU Control bit 0. This bit is updated by each ARU read access and its value is sent to ARU by each ARU write access on bit[48] of the ARU word.

13.3.1.8 MCS[i] Clear Trigger Bits Register (CTRG)

A write access to MCS[i]_CTRG may take up to 9 clock cycles, since the write access is scheduled to the next CPU time slot determined by the MCS scheduler.

Address: 0h base + 3_0028h offset = 3_0028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0								TRG															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

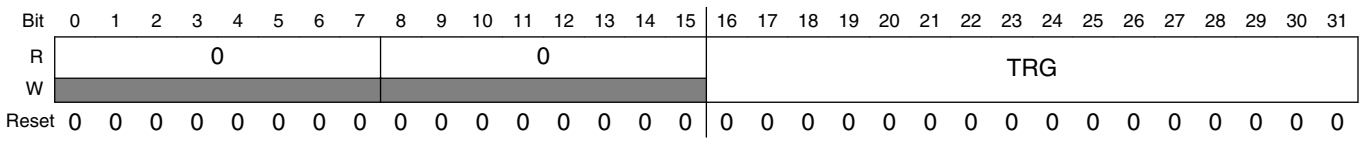
CTRG field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TRG	Trigger bit[15:0]. Trigger bits can be used for signalizing specific events to MCS-channels or the CPU. An MCS-channel suspended with a WURM instruction can be resumed by setting the appropriate trigger bit. **NOTE:** The trigger bits TRGx (x = 0..15) are accessible by all MCS channels as well as the CPU. Setting a trigger bit can be performed with the STRG register, in the case of an MCS-channel or the MCS[i]_STRG register in the case of the CPU. Clearing a trigger bit can be performed with the CTRG register, in the case of an MCS-channel or the MCS[i]_CTRG register in the case of the CPU. 0 READ: trigger bit is cleared / WRITE : do nothing 1 READ: trigger bit is set / WRITE : clear trigger bit.

13.3.1.9 MCS[i] Set Trigger Bits Register (STRG)

A write access to MCS[i]_STRG may take up to 9 clock cycles, since the write access is scheduled to the next CPU time slot determined by the MCS scheduler.

Address: 0h base + 3_002Ch offset = 3_002Ch



STRG field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–31 TRG	Trigger bit[15:0]. The trigger bits TRGn (n = 0..15) are accessible by all MCS channels as well as the CPU. Setting a trigger bit can be performed with the STRG register, in the case of an MCS-channel or the MCS[i]_STRG register in the case of the CPU. Clearing a trigger bit can be performed with the CTRG register, in the case of an MCS-channel or the MCS[i]_CTRG register in the case of the CPU. Trigger bits can be used for signalizing specific events to MCS-channels or the CPU. An MCS-channel suspended with a WURM instruction can be resumed by setting the appropriate trigger bit. 0 READ: trigger bit is cleared / WRITE : do nothing. 1 READ: trigger bit is set / WRITE : set trigger bit.

13.4 MCS Configuration Registers

This section describes the configuration registers of the MCS submodule.

These registers can only be accessed by the CPU using AEI, but not within the MCS-channel using MCS instructions.

13.4.1 Memory Map and Registers

The Multi Channel Sequencer (MCS0) module registers are described as follows.

These registers can only be accessed by the CPU using AEI, but not within the MCS-channel using MCS instructions.

MCS_0 memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	MCS0 Channel n General Purpose 0 Register, n=0:8 (MCS_0_CH0_R0)	32	R/W	0000_0000h	13.4.1.1/ 440
4	MCS0 Channel n General Purpose 1 Register, n=0:8 (MCS_0_CH0_R1)	32	R/W	0000_0000h	13.4.1.2/ 440
8	MCS0 Channel n General Purpose 2 Register, n=0:8 (MCS_0_CH0_R2)	32	R/W	0000_0000h	13.4.1.3/ 441
C	MCS0 Channel n General Purpose 3 Register, n=0:8 (MCS_0_CH0_R3)	32	R/W	0000_0000h	13.4.1.4/ 441
10	MCS0 Channel n General Purpose 4 Register, n=0:8 (MCS_0_CH0_R4)	32	R/W	0000_0000h	13.4.1.5/ 442
14	MCS0 Channel n General Purpose 5 Register, n=0:8 (MCS_0_CH0_R5)	32	R/W	0000_0000h	13.4.1.6/ 442
18	MCS0 Channel n General Purpose 6 Register, n=0:8 (MCS_0_CH0_R6)	32	R/W	0000_0000h	13.4.1.7/ 443
1C	MCS0 Channel n General Purpose 7 Register, n=0:8 (MCS_0_CH0_R7)	32	R/W	0000_0000h	13.4.1.8/ 443
20	MCS0 Channel n Control Register, n=0:8 (MCS_0_CH0_CTRL)	32	R/W	0000_0000h	13.4.1.9/ 444
24	MCS0 Channel n ARU Control Bit Register, n=0:8 (MCS_0_CH0_ACB)	32	R	0000_0000h	13.4.1.10/ 446
28	MCS0 Clear Trigger Control Register (MCS_0_CTRG)	32	R/W	0000_0000h	13.4.1.11/ 447
2C	MCS0 Set Trigger Control Register (MCS_0_STRG)	32	R/W	0000_0000h	13.4.1.12/ 448
40	MCS0 Channel n Program Counter Register, n=0:8 (MCS_0_CH0_PC)	32	R/W	0000_0000h	13.4.1.13/ 450
44	MCS0 Channel n Interrupt Request Notification Register, n=0:8 (MCS_0_CH0_IRQ_NOTIFY)	32	w1c	0000_0000h	13.4.1.14/ 451
48	MCS0 Channel n Interrupt Request Enable Register, n=0:8 (MCS_0_CH0_IRQ_EN)	32	R/W	0000_0000h	13.4.1.15/ 452
4C	MCS0 Channel n Force Interrupt Request Register, n=0:8 (MCS_0_CH0_IRQ_FORCINT)	32	R/W	0000_0000h	13.4.1.16/ 453
50	MCS0 Channel n Interrupt Request Mode Register, n=0:8 (MCS_0_CH0_IRQ_MODE)	32	R/W	See section	13.4.1.17/ 454

Table continues on the next page...

MCS_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
54	MCS0 Channel n Error Interrupt Request Enable register, n=0:8 (MCS_0_CH0_EIRQ_EN)	32	R/W	0000_0000h	13.4.1.18/ 455
74	MCS0 Control Register (MCS_0_CTRL)	32	R/W	0000_0000h	13.4.1.19/ 456
78	MCS0 Reset Register (MCS_0_RST)	32	R/W	0000_0000h	13.4.1.20/ 457
7C	MCS0 Error Register (MCS_0_ERR)	32	w1c	0000_0000h	13.4.1.21/ 459
80	MCS0 Channel n General Purpose 0 Register, n=0:8 (MCS_0_CH1_R0)	32	R/W	0000_0000h	13.4.1.1/ 440
84	MCS0 Channel n General Purpose 1 Register, n=0:8 (MCS_0_CH1_R1)	32	R/W	0000_0000h	13.4.1.2/ 440
88	MCS0 Channel n General Purpose 2 Register, n=0:8 (MCS_0_CH1_R2)	32	R/W	0000_0000h	13.4.1.3/ 441
8C	MCS0 Channel n General Purpose 3 Register, n=0:8 (MCS_0_CH1_R3)	32	R/W	0000_0000h	13.4.1.4/ 441
90	MCS0 Channel n General Purpose 4 Register, n=0:8 (MCS_0_CH1_R4)	32	R/W	0000_0000h	13.4.1.5/ 442
94	MCS0 Channel n General Purpose 5 Register, n=0:8 (MCS_0_CH1_R5)	32	R/W	0000_0000h	13.4.1.6/ 442
98	MCS0 Channel n General Purpose 6 Register, n=0:8 (MCS_0_CH1_R6)	32	R/W	0000_0000h	13.4.1.7/ 443
9C	MCS0 Channel n General Purpose 7 Register, n=0:8 (MCS_0_CH1_R7)	32	R/W	0000_0000h	13.4.1.8/ 443
A0	MCS0 Channel n Control Register, n=0:8 (MCS_0_CH1_CTRL)	32	R/W	0000_0000h	13.4.1.9/ 444
A4	MCS0 Channel n ARU Control Bit Register, n=0:8 (MCS_0_CH1_ACB)	32	R	0000_0000h	13.4.1.10/ 446
C0	MCS0 Channel n Program Counter Register, n=0:8 (MCS_0_CH1_PC)	32	R/W	0000_0000h	13.4.1.13/ 450
C4	MCS0 Channel n Interrupt Request Notification Register, n=0:8 (MCS_0_CH1_IRQ_NOTIFY)	32	w1c	0000_0000h	13.4.1.14/ 451
C8	MCS0 Channel n Interrupt Request Enable Register, n=0:8 (MCS_0_CH1_IRQ_EN)	32	R/W	0000_0000h	13.4.1.15/ 452
CC	MCS0 Channel n Force Interrupt Request Register, n=0:8 (MCS_0_CH1_IRQ_FORCINT)	32	R/W	0000_0000h	13.4.1.16/ 453
D0	MCS0 Channel n Interrupt Request Mode Register, n=0:8 (MCS_0_CH1_IRQ_MODE)	32	R/W	See section	13.4.1.17/ 454
D4	MCS0 Channel n Error Interrupt Request Enable register, n=0:8 (MCS_0_CH1_EIRQ_EN)	32	R/W	0000_0000h	13.4.1.18/ 455
100	MCS0 Channel n General Purpose 0 Register, n=0:8 (MCS_0_CH2_R0)	32	R/W	0000_0000h	13.4.1.1/ 440
104	MCS0 Channel n General Purpose 1 Register, n=0:8 (MCS_0_CH2_R1)	32	R/W	0000_0000h	13.4.1.2/ 440

Table continues on the next page...

MCS_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
108	MCS0 Channel n General Purpose 2 Register, n=0:8 (MCS_0_CH2_R2)	32	R/W	0000_0000h	13.4.1.3/ 441
10C	MCS0 Channel n General Purpose 3 Register, n=0:8 (MCS_0_CH2_R3)	32	R/W	0000_0000h	13.4.1.4/ 441
110	MCS0 Channel n General Purpose 4 Register, n=0:8 (MCS_0_CH2_R4)	32	R/W	0000_0000h	13.4.1.5/ 442
114	MCS0 Channel n General Purpose 5 Register, n=0:8 (MCS_0_CH2_R5)	32	R/W	0000_0000h	13.4.1.6/ 442
118	MCS0 Channel n General Purpose 6 Register, n=0:8 (MCS_0_CH2_R6)	32	R/W	0000_0000h	13.4.1.7/ 443
11C	MCS0 Channel n General Purpose 7 Register, n=0:8 (MCS_0_CH2_R7)	32	R/W	0000_0000h	13.4.1.8/ 443
120	MCS0 Channel n Control Register, n=0:8 (MCS_0_CH2_CTRL)	32	R/W	0000_0000h	13.4.1.9/ 444
124	MCS0 Channel n ARU Control Bit Register, n=0:8 (MCS_0_CH2_ACB)	32	R	0000_0000h	13.4.1.10/ 446
140	MCS0 Channel n Program Counter Register, n=0:8 (MCS_0_CH2_PC)	32	R/W	0000_0000h	13.4.1.13/ 450
144	MCS0 Channel n Interrupt Request Notification Register, n=0:8 (MCS_0_CH2_IRQ_NOTIFY)	32	w1c	0000_0000h	13.4.1.14/ 451
148	MCS0 Channel n Interrupt Request Enable Register, n=0:8 (MCS_0_CH2_IRQ_EN)	32	R/W	0000_0000h	13.4.1.15/ 452
14C	MCS0 Channel n Force Interrupt Request Register, n=0:8 (MCS_0_CH2_IRQ_FORCINT)	32	R/W	0000_0000h	13.4.1.16/ 453
150	MCS0 Channel n Interrupt Request Mode Register, n=0:8 (MCS_0_CH2_IRQ_MODE)	32	R/W	See section	13.4.1.17/ 454
154	MCS0 Channel n Error Interrupt Request Enable register, n=0:8 (MCS_0_CH2_EIRQ_EN)	32	R/W	0000_0000h	13.4.1.18/ 455
180	MCS0 Channel n General Purpose 0 Register, n=0:8 (MCS_0_CH3_R0)	32	R/W	0000_0000h	13.4.1.1/ 440
184	MCS0 Channel n General Purpose 1 Register, n=0:8 (MCS_0_CH3_R1)	32	R/W	0000_0000h	13.4.1.2/ 440
188	MCS0 Channel n General Purpose 2 Register, n=0:8 (MCS_0_CH3_R2)	32	R/W	0000_0000h	13.4.1.3/ 441
18C	MCS0 Channel n General Purpose 3 Register, n=0:8 (MCS_0_CH3_R3)	32	R/W	0000_0000h	13.4.1.4/ 441
190	MCS0 Channel n General Purpose 4 Register, n=0:8 (MCS_0_CH3_R4)	32	R/W	0000_0000h	13.4.1.5/ 442
194	MCS0 Channel n General Purpose 5 Register, n=0:8 (MCS_0_CH3_R5)	32	R/W	0000_0000h	13.4.1.6/ 442
198	MCS0 Channel n General Purpose 6 Register, n=0:8 (MCS_0_CH3_R6)	32	R/W	0000_0000h	13.4.1.7/ 443
19C	MCS0 Channel n General Purpose 7 Register, n=0:8 (MCS_0_CH3_R7)	32	R/W	0000_0000h	13.4.1.8/ 443

Table continues on the next page...

MCS_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1A0	MCS0 Channel n Control Register, n=0:8 (MCS_0_CH3_CTRL)	32	R/W	0000_0000h	13.4.1.9/ 444
1A4	MCS0 Channel n ARU Control Bit Register, n=0:8 (MCS_0_CH3_ACB)	32	R	0000_0000h	13.4.1.10/ 446
1C0	MCS0 Channel n Program Counter Register, n=0:8 (MCS_0_CH3_PC)	32	R/W	0000_0000h	13.4.1.13/ 450
1C4	MCS0 Channel n Interrupt Request Notification Register, n=0:8 (MCS_0_CH3_IRQ_NOTIFY)	32	w1c	0000_0000h	13.4.1.14/ 451
1C8	MCS0 Channel n Interrupt Request Enable Register, n=0:8 (MCS_0_CH3_IRQ_EN)	32	R/W	0000_0000h	13.4.1.15/ 452
1CC	MCS0 Channel n Force Interrupt Request Register, n=0:8 (MCS_0_CH3_IRQ_FORCINT)	32	R/W	0000_0000h	13.4.1.16/ 453
1D0	MCS0 Channel n Interrupt Request Mode Register, n=0:8 (MCS_0_CH3_IRQ_MODE)	32	R/W	See section	13.4.1.17/ 454
1D4	MCS0 Channel n Error Interrupt Request Enable register, n=0:8 (MCS_0_CH3_EIRQ_EN)	32	R/W	0000_0000h	13.4.1.18/ 455
200	MCS0 Channel n General Purpose 0 Register, n=0:8 (MCS_0_CH4_R0)	32	R/W	0000_0000h	13.4.1.1/ 440
204	MCS0 Channel n General Purpose 1 Register, n=0:8 (MCS_0_CH4_R1)	32	R/W	0000_0000h	13.4.1.2/ 440
208	MCS0 Channel n General Purpose 2 Register, n=0:8 (MCS_0_CH4_R2)	32	R/W	0000_0000h	13.4.1.3/ 441
20C	MCS0 Channel n General Purpose 3 Register, n=0:8 (MCS_0_CH4_R3)	32	R/W	0000_0000h	13.4.1.4/ 441
210	MCS0 Channel n General Purpose 4 Register, n=0:8 (MCS_0_CH4_R4)	32	R/W	0000_0000h	13.4.1.5/ 442
214	MCS0 Channel n General Purpose 5 Register, n=0:8 (MCS_0_CH4_R5)	32	R/W	0000_0000h	13.4.1.6/ 442
218	MCS0 Channel n General Purpose 6 Register, n=0:8 (MCS_0_CH4_R6)	32	R/W	0000_0000h	13.4.1.7/ 443
21C	MCS0 Channel n General Purpose 7 Register, n=0:8 (MCS_0_CH4_R7)	32	R/W	0000_0000h	13.4.1.8/ 443
220	MCS0 Channel n Control Register, n=0:8 (MCS_0_CH4_CTRL)	32	R/W	0000_0000h	13.4.1.9/ 444
224	MCS0 Channel n ARU Control Bit Register, n=0:8 (MCS_0_CH4_ACB)	32	R	0000_0000h	13.4.1.10/ 446
240	MCS0 Channel n Program Counter Register, n=0:8 (MCS_0_CH4_PC)	32	R/W	0000_0000h	13.4.1.13/ 450
244	MCS0 Channel n Interrupt Request Notification Register, n=0:8 (MCS_0_CH4_IRQ_NOTIFY)	32	w1c	0000_0000h	13.4.1.14/ 451
248	MCS0 Channel n Interrupt Request Enable Register, n=0:8 (MCS_0_CH4_IRQ_EN)	32	R/W	0000_0000h	13.4.1.15/ 452
24C	MCS0 Channel n Force Interrupt Request Register, n=0:8 (MCS_0_CH4_IRQ_FORCINT)	32	R/W	0000_0000h	13.4.1.16/ 453

Table continues on the next page...

MCS_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
250	MCS0 Channel n Interrupt Request Mode Register, n=0:8 (MCS_0_CH4_IRQ_MODE)	32	R/W	See section	13.4.1.17/ 454
254	MCS0 Channel n Error Interrupt Request Enable register, n=0:8 (MCS_0_CH4_EIRQ_EN)	32	R/W	0000_0000h	13.4.1.18/ 455
280	MCS0 Channel n General Purpose 0 Register, n=0:8 (MCS_0_CH5_R0)	32	R/W	0000_0000h	13.4.1.1/ 440
284	MCS0 Channel n General Purpose 1 Register, n=0:8 (MCS_0_CH5_R1)	32	R/W	0000_0000h	13.4.1.2/ 440
288	MCS0 Channel n General Purpose 2 Register, n=0:8 (MCS_0_CH5_R2)	32	R/W	0000_0000h	13.4.1.3/ 441
28C	MCS0 Channel n General Purpose 3 Register, n=0:8 (MCS_0_CH5_R3)	32	R/W	0000_0000h	13.4.1.4/ 441
290	MCS0 Channel n General Purpose 4 Register, n=0:8 (MCS_0_CH5_R4)	32	R/W	0000_0000h	13.4.1.5/ 442
294	MCS0 Channel n General Purpose 5 Register, n=0:8 (MCS_0_CH5_R5)	32	R/W	0000_0000h	13.4.1.6/ 442
298	MCS0 Channel n General Purpose 6 Register, n=0:8 (MCS_0_CH5_R6)	32	R/W	0000_0000h	13.4.1.7/ 443
29C	MCS0 Channel n General Purpose 7 Register, n=0:8 (MCS_0_CH5_R7)	32	R/W	0000_0000h	13.4.1.8/ 443
2A0	MCS0 Channel n Control Register, n=0:8 (MCS_0_CH5_CTRL)	32	R/W	0000_0000h	13.4.1.9/ 444
2A4	MCS0 Channel n ARU Control Bit Register, n=0:8 (MCS_0_CH5_ACB)	32	R	0000_0000h	13.4.1.10/ 446
2C0	MCS0 Channel n Program Counter Register, n=0:8 (MCS_0_CH5_PC)	32	R/W	0000_0000h	13.4.1.13/ 450
2C4	MCS0 Channel n Interrupt Request Notification Register, n=0:8 (MCS_0_CH5_IRQ_NOTIFY)	32	w1c	0000_0000h	13.4.1.14/ 451
2C8	MCS0 Channel n Interrupt Request Enable Register, n=0:8 (MCS_0_CH5_IRQ_EN)	32	R/W	0000_0000h	13.4.1.15/ 452
2CC	MCS0 Channel n Force Interrupt Request Register, n=0:8 (MCS_0_CH5_IRQ_FORCINT)	32	R/W	0000_0000h	13.4.1.16/ 453
2D0	MCS0 Channel n Interrupt Request Mode Register, n=0:8 (MCS_0_CH5_IRQ_MODE)	32	R/W	See section	13.4.1.17/ 454
2D4	MCS0 Channel n Error Interrupt Request Enable register, n=0:8 (MCS_0_CH5_EIRQ_EN)	32	R/W	0000_0000h	13.4.1.18/ 455
300	MCS0 Channel n General Purpose 0 Register, n=0:8 (MCS_0_CH6_R0)	32	R/W	0000_0000h	13.4.1.1/ 440
304	MCS0 Channel n General Purpose 1 Register, n=0:8 (MCS_0_CH6_R1)	32	R/W	0000_0000h	13.4.1.2/ 440
308	MCS0 Channel n General Purpose 2 Register, n=0:8 (MCS_0_CH6_R2)	32	R/W	0000_0000h	13.4.1.3/ 441
30C	MCS0 Channel n General Purpose 3 Register, n=0:8 (MCS_0_CH6_R3)	32	R/W	0000_0000h	13.4.1.4/ 441

Table continues on the next page...

MCS_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
310	MCS0 Channel n General Purpose 4 Register, n=0:8 (MCS_0_CH6_R4)	32	R/W	0000_0000h	13.4.1.5/ 442
314	MCS0 Channel n General Purpose 5 Register, n=0:8 (MCS_0_CH6_R5)	32	R/W	0000_0000h	13.4.1.6/ 442
318	MCS0 Channel n General Purpose 6 Register, n=0:8 (MCS_0_CH6_R6)	32	R/W	0000_0000h	13.4.1.7/ 443
31C	MCS0 Channel n General Purpose 7 Register, n=0:8 (MCS_0_CH6_R7)	32	R/W	0000_0000h	13.4.1.8/ 443
320	MCS0 Channel n Control Register, n=0:8 (MCS_0_CH6_CTRL)	32	R/W	0000_0000h	13.4.1.9/ 444
324	MCS0 Channel n ARU Control Bit Register, n=0:8 (MCS_0_CH6_ACB)	32	R	0000_0000h	13.4.1.10/ 446
340	MCS0 Channel n Program Counter Register, n=0:8 (MCS_0_CH6_PC)	32	R/W	0000_0000h	13.4.1.13/ 450
344	MCS0 Channel n Interrupt Request Notification Register, n=0:8 (MCS_0_CH6_IRQ_NOTIFY)	32	w1c	0000_0000h	13.4.1.14/ 451
348	MCS0 Channel n Interrupt Request Enable Register, n=0:8 (MCS_0_CH6_IRQ_EN)	32	R/W	0000_0000h	13.4.1.15/ 452
34C	MCS0 Channel n Force Interrupt Request Register, n=0:8 (MCS_0_CH6_IRQ_FORCINT)	32	R/W	0000_0000h	13.4.1.16/ 453
350	MCS0 Channel n Interrupt Request Mode Register, n=0:8 (MCS_0_CH6_IRQ_MODE)	32	R/W	See section	13.4.1.17/ 454
354	MCS0 Channel n Error Interrupt Request Enable register, n=0:8 (MCS_0_CH6_EIRQ_EN)	32	R/W	0000_0000h	13.4.1.18/ 455
380	MCS0 Channel n General Purpose 0 Register, n=0:8 (MCS_0_CH7_R0)	32	R/W	0000_0000h	13.4.1.1/ 440
384	MCS0 Channel n General Purpose 1 Register, n=0:8 (MCS_0_CH7_R1)	32	R/W	0000_0000h	13.4.1.2/ 440
388	MCS0 Channel n General Purpose 2 Register, n=0:8 (MCS_0_CH7_R2)	32	R/W	0000_0000h	13.4.1.3/ 441
38C	MCS0 Channel n General Purpose 3 Register, n=0:8 (MCS_0_CH7_R3)	32	R/W	0000_0000h	13.4.1.4/ 441
390	MCS0 Channel n General Purpose 4 Register, n=0:8 (MCS_0_CH7_R4)	32	R/W	0000_0000h	13.4.1.5/ 442
394	MCS0 Channel n General Purpose 5 Register, n=0:8 (MCS_0_CH7_R5)	32	R/W	0000_0000h	13.4.1.6/ 442
398	MCS0 Channel n General Purpose 6 Register, n=0:8 (MCS_0_CH7_R6)	32	R/W	0000_0000h	13.4.1.7/ 443
39C	MCS0 Channel n General Purpose 7 Register, n=0:8 (MCS_0_CH7_R7)	32	R/W	0000_0000h	13.4.1.8/ 443
3A0	MCS0 Channel n Control Register, n=0:8 (MCS_0_CH7_CTRL)	32	R/W	0000_0000h	13.4.1.9/ 444
3A4	MCS0 Channel n ARU Control Bit Register, n=0:8 (MCS_0_CH7_ACB)	32	R	0000_0000h	13.4.1.10/ 446

Table continues on the next page...

MCS_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3C0	MCS0 Channel n Program Counter Register, n=0:8 (MCS_0_CH7_PC)	32	R/W	0000_0000h	13.4.1.13/ 450
3C4	MCS0 Channel n Interrupt Request Notification Register, n=0:8 (MCS_0_CH7_IRQ_NOTIFY)	32	w1c	0000_0000h	13.4.1.14/ 451
3C8	MCS0 Channel n Interrupt Request Enable Register, n=0:8 (MCS_0_CH7_IRQ_EN)	32	R/W	0000_0000h	13.4.1.15/ 452
3CC	MCS0 Channel n Force Interrupt Request Register, n=0:8 (MCS_0_CH7_IRQ_FORCINT)	32	R/W	0000_0000h	13.4.1.16/ 453
3D0	MCS0 Channel n Interrupt Request Mode Register, n=0:8 (MCS_0_CH7_IRQ_MODE)	32	R/W	See section	13.4.1.17/ 454
3D4	MCS0 Channel n Error Interrupt Request Enable register, n=0:8 (MCS_0_CH7_EIRQ_EN)	32	R/W	0000_0000h	13.4.1.18/ 455

13.4.1.1 MCS0 Channel n General Purpose 0 Register, n=0:8 (MCS_0_CH*n*_R0)

Address: 3_0000h base + 0h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								DATA																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCS_0_CH*n*_R0 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 DATA	Data of MCS general purpose register R0.

13.4.1.2 MCS0 Channel n General Purpose 1 Register, n=0:8 (MCS_0_CH*n*_R1)

Address: 3_0000h base + 4h offset + (128d × i), where i=0d to 7d

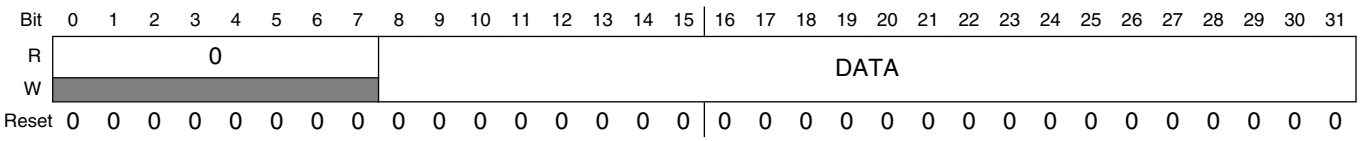
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								DATA																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCS_0_CH*n*_R1 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 DATA	Data of MCS general purpose register R1.

13.4.1.3 MCS0 Channel n General Purpose 2 Register, n=0:8 (MCS_0_CH*n*_R2)

Address: 3_0000h base + 8h offset + (128d × i), where i=0d to 7d

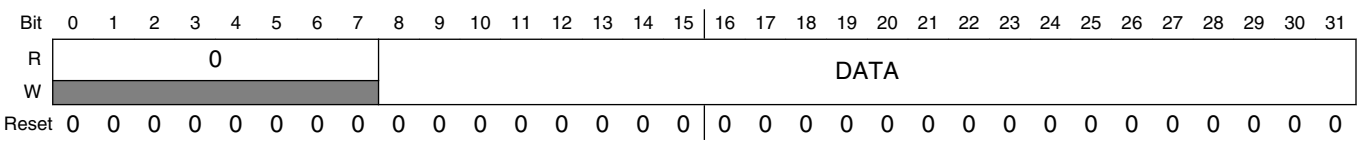


MCS_0_CH*n*_R2 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 DATA	Data of MCS general purpose register R2.

13.4.1.4 MCS0 Channel n General Purpose 3 Register, n=0:8 (MCS_0_CH*n*_R3)

Address: 3_0000h base + Ch offset + (128d × i), where i=0d to 7d



MCS_0_CH*n*_R3 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 DATA	Data of MCS general purpose register R3.

13.4.1.5 MCS0 Channel n General Purpose 4 Register, n=0:8 (MCS_0_CH*n*_R4)

Address: 3_0000h base + 10h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								DATA																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCS_0_CH*n*_R4 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 DATA	Data of MCS general purpose register R4.

13.4.1.6 MCS0 Channel n General Purpose 5 Register, n=0:8 (MCS_0_CH*n*_R5)

Address: 3_0000h base + 14h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								DATA																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCS_0_CH*n*_R5 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 DATA	Data of MCS general purpose register R5.

13.4.1.7 MCS0 Channel n General Purpose 6 Register, n=0:8 (MCS_0_CH*n*_R6)

This register is write protected during an active ARDI or NARDI instruction.

Address: 3_0000h base + 18h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								DATA																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCS_0_CH*n*_R6 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 DATA	Data of MCS general purpose register R6.

13.4.1.8 MCS0 Channel n General Purpose 7 Register, n=0:8 (MCS_0_CH*n*_R7)

Address: 3_0000h base + 1Ch offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								DATA																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCS_0_CH*n*_R7 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 DATA	Data of MCS general purpose register R7.

13.4.1.9 MCS0 Channel n Control Register, n=0:8 (MCS_0_CH*n*_CTRL)

Address: 3_0000h base + 20h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0													SP_CNT		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0					SAT	CWT	CAT	N	V	Z	CY	0	ERR	IRQ	EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCS_0_CH*n*_CTRL field descriptions

Field	Description
0–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–15 SP_CNT	Stack pointer counter value. Actual stack depth of channel. The bit field is incremented on behalf of a CALL or PUSH instruction and decremented on behalf of a RET or POP instruction. The MCS channel STK_ERR_IRQ is raised, when an overflow or underflow is detected on this bit field.
16–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 SAT	Successful ARU transfer bit. 0 Non-blocking ARU transfer failed due to missing data. 1 Non-blocking ARU transfer finished successfully.
22 CWT	Cancel WURM instruction state. This bit is read only and it mirrors the internal cancel WURM instruction status flag CWT.
23 CAT	Cancel ARU transfer state. This bit is read only and it mirrors the internal cancel ARU transfer status flag CAT.
24 N	Negative bit state. This bit is read only and it mirrors the internal zero flag N.
25 V	Overflow bit state. This bit is read only and it mirrors the internal carry flag V.
26 Z	Zero bit state. This bit is read only and it mirrors the internal zero flag Z.
27 CY	Carry bit state. This bit is read only and it mirrors the internal carry flag CY.

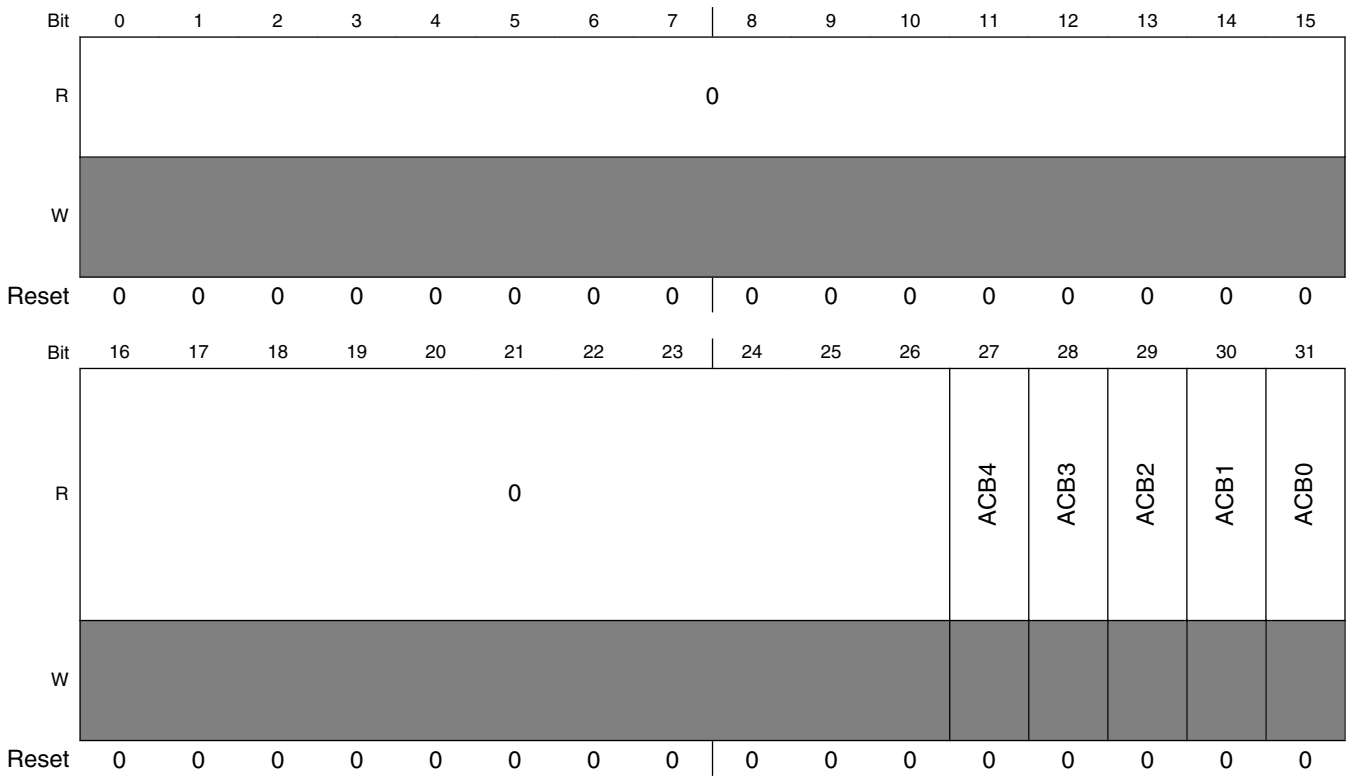
Table continues on the next page...

MCS_0_CH*n*_CTRL field descriptions (continued)

Field	Description
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 ERR	Error state. This bit is read only and it mirrors the internal error state. 0 error signal pending in MCS-channel x. 1 Error signal is pending in MCS-channel x.
30 IRQ	Interrupt state. This bit is read only and it mirrors the internal IRQ state. 0 No interrupt pending in MCS-channel x. 1 Interrupt is pending in MCS-channel x.
31 EN	Enable MCS-channel. Enabling or disabling of an MCS-channel may take several clock cycles, e.g. active memory transfers have to be finished before disabling the MCS-channel. The internal state of a channel can be obtained by reading the bit EN. **NOTE:** The EN bit is write protected during RAM reset phase. 0 Disable current MCS-channel. 1 Enable current MCS-channel.

13.4.1.10 MCS0 Channel n ARU Control Bit Register, n=0:8 (MCS_0_CH*n*_ACB)

Address: 3_0000h base + 24h offset + (128d × i), where i=0d to 7d



MCS_0_CH*n*_ACB field descriptions

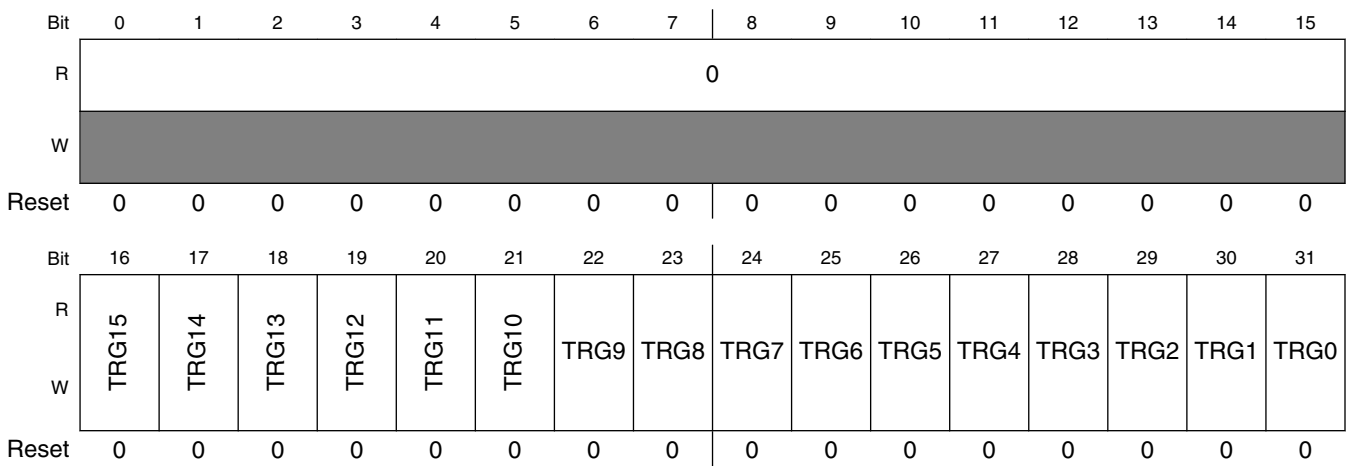
Field	Description
0–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 ACB4	See ACB0.
28 ACB3	See ACB0.
29 ACB2	See ACB0.
30 ACB1	See ACB0.
31 ACB0	ARU Control bit[4:0]. These bits are read only and mirrors the internal state.

13.4.1.11 MCS0 Clear Trigger Control Register (MCS_0_CTRG)

The trigger bits TRGn (n = 0..15) are accessible by all MCS channels as well as the CPU. Setting a trigger bit can be performed with the STRG register, in the case of an MCS-channel or the MCS[i]_STRG register in the case of the CPU. Clearing a trigger bit can be performed with the CTRG register, in the case of an MCS-channel or the MCS[i]_CTRG register in the case of the CPU.

Trigger bits can be used for signalizing specific events to MCS-channels or the CPU. An MCS-channel suspended with a WURM instruction can be resumed by setting the appropriate trigger bit.

Address: 3_0000h base + 28h offset = 3_0028h



MCS_0_CTRG field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 TRG15	Trigger bit.
17 TRG14	Trigger bit.
18 TRG13	Trigger bit.
19 TRG12	Trigger bit.
20 TRG11	Trigger bit.
21 TRG10	Trigger bit.
22 TRG9	Trigger bit.

Table continues on the next page...

MCS_0_CTRG field descriptions (continued)

Field	Description
23 TRG8	Trigger bit.
24 TRG7	Trigger bit.
25 TRG6	Trigger bit.
26 TRG5	Trigger bit.
27 TRG4	Trigger bit.
28 TRG3	Trigger bit.
29 TRG2	Trigger bit.
30 TRG1	Trigger bit.
31 TRG0	Trigger bit. 0 READ: trigger bit is cleared / WRITE : do nothing. 1 READ: trigger bit is set / WRITE : clear trigger bit.

13.4.1.12 MCS0 Set Trigger Control Register (MCS_0_STRG)

The trigger bits TRGn (n = 0..15) are accessible by all MCS channels as well as the CPU. Setting a trigger bit can be performed with the STRG register, in the case of an MCS-channel or the MCS[i]_STRG register in the case of the CPU. Clearing a trigger bit can be performed with the CTRG register, in the case of an MCS-channel or the MCS[i]_CTRG register in the case of the CPU.

Trigger bits can be used for signalizing specific events to MCS-channels or the CPU. An MCS-channel suspended with a WURM instruction can be resumed by setting the appropriate trigger bit.

Address: 3_0000h base + 2Ch offset = 3_002Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TRG15	TRG14	TRG13	TRG12	TRG11	TRG10	TRG9	TRG8	TRG7	TRG6	TRG5	TRG4	TRG3	TRG2	TRG1	TRG0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCS_0_STRG field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 TRG15	Trigger bit.
17 TRG14	Trigger bit.
18 TRG13	Trigger bit.
19 TRG12	Trigger bit.
20 TRG11	Trigger bit.
21 TRG10	Trigger bit.
22 TRG9	Trigger bit.
23 TRG8	Trigger bit.
24 TRG7	Trigger bit.
25 TRG6	Trigger bit.
26 TRG5	Trigger bit.
27 TRG4	Trigger bit.
28 TRG3	Trigger bit.
29 TRG2	Trigger bit.
30 TRG1	Trigger bit.
31 TRG0	Trigger bit. 0 READ: trigger bit is cleared / WRITE : do nothing. 1 READ: trigger bit is set / WRITE : set trigger bit.

13.4.1.13 MCS0 Channel n Program Counter Register, n=0:8 (MCS_0_CH*n*_PC)

NOTE

PC register reset value for each channel is 0x00000000 + 4*n

Address: 3_0000h base + 40h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																PC															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCS_0_CH*n*_PC field descriptions

Field	Description
0–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–31 PC	Current Program Counter. The program counter is only writable if the corresponding MCS-channel is disabled. Bits 0 and 1 are always written as zeros.

13.4.1.14 MCS0 Channel n Interrupt Request Notification Register, n=0:8 (MCS_0_CH*n*_IRQ_NOTIFY)

Address: 3_0000h base + 44h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0													MEM_ERR_IRQ	STK_ERR_IRQ	MCS_IRQ
W														w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCS_0_CH*n*_IRQ_NOTIFY field descriptions

Field	Description
0–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 MEM_ERR_IRQ	Memory access out of range in channel n. This bit will be cleared on a CPU write access with a value '1'. A read access leaves the bit unchanged. **NOTE:** In the case of a memory overflow, any read or write access to the RAM is always blocked. The read data is unpredictable. 0 No IRQ released. 1 MCS-channel requests a memory location that is out of range.
30 STK_ERR_IRQ	Stack counter overflow/underflow of channel n. This bit will be cleared on a CPU write access with a value '1'. A read access leaves the bit unchanged.

Table continues on the next page...

MCS_0_CH*n*_IRQ_NOTIFY field descriptions (continued)

Field	Description
	0 No IRQ released. 1 A stack counter overflow or underflow occurred.
31 MCS_IRQ	Interrupt request by MCS-channel n. This bit will be cleared on a CPU write access with a value '1'. A read access leaves the bit unchanged. 0 No IRQ released. 1 IRQ released by MCS-channel.

13.4.1.15 MCS0 Channel n Interrupt Request Enable Register, n=0:8 (MCS_0_CH*n*_IRQ_EN)

Address: 3_0000h base + 48h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0													MEM_ERR_IRQ_EN	STK_ERR_IRQ_EN	MCS_IRQ_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCS_0_CH*n*_IRQ_EN field descriptions

Field	Description
0–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 MEM_ERR_IRQ_EN	MCS channel n MEM_ERR_IRQ interrupt enable. 0 Disable interrupt 1 Enable interrupt
30 STK_ERR_IRQ_EN	MCS channel n STK_ERR_IRQ interrupt enable. 0 Disable interrupt 1 Enable interrupt
31 MCS_IRQ_EN	MCS channel n MCS_IRQ interrupt enable.

Table continues on the next page...

MCS_0_CH*n*_IRQ_EN field descriptions (continued)

Field	Description
	0 Disable interrupt 1 Enable interrupt

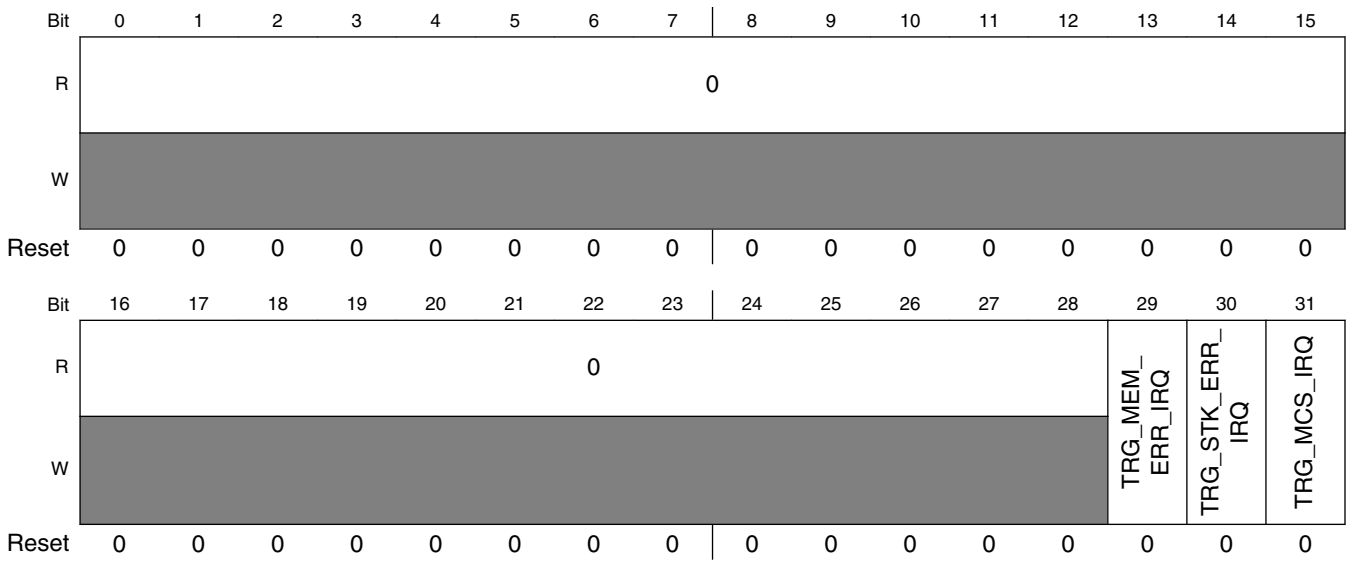
13.4.1.16 MCS0 Channel n Force Interrupt Request Register, n=0:8 (MCS_0_CH*n*_IRQ_FORCINT)

NOTE

Bit[29:31] is write protected by bit GTM_CTRL[RF_PROT].

Bit[29:31] is cleared automatically after write to the bit.

Address: 3_0000h base + 4Ch offset + (128d × i), where i=0d to 7d



MCS_0_CH*n*_IRQ_FORCINT field descriptions

Field	Description
0–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 TRG_MEM_ERR_IRQ	Trigger IRQ bit in MCS_CH_[x]_IRQ_NOTIFY register by software. 0 No interrupt triggering. 1 Assert corresponding field in MCS[i]_CH[x]_IRQ_NOTIFY register.
30 TRG_STK_ERR_IRQ	Trigger IRQ bit in MCS_CH_[x]_IRQ_NOTIFY register by software. 0 No interrupt triggering. 1 Assert corresponding field in MCS[i]_CH[x]_IRQ_NOTIFY register.
31 TRG_MCS_IRQ	Trigger IRQ bit in MCS_CH_[x]_IRQ_NOTIFY register 0 No interrupt triggering. 1 Assert corresponding field in MCS[i]_CH[x]_IRQ_NOTIFY register.

13.4.1.17 MCS0 Channel n Interrupt Request Mode Register, n=0:8 (MCS_0_CH*n*_IRQ_MODE)

Address: 3_0000h base + 50h offset + (128d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0														IRQ_MODE	
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

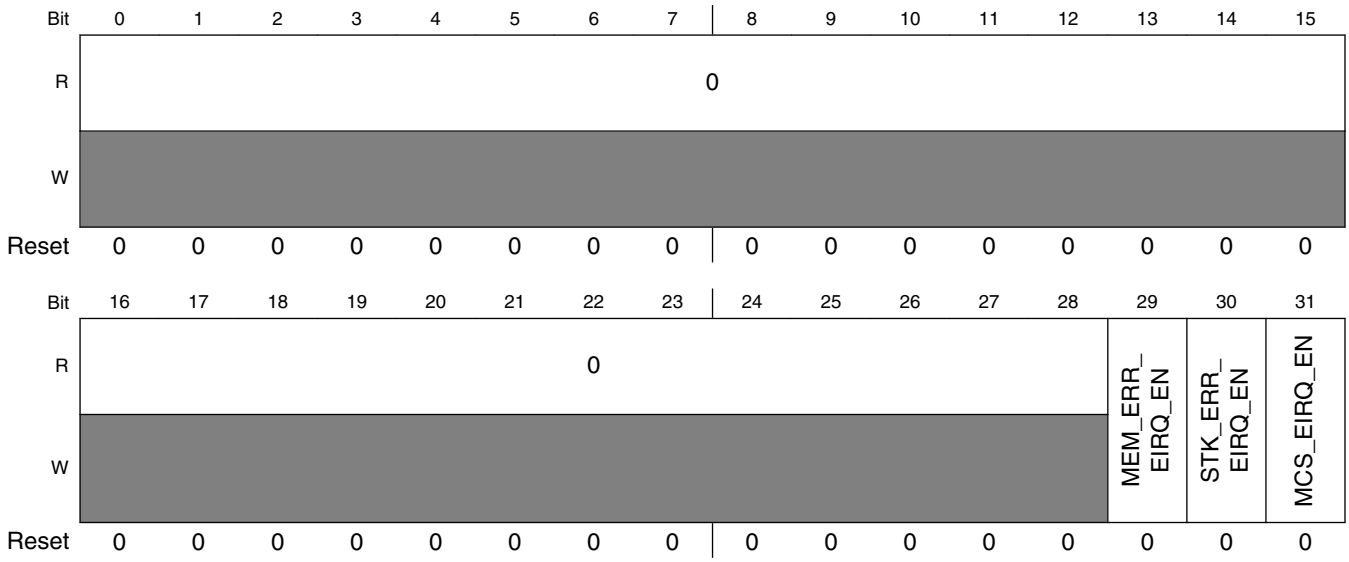
* Notes:
- Bits 28-31 do not reset to a given value.

MCS_0_CH*n*_IRQ_MODE field descriptions

Field	Description
0–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–31 IRQ_MODE	IRQ mode selection. The interrupt modes are described in section 2.5. 00 Level mode. 01 Pulse mode. 10 Pulse-Notify mode. 11 Single-Pulse mode.

13.4.1.18 MCS0 Channel n Error Interrupt Request Enable register, n=0:8 (MCS_0_CH*n*_EIRQ_EN)

Address: 3_0000h base + 54h offset + (128d × i), where i=0d to 7d



MCS_0_CH*n*_EIRQ_EN field descriptions

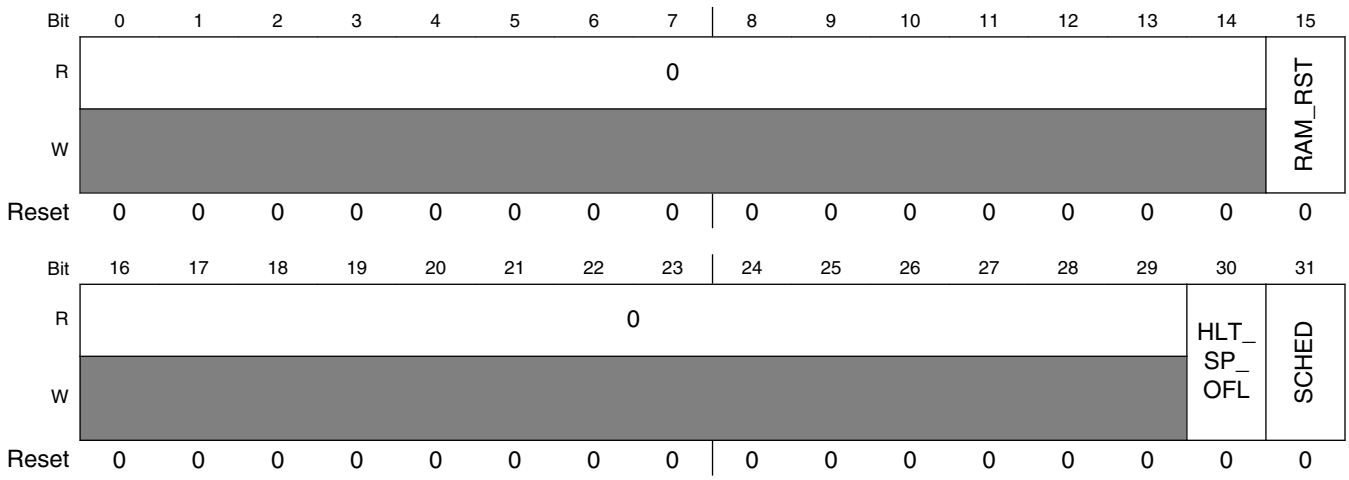
Field	Description
0–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 MEM_ERR_ EIRQ_EN	MCS[i] channel n MEM_ERR_EIRQ error interrupt enable. 0 Disable error interrupt 1 Enable error interrupt
30 STK_ERR_ EIRQ_EN	MCS[i] channel n STK_ERR_IRQ error interrupt enable. 0 Disable error interrupt 1 Enable error interrupt
31 MCS_EIRQ_EN	Trigger IRQ bit in MCS_CH_[x]_IRQ_NOTIFY register 0 Disable error interrupt 1 Enable error interrupt

13.4.1.19 MCS0 Control Register (MCS_0_CTRL)

NOTE

Reset value of bit 16 is undefined.

Address: 3_0000h base + 74h offset = 3_0074h



MCS_0_CTRL field descriptions

Field	Description
0–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 RAM_RST	RAM reset bit. The RAM reset initializes the memory content with zeros. RAM access and enabling of MCS channels is disabled during active RAM reset. **NOTE:** This bit is only writable if bit GTM_CTRL[RF_PROT] is cleared and all MCS-channels are disabled. The actual reset values of this bit depends on the silicon vendor configuration. The reset value is 1, if the RAM reset is performed together with the submodule reset, otherwise the reset value is 0. If the reset value is 1, the reset value is changed to 0 by hardware, when the RAM reset finished. 0 READ: no RAM reset is active / WRITE : do nothing. 1 READ: MCS currently resets RAM content / WRITE : trigger RAM reset.
16–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 HLT_SP_OFL	Halt on stack pointer overflow. 0 No halt on MCS-channel stack pointer counter over/underflow. 1 MCS-channel is disabled if a stack pointer counter over/underflow occurs.
31 SCHED	MCS submodule scheduling scheme. 0 Accelerated scheduling scheme. 1 Round-Robin scheduling scheme.

13.4.1.20 MCS0 Reset Register (MCS_0_RST)

The CWTn (n = 0 ...7) bit inside the STA register of the corresponding MCS-channel is set and any pending WURM instruction is canceled. The MCS-channel resumes with the instruction after the WURM instruction.

The CWTn (n = 0 ...7) bit is cleared by the corresponding MCS channel, when the channel reaches a WURM instruction.

The CATn (n = 0 ...7) bit inside the STA register of the corresponding MCS-channel is set and any pending ARU read or write request is canceled. The MCS-channel resumes with the instruction after the ARU transfer instruction.

The CATn (n = 0 ...7) bit is cleared by the corresponding MCS channel, when the channel reaches an ARU read or write instruction.

The RSTn (n= 0 ...7) bits is cleared automatically after write access by the CPU. All channel related registers are set to their reset values and channel operation is stopped immediately. The reset action RSTn has a higher priority than setting the bits CWTn or CATn (n = 0 ...7).

Channel related registers are all registers MCS[i]_CH[n]_*, all MCS internal registers accessible by the correpsonding channel, with exception of the common trigger register (accessed by CTRG/STRG).

Address: 3_0000h base + 78h offset = 3_0078h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0								CWT7	CWT6	CWT5	CWT4	CWT3	CWT2	CWT1	CWT0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	CAT7	CAT6	CAT5	CAT4	CAT3	CAT2	CAT1	CAT0	RST7	RST6	RST5	RST4	RST3	RST2	RST1	RST0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCS_0_RST field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 CWT7	Cancel WURM instruction for channel.
9 CWT6	Cancel WURM instruction for channel.
10 CWT5	Cancel WURM instruction for channel.
11 CWT4	Cancel WURM instruction for channel.
12 CWT3	Cancel WURM instruction for channel.
13 CWT2	Cancel WURM instruction for channel.
14 CWT1	Cancel WURM instruction for channel.
15 CWT0	Cancel WURM instruction for channel. 0 Do nothing. 1 Cancel any pending WURM instruction.
16 CAT7	Cancel ARU transfer for channel.
17 CAT6	Cancel ARU transfer for channel.
18 CAT5	Cancel ARU transfer for channel.
19 CAT4	Cancel ARU transfer for channel.
20 CAT3	Cancel ARU transfer for channel.
21 CAT2	Cancel ARU transfer for channel.
22 CAT1	Cancel ARU transfer for channel.
23 CAT0	Cancel ARU transfer for channel. 0 Do nothing. 1 Cancel any pending ARU read or write transfer.
24 RST7	Software reset of channel.
25 RST6	Software reset of channel.
26 RST5	Software reset of channel.
27 RST4	Software reset of channel.

Table continues on the next page...

MCS_0_RST field descriptions (continued)

Field	Description
28 RST3	Software reset of channel.
29 RST2	Software reset of channel.
30 RST1	Software reset of channel.
31 RST0	Software reset of channel. 0 No action. 1 Reset channel.

13.4.1.21 MCS0 Error Register (MCS_0_ERR)

The CPU can read the ERRn (n = 7...0) bits in order to determine the current error state of the corresponding MCS-channel n. The error state is also evaluated by the module MON.

Writing a value 1 to these bits resets the corresponding error state and resets the channel internal ERR bit in the STA and channel CTRL registers.

Address: 3_0000h base + 7Ch offset = 3_007Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								ERR7	ERR6	ERR5	ERR4	ERR3	ERR2	ERR1	ERR0
W									w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCS_0_ERR field descriptions

Field	Description
0–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 ERR7	Error State of MCS-channel.
25 ERR6	Error State of MCS-channel.
26 ERR5	Error State of MCS-channel.

Table continues on the next page...

MCS_0_ERR field descriptions (continued)

Field	Description
27 ERR4	Error State of MCS-channel.
28 ERR3	Error State of MCS-channel.
29 ERR2	Error State of MCS-channel.
30 ERR1	Error State of MCS-channel.
31 ERR0	Error State of MCS-channel. 0 No error signal. 1 Error signal is pending.

Chapter 14
Memory Configuration (MCFG)

14.1 MCFG Overview

The Memory Configuration submodule (MCFG) is an infrastructure module that organizes physical memory blocks, and maps them to the instances of the Multi Channel Sequencer (MCS) submodules. The default configuration maps a memory of size 1K * 32 bit = 4KB to MCS memory page 0 and a memory of size 0.5K * 32 bit = 2 KB to MCS memory page 1.

In order to support different memory sizes for different MCS instances, the MCFG provides two additional layout configurations for reorganization of memory pages between neighboring MCS submodules:

- The SWAP layout configuration swaps the 2KB memory page of the current MCS instance with the 4KB memory page of the successive MCS instance. Thus, the memory of the current MCS module is increased by 2KB, but the memory of the successor is decreased by 2KB.
- The BORROW layout configuration borrows the 4KB memory page of the successive MCS instance for the current instance. Thus, the memory of the current MCS module is increased by 4KB, but the memory of the successor is decreased by 4KB.

NOTE

The successor of the last MCS instance is the first MCS instance MCS0.

The actual size of the memory pages for an MCS instance depends on the layout configuration for the current instance MCS[i] and the layout configuration of the preceding memory instance MCS[i - 1].

Memory Layout Parameters (MCS_RAM1_EN_ADDR_MSB = 0) summarizes the layout parameters MP0 and MP1 for MCS instance MCS[i].

The addressing of memory page 0 ranges from 0 to MP0 - 4 and the addressing of memory page 1 ranges from MP0 to MP1 - 4.

Besides these software related layout configurations, the MCFG submodule has an additional input port MCS_RAM1_EN_ADDR_MSB which is routed to the top level of the GTM-IP. The above mentioned behaviour of the MCFG submodule is applied if this port is connected to a constant logic level of zero (0).

If a constant logic level of one (1) is applied to this port, the MCFG submodule assumes that a memory of size 1K*32 bit = 4KB is also mapped to MCS memory page 1.

In this case the memory layout configurations of the MCFG submodule change as shown in Memory Layout Configurations (MCS_RAM1_EN_ADDR_MSB = 1) and the memory layout parameters are modified according to Memory layout parameters (MCS_RAM1_EN_ADDR_MSB = 1).

This document assumes that the GTM implementation embeds 7 MCS instances. However, the actual number of implemented MCS instances can be obtained from [1].

14.1.1 Memory layout configurations (MCS_RAM1_EN_ADDR_MSB = 0)

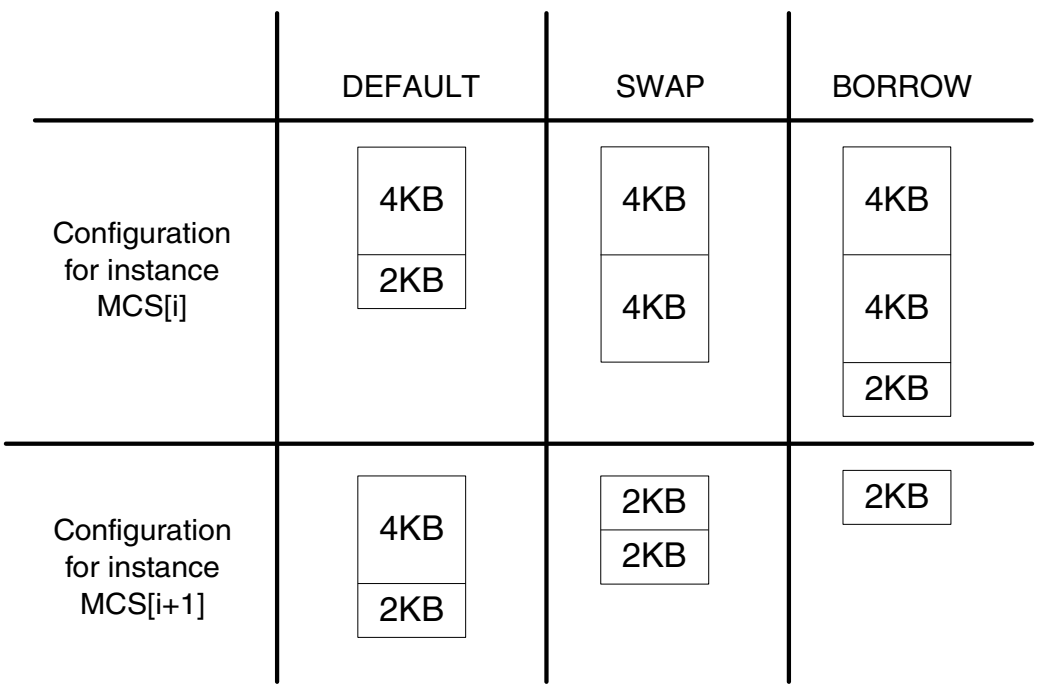


Figure 14-1. Memory Layout Configurations (MCS_RAM1_EN_ADDR_MSB = 0)

14.1.2 Memory Layout Parameters (MCS_RAM1_EN_ADDR_MSB = 0)

Table 14-1. Memory Layout Parameters (MCS_RAM1_EN_ADDR_MSB = 0)

			Memory Layout Option of preceding MCS instance MCS[i-1]		
			DEFAULT	SWAP	BORROW
Memory Layout Option of current MCS instance MCS[i]	DEFAULT	MP0	0x1000	0x0800	0x0000
		MP1	0x1800	0x1000	0x0800
	SWAP	MP0	0x1000	0x0800	0x0000
		MP1	0x2000	0x1800	0x1000
	BORROW	MP0	0x1000	0x0800	0x0000
		MP1	0x2800	0x2000	0x1800

14.1.3 Memory Layout Configurations (MCS_RAM1_EN_ADDR_MSB = 1)

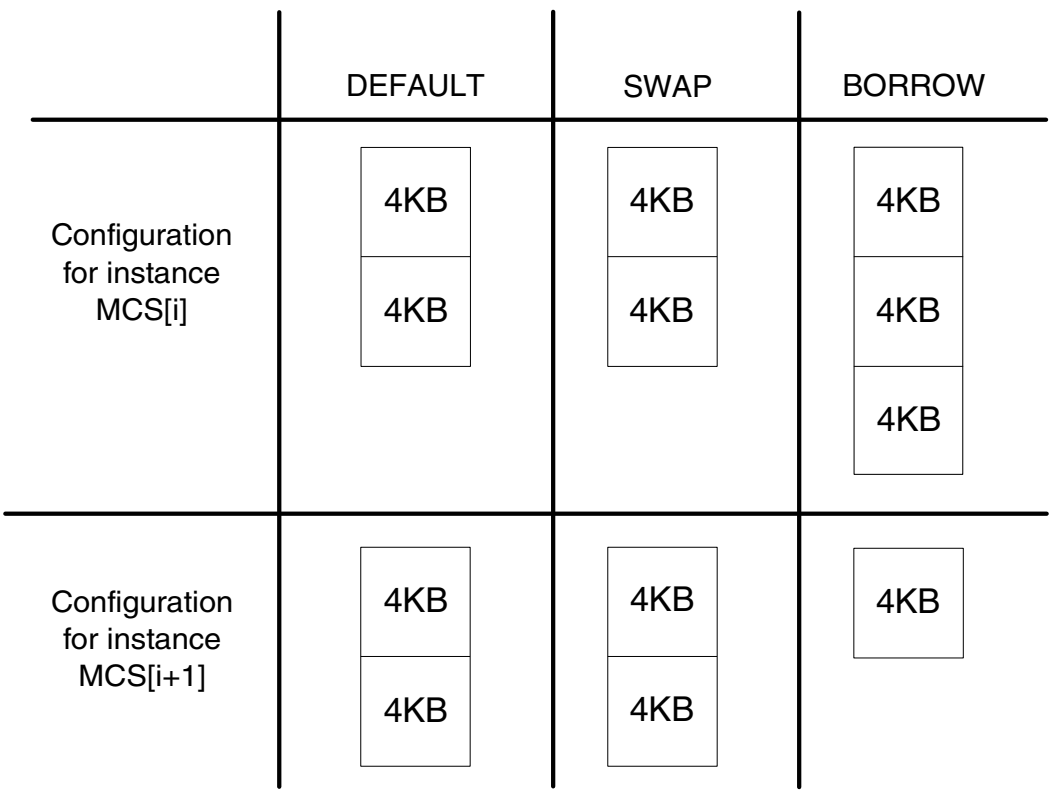


Figure 14-2. Memory Layout Configurations (MCS_RAM1_EN_ADDR_MSB = 1)

14.1.4 Memory layout parameters (MCS_RAM1_EN_ADDR_MSB = 1)

Table 14-2. Memory layout parameters (MCS_RAM1_EN_ADDR_MSB = 1)

			Memory Layout Option of preceding MCS instance MCS[i-1]		
			DEFAULT	SWAP	BORROW
Memory Layout Option of current MCS instance MCS[i]	DEFAULT	MP0	0x1000	0x1000	0x0000
		MP1	0x2000	0x2000	0x1000
	SWAP	MP0	0x1000	0x1000	0x0000
		MP1	0x2000	0x2000	0x1000
	BORROW	MP0	0x1000	0x1000	0x0000
		MP1	0x3000	0x3000	0x2000

14.2 Memory Map and Registers

The Memory Configuration (MCFG) module register is described as follows:

MCFG memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Memory Configuration Register (MCFG_CTRL)	32	R/W	0000_0000h	14.2.1/464

14.2.1 Memory Configuration Register (MCFG_CTRL)

Address: F40h base + 0h offset = F40h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0		MEM6		MEM5		MEM4		MEM3		MEM2		MEM1		MEM0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCFG_CTRL field descriptions

Field	Description
0–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–19 MEM6	Configure Memory pages for MCS-instance MCS6. 00 DEFAULT configuration 01 SWAP configuration 10 BORROW configuration 11 DEFAULT configuration
20–21 MEM5	Configure Memory pages for MCS-instance MCS5. 00 DEFAULT configuration 01 SWAP configuration 10 BORROW configuration 11 DEFAULT configuration
22–23 MEM4	Configure Memory pages for MCS-instance MCS4. 00 DEFAULT configuration 01 SWAP configuration 10 BORROW configuration 11 DEFAULT configuration
24–25 MEM3	Configure Memory pages for MCS-instance MCS3 00 DEFAULT configuration 01 SWAP configuration 10 BORROW configuration 11 DEFAULT configuration
26–27 MEM2	Configure Memory pages for MCS-instance MCS2 00 DEFAULT configuration 01 SWAP configuration 10 BORROW configuration 11 DEFAULT configuration
28–29 MEM1	Configure Memory pages for MCS-instance MCS1. 00 DEFAULT configuration 01 SWAP configuration. 10 BORROW configuration. 11 DEFAULT configuration
30–31 MEM0	Configure Memory pages for MCS-instance MCS0. 00 DEFAULT configuration 01 SWAP configuration. 10 BORROW configuration. 11 DEFAULT configuration

Chapter 15
TIM0 Input Mapping Module (MAP)

15.1 MAP Overview

The TIM0 Input Mapping Module (MAP) submodule evaluates the TIM0_CHn outputs from TIM0 channels 0:5 and generates two DPLL submodule input signals, TRIGGER and STATE. The filtering of the signals can be accomplished inside the TIM channels themselves. Figure 15-1 shows the architecture of the MAP.

15.1.1 MAP Architecture

Figure 15-1 shows the MAP architecture.

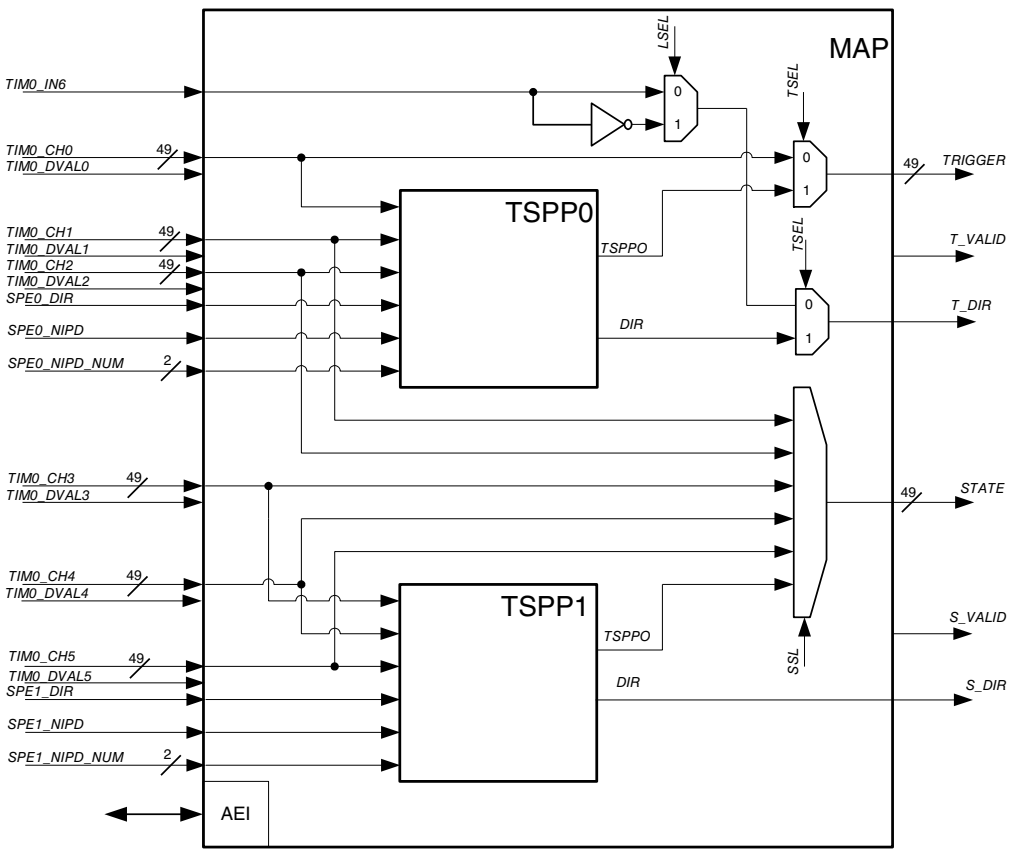


Figure 15-1. MAP architecture

The MAP submodule can route the signals (coming from the Tim channel 0:5 outputs) three ways:

- Route the 49 bits of data coming from TIM0 channel 0 (TIM0_CH0) to the *TRIGGER* signal, which is then routed to the DPLL along with the *T_VALID* signal.
- Route one of the five signals coming from TIM0 channels 1:5 to the *STATE* signal, which is then routed to the DPLL along with the *S_VALID* signal.

- The *TRIGGER*, *T_VALID*, *STATE* and *S_VALID* signals can be generated in the TIM Signal Preprocessing (TSPP) subunits. This is accomplished in combination with the Sensor Pattern Evaluation (SPE) submodule (see SPE Overview) and only when the TSSPx subunits are enabled and the *SPEi_NIPD* signal is asserted by the SPE. The *SPEi_NIPD_NUM* signal encodes which of the three *TIMi_CHn* input signals has changed. The *SPEi_DIR* signal is routed through the TSPPx subunit to implement the *T_DIR* or *S_DIR* signal.

  The TSPP0 subunit uses the outputs from TIM0 channels 0:2 to generate the *TRIGGER* signal.

  The TSPP1 subunit uses the outputs from TIM0 channels 3:5 to generate the *STATE* signal.

Another method to provide a direction signal to the DPLL is to route the TIM0 channel 6 input (TIM0_IN6) (instead of the DIR signal coming from TSSOP0) to the MAP's T_DIR output (set TSEL = 0).

15.2 TIM Signal Preprocessing (TSPP)

The TSPP combines the three 49-bit input streams coming from the TIM0 submodule and generates one combined 49-bit *TSPPO* output stream. The input stream combination is accomplished in the Bit Stream Combination (BSC) subunit. The architecture of the TSPP is shown in Figure 15-2.

15.2.1 TIM Signal Preprocessing (TSPP) subunit architecture

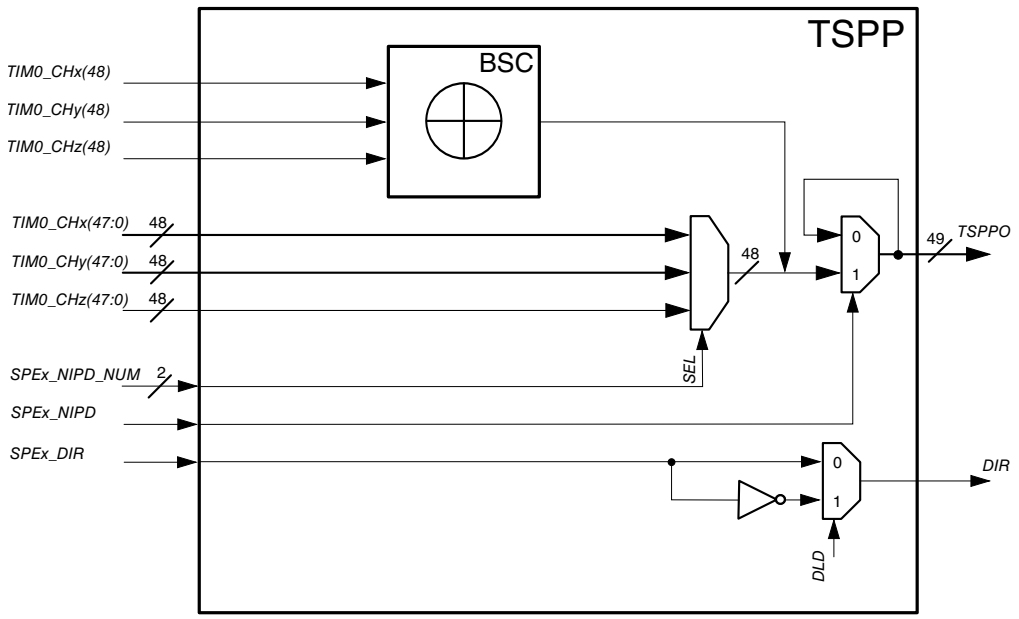


Figure 15-2. TSPP subunit architecture

15.2.2 Bit Stream Combination

The BSC subunit is used to XOR-combine the three most significant bits of the *TIM0_CHx(48)*, *TIM0_CHy(48)* and *TIM0_CHz(48)* inputs. The XOR-combined signal is merged with the remaining 48 bits of one of the three *TIM0_CHx(47…0)*, *TIM0_CHy(47…0)* or *TIM0_CHz(47…0)* input signals to generate the *TSPPO* signal. The input signal selection is based on the *SPEx_NIPD_NUM* input signal (coming from the SPE submodule). The action, when the 49 bits are transferred to the TSPPO and the T_VALID or S_VALID signal is raised, is determined by the SPEx_NIPD signal (coming from the SPE submodule). An example of *TSPPO* output signal generation is shown in Figure 15-3.

15.2.2.1 TSPP Signal generation for signal TSPPO

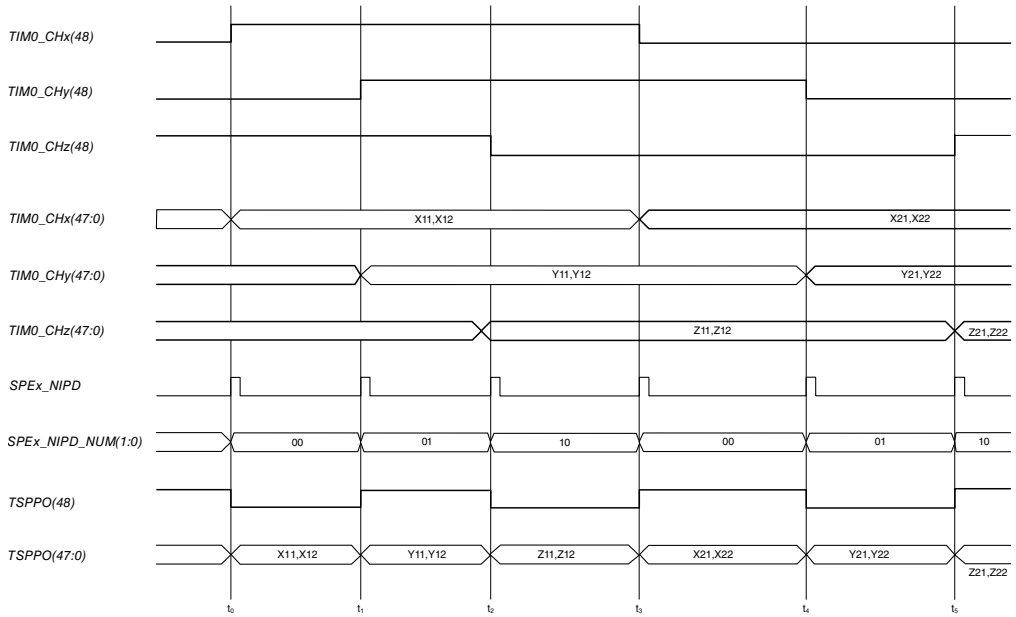


Figure 15-3. TSPP signal generation for the TSPPO signal

The *SPEx_NIPD_NUM* input signal determines which data is routed to the *TSPPO* signal. When the first edge of *TIM0_CHx(48)* is detected, the new data (X11 and X12) is routed to the *TSPPO(47:0)* signal. The X11 and X12 values are the two 24 bit values coming from the TIM0_CHx input. The next edge of the *TIM0_CHy(48)* signal is detected at time t_1. At time t_1, the level of the *TSPPO(48)* signal changes and the *TSPPO(47:0)* signal is set to Y11 and Y12, and so forth.

15.3 Memory Map and Registers

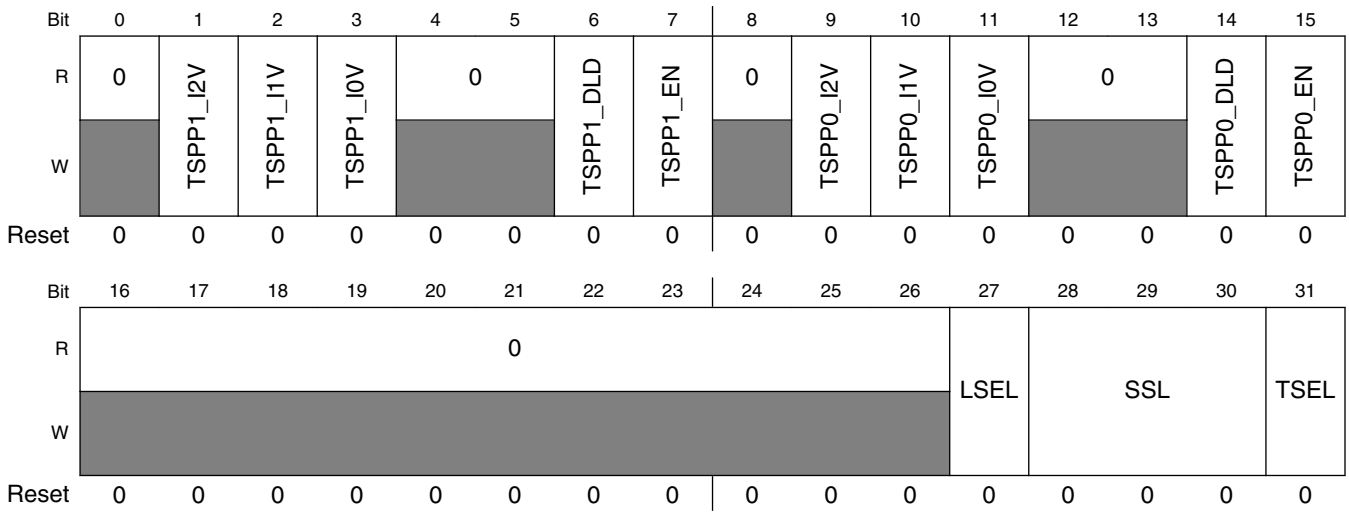
The TIM0 Input Mapping Module (MAP) register is described as follows:

MAP memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	TIM0 Input MAP Control Register (MAP_CTRL)	32	R/W	0000_0000h	15.3.1/472

15.3.1 TIM0 Input MAP Control Register (MAP_CTRL)

Address: F00h base + 0h offset = F00h



MAP_CTRL field descriptions

Field	Description
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 TSPP1_I2V	Disable of TSPP1 TIM0_CHz(48) input line. 0 Input line enabled. 1 Input line disabled; input for TSPP1 is set to zero (0).
2 TSPP1_I1V	Disable of TSPP1 TIM0_CHy(48) input line. 0 Input line enabled. 1 Input line disabled; input for TSPP1 is set to zero (0).
3 TSPP1_I0V	Disable of TSPP1 TIM0_CHx(48) input line. 0 Input line enabled. 1 Input line disabled; input for TSPP1 is set to zero (0).
4–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 TSPP1_DLD	DIR level definition bit. 0 SPEx_DIR signal is routed through as is. 1 SPEx_DIR signal is inverted.
7 TSPP1_EN	Enable of TSPP1 subunit. 0 TSPP1 disabled. 1 TSPP1 enabled.
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

MAP_CTRL field descriptions (continued)

Field	Description
9 TSPP0_I2V	Disable of TSPP0 TIM0_CHz(48) input line. 0 Input line enabled. 1 Input line disabled; input for TSPP0 is set to zero (0).
10 TSPP0_I1V	Disable of TSPP0 TIM0_CHy(48) input line. 0 Input line enabled. 1 Input line enabled.Input line disabled; input for TSPP0 is set to zero (0).
11 TSPP0_I0V	Disable of TSPP0 TIM0_CHx(48) input line. 0 Input line enabled. 1 Input line enabled.Input line disabled; input for TSPP0 is set to zero (0).
12–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 TSPP0_DLD	DIR level definition bit. 0 SPEx_DIR signal is routed through as is. 1 SPEx_DIR signal is inverted.
15 TSPP0_EN	Enable of TSPP0 subunit. 0 TSPP0 disabled. 1 TSPP0 enabled.
16–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 LSEL	TIM0_IN6 input level selection. 0 TIM0_IN6 input level '0' encodes TRIGGER in forward direction. 1 TIM0_IN6 input level '1' encodes TRIGGER in forward direction.
28–30 SSL	STATE signal output select. 000 TIM0_CH1 selected as STATE output signal. 001 TIM0_CH2 selected as STATE output signal. 010 TIM0_CH3 selected as STATE output signal. 011 TIM0_CH4 selected as STATE output signal. 100 TIM0_CH5 selected as STATE output signal. 101 TSPP1_TSPPO selected as STATE output signal. 110 TIM0_CH1 selected as STATE output signal. 111 TIM0_CH1 selected as STATE output signal.
31 TSEL	TRIGGER signal output select. 0 TIM0_CH0 selected as TRIGGER output signal. TIM0_IN6 (TIM0 chanel 6 input) is used as direction signal T_DIR. 1 TSPP0_TSPPO selected as TRIGGER output signal.

Chapter 16
Digital PLL Module (DPLL)

16.1 DPLL Overview

The digital PLL (DPLL) submodule is used for frequency multiplication. The DPLL provides a higher precision of position or value information in applications that have rapidly changing input frequencies. There are two input signals, TRIGGER and STATE, for which periodic events are processed. The time period between two valid events is called an increment, where each increment is divided by SUB_INC pulses into a given number of sub-increments. The resolution of the generated SUB_INC pulses is restricted by the period of the CMU_CLK0 clock (see CMU Overview) or the TS_CLK (see TBU Overview). The TRIGGER and STATE input signals can represent:

- position information of linear or angle motions,
- mass flow values,
- temperature values, or
- liquids level values.

The DPLL can reduce the load on the CPU by relieving it from doing repeated or periodic standard tasks.

The DPLL can perform tasks that:

- predict the duration of the current increment (see Prediction of current increment duration),
- generate SUB_INC1,2 pulses for up to two position counters in normal or emergency mode (see Sub pulse generation for SMC=0),
- synchronize the actual position (under CPU control (see Synchronization description),
- seamlessly switch (possibility) to emergency mode and back under CPU control (see DPLL_CTRL_0 register), and
- predict position and time related events (see Calculations for actions).

16.2 Requirements and demarcation

The two input signals *TRIGGER* and *STATE* can be sensor signals from the same engine (or system) or from two independent engines (or systems). When they come from the same engine (or system) the *TRIGGER* input is typically a more frequent signal and *STATE* is a less frequent signal. In such a case the *STATE* signal can support an emergency mode, when no *TRIGGER* signal is available. There are also applications supported when *STATE* and *TRIGGER* are independent signals from different engines (or systems). Both input signals are combined with a validation signal *T_VALID* or *S_VALID* respectively, which shows the appearance of new data and must result in a data fetch and a start of the correspondent state machine to perform the calculations (see explanation below).

When *STATE* is a redundant signal of the same engine (or system), only the *TRIGGER* input is used to generate the SUB_INC1 pulses in normal mode. There is a configuration possible, called emergency mode, for which the SUB_INC1 pulses are generated using the *STATE* input signal.

Because a need to switch to emergency mode can appear suddenly, the CPU should store (as a precaution) the information regarding the last increment durations of the *STATE* input (up to FULL_SCALE). The decision to switch to the emergency mode or back again is made by the CPU. It then configures the DPLL_CTRL_0[RMO] bit accordingly.

The MAP submodule generates the two TRIGGER and STATE input signals for the DPLL submodule by evaluating the output signals from:

- channels 0:5 of the TIM0 submodule, or
- channels of the SPE0 and/or SPE1 submodules.

A configurable filter algorithm is provided in the TIM0 channels to filter each slope and signal.

The Sensor Pattern Evaluation (SPE) submodule uses an anti-valence operation to evaluate three hall sensor inputs. SPE outputs are routed to the inputs of the MAP and TOM submodules.

TIM0 channel output signals are routed to the MAP. These 49-bit TIM0 channel output signals contain three fields, where:

- TIM0_MAP_DATAn(48) contains the signal level,

- TIM0_MAP_DATAn(47:24) contain the actual filter value TIM0_CH0_FLT_RE/ TIM0_CH0_FLT_FE, and
- TIM0_MAP_DATAn(23:0) contain the mark TIM0_MAP_DATAn valid for one clock cycle.

Because the delay conditions of the signals can change during an application, the filter delay value is output from the TIM0 module in the TIM0_MAP_DATAn(47:24) part of the output signal. Filter delays depend on the filter algorithms used. Only the effective filter delay can be considered in the DPLL.

In order to provide timing conditions to the DPLL, the *TRIGGER* and *STATE* input signals should have a time stamp (and an optional filter value and signal level value as stated above) with an appropriate resolution. The resolution of the time stamps can be either the same resolution as the TBU_TS0 input time base or eight times higher. The CPU configures the time stamp resolution by writing to the DPLL_CTRL_1[TS0_HRT, TS0_HRS] bits. The TBU_TS0 time base is used to predict future events, which are called actions.

A predefined number of pulses between each active slope of the *TRIGGER/STATE* signal is generated (if the corresponding pulse generator bit is set to one (enabled) in the DPLL_CTRL_1[SGEx] bit field) in the DPLL and routed to the SUB_INCx outputs. Depending on the configuration, different strategies can be used to correct an incorrect amount of pulses.

The FULL_SCALE range is divided into a fix number of nominal increments, which have the same size. The number of nominal increments in the HALF_SCALE range is configured by the CPU writing to the DPLL_CTRL_0[TNU, SNU] bit fields.

For synchronization purposes, some *TRIGGER*/*STATE* input signals (from detection of missing and/or additional teeth in a crank signal) can be suppressed, dependent on the current position. Therefore, an increment duration between two valid input events can be either one nominal increment or more than one nominal increment.

While a true nominal increment starts with a valid event, a virtual increment (always of nominal size) is an increment which starts from detection of missing and/or additional teeth in a crank signal. Each increment which represents a gap (e.g. for synchronization purposes), consists of exactly one true nominal increment and at least one virtual increment, where each have the same nominal duration (see Figure 16-1).

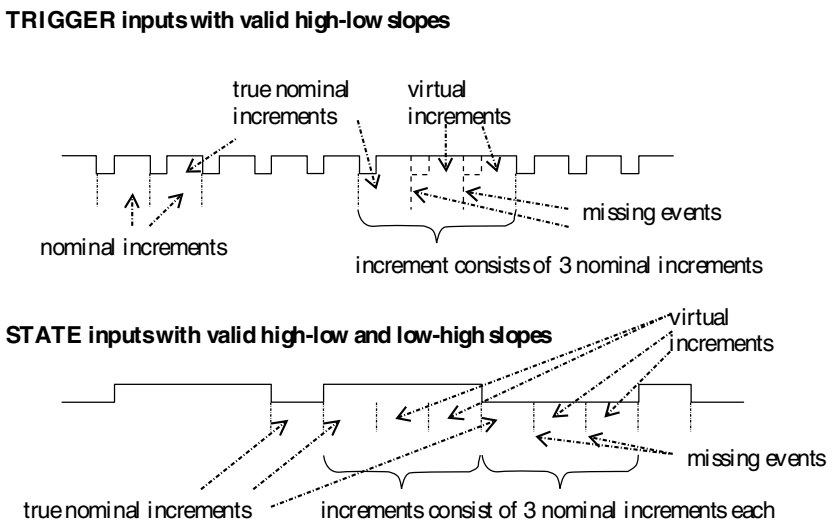


Figure 16-1. Input signal courses

16.3 DPLL block diagram and interface description

The DPLL block diagram is shown in Figure 16-2 and the DPLL interface description is shown in Table 16-1.

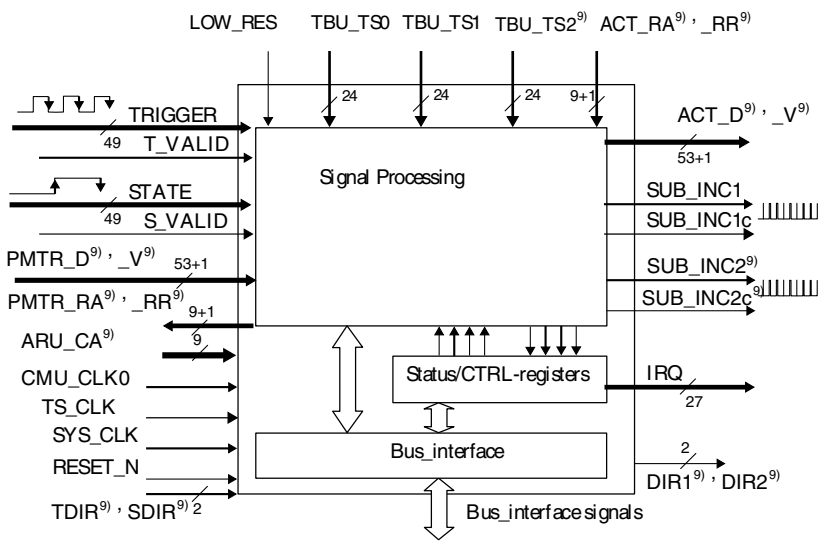


Figure 16-2. DPLL block diagram

Table 16-1. DPLL interface description

Name	Width	I/O	Description	Comment
TRIGGER	49	I	Normal Signal for triggering DPLL by positions/values Bit(48)= TRIGGER_S	One bit signal value (SV), 24 bits filter delay value info and 24 bits time stamp, filtered in different modes.

Table continues on the next page...

Table 16-1. DPLL interface description (continued)

Name	Width	I/O	Description	Comment
			Bits(47:24)= TRIGGER_FT Bits(23:0)= TRIGGER_TS	
T_VALID	1	I	The values of *TRIGGER* are valid	Announces the arrival of a new *TRIGGER* value
STATE	49	I	Assistance signal for synchronisation: STATE(48)= STATE_S STATE(47:24)= STATE_FT STATE(23:0)= STATE_TS	Replacement of signal *TRIGGER* for emergency situations, or signal from an independent device; bits like above, corresponding
S_VALID	1	I	The values of STATE are valid	Announces the arrival of a new STATE value
PMTR_D	53	I	Position minus time request data, delivered by ARU on request for up to 24 requests PMTR_RR; SV_i=PMTR_D(52:48): ACB bits, directly written to the correspondent DPLL_ACB_j registers PSA[i]=PMTR_D(47:24): position value for action DLA[i]=PMTR_D(23:0) time delay value for action	Data values for calculation of actual ACTIONs; the values are requested by AENi=1[1)]and CAIP=0[, 2)]; a served request is shown by PMTR_V which signals that valid PMTR data arrived and they are written immediately after that to the corresponding RAM regions and registers; The DLA[i] values must have the same resolution as the TBU_TS0 input.
PMTR_V	1	I	Signals a valid PMTR_D value, that means data is delivered on request	When valid: PMTR_D overwrites data in the PSA[i] and DLA[i] registers, also when the corresponding ACT_N[i][3)] bit =1;
ARU_CA	9	I	Channel address; for valid PMTR addresses: demand data by setting PMTR_RR=1 when enabled by AENi=1[1)] and CAIP=0[, 2)] ;	Counter value of ARU selects PMTR_RA and PMTR_RR when a valid address
PMTR_RA	9	O	Read address of PMTR access	Reflects ID_PMTR_i according to the selected channel address
PMTR_RR	1	O	Read request of PMTR access; suppressed for CAIPi=1 (see DPLL_STATUS register)	Reflects the value of the corresponding AENi[1)] bit while the correspondent bit CAIPi=0[, 2)]
ACT_D	53	O	Output of a time stamp, a position and a control signal for a calculated action; SV_i=ACT_D(52:48): ACB bits, directly written from the correspondent PMTR_D signals; ACT_D(47:24) is the calculated position value PSAC[i] for the action in relation to TBU_TS1 or 2 [6)] and ACT_D(23:0) is the time stamp value TSAC[i] for the action in relation to TBU_TS0 [, 6)]	Future time stamp with the resolution as TBU_TS0 input, additional position information and additional control bits;
ACT_V	1	O	ACT_D value is available and valid; blocking read access	For a valid action address: ACT_V reflects the shadow value of ACT_N[i][3)]

Table continues on the next page...

Table 16-1. DPLL interface description (continued)

Name	Width	I/O	Description	Comment
				(ACT_N[i] is 1 when new PMTR values are available and the shadow register is updated, when a calculation of the actual PMTR values was done); reset after reading of the ACT_D values
ACT_RA	9	I	ACTION read address	Address bits for selection of all 24 action channels
ACT_RR	1	I	Read request of selected action	The action data is demanded from an other module
IRQ	27	O	Interrupt request output	Interrupts from DPLL
SUB_INC1	1	O	Pulse output for *TRIGGER* input filter	Sub-position increment provided continuously
SUB_INC2	1	O	Pulse output for *STATE* input filter (when *TRIGGER* and *STATE* are used for 2 independent devices)	Sub-position increment provided continuously
SUB_INC1c	1	O	Pulse output for time base unit 1 in compensation mode (can stop in automatic end mode)	Sub-position increment related to *TRIGGER* input
SUB_INC2c	1	O	Pulse output for time base unit 2 in compensation mode (can stop in automatic end mode)	Sub-position increment related to *STATE* input (when *TRIGGER* and *STATE* are used for 2 independent devices)
TS_CLK	1	I	Time stamp clock	Used for generation of the time stamps; this clock is used to generate the SUB_INC1,2 pulses
CMU_CLK0	1	I	CMU clock 0	used for rapid pulse correction of SUB_INC1,2
SYS_CLK	1	I	System clock	High frequency clock
RESET_N	1	I	Asynchronous reset signal	Low active; After Reset he DPLL is available only after performing the RAM reset procedures by the DPLL hardware.
LOW_RES	1	I	Low resolution of TBU_TS0 selected; shows which of the 27 bits of TBU_TS0 are connected to the DPLL	LOW_RES=0: TBU_TS0(DPLL)= lower 24 Bits of TBU_TS0(TBU); LOW_RES=1: TBU_TS0(DPLL)= higher 24 bits of TBU_TS0(TBU); For LOW_RES=1, the TS0_HRT and/or TS0_HRS bits can be set[5)]
TBU_TS0	24	I	Actual time stamp from TBU; needed to decide if a calculated action is already in the past	24-bit time input, with a resolution of the time stamp clock
TBU_TS1	24	I	Actual position/value stamp 1; for calculation of position stamps (*TRIGGER*/*STATE*)	24-bit position/value input, with a resolution of the SUB_INC1 pulses for calculation of position stamps (*STATE*) for SMC[5)]=RMO,[4)]=1
TBU_TS2	24	I	Actual position/value stamp 2; to be implemented for an additional independent position	24-bit position/value input, with a resolution of the SUB_INC2

Table continues on the next page...

Table 16-1. DPLL interface description (continued)

Name	Width	I/O	Description	Comment
				for calculation of position stamps (*STATE*) for SMC[5)]=RMO,[4)]=1
TDIR	1	I	Direction of *TRIGGER* input values (TDIR=0 does mean a forward direction and TDIR=1 a backward direction)	Direction information from multiple sensors valid only for SMC[5)]=1
SDIR	1	I	Direction of *STATE* input values (SDIR=0 does mean a forward direction and SDIR=1 a backward direction)	Direction information from multiple sensors valid only for SMC[5)]=1
DIR1	1	O	Direction information of SUB_INC1 (count forward for DIR1=0 and backward for DIR1=1)	Count direction of TBU_TS1; DIR1 changes always after the evaluation of the corresponding valid *TRIGGER* slope and after incrementing/decrementing of the address pointer
DIR2	1	O	Direction information of SUB_INC2 (count forward for DIR2=0 and backward for DIR2=1)	Count direction of TBU_TS2; DIR2 changes always after the evaluation of the corresponding valid *STATE* slope and after incrementing/decrementing of the address pointer

1). see DPLL_CTRL_x register, x=2,3,4
2). see DPLL_STATUS register
3). see DPLL_ACT_STA register
6). see DPLL input signal description
5). see DPLL_CTRL_1 register
5). See DPLL_CTRL_1 register
4). See DPLL_CTRL_0 register

16.4 DPLL Architecture

16.4.1 Purpose of the module

The DPLL submodule generates a predefined number of incremental signal pulses within the period between two events of an input *TRIGGER* or *STATE* signal (when the corresponding pulse generator is enabled). The resolution of the pulses is restricted by the frequency of the TS_CLK time stamp clock. To get the same number of pulses, changes in the period length of the predicted time period of the current increment require a change of the pulse frequency. This frequency change can be performed by DPLL hardware, software, or with the support of DPLL hardware (in different modes).

The basic purpose of the DPLL is to make a prediction of the current period between two *TRIGGER* and/or *STATE* signal edges. Disturbances and systematic failures must be considered as well as changes of increment durations caused by acceleration and

deceleration of the supervised process. Therefore, a good prediction uses some measured values from the past. When the process is steady and differentiable, the current increment and some future increments can be predicted.

16.4.2 Explanation of the prediction methodology

The basic function of all DPLL tasks is the prediction of the current increment, which is based on a relation between past increments. Because the relation between two succeeding intervals at a fixed position remains also valid in the case of acceleration or deceleration, the prediction of the duration of the current time interval is accomplished by a similarity transformation. Having a good estimation of the current time interval, all the other DPLL tasks can be easily accomplished by calculations explained in Prediction of current increment duration.

16.4.3 Clock topology

All registers are read using the *SYS_CLK* system clock. The SUB_INC1,2 pulses that are generated have (in the normal case) the highest frequency not higher then CMU_CLK0 or half of TS_CLK respectively. For individual pulses, the frequency can be doubled. All operations can be performed using the system clock.

16.4.4 Clock generation

The clock is generated outside of the DPLL.

16.4.5 Typical frequencies

For the system clock, a reasonable clock frequency should be applied to give the DPLL module sufficient computational power to calculate all needed values (prediction of next increment, actions) in time. The typical system clock frequency is in the range from 40 MHz up to 150 MHz.

16.4.6 Time stamps and systematic corrections

The time stamps for the *TRIGGER* and *STATE* input signals have a width of 24 bits each. These bits represent the value of the 24-bit free running counter, which runs with a clock frequency that is configured in the TBU submodule. Using a typical frequency of 20 MHz, each time stamp has a 50 ns resolution.

The *TRIGGER* and *STATE* input signals have to be filtered in the TIM0 submodule before they are routed, via the MAP submodule, to the DPLL. The time stamps can have a delay caused by the filter algorithm that is used. Delayed and un-delayed filter algorithms are available. The delay value can depend on a time or a position value.

Systematic deviations of the *TRIGGER* inputs can be corrected by a profile, which also considers systematic missing *TRIGGER*s. The increments containing missing *TRIGGER*s are divided into the corresponding number of nominal increments, where the duration of a nominal increment is the greatest divider of all increment durations.

For each increment, the number of enclosed nominal increments is stored in a profile as an NT value for *TRIGGER*. When the increment is a nominal increment, the NT value is 1.

The NT value for the *TRIGGER* input is stored in the ADT_T field of RAM region 2c.

If Adapt Mode TRIGGER (AMT) is configured by setting the DPLL_CTRL_0(AMT) bit = 1, the **ADT_T[i]** values in RAM region 2c must also contain the adapting information for the *TRIGGER* signal, which considers (for each increment) a systematic physical deviation (PD) from the perfect increment value with a resolution according to the chosen value of MLT+1 (which describes the number of SUB_INC1 pulses for a nominal increment).

The **PD** value (sint13 value) for the *TRIGGER* describes the amount of missing or surplus pulses, which are to be directly added to MLT+1. The correction value is (in this way) also applicable in the case of missing *TRIGGER* inputs for the synchronization gaps, where the amount of SUB_INC1 pulses that are provided for a nominal increment (MLT +1) is multiplied by NT first before the PD value is added.

The NT value of the current increment is stored in the DPLL_NUTC[SYN_T] bit field.

If Reference Mode (RM) is configured to Emergency Mode by setting the DPLL_CTRL_0(RMO) bit = 1 and Synchronous Motor Control (SMC) is configured by setting the DPLL_CTRL_1(SMC) bit = 1, the time stamp of the *STATE* input is used to generate the SUB_INC1 output.

In emergency mode, more inaccuracy is acceptable because there are usually less events available for FULL_SCALE, according to the State Number that is in the DPLL_CTRL_0[SNU] bit field.

For the *STATE* input, the systematic deviations of the increments can be corrected in the same way as for the *TRIGGER* input. That is, by profile and adaptation information that is described below.

Systematic deviations of *STATE* inputs can be corrected by a profile, which also considers systematic missing *STATE* events. The increments containing missing *STATE*s are divided into the corresponding number of nominal increments, where the duration of a nominal increment is the greatest divider of all increment durations.

For each increment, the number of enclosed nominal increments is stored in a profile as an NS value for *STATE*. When the increment is a nominal increment, the NS value is 1.

For the *STATE* input, the NS value is stored in the ADT_S field of RAM region 1c3.

If Adapt Mode STATE (AMS) is configured by setting the DPLL_CTRL_0[AMS] bit = 1, the **ADT_S[i]** values in RAM region 1c3 must contain the adapting information for the STATE signal, which considers (for each increment) a systematic physical **PD_S** deviation from the perfect increment value with a resolution according to the chosen value of MLS1 (which describes the number of SUB_INC1 pulses for a nominal increment).

In emergency mode (for SMC=0) and to get the same number of pulses in FULL_SCALE for normal and emergency mode, the amount of SUB_INC1 pulses for a nominal *STATE* increment is given by

MLS1= (MLT + 1) * (TNU + 1) / (SNU + 1).

This value has to be configured by the CPU.

The **PD_S** value (sint16 value) for the *STATE* input describes the amount of missing or surplus pulses that are to be directly added to MLS1. The correction value is (in this way) also applicable to missing *STATE* inputs for synchronization gaps. In this case, the amount of SUB_INC1 pulses that are provided for a nominal increment (MLS1) is multiplied by NS before the PD_S value is added.

The current NS value is stored in the DPLL_NUSC[SYN_S] bit field.

16.4.7 DPLL Architecture overview

The DPLL can process different input signals. The *TRIGGER* signal is the normal input signal, which provides detailed information about the supervised process. It can be, for example, information about the volume of water or other liquid levels, where each increasing millimeter during the filling process results in generation of the *TRIGGER* input signal. In order to get a predefined filling level, without overflow, the inertia of the

system must be taken into account. Therefore, some delay that accounts for the remaining amount of water in the pipe must be considered so that the inlet valve can be closed before the desired filling level is reached.

The *STATE* input signal provides additional (redundant) information. For example, at some fill level (in centimeters) and because of intervals with different distances, it provides information about the system state regarding the in or out direction of the water flow. In some applications, the inactive slope of the *TRIGGER* signal can be utilized to provide direction information. In the case of faults in the *TRIGGER* signal, the *STATE* signal can be processed (with some loss of accuracy) in order to reach the desired value.

The measuring scale can have systematic failures because not all measured distances (in millimeters or centimeters) are as accurate as desired. This inaccuracy could be due to changes in the thickness of the measuring cylinder or inaccurate mark positions. However, these systematic failures are well known by the system and, for improvement of the prediction, the *ADT_T* and *ADT_S* signals (stored in the internal RAM) are provided for correction of systematic *TRIGGER* and *STATE* signal failures.

The *TRIGGER* and *STATE* input signals are each represented by a time stamp signal, which is stored in the TS-part (24 bits) of the corresponding signal. Information concerning the delay of the signal by filtering of disturbances is stored in the FT-part (24 bit) of the signal.

In order to establish the relationship of time stamps to the actual time, the TBU_TS0 time stamp value is provided to show the actual time value that is used for the prediction of future actions.

After reaching the desired water level, the water is is drained into a bottle. After that, the water filling process is repeated. The water level at draining is observed by the same sensor signals (the same number of *TRIGGER* pulses), but the duration of the draining could be different from the filling time. Both times together form the FULL_SCALE region, while one of them is a HALF_SCALE region, which can differ in time but not in the number of pulses, especially for *TRIGGER*.

For synchronization purposes, some *TRIGGER* marks can be omitted in order to set the system to a proper synchronisation value (maybe before the upper filling value is reached).

In emergency situations, when *TRIGGER* signals are missed, the *STATE* signal is used instead.

The PMTR_i signals announce the request for a position minus time calculation for up to 24 events.

All 24 events can be activated by setting the 12 DPLL_CTRL_3[AENn] and 12 DPLL_CTRL_4[AENn] action enable bits to 1. Each of these enable bits are asked by the routing engine for a read access. The corresponding read access is generated by when the AENi bit is set to 1 while the corresponding DPLL_STATUS[CAIP1, CAIP2] bit = zero, where CAIP1 = 0 and/or CAIP2 = 0 indicates that "calculation of actions in progress" is being controlled by the state machine (see Requirements and demarcation) for scheduling the operations.

When such a request is serviced by the ARU submodule (if CAIPn = 0), the values for position and time are written to the corresponding RAM 1a region (0x0200…0x025C for the position value and 0x0260…0x02BC for the delay value) and the control bits for the corresponding action are set accordingly. When a new PMTR value arrives, the old value is overwritten without notice and the shadow bit of ACT_N[i] is cleared while the ACT_N[i] (new action) bit is set in the DPLL_ACT_STA register.The ACT_N[i] bit is cleared, when the currently calculated action value is in the past. Overwriting old information is possible without data inconsistency because the read request to the ARU is suppressed during action calculations by the CAIP1,2 bits. Therefore, the last PMTR value is used consistently.

16.4.8 DPLL Architecture description

The DPLL block diagram will now be explained in combination with some example configurations of the control registers. There are different configuration bits available which can adapt the DPLL to the use case.

For example:

- let the*TRIGGER* number (TNU) be configured by setting the DPLL_CTRL_0[TNU] bit field = 0x3B (TNU+1 = 0x3B + 0x1 = 0x3C = 60 (decimal) events for half scale and 120 (decimal) events for FULL_SCALE), and
- and let the number of SUB_INC1 pulses between two *TRIGGER*s be configured by setting the TRIGGER multipiler (MLT) in the DPLL_CTRL_0[MLT] bit field = 0x257 (600 pulses per *TRIGGER* event).

The FULL_SCALE region can be divided into 72000 parts, each of them associated with its own SUB_INC1 pulse. For a run through FULL_SCALE, all 72000 pulses should appear with the possibility of a different pulse frequency between two *TRIGGER* events. After each 600 pulses at the *SUB_INC1* output, the next *TRIGGER* event is expected (with a new time stamp).

Missing SUB_INC1 pulses, due to acceleration, have to be taken into account within the next increment. Not one pulse has to be missed or added because of calculation inaccuracy in average for a sufficient number of FULL_SCALE periods. This means that not one pulse is sent in addition and all missing pulses are to be caught up on afterwards.

For the systematic arrangement of *TRIGGER* inputs, **the profile** (described in Time stamps and systematic corrections) is stored in the Adapt Values for all TRIGGER Increments Register (ADT_Tn), which is located in RAM region 2c. The relative position of gaps (NT) and physical deviations (PD) can be stored in the ADT_Tn[NT, PD] bit fields.

For the consideration of systematic missing *TRIGGER*s, the actual NT value of the profile is stored in the DPLL_NUTC[SYN_T] bit field.

In normal mode, the physical deviation (PD) values in the ADT_Tn[PD] bit field could be used to balance the local systematic inaccuracy of the *TRIGGER* signal, where the PD value is the pulse difference in the corresponding increment,, i.e. the number of sub pulses to be added to the nominal number of pulses. The PD bit field contains a signed 13-bit integer value for up to +/-4096 pulses, which can be added for each increment.

When a nominal increment is assumed, the NT value in the ADT_Tn[NT] bit field of the profile has the value 1. An integer number > 1 is the number of nominal increments that are to be considered for a gap. For the actual increment after synchronization, the corresponding NT value is stored in the DPLL_NUTC[SYN_T] bit field.

Using the *STATE* input, similar configuration bits are available. For example:

- let the*STATE* number (SNU) be configured by setting the DPLL_CTRL_0[SNU] bit field = 0xB (SNU + 1 = 0xB = 0x1 = 0xC = decimal 12), and
- let the sum of all systematic missing STATE events be configured for HALF_SCALE by setting the DPLL_CTRL_1[SYSF] bit = 0.

This configuration handles 12 events in HALF_SCALE and 24 events in FULL_SCALE. In order to get the same number of SUB_INC1 pulses for FULL_SCALE (as above for *TRIGGER*s), the value

(MLT + 1 = 600) is divided by (2 * (SNU + 1) = 24) and multiplied by (2 * (TNU + 1) = 120) to give 3000. The CPU must store the value 3000

The CPU must store the result, 3000, in the MLS1 bit field of the Calculated Number of Sub-Pulses Between Two STATE Events Register (MLS1), which is located in RAM Region 1b.

For the systematic arrangement of *STATE* inputs, **the profile** is stored the Adapt Values STATE n Increments Register (ADT_Sn), which is located in RAM region 1c3 . The relative position of gaps can be stored in the NS bit field and the physical deviations can be stored in the PD_S bit field.

For the consideration of systematic missing *TRIGGER*s, the actual NS value of the profile is stored in the DPLL_NUSC[SYN_S] bit field .

In emergency mode, the physical deviation values in the ADT_S[PD_S] bit field can be used to balance the local systematic inaccuracy of the *STATE* signal. The PD_S value shows the pulse difference in the corresponding increment, i.e. the number of sub pulses to be added to the nominal number of pulses. The PD_S value is a signed 16-bit integer for up to +/-32768 pulses, which can be added for each increment.

When a nominal increment is assumed, the ADT_S[NS] bit field of the profile contains the value 1. An integer number > 1 shows the number of nominal increments to be considered for a gap. For the actual increment after synchronization, the corresponding NS value is stored the DPLL_NUSC[SYN_S] bit field.

16.4.9 Block diagrams of time stamp processing

The time stamp difference of two successive TRIGGER input events is calculated as shown in Figure 16-3.

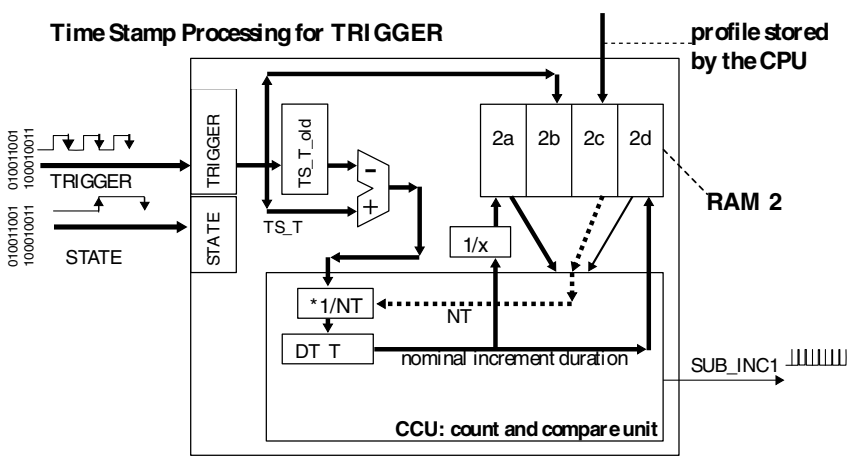


Figure 16-3. Time stamp processing for TRIGGER

For the prediction of the current increment duration, values from the past are used. For this purpose, the measured and calculated values of the last FULL_SCALE period are stored in RAM Region 2. For the *TRIGGER* inputs, RAM Region 2 is divided into four partitions as shown in Table 16-2.

Table 16-2. RAM Region 2 partitions

2a	Stores reciprocals of each nominal increment duration RDT_T
2b	Stores time stamps of each valid input event TSF_T
2c	Used for profile ADT_T
2d	Used for nominal increment durations DT_T

The prediction can be calculated by the multiplication of increment duration values with the reciprocal value of another increment. In order to not be forced to distinguish between gaps and "normal" increment durations or gaps only, the nominal duration and the corresponding reciprocal values are stored in the RAM Region 2. This is possible by consideration of the NT value in the profile: the measured increment duration is divided by NT.

The time stamp difference of two successive STATE input events is calculated as shown in Figure 16-4.

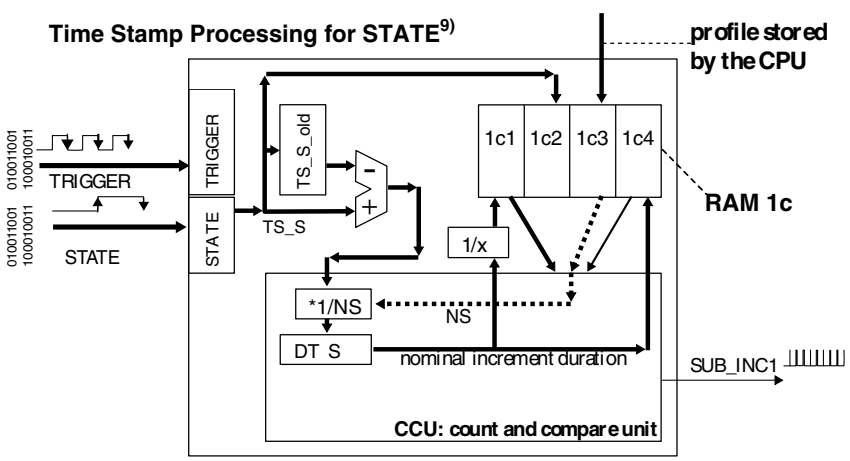


Figure 16-4. Time stamp processing for STATE

For the *STATE* inputs, RAM Region 1c is divided into four partitions as shown in Table 16-3.

Table 16-3. RAM Region 1 partitions

1c1	stores reciprocals of each nominal increment duration RDT_S
1c2	stores time stamps of each valid input event TSF_S
1c3	used for profile ADT_S
1c4	used nominal increment durations DT_S

The calculations are performed like those for the *TRIGGER* input events. The NS value in the profile shows the appearance of a gap.

16.4.10 Register and RAM address overview

The DPLL address map is divided into register and memory regions as defined in Table 16-4. The addresses from 0x0000 to 0x00FC are reserved for registers; the addresses from 0x0100 to 0x01FC are reserved for action registers that immediately serve the ARU for read requests.

The RAM is divided into three independent and accessible regions: 1a, 1b+c, and 2.

Region 1a has a size of 288 bytes and its addresses range from 0x0200 through 0x037C. These region 1a locations are used for PMTR values from the ARU and for intermediate calculation values.

NOTE

The CPU cannot access the RAM when the DPLL is enabled.

Region 1b addresses range from 0x0400 through 0x05FC. These region 1b locations are reserved for RAM variables. Region 1c addresses range from 0x0600 through 0x09FC. These region 1c locations are used for *STATE* signal values. Regions b+c together have a size of 1,125 Kbytes.

Region 2 has a size that is configurable from 1,5 to 12 Kbytes (depending on the number of *TRIGGER* events in FULL_SCALE) and its addresses range from 0x4000 through 0x7FFC. These region 2 locations are reserved for the *TRIGGER* signal values.

Region 2 is subdivided into four partitions: 2a, 2b, 2c, and 2d. The DPLL_AOSV_2 register is used to determine the beginning of each partition.

16.4.10.1 Register and RAM address map

Registers are used to control the DPLL and to show its status. Parameters are stored in registers. Table 16-4 shows the addresses for status and control registers as well as values stored in additional registers.

Time stamps for TRIGGER and *STATE* can have either the same resolution as the TBU_TS0 time stamp input or eight times higher. The resolution is configured by writing to the DPLL_CTRL_1[TS0_HRT] bit field. While the TBU_TS0 time stamp is used for action predictions, the higher resolution of *TRIGGER* and *STATE* inputs can be used for a more accurate pulse generation.

The time stamp fields of the *TRIGGER* and *STATE* inputs are stored in the corresponding RAM regions, where virtual increments are provided for gaps. Virtual increments for gaps are necessary in order to calculate time differences between a given number of (real and virtual) input events, independent of a gap. Therefore, the gap is extended in RAM 2b and RAM 1c2.

For all other regions in RAM 2 and RAM 1c, the gap is considered as one increment.

Address pointers are used to access RAM fields. When the GTM starts, all address pointers have a zero value and the first measured and calculated values are stored in the beginning of the corresponding RAM field.

Because the position of the device is usually unknown at the beginning, profile information is not available. The profile regions must have their own address pointers, which are configured by the microcontroller as soon as the position is known. The DPLL_STATUS[SYT, SYS] synchronization bits are set by configuring the appropriate value for the APT_2c address pointer (DPLL_APT_2c[APT_2c] bit field) for the *TRIGGER*profile or the APS_1c3 address pointer (DPLL_APS_1c3[APS_1c3] bit field) for the *STATE* profile.

Because the time stamp fields are extended at the gaps, the APT_2b address pointer (DPLL_APT_2b[APT_2b] bit field) is provided for *TRIGGER* time stamps and the APS_1c2 address pointer (DPLL_APS_1c2[APT_1c2] bit field) is provided for *STATE* time stamps. These address pointers must be incremented by NT or NS, respectively, when a gap appears.

Table 16-4. DPLL register and RAM address map overview

Addr. range Start	Addr. range End	Value number	Byte #	Content	Indication	Region	RAM size
0x0000	0x0FC	64	256	Register	used/reserved	0	no RAM
0x100	0x1FC	64	192	ACTION registers	direct read from ARU	0	no RAM
0x0200	0x037C	96	288	PMTR values RAM 1a	CPU R/Pw access, when DPLL disabled; ARU has highest priority	1a with own ports	RAM part 1a: 288 bytes
0x0400	0x05FC	128	384	Variables RAM 1b	R and monitored W access by the CPU	1b	RAM part 1b+c: 1,125 Kbytes
0x0600	0x09FC	256	768	*STATE* data	R and monitored W access by the CPU	1c	
0x0600	0x06FC	64	192	RDT_S[i]	*STATE* reciprocal values	1c1	

Table continues on the next page...

Table 16-4. DPLL register and RAM address map overview (continued)

Addr. range Start	Addr. range End	Value number	Byte #	Content	Indication	Region	RAM size
0x0700	0x07FC	64	192	TSF_S[i]	*STATE* TS values	1c2	
0x0800	0x08FC	64	192	ADT_S[i]	adapt values of *STATE*	1c3	
0x0900	0x09FC	64	192	DT_S[i]	nom. STATE inc	1c4	
0x4000	0x47FC… 0x7FFC	512 … 4096	1536 … 12288	TRIGGER data	R and monitored W access of CPU	2	RAM part 2: 1,5… 12 Kbytes
0x4000	0x41FC… 4FFC	128… 1024	384…3072	RDT_T[i]	TRIGGER reciprocal values	2a	
0x4200… 5000	0x43FC… 5FFC	128… 1024	384…3072	TSF_T[i]	*TRIGGER* TS values	2b	
0x4400… 6000	0x45FC… 6FFC	128… 1024	384…3072	ADT_T[i]	adapt values of *TRIGGER*	2c	
0x4600… 7000	0x47FC… 7FFC	128… 1024	384…3072	DT_T[i]	nom. *TRIGGER* increments	2d	

16.4.10.2 RAM Region 1

RAM region 1 has a size of 1.5 Kbytes. It is used to store variables and parameters as well as the measured and calculated values for increments of *STATE*. RAM region 1 is divided into two independent RAM parts (a and b+c), each with its own access port. The address information is shown in Table 16-4 and detailed descriptions are provided in subsequent DPLL sections.

- **RAM Region 1a:** Used to store the PMTR values that are read from the ARU and to store intermediate calculation results of actions; read and write access by the CPU is possible only when the DPLL is disabled; address range is 0x0200 through 0x037C.
- **RAM Region 1b:** Used for intermediate calculations and auxiliary values; data width of three bytes is used for 24-bit values; a write access results in assertion of an interrupt request to the CPU (when enabled); address range is 0x0400 through 0x05FC.
- **RAM Region 1c**: Used to store time stamps, the profile, and durations for all *STATE* inputs of the last FULL_SCALE region; data width of three bytes is used for 24-bit values; a write access results in assertion of an interrupt request to the CPU (when enabled); address range is 0x0600 through 0x09FC.

The RAM is initialized by the DPLL after a HW-reset. After performing the initialization procedure by setting the DPLL_RAM_INI[Init_RAM] bit, all RAM cells must have a zero value. The DPLL is not available until completion of the initialization procedure. Initialization progress is shown in the INIT_n status bits of the same register.

RAM regions 1c1, 1c3 and 1c4 have **2 * (SNU + 1 – SYN_NS)** entries for SYSF = 0 or **2 * (SNU + 1) – SYN_NS** entries for SYSF = 1 (see DPLL_CTRL_0[SNU] and DPLL_CTRL_1[SYN_NS]). RAM region 1c2 has **2 * (SNU + 1)** entries. For the latter, virtual events are considered, that is, the gap is divided into equidistant parts that have the same position share as increments without a gap. For this reason, the CPU must extend the stored TSF_S[i] values in RAM region 1c2 before the APS_1c3 address pointer is configured (by writing to the DPLL_ APS_1c3[APS_1c3] address pointer bit field). Writing to the APS_1c3 bit field results in the setting of the DPLL_STATUS[SYS] bit, which marks the end of the synchronization process. After the SYS bit is set, the PMTR values can consider more than the last increment duration for the action prediction by setting the DPLL_NUSC[NUSE] bit field to a corresponding value.

The address pointers for RAM region 1c are shown in Figure 16-5.

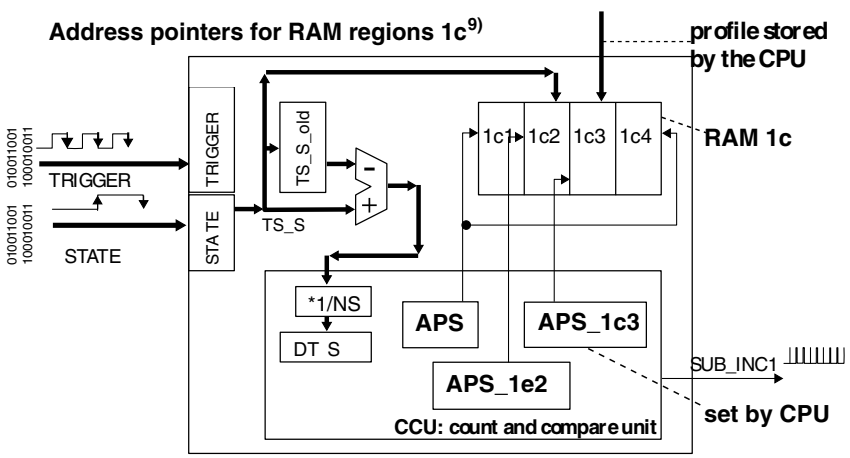


Figure 16-5. Address pointers for RAM regions 1c

The APS address pointer points to RAM regions 1c1 and 1c4. The APS_1c2 address pointer points to the time stamp field in RAM region 1c2. This is necessary because, in the time stamp field, the gaps are extended to the number of nominal increments (see explanation above and in Synchronization description).

The APS_1c3 profile address pointer is configured by the CPU when the position is known, and therefore the relationship to the other address pointers is calculated. This configuration action results in synchronization of the RAM fields to each other. Completion of the synchronization is indicated when the DPLL_STATUS[SYS] bit is set to 1.

16.4.10.3 RAM Region 2

RAM region 2 has a configurable size of 1.5 to 12 Kbytes. It is used to store measured and calculated values for increments of *TRIGGER*. Address information is shown in Table 16-4.

Up to 512 *TRIGGER* events can occur in HALF_SCALE. Therefore, RAM region 2a, b, c, and d must each have up to 1024 storage locations. Considering 3-byte word sizes, RAM region 2 provides up to12 Kbytes of storage.

In order to save RAM size for configurations that have fewer *TRIGGER* events, the RAM size can be configurable by writing to the Offset Switch Register (OSW) at 0x2801C (DPLL_OSW[OSS] bit field) and the Address Offset RAM2 Register (AOSV2) at 0x28020 (DPLL_AOSV2[AOSV_2d, AOSV_2c, AOSV_2b, AOSV_2a] bit fields. After HW-reset, the DPLL must initialize all RAM location to 0. The DPLL is available only after completion of the initialization procedure.

RAM regions 2a, 2c and 2d each have **2 * (TNU + 1 – SYN_NT)** entries. RAM region 2b has **2 * (TNU + 1)** entries (see DPLL_CTRL_0[TNU] and DPLL_CTRL_1[SYN_NT] bit fields). For the latter, virtual events are considered, that is, the gap is divided into equidistant parts where each has the same position share as increments that do not have a gap. For this reason, the CPU must extend the stored TSF_T[i] values in RAM region 2b before the APT_2c address pointer is configured.

Configuring the APT_2c address pointer results in the DPLL_STATUS[SYT] bit being set to 1, which indicates the end of the synchronization process. After the SYT bit is set to 1, the PMTR values can consider more than the last increment duration for the action prediction by setting the DPLL_NUTC[NUTE] bit field to a value > 1.

Up to 512 TRIGGER events can occur in HALF_SCALE. Therefore, RAM region 2a, b, c, and d must each have up to 1024 storage locations. Considering 3-byte word sizes, RAM region 2 provides up to12 Kbytes of storage.

In order to save RAM size for configurations that have fewer TRIGGER events, the RAM size can be configurable by writing to the Offset Switch Register (OSW) at 0x2801C (DPLL_OSW[OSS] bit field) and the Address Offset RAM2 Register (AOSV2) at 0x28020 (DPLL_AOSV2[AOSV_2d, AOSV_2c, AOSV_2b, AOSV_2a] bit fields. After HW-reset, the DPLL must initialize all RAM location to 0. The DPLL is available only after completion of the initialization procedure. RAM regions 2a, 2c and 2d each have 2 * (TNU + 1 – SYN_NT) entries. RAM region 2b has 2 * (TNU + 1) entries (see DPLL_CTRL_0[TNU] and DPLL_CTRL_1[SYN_NT] bit fields). For the latter, virtual events are considered, that is, the gap is divided into equidistant parts where each has the same position share as increments that do not have a gap. For this reason, the CPU must

extend the stored TSF_T[i] values in RAM region 2b before the APT_2c address pointer is configured. Configuring the APT_2c address pointer results in the DPLL_STATUS[SYT] bit being set to 1, which indicates the end of the synchronization process. After the SYT bit is set to 1, the PMTR values can consider more than the last increment duration for the action prediction by setting the DPLL_NUTC[NUTE] bit field to a value 1.

The address pointers for RAM region 2 are shown in Figure 16-6

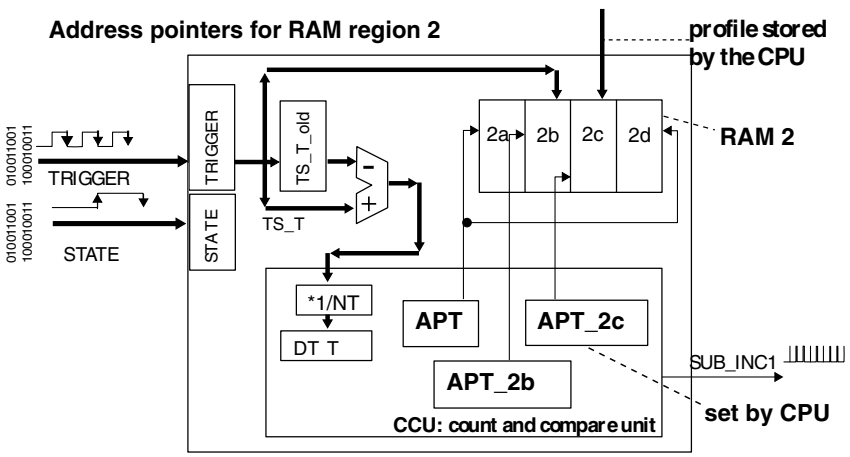


Figure 16-6. Address pointers for RAM region 2

The APT address pointer points to the RAM regions 2a and 2d. The APT_2b address pointer points to the time stamp field in region 2b. This is necessary because, in the time stamp field, the gaps are extended to the number of nominal increments (see explanation above and in Synchronization description).

The APT_2c profile address pointer is configured by the CPU when the position is known. Therefore, the relation to the other address pointers is calculated. Configuring this APT_2c profile address pointer results in synchronization the RAM fields to each another. Synchronization is complete when the DPLL_STATUS[SYT] bit = 1.

16.5 Prediction of current increment duration

16.5.1 The use of increments in the past

Past values to be considered for the prediction of TRIGGER

In order to take into account values of increments for *TRIGGER*s from the past, the DPLL_NUTC[NUTE] bit field value must be configured to determine the number of past values. The DPLL_NUTC[VTN] bit field contains a value for the number of virtual increments in the NUTE region. Because gaps come into the NUTE region or leave it, the VTN value must be updated by the CPU until the NUTE value is set to HALF_SCALE or FULL_SCALE. For RAM regions 2a and 2d, the NUTE – VTN value is to be considered; for RAM region 2b, only the NUTE value is to be considered. This is due to the fact that time stamp entries in a gap are extended to the number of nominal increments, but duration entries are not.

Past values to be considered for the prediction of STATE

In order to take into account values of increments for *STATE* from the past, the DPLL_NUSC[NUSE] bit field value must be configured to determine the number of past values. The DPLL_NUSC[VSN] bit field contains a value for the number of virtual increments in the NUSE region. Because gaps come in to the NUSE region or leave it, the VSN bit field value must be updated by the CPU until the NUSE bit field value is set to HALF_SCALE or FULL_SCALE. For RAM regions 1c1 and 1c4 (in the past), the NUSE – VSN value is to be considered; for the RAM region 1c2, only the NUSE value is to be considered. This is due to the fact that time stamp entries in a gap are extended to the number of nominal increments, but duration entries are not.

16.5.2 Increment prediction in Normal Mode forwards (DIR1=0)

For the prediction of increments and actions in normal mode, the values are calculated by the equations that are described in sections:

- Equations DPLL-1a to calculate TRIGGER time stamps
- Equation DPLL-1b to calculate DT_T_ACT (nominal value)
- Equation DPLL-1c to calculate RDT_T_ACT (nominal value)
- Equation DPLL-2a1 to calculate QDT_T_ACT
- Equation DPLL-3 to calculate the error of last prediction
- Equation DPLL-4 to calculate the weighted average error
- Equations DPLL-5 to calculate the current increment value

The ascending order of calculations (in the following sections) must be followed in order to avoid the lose of results that are still needed. For TRIGGER values, it is important to calculate and store (in the RAM region 2) all values from equations DPLL-1a through DPLL-14 before equations DPLL-1a4…7, DPLL-1b1, and DPLL-1c1 are calculated and stored. Calculation of equation DPLL-1c1 overwrites the DT_Tn[DT_T] bit field value (RAM region 2 register) when the DPLL_NUTC[NUTE] bit field value is set to the FULL_SCALE range. Because the old value of DT_Tn is also needed for equations

DPLL-10 and DPLL-11, this value is temporarily stored at DT_T_ACT (as described in equation DPLL-1a or DPLL-1b, respectively) until all prediction calculations are completed. After that, equations DPLL-1a4…7, DPLL-1b1 and DPLL-1b1 are calculated to produce an update value for the DT_Tn[DT_T] bit field. The actual update of the DT_Tn[DT_T] bit field value occurs after equation DPLL-14 is calculated. When p = APT (Address Pointer TRIGGER value in DPLL_APT[APT] bit field), calculations are completed for normal mode. When filter information from the FTV_T[TRIGGER_FT] bit field (RAM region 1b register) is selected by setting the DPLL_CTRL_0[IDT] bit to 1, the DPLL_CTRL_0[IFP] bit must be configured to specify that TRIGGER_FT is either time-related or position-related.

In order to make it possible to perform the automatic resolution corrections of equations DPLL-1a1, the filter unit in the TIM module must operate using the time stamp clock.

16.5.2.1 Equations DPLL-1a to calculate TRIGGER time stamps

For calculation of time stamps, use the filter delay information:

TS_T = TS_T – FTV_; for IDT = 1 and IFP = 0..... Equation DPLL-1a1

TS_T = TS_T – FTV_T * (CDT_TX / NMB_T)_old[10]); for IDT = 1 and IFP = 1.....Equation DPLL-1a2

Calculation of time stamps can also the ADD_IN_CALN value:

TS_T = TS_T – FTV_T * (1 / ADD_IN_CALN_old[10]); for IDT = 1 and IFP = 1.....Equation DPLL-1a3

[10)] Consider values calculated for the last increment; position related filter values are only considered up to (at least) 1 ms time between two *TRIGGER* events. The reciprocal value is stored using a 32-bit fractional part while only the 24 lower bits are used (see footnote[4] in DPLL_CTRL_0 register details). The value of 1/ADD_IN_CALN_old or (CDT_TX/ NMB_T)_old is set to 0xFFFFFF in the case of an overflow.

NOTE

CDT_TX is the predicted duration of the last *TRIGGER* increment and NMB_T is the calculated number of SUB_INC1 events in the last increment, because the new calculations are done by equation DPLL-5 and equation 16.21 (see Equation DPLL-21 to calculate the number of pulses to be sent in normal mode using the automatic end mode condition) for the current increment after that. Therefore in equation DPLL-1a3, the

ADD_IN value of the last increment is used (see equation DPLL-25 in Equation DPLL-25 to calculate ADD_IN in normal mode for SMC = 0). SYN_T_old is the number of TRIGGER events including missing TRIGGERs as specified in the DPLL_NUTC register for the last increment, with the initial value of 1.

For storage of time stamps in the RAM, see equations DPLL-1a4 ff in after calculation of actions.

16.5.2.2 Equation DPLL-1b to calculate DT_T_actual (nominal value)

DT_T_actual = (TS_T − TS_T_old) / SYN_T_old;.....Equation DPLL-1b

For the case SYT = 0 (still no synchronization to the profile), the SYN_T and SYN_T_old values are still assumed to be 1.

16.5.2.3 Equation DPLL-1c to calculate RDT_T_actual (nominal value)

RDT_T_actual = 1 / DT_T_actual;.....Equation DPLL-1c

16.5.2.4 Equation DPLL-2a1 to calculate QDT_T_actual

Relation of the recent last two increment values for p = APT in forward direction (DIR1 = 0):

QDT_T_actual = DT_T_actual * RDT_T[p − 1];.....Equation DPLL2a1

QDT_T_actual as well as QDT_Tn have a 24-bit value using a 6-bit integer part and an 18-bit fractional part.

16.5.2.5 Equation DPLL-3 to calculate the error of last prediction

When q = NUTE − VTN, only the last valid prediction values for DIR1=0 are considered for the error calculation:

Calculate the error of the last prediction using RDT_T_FS1, DT_T[p − q] and DT_T[p − 1] for the prediction of DT_T[p]:

EDT_T = DT_T_actual − (DT_T[p − 1] * QDT_T[p − q]);.....Equation DPLL-3

with

QDT_T[p − q] = DT_T[p − q] * RDT_T[p − q − 1]; for FST = 0.....Equation DPLL-2b1

QDT_T[p − q] = DT_T[p − q] * RDT_T_FS1; for FST=1.....Equation DPLL-2b2

and FST has the meaning: NUTE = FULL_SCALE (see DPLL_NUTC register)

while

QDT_T_actual as well as QDT_T[i] have a 24-bit value using a 6-bit integer part and an 18-bit fractional part.

Note

QDT_T[p − q] has a 6-bit integer part and an 18-bit fractional part.

16.5.2.6 Equation DPLL-4 to calculate the weighted average error

For SYT = 1, calculate:

MEDT_T := (EDT_T + MEDT_T) / 2;.....Equation DPLL-4

16.5.2.7 Equation DPLL-5 to calculate the current increment value

CDT_TX _nom = (DT_T_actual + MEDT_T) * QDT_T[p − q + 1];.....Equation DPLL-5a

with (for q > 1):

QDT_T[p − q + 1] = DT_T[p − q + 1] * RDT_T[p − q];.....Equation DPLL-2c

For q = 1, use equation 16.2a1.

while

QDT_T_actual as well as QDT_T[i] have a 24-bit value using a 6-bit integer part and an 18-bit fractional part,

and the expected duration to the next *TRIGGER* event is

CDT_TX = CDT_TX _nom *
SYN_T;.....Equation DPLL-5b

Note

QDT_T[p-q+1] uses a 6 bit integer part and a 18 bit fractional part.

NOTE

In the case of an overflow in equations DPLL-5a or DPLL-5b, set the value to 0xFFFFFF and the corresponding CTO bit in the DPLL_STATUS register. In the case of negative values, set CDT_TX to 0x0 without any effect to the CTO bit.

16.5.3 Increment prediction in Emergency Mode forward (DIR2=0)

The ascending order of calculations for *STATE* and storage of the values in the RAM region 1c must be followed in order to avoid lose of results that are still needed. The same considerations for DT_T_actual are valid for DT_S_actual (equations 16.6a4....7, 16.6b1, and 16.6b1): update TD_Sn only after calculating equation 16.14.

When using filter information from the FTV_S[STATE_FT] bit field in RAM region 1b (selected by configuring the DPLL_CTRL_0[IDS] bit = 1), the DPLL_CTRL_0[[IFP] must be configured to show that the filter information is time-related or position-related.

In order to make it possible to perform the automatic resolution corrections of equations 16.6a1, the filter unit in the TIM module must operate using the time stamp clock.

16.5.3.1 Equations DPLL-6a to calculate STATE time stamps

For calculation of time stamps, use the filter delay information and use p = APS (DPLL_APS[APS] bit field) while DIR2 = 0:

- TS_S = STATE_TS; for IDS = 0, received unchanged value.....Equation DPLL-6a0

- TS_S = TS_S − FTV_S; for IDS = 1 and IFP = 0.....Equation DPLL-6a1
- TS_S = TS_S − FTV_S * (CDT_SX / NMB_S)_old(10); for IDS = 1 and IFP = 1.....Equation DPLL-6a2

This can be also calculated using the ADD_IN_CALE value:

TS_S = TS_S − FTV_S * (1 / ADD_IN_CALE)_old(10); for IDS = 1 and IFP = 1.....Equation DPLL-6a3

with . See equation 16.6a4 ff in Increment prediction in Normal Mode forwards (DIR1=0) for TRIGGER.

(10)Consider values calculated for the last increment; position-related filter values are only considered up to (at least) 1 ms time between two STATE events. The reciprocal value is stored using a 32-bit fractional part, while only the 24 lower bits are used (see footnote4in the DPLL_CTRL_0 register details). In the case of an overflow, the value of 1 / ADD_IN_CALE_old or (CDT_SX / NMB_S)_old) is set to 0xFFFFFF.

NOTE

CDT_SX is the predicted duration of the last STATE increment and NMB_S is the calculated number of SUB_INC1 events in the last increment because the new calculations are done by equations DPLL-10 and DPLL-22, respectively, for the current increment after that. Therefore, in equation DPLL-6a3, the ADD_IN value of the last increment is used (see equation DPLL-26 in Equation DPLL-26 to calculate ADD_IN in emergency mode for SMC=0). SYN_S_old is the number of increments, including missing STATEs (as specified in the DPLL_NUSC register) for the last increment with the initial value of 1. The update to RAM region 1c4 is done after all related calculations are completed. For this reason, see equation DPLL-6d after equation DPLL-14 (see Equation DPLL-14 to calculate the duration value for an action).

16.5.3.2 Equation DPLL-6b to calculate DT_S_actual (nominal value)

DT_S_actual = (TS_S − TS_S_old) / SYN_S_old;.....Equation DPLL-6b

For the case SYS = 0 (still no synchronization to the profile), the SYN_S and SYN_S_old values are still assumed to be 1.

16.5.3.3 Equation DPLL_6c to calculate RDT_S_actual (nominal value)

RDT_S_actual = 1 / DT_S_actual;.....Equation DPLL_6c

16.5.3.4 Equation DPLL-7a1 to calculate QDT_S_actual

For p = APS (see DPLL_APS[APS] bit field) in forward direction (DIR2=0):

QDT_S_actual = DT_S_actual * RDT_S[p − 1];.....Equation DPLL-7a1

QDT_S_actual as well as QDT_S[i] have a 24-bit value using a 6-bit integer part and an 18-bit fractional part.

16.5.3.5 Equation DPLL-8 to calculate the error of last prediction

With q = NUSE – VSN when using QDT_S[p – q] and DT_S[p – 1] for the prediction of DT_S[p]:

EDT_S = DT_S_actual – (DT_S[p – 1] * QDT_S[p – q]).....Equation DPLL-8

and with

QDT_S[p – q] = DT_S[p – q] * RDT_S[p – q – 1]; for FSS = 0.....Equation DPLL-7b1

QDT_S[p – q] = DT_S[p – q] * RDT_S_FS1; for FSS=1.....Equation DPLL-7b2

and FSS has the meaning: NUSE = FULL_SCALE (see DPLL_NUSC[NUSE] bit field).

QDT_S_actual as well as QDT_Sn have a 24-bit value using a 6-bit integer part and an 18-bit fractional part.

Note

QDT_S[p – q] uses a 6-bit integer part and an 18-bit fractional part.

16.5.3.6 Equation DPLL-9 to calculate the weighted average error

For SYS = 1, calculate:

MEDT_S := (EDT_S + MEDT_S) / 2;.....Equation DPLL-9

16.5.3.7 Equations DPLL-10 to calculate the current increment (nominal value)

Table 16-5. Equation DPLL-10a

Equation	Description
CDT_SX _nom = (DT_S_actual + MEDT_S) * QDT_S[p – q + 1]	None

with

Table 16-6. Equation DPLL-7c

Equation	Description
QDT_S[p – q + 1] = DT_S[p – q + 1] * RDT_S[p – q]	For q>1 (for q = 0, see equation 16.7a)

while

QDT_S_actual as well as QDT_Sn have a 24-bit value using a 6-bit integer part and an 18-bit fractional part,

and the expected duration to the next *STATE* event:

Table 16-7. Equation DPLL-10b

Equation	Description
CDT_SX = CDT_SX _nom * SYN_S	None

Note

QDT_S[p – q + 1] uses a 6-bit integer part and an 18-bit fractional part.

Note

In the case of an overflow in equations DPLL-10a or DPLL-10b set the value to 0xFFFFFF and the corresponding CSO bit in the DPLL_STATUS register. In the case of negative values set CDT_SX to 0x0 without any effect to the CSO bit.

All five steps above (equation16.6 to equation DPLL-10) are only needed in emergency mode. For normal mode, the calculations of equations DPLL-6 and DPLL-7 are done to get the values needed for a sudden switch to emergency mode.

16.5.4 Increment prediction in Normal Mode backward (DIR1=1)

16.5.4.1 Equation DPLL-2a2 to calculate QDT_T_actual backwards

QDT_T_actual = DT_T_actual * RDT_T[p + 1];.....Equation DPLL-2a2

QDT_T_actual as well as QDT_Tn have a 24 bit value using a 6-bit integer part and an 18-bit fractional part.

16.5.4.2 Equation DPLL-3a to calculate the error of last prediction

When q = NUTE − VTN and DIR1 = 1 using only QT_T[p + q] and DT_T[p + 1] for the prediction of DT_T[p]:

EDT_T = DT_T_actual − (DT_T[p + 1] * QDT_T[p + q];.....Equation DPLL-3a

with

QDT_T[p + q] = DT_T[p + q] * RDT_T[p + q + 1]; for FST = 0.....Equation DPLL-2b3

QDT_T[p + q] = DT_T[p + q] * RDT_T_FS1; for FST=1.....Equation DPLL-2b4

and FST has the meaning: NUTE = FULL_SCALE (see DPLL_NUTC[NUTE] bit field)

QDT_T_actual as well as QDT_Tn have a 24-bit value using a 6-bit integer part and an 18-bit fractional part.

16.5.4.3 Equation DPLL-4 to calculate the weighted average error

For SYT = 1, calculate:

MEDT_T := (EDT_T + MEDT_T) / 2;.....Equation DPLL-4

16.5.4.4 Equation DPLL-5 to calculate the current increment value

CDT_TX _nom = (DT_T_actual + MEDT_T) * QDT_T[p − q + 1];.....Equation DPLL-5a

with (for q > 1):

QDT_T[p − q + 1] = DT_T[p − q + 1] * RDT_T[p − q];.....Equation DPLL-2c

For q = 1, use equation 16.2a1.

while

QDT_T_actual as well as QDT_T[i] have a 24-bit value using a 6-bit integer part and an 18-bit fractional part,

and the expected duration to the next *TRIGGER* event is

CDT_TX = CDT_TX _nom * SYN_T;.....Equation DPLL-5b

Note

QDT_T[p-q+1] uses a 6 bit integer part and a 18 bit fractional part.

NOTE

In the case of an overflow in equations DPLL-5a or DPLL-5b, set the value to 0xFFFFFF and the corresponding CTO bit in the DPLL_STATUS register. In the case of negative values, set CDT_TX to 0x0 without any effect to the CTO bit.

16.5.5 Increment prediction in Emergency Mode backward (DIR2=1)

16.5.5.1 Equation DPLL-7a2 to calculate QDT_S_actual backward

QDT_S_actual = DT_S_actual * RDT_S[p + 1];.....Equation DPLL-7a2

QDT_S_actual as well as QDT_Sn have a 24-bit value using a 6-bit integer part and an 18-bit fractional part.

16.5.5.2 Equation DPLL-8a to calculate the error of the last prediction

While q = NUSE − VSN, use only QDT_S[p + q] and DT_S[p +1] for the prediction of DT_S[p]:

EDT_S = DT_S_actual − (DT_S[p − 1] * QDT_S[p − q]);.....Equation DPLL-8a

with

QDT_S[p − q] = DT_S[p − q] * RDT_S[p − q − 1]; for FSS = 0.....Equation DPLL-7b1

or

QDT_S[p − q] = DT_S[p − q] * RDT_S_FS1; for FSS = 1.....Equation DPLL-7b2

FSS has the meaning: NUSE = FULL_SCALE (see DPLL_NUSC[NUSE] bit field).

QDT_S_actual as well as QDT_S[i] have a 24-bit value using a 6-bit integer part and an 18-bit fractional part.

16.5.5.3 Equation DPLL-9 to calculate the weighted average error

For SYS = 1, calculate:

MEDT_S := (EDT_S + MEDT_S) / 2;.....Equation DPLL-9

16.5.5.4 Equations DPLL-10 to calculate the current increment value

Table 16-8. Equation DPLL-10a

Equation	Description
CDT_SX _nom = (DT_S_actual + MEDT_S) * QDT_S[p +q − 1]	None

with, for q > 1:

Table 16-9. Equation DPLL-7c1

Equation	Description
QDT_S[p + q – 1] = DT_S[p + q – 1] * RDT_S[p + q]	None

and for q = 1, use equation 16.7a.

while

QDT_S_actual as well as QDT_Sn have a 24-bit value using a 6-bit integer part and an 18-bit fractional part,

Calculate the expected duration to the next *STATE* event:

Table 16-10. Equation DPLL-10b

Equation	Description
CDT_SX = CDT_SX _nom * SYN_S	None

Note

In the case of an overflow in equations DPLL-10a1 or DPLL-10b, set the value to 0xFFFFFF and set the corresponding DPLL_STATUS[CSON, CSO] bits. In the case of negative values, set CDT_SX(_nom) to 0x0.

All five steps above (equation DPLL-6 through equation DPLL-10) are only needed in emergency mode. For normal mode, the calculations of equations DPLL-6 and DPLL-7 are done to get the values needed for a sudden switch to emergency mode.

16.6 Calculations for actions

As already shown for the calculation of the current interval (by equations DPLL-1 through DPLL-10), a similar calculation has to be done for the prediction of actions (by equations DPLL-11 through DPLL-14). The calculation of actions is also needed when the DPLL is used for synchronous motor control applications (when the DPLL_CTRL_1[SMC] bit is set to 1). For action prediction purposes, the measured time periods of the past (one FULL_SCALE back) are used (when the corresponding DPLL_NUTC[NUTE] or DPLL_NUSC[NUSE] values are set by the CPU). The calculation can be described by these assumptions:

1. Take the corresponding increments for prediction in the past and put the sum of them in relation to the increment (DT_T[k], DT_S[k], with k ≥ 0, which is represented by the time stamp difference) which is exactly one FULL_SCALE period in the past (see equationDPLL-11 or DPLL-13, respectively).
2. Make a prediction for the coming sum of increments by using the current measured increment (DT_T_ACT or DT_S_ACT by equation DPLL-1 or equation DPLL-6, respectively) and add a weighted average error (by equation DPLL-3 and DPLL-4 or equation DPLL-8 and DPLL-9, respectively, calculated for one increment prediction) before multiplication by the relation of equation DPLL-11 or DPLL-13, respectively, in order to get the result as described by equations DPLL-12 or DPLL-14, respectively.

In order to avoid division operations instead of the increment (DT_T_,[k], DT_S_[k], with k > 0) in the past, its reciprocal value (RDT_T_,[k], RDT_S_[k], with k > 0) is used, which is stored also in RAM.

For the calculation of actions, always perform a new refined calculation as long as the resulting time stamp is not in the past. In the other case:

- the tsac/psac values (time/position stamp of action calculated) is set to the time/ position stamp of the last input event (TRIGGER/STATE),
- the DPLL_ACT_STA[ACT_N(i)] bit is reset to 0, while
- the corresponding ACT_N[i] bit in the DPLL_ACT_STA_shadow register is set to 1.

Each new PMTR_i value will set this ACT_N[i] bit again and reset the corresponding shadow bit until a new calculation is performed.

Action updates at highest speed

Up to 32 action values can be calculated. For the shortest increment duration (23.4 µs), not all action values can be updated with each valid input event. The following conditions and parameters are used to estimate possible results.

All time estimation values are given for a system clock frequency of 100 MHz. (When using a different system clock frequency, the calculation duration is changed accordingly.) It is assumed that the calculations are not impeded by a remote read or write access to the DPLL RAMs. Each RAM access should consider an additional delay of about 40 ns. The time estimations values are:

1. Time needed for basic operations (RAM update, pointer calculation and SUB_INC generation for normal, emergency mode or one PMSM:

   t_{basic_0} = 11.5 µs.

2. Time needed basic operations (RAM update, pointer calculation and SUB_INC generation for two PMSMs:

t_{basic_1} = 13.5 µs.

3. Time needed to calculate one action

   t_{action_i} = 4 µs.

These values allow the calculation of at least 2 action values for each input event (for all of the specified increment durations). The complete time needed for the basic operation, n action calculations and k remote RAM access operations, can be calculated as:

$t_{complete} = t_{basic\_0/1} + n * t_{action\_i} + k * t_{remote\_RAM\_access}$.

Typical applications

Normal and emergency mode

For a typical application with the shortest increment duration of 100 µs in normal or emergency mode, the calculation of up to 22 action values can be performed for each valid input event.

One PMSM

For one PMSM and a typical shortest increment time of 39 µs, the calculation of up to six action values can be performed for each valid input event.

Two PMSMs with restricted action calculations

When only one PMSM uses the action calculation service an the shortest increment duration is 39 µs, there can up to 6 actions served for each input event.

Two PMSMs with unrestricted action calculations

When two PMSMs are used and both use the action calculation service at a minimal increment duration of 39 µs, up to five action calculations are possible for each of the two engines - that means an average of up to 10 action calculations per increment.

16.6.1 Action calculations for TRIGGER forward

Valid for RMO = 0 or SMC = 1 with

p = APT_2b, t = APT, m = NA[i] (part w), mb = NA[i](part b) / 1024, NUTE – VTN = q, NUTE = n.

16.6.1.1 Equation DPLL-11a1 to calculate the time prediction for an action

For DIR1 = 0 and q > m, calculate:

Table 16-11. Equation DPLL-11a1

Equation	Description
PDT_T[i] = (TSF_T[p + m – n] – TSF_T[p – n] + mb * DT_Tx[t – q + 1]) * RDT_T[t – q]	None

with

Table 16-12. Equation DPLL-11b2

Equation	Description
DT_Tx[t – q + 1] = DT_T[t – q + 1]	For TS0_HRT = 0

or

Table 16-13. Equation DPLL-11b3

Equation	Description
DT_Tx[t – q + 1] = DT_T[t – q +1] / 8	For TS0_HRT = 1

where mb is the the fractional part of NA[i].

For SMC = 0 and RMO = 0, calculate all 24 actions in forward direction for DIR1 = 0, if requested; for SMC = 1, calculate up to 12 actions (0 through 11) dependent on the *TRIGGER* input.

16.6.1.2 Equation DPLL-11a2 to calculate the time prediction for an action

For SYT = 1,NUTE = 2 * (TNU + 1), q > m, and DIR1 = 0:

Table 16-14. Equation DPLL-11a2

Equation	Description
PDT_T[i] = (TSF_T[p + m] – TSF_T[p] + mb * DT_Tx[t – q + 1]) * RDT_T[t]	None

with

Table 16-15. Equation DPLL-11b2

Equation	Description
DT_Tx[t – q + 1] = DT_T[t – q + 1]	For TS0_HRT = 0

or

Table 16-16. Equation DPLL-11b3

Equation	Description
DT_Tx[t – q + 1] = DT_T[t – q + 1] / 8	For TS0_HRT = 1

16.6.1.3 Equation DPLL-11b to calculate the time prediction for an action

For DIR1 = 0, NUTE – VTN = q <= m, q > 1 and t = APT:

Table 16-17. Equation DPLL-11b

Equation	Description
PDT_T[i] = (m + mb) * DT_Tx[t – q + 1] * RDT_T[t – q]	None

with

Table 16-18. Equation DPLL-11b2

Equation	Description
DT_Tx[t – q + 1] = DT_T[t – q + 1]	For TS0_HRT = 0

or

Table 16-19. Equation DPLL-11b3

Equation	Description
DT_Tx[t – q + 1] = DT_T[t – q + 1] / 8	For TS0_HRT = 1

Note

Make the above calculations before updating the TSF_T[i] values according to equation DPLL-1c3 ff.

16.6.1.4 Equation DPLL-11c to calculate the time prediction for an action

For n = 1 (this is always valid for SYT = 0) or q = 1:

Table 16-20. Equation DPLL-11c

Equation	Description
PDT_T[i] = (m + mb) * DT_T_ax * RDT_T[t – 1]	None

with

Table 16-21. Equation DPLL-1a4a

Equation	Description
DT_T_ax = DT_T_actual	For TS0_HRT = 0

or

Table 16-22. Equation DPLL-1a4b

Equation	Description
DT_T_ax = DT_T_actual / 8	For TS0_HRT = 1

Note

For the relevant last increment, add the fractional part of DT_T_actual as described in NA[i].

16.6.1.5 Equation DPLL-12 to calculate the duration value until action

Table 16-23. Equation DPLL-12

Equation	Description
DTA[i] = (DT_T_actual + MEDT_T) * PDT_T[i]	None

Note

All 5 steps in equations DPLL-11 to DPLL-12 are only calculated in normal mode.

16.6.2 Action calculations for TRIGGER backward

Valid for RMO = 0 or for SMC = 1 with

p = APT_2b, t = APT, m = NA[i] (part w), mb = NA[i](part b) / 1024, q = NUTE – VTN and n = NUTE

For SMC = 0 and RMO = 0, calculate (for DIR1=1) all 24 actions in backward direction for special purposes; for SMC=1, calculate up to 12 actions (0 through 11) dependent on the TRIGGER input.

16.6.2.1 Equation DPLL-11a3 to calculate the time prediction for an action

For DIR1 = 1 and q > m, calculate:

Table 16-24. Equation DPLL-11a3

Equation	Description
PDT_T[i] = (TSF_T[p – m + n] – TSF_T[p + n] + mb * DT_Tx[t + q – 1]) * RDT_T[t + q]	None

with

Table 16-25. Equation DPLL-11b4

Equation	Description
DT_Tx[t + q – 1] = DT_T[t + q – 1]	For TS0_HRT = 0

or

Table 16-26. Equation DPLL-11b5

Equation	Description
DT_Tx[t + q – 1] = DT_T[t + q – 1] / 8	For TS0_HRT = 1

16.6.2.2 Equation DPLL-11a4 to calculate the time prediction for an action

For SYT = 1 and NUTE = 2 * (TNU + 1), q > m, VTN = 2 * SYN_NT and hence **NUTE – VTN = 2 * (TNU + 1 – SYN_NT).**

For DIR1 = 1, this is equal to:

Table 16-27. Equation DPLL-11a4

Equation	Description
PDT_T[i] = (TSF_T[p – m] – TSF_T[p] + mb * DT_Tx[t + q – 1]) * RDT_T[t]	None

with

Table 16-28. Equation DPLL-11b4

Equation	Description
DT_Tx[t + q – 1] = DT_T[t + q – 1]	For TS0_HRT = 0

or

Table 16-29. Equation DPLL-11b5

Equation	Description
DT_Tx[t + q – 1] = DT_T[t + q – 1] / 8	For TS0_HRT = 1

Note

Make the calculations above before updating the TSF_T[i] values according to equation DPLL-1c3 ff.

16.6.2.3 Equation DPLL-11b1 to calculate the time prediction for an action

For NUTE – VTN = q ≤ m, q > 1, and t = APT:

Table 16-30. Equation DPLL-11b1

Equation	Description
PDT_T[i] = (m+ mb) * DT_Tx[t + q –1] * RDT_T[t + q]	None

with

Table 16-31. Equation DPLL-11b4

Equation	Description
DT_Tx[t + q – 1] = DT_T[t + q – 1]	For TS0_HRT = 0

or

Table 16-32. Equation DPLL-11b5

Equation	Description
DT_Tx[t + q – 1] = DT_T[t + q – 1] / 8	For TS0_HRT = 1

16.6.2.4 Equation DPLL-11c1 to calculate the time prediction for an action

For n = 1 (this is always valid for SYT = 0) or q = 1:

Table 16-33. Equation DPLL-11c1

Equation	Description
PDT_T[i] = (m+ mb) * DT_T_ax * RDT_T[t + 1]	None

with

Table 16-34. Equation DPLL-1a4a

Equation	Description
DT_T_ax = DT_T_actual	For TS0_HRT = 0

or

Table 16-35. Equation DPLL-1a4b

Equation	Description
DT_T_ax = DT_T_actual / 8	For TS0_HRT = 1

Note

For the relevant last increment, add the fractional part of DT_T_actual as described in NA[i].

16.6.2.5 Equation DPLL-16.12 to calculate the duration value for an action

Use the results of equations DPLL-16.1a, DPLL-16.1b, DPLL-16.3 and DPLL-16.4 for the calculation:

Table 16-36. Equation DPLL-16.12

Equation	Description
DTA[i] = (DT_T_actual + MEDT_T) * PDT_T[i]	None

Note

All five steps in equations 16.11 through 16.12 are calculated in normal mode only.

16.6.3 Action calculations for STATE forwards

Valid for RMO = 1 with

p = APS_1c2, t = APS, m = NA[i](part w), mb = NA[i](part b) / 1024, NUSE – VSN = q and NUSE = n > m.

For SMC = 0 and RMO = 1, calculate all 24 actions in forward direction for DIR2=0, if requested; for SMC = 1 and RMO = 1, calculate up to 12 actions (12 through 23) dependent on the STATE input.

16.6.3.1 Equation DPLL-13a1 to calculate the time prediction for an action

For DIR2 = 0 and q > m, calculate:

Table 16-37. Equation DPLL-13a1

Equation	Description
PDT_S[i] = (TSF_S[p + m – n] – TSF_S[p – n] + mb * DT_Sx[t – q + 1] * RDT_S[t – q]	None

with

Table 16-38. Equation DPLL-13b2

Equation	Description
DT_Sx[t – q + 1] = DT_S[t – q + 1]	For TS0_HRS = 0

or

Table 16-39. Equation DPLL-13b3

Equation	Description
DT_Sx[t – q + 1] = DT_S[t – q + 1] / 8	For TS0_HRS = 1

16.6.3.2 Equation DPLL-13a2 to calculate the time prediction for an action

For **SYS = 1 and NUSE = 2 * (SNU + 1)**, q > m, SYSF = 0, VSN = 2 * SYN_NS and hence NUSE – VSN = 2 * (SNU + 1 – SYN_NS):

Table 16-40. Equation DPLL-13a2

Equation	Description
PDT_S[i] = (TSF_S[p + m] – TSF_S[p] + mb * DT_Sx[t – q + 1]) * RDT_S[t]	None

with

Table 16-41. Equation DPLL-13b2

Equation	Description
DT_Sx[t – q + 1] = DT_S[t – q + 1]	For TS0_HRS = 0

or

Table 16-42. Equation DPLL-13b3

Equation	Description
DT_Sx[t – q + 1] = DT_S[t – q + 1] / 8	For TS0_HRS = 1

16.6.3.3 Equation DPLL-13b to calculate the time prediction for an action

For NUSE – VTN = q <= m and q >1:

Table 16-43. Equation DPLL-13b

Equation	Description
PDT_S[i] = (m + mb) * DT_Sx[t – q + 1] * RDT_S[t – q]	None

with

Table 16-44. Equation DPLL-13b2

Equation	Description
DT_Sx[t – q + 1] = DT_S[t – q + 1]	For TS0_HRS = 0

or

Table 16-45. Equation DPLL-13b3

Equation	Description
DT_Sx[t – q + 1] = DT_S[t – q + 1] / 8	For TS0_HRS=1

16.6.3.4 Equation DPLL-13c to calculate the time prediction for an action

For n = 1:

Table 16-46. Equation DPLL-13c

Equation	Description
PDT_S[i] = (m + mb) * DT_S_ax * RDT_S[t – 1]	None

with

Table 16-47. Equation DPLL-6a4a

Equation	Description
DT_S_ax = DT_S_actual	For TS0_HRS = 0

or

Table 16-48. Equation DPLL-6a4b

Equation	Description
DT_S_ax = DT_S_actual / 8	For TS0_HRS = 1

16.6.3.5 Equation DPLL-14 to calculate the duration value for an action

Use the results of equation DPLL-7, DPLL-8, and DPLL-9 in this calculation:

Table 16-49. Equation DPLL-14

Equation	Description
DTA[i] = (DT_S_actual + MEDT_S) * PDT_S[i]	None

Note

All five steps of equations DPLL-13 through DPLL-14 are calculated in emergency mode only.

16.6.4 Action calculations for STATE backwards

Valid for RMO = 1 with

p = APS_1c2, t = APS, m = NA[i](part w), mb = NA[i](part b) / 1024, NUSE – VSN = q and NUSE = n.

For SMC = 0 and RMO = 1, calculate (for DIR1=1) all 24 actions in backward mode for special purposes; for SMC = 1 and RMO = 1 calculate up to 12 actions (12 to 23) dependent on the STATE input.

16.6.4.1 Equation DPLL-13a3 to calculate the time prediction for an action

For DIR2 = 1 and q > m, calculate:

Table 16-50. Equation DPLL-13a3

Equation	Description
PDT_S[i] = (TSF_S[p – m + n] – TSF_S[p + n] + mb * DT_Sx[p + q – 1]) * RDT_S[t + q]	None

with

Table 16-51. Equation DPLL-13b4

Equation	Description
DT_Sx[p + q – 1]= DT_S[p + q – 1]	For TS0_HRS = 0

or

Table 16-52. Equation DPLL-13b5

Equation	Description
DT_Sx[p + q – 1] = DT_S[p + q – 1] / 8	For TS0_HRS = 1

16.6.4.2 Equation DPLL-13a4 to calculate the time prediction for an action

For **SYS = 1, NUSE = 2 * (SNU + 1)**, q > m, SYSF = 0, VSN = 2 *SYN_NS and hence NUSE – VSN = 2 * (SNU + 1 – SYN_NS):

Table 16-53. Equation DPLL-13a4

Equation	Description
PDT_S[i] = (TSF_S[p – m] – TSF_S[p] + mb * DT_Sx[p + q – 1]) * RDT_S[t]	None

with

Table 16-54. Equation DPLL-13b4

Equation	Description
DT_Sx[p + q – 1] = DT_S[p + q – 1]	For TS0_HRS = 0

or

Table 16-55. Equation DPLL-13b5

Equation	Description
DT_Sx[p + q – 1] = DT_S[p + q –1] / 8	For TS0_HRS = 1

16.6.4.3 Equation DPLL-13b1 to calculate the time prediction for an action

For NUSE – VSN = q <= m, NUSE = n and q > 1:

Table 16-56. Equation DPLL-13b1

Equation	Description
PDT_S[i] = (m + mb) * DT_Sx[t – q + 1] * RDT_S[t – q]	None

with

Table 16-57. Equation DPLL-13b4

Equation	Description
DT_Sx[t – q + 1] = DT_S[t – q + 1]	For TS0_HRS = 0

or

Table 16-58. Equation DPLL-13b5

Equation	Description
DT_Sx[t – q + 1] = DT_S[t – q + 1] / 8	For TS0_HRS = 1

16.6.4.4 Equation DPLL-13c1 to calculate the time prediction for an action

For n = 1 or q = 1,

Table 16-59. Equation DPLL-13c1

Equation	Description
PDT_S[i] = (m + mb) * DT_S_ax * RDT_S[t +1]	None

with

Table 16-60. Equation DPLL-6a4a

Equation	Description
DT_S_ax = DT_S_actual	For TS0_HRS = 0

or

Table 16-61. Equation DPLL-6a4b

Equation	Description
DT_S_ax = DT_S_actual / 8	for TS0_HRS = 1

16.6.4.5 Equation DPLL-14 to calculate the duration value until action

Use the results from equations DPLL-7, DPLL-8, and DPLL-9 to calculate:

Table 16-62. Equation DPLL-14

Equation	Description
DTA[i] = (DT_S_actual + MEDT_S) * PDT_S[i]	None

Note

All five steps of equations DPLL-13 and DPLL-14 are calculated in emergency mode only.

16.6.5 Update of RAM in Normal and Emergency Mode

After considering the calculations for up to all 24 actions according to equations DPLL-11 and DPLL-12 and only when going back to state 1 or 21 (because of a new TRIGGER or STATE event, meaning that no further PMTR values are to be considered), set time stamp values and durations of increments in the RAM.

16.6.5.1 Equation DPLL-1a4 to update the time stamp values for TRIGGER

Table 16-63. Equation DPLL-1a4

Equation	Description
TSF_T[s] = TS_Tx	None

using these equations for the determination of TS_Tx:

For TS0_HRT = 0:

- **Table 16-64. Equation DPLL-1a4w**

Equation	Description
TS_Tx = TS_T	None

- **Table 16-65. Equatio n DPLL-1a4a**

Equation	Description
DT_T_ax = DT_T_actual	None

For TS0_HRT = 1:

- **Table 16-66. Equation DPLL-1a4x**

Equation	Description
TS_Tx(20:0) = TS_T / 8	None

- **Table 16-67. Equation DPLL-1a4y**

Equation	Description
TS_Tx(23:21) = TBU_TS0_T(23:21)	None

For TBU_TS0_T(20:0) ≥ TS_Tx(20:0):

- **Table 16-68. Equation DPLL-1a4z**

Equation	Description
TS_Tx(23:21) = TBU_TS0_T(23:21) –1	None

For TBU_TS0_T(20:0) < TS_Tx(20:0):

- **Table 16-69. Equation DPLL-1a4b**

Equation	Description
DT_T_ax = DT_T_actual / 8	None

NOTE

The combination of the LOW_RES = 0 and TS0_HRT = 1 values is not possible.

Store the time stamp values in the time stamp field according to the address pointer, APT_2b = s, but make this update only after the calculation of actions by equation DPLL-7 because the old TSF_T[i] values are still needed for these calculations. The address pointer after a gap is still incremented by SYN_T_old (see state machine step 1 in Scheduling of the Calculation).

16.6.5.2 Equation DPLL-1a5-7 to extend the time stamp values for TRIGGER in forward direction

When SYT = 1 and SYN_T_old = r > 1 and DIR1 = 0:

TSF_T[s – 1] = TSF_T[s] –
DT_T_ax;.....Equation DPLL-1a5

TSF_T[s – 2] = TSF_T[s – 1] –
DT_T_ax;.....Equation DPLL-1a6

until

TSF_T[s – r + 1] = TSF_T[s – r + 2] –
DT_T_ax;.....Equation DPLL-1a7

after incrementation of the APT_2b pointer by SYN_T_old.

16.6.5.3 Equations DPLL-1a5-7 for backward direction

When SYT = 1 and SYN_T_old = r > 1 and DIR1 = 1

TSF_T[s + 1] = TSF_T[s] –
DT_T_ax;.....Equation DPLL-1a5

TSF_T[s + 2] = TSF_T[s + 1] –
DT_T_ax;.....Equation DPLL-1a6

until

TSF_T[s + r – 1] = TSF_T[s + r – 2] –
DT_T_ax;.....Equation DPLL-1a7

after decrementing the APT_2b by SYN_T_old pointer.

16.6.5.4 Equations DPLL-1b1 and DPLL-1c1 to update the RAM after calculation

DT_T[p] =
DT_T_actual;.....Equation DPLL-1b1

Save the old reciprocal value from RAM before overwriting:

RDT_T_FS1 =
RDT_T[p];.....Equation DPLL-1c1

then store the new value in RAM:

RDT_T[p] =
RDT_T_actual;.....Equation DPLL-1c2

Store the increment duration and reciprocal value in RAM region 2 (in normal mode) after calculation of actions only when a new valid *TRIGGER* slope is detected and (in emergency mode) directly after calculation of DT_T_actual or RDT_T_actual, respectively.

16.6.5.5 Equation DPLL-6a4 to update the time stamp values for STATE

TSF_S[s] =
TS_Sx;.....Equation DPLL-6a4

using the following equations for the determination of TS_Sx:

For TS0_HRS = 0:

- TS_Sx =
  TS_S;.....Equation DPLL-6a4
- DT_S_ax =
  DT_S_actual;.....Equation DPLL-6a4a

For TS0_HRS = 1:

- TS_Sx(20:0) = TS_S /
  8;.....Equation DPLL-6a4x
- TS_Sx(23:21) =
  TBU_TS0_S(23:21);.....Equation DPLL-6a4y

For TBU_TS0_S(20:0) ≥ TS_Sx(20:0):

- TS_Sx(23:21) = TBU_TS0_S(23:21) – 1;.....Equation DPLL-6a4z

For TBU_TS0_S(20:0) < TS_Sx(20:0):

- DT_S_ax = DT_S_actual / 8;.....Equation DPLL-6a4b

Note

The combination of the LOW_RES=0 and TS0_HRS=1 values is not possible.

Store the time stamp value in the time stamp field according to the address pointer, APS_1c2 = s, but make this update only after the calculation of actions (equations DPLL-13a2 or DPLL-13a4 to calculate the time prediction for an action, if applicable) because the old TSF_S[i] values are still needed for these calculations. After a gap, the address pointer is still incremented by SYN_S_old (see state machine step 21 in Scheduling of the Calculation).

16.6.5.6 Equations DPLL-6a5-7 to extend the time stamp values for STATE

When SYS = 1, SYN_S_old = r > 1, and DIR2 = 0 or DIR1 = 0, respectively, calculate:

TSF_S[s – 1] = TSF_S[s] – DT_S_ax;.....Equation DPLL-6a5

TSF_S[s – 2] = TSF_S[s – 1] - DT_S_ax;.....Equation DPLL-6a6

until

TSF_S[s – r + 1] = TSF_S[s – r + 2] – DT_S_ax;.....Equation DPLL-6a7

after the APS_2b pointer is incremented by SYN_S_old.

16.6.5.7 Equations DPLL-6a5-7 for backward direction

When SYS = 1 and SYN_S_old = r > 1 and DIR2 = 1, calculate:

TSF_S[s + 1] = TSF_S[s] – DT_S_ax;.....Equation DPLL-6a5

TSF_S[s + 2] = TSF_S[s + 1] –
DT_S_ax;.....Equation DPLL-6a6

until

TSF_S[s + r – 1] = TSF_S[s + r – 2] –
DT_S_ax;.....Equation DPLL-6a7

after the APS_1c2 pointer is incremented by SYN_S_old.

16.6.5.8 Equations DPLL-6b1, DPLL-6c1, and DPLL-1c2 to update RAM after calculation

DT_S[p] = DT_S_actual;.....Equation DPLL-6b1

Save old reciprocal value from RAM before overwriting:

RDT_S_FS1= RDT_S[p];.....Equation DPLL-6c1

Then store new value in RAM:

RDT_S[p] = RDT_S_actual;..... Equation DPLL-6c2

Store the increment duration and reciprocal value in RAM region 2 (in normal mode) after calculation of actions only when a new valid TRIGGER slope is detected and (in emergency mode) directly after the calculation of DT_T_actual or RDT_T_actual.

16.6.6 Time and position stamps for actions in Normal Mode

16.6.6.1 Equation DPLL-15 to calculate the action time stamp

Table 16-70. Equation DPLL-15a

Equation	Description
TSAC[i] = DTA[i] – DLA[i] + TS_Tx	For DTA[i] > DLA[i] and DTA[i] – DLA[i] < 0x800000

Table 16-71. Equation DPLL-15b

Equation	Description
TSAC[i] = TS_Tx	For DTA[i] < DLA[i]

Table 16-72. Equation DPLL-15c

Equation	Description
TSAC[i] = 0x7FFFFF + TS_Tx	For DTA[i] > DLA[i] and DTA[i] – DLA[i] > 0x7FFFFF

Note

For TS_Tx, see equation DPLL-1a4 in Equation DPLL-1a4 to update the time stamp values for TRIGGER.

The calculation is done after the calculation of the current expected duration value according to equation DPLL-12 in Equation DPLL-16.12 to calculate the duration value for an action. The time stamp of the action can be calculated as shown above in equation DPLL-15. using the delay value of the action and the current time stamp.

16.6.6.2 Equations DPLL-17 to calculate the position stamp forward

For DIR1 = 0 and TS0_HRT = 0:

Table 16-73. Equation DPLL-17

Equation	Description
PSAC[i] = PSA[i] – (DLA[i] * RCDT_TX_nom) * (MLT + 1)	None

with

Table 16-74. Equation DPLL-17a

Equation	Description
RCDT_TX_nom = RCDT_TX * SYN_T	None

and

Table 16-75. Equation DPLL-17b

Equation	Description
RCDT_TX = 1 / CDT_TX	None

For **DIR1 = 0** and TS0_HRT = 1:

Table 16-76. Equation DPLL-17d

Equation	Description
PSAC[i] = PSA[i] – (8 * DLA[i] * RCDT_TX_nom) * (MLT + 1)	None

with

Table 16-77. Equation DPLL-17a

Equation	Description
RCDT_TX_nom = RCDT_TX * SYN_T	None

and

Table 16-78. Equation DPLL-17b

Equation	Description
RCDT_TX = 1 / CDT_TX	None

For SMC = 1, replace (MLT + 1) in equations DPLL-17 and DPLL-17d with MLS1.

Also, use the calculated value of equationDPLL-17b for the generation of SUB_INCi and serve the action by transmission of TSAC[i] and PSAC[i] to ACT_D_i.

The action is to be updated for each new *TRIGGER* event until the calculated time stamp is in the past. In this case, use the time stamp of the last input event instead of the calculated value and use the calculated position stamp of the actual increment as the target position value. Set the corresponding shadow bit in the DPLL_ACT_STA register. Because of the blocking read operation, the ACT_D values can be read only once.

16.6.6.3 Equations DPLL-17 to calculate the position stamp backward

For **DIR1=1** and TS0_HRT=0:

Table 16-79. Equation DPLL-17c

Equation	Description
PSAC[i] = PSA[i] + (DLA[i] * RCDT_TX_nom) * (MLT + 1)	None

with

Table 16-80. Equation DPLL-17a

Equation	Description
RCDT_TX_nom = RCDT_TX * SYN_T	None

and

Table 16-81. Equation DPLL-17b

Equation	Description
RCDT_TX = 1/CDT_TX	None

For **DIR1=1** and TS0_HRT=1:

Table 16-82. Equation DPLL-17e

Equation	Description
PSAC[i] = PSA[i] + (8 * DLA[i] * RCDT_TX_nom) * (MLT + 1)	None

with

Table 16-83. Equation DPLL-17a

Equation	Description
RCDT_TX_nom = RCDT_TX * SYN_T	None

and

Table 16-84. EquationDPLL-17b

Equation	Description
RCDT_TX = 1 / CDT_TX	None

For SMC = 1, replace (MLT+1) in equations DPLL-17c and DPLL-17e with MLS1.

Also, use the calculated value from equation DPLL-17b for the generation of SUB_INCi and serve the action by transmission of TSAC[i] and PSAC[i] to ACT_D_i.

The action is to be updated for each new *TRIGGER* event until the calculated time stamp is in the past. In this case, use the time stamp of the last input event instead of the calculated value and use the calculated position stamp of the actual increment as the target position value. Set the corresponding shadow bit in the DPLL_ACT_STA register. Because of the blocking read operation, the ACT_D values can be read only once.

16.6.7 The use of the RAM

The RAM (external to the GTM) is used to store the data of the last FULL_SCALE period. A single port RAM is recommended. The data width of the RAM is usually three bytes, but it could be extended to four bytes in future applications. There are three different RAMs, each with separate access ports. RAM 1a is used to store the position minus time requests received from the ARU. The CPU can access this RAM only when the DPLL is disabled.

RAM 1b is used for configuration parameters and variables that are needed for calculations. RAM 1c is used to store the values of the *STATE* events. RAM 1b and RAM 1c have a common access port and they are marked as RAM 1bc to clarify this fact.

RAM 2 is used to store the values of the *TRIGGER* events.

Because RAMs 1bc and 2 can be accessed by both the DPLL internal state machine and the CPU, the access priority (see blow) has to be controlled. If a CPU access (via the AEI bus)) collides with a state machine access, the CPU goes into a wait state until it receives a data valid status, indicating (in this case) that the state machine access has completed. In order to avoid provoking unexpected behavior of the algorithms, CPU write accesses to RAM regions 1b, 1c or 2 will generate interrupt requests (when enabled) so that the write accesses can be monitored.

Access priority:

1. The CPU has the highest priority for a single read/write access. The DPLL algorithm is stalled during RAM accesses.
2. After serving a CPU access to the RAM, the DPLL gets the highest RAM access priority for eight clock cycles. Afterwards, it has access priority for only one clock cycle.

The RAM address space must be implemented in the address space of the CPU.

16.6.8 Time and position stamps for actions in Emergency Mode

16.6.8.1 Equation DPLL-18 to calculate the action time stamp

Table 16-85. Equation DPLL-18a

Equation	Description
TSAC[i] = DTA[i] – DLA[i] + TS_Sx	For DTA[i] > DLA[i] and DTA[i] – DLA[i] < 0x800000

Table 16-86. Equation DPLL-18b

Equation	Description
TSAC[i] = TS_Sx	For DTA[i] < DLA[i]

Table 16-87. Equation DPLL-18c

Equation	Description
TSAC[i] = 0x7FFFFF + TS_Sx	For DTA[i] > DLA[i] and DTA[i] – DLA[i] > 0x7FFFFF

Note

For TS_Sx, see equation DPLL-6a4 and following section.

The calculation is done after the calculation of the current expected duration value according to equation DPLL-14 in Equation DPLL-6a4 to update the time stamp values for STATE. The time stamp of the action can be calculated using the delay value of the action and the current time stamp as shown in equation DPLL-18.

16.6.8.2 Equations DPLL-20 to calculate the position stamp forward

For **DIR2 = 0** and TS0_HRS = 0:

PSAC[i] = PSA[i] – (DLA[i] * RCDT_SX_nom) * MLS1;.....Equation DPLL-20

with

RCDT_SX_nom = RCDT_SX * SYN_S;.....Equation DPLL-20a

and

RCDT_SX = 1 /
CDT_SX;.....Equation DPLL-20b

For **DIR2 = 0** and TS0_HRS = 1:

PSAC[i] = PSA[i] – (8 * DLA[i] * RCDT_SX_nom) *
MLS1;.....Equation DPLL-20d

with

RCDT_SX_nom = RCDT_SX *
SYN_S;.....Equation DPLL-20a

and

RCDT_SX = 1 /
CDT_SX;.....Equation DPLL-20b

Replace MLS1 in equations DPLL-20 and DPLL-20d with MLS2 for SMC = 1 and RMO = 1.

Also use the calculated value of equation DPLL-20b for the generation of SUB_INCi and serve the action by transmission of TSAC[i] and PSAC[i] to ACT_D.

The action is to be updated for each new *STATE* event until the event is in the past. In this case, use the time stamp of the last input event instead of the calculated value and use the calculated position stamp of the actual increment as the target position value. Set the corresponding shadow bit in the DPLL_ACT_STA register. Because of the blocking read operation, the ACT_D values can be read only once.

16.6.8.3 Equations DPLL-20 to calculate the position stamp backward

For DIR2 = 0 or DIR1 = 0, and TS0_HRS = 0:

PSAC[i] = PSA[i] − (DLA[i] * RCDT_SX_nom) *
MLS1;.....Equation DPLL-20

with

RCDT_SX_nom = RCDT_SX *
SYN_S;.....Equation DPLL-20a

and

RCDT_SX = 1 / CDT_SX;.....Equation DPLL-20b

For DIR2 = 0 or DIR1 = 0, and TS0_HRS = 1:

PSAC[i] = PSA[i] − (8 * DLA[i] * RCDT_SX_nom) * MLS1;.....Equation DPLL-20d

with

RCDT_SX_nom = RCDT_SX * SYN_S;.....Equation DPLL-20a

and

RCDT_SX = 1 / CDT_SX;.....Equation DPLL-20b

Replace MLS1 in equations DPLL-20 and DPLL-20d with MLS2 for SMC = 1 and RMO = 1.

Use the calculated value of equation DPLL-20b for the generation of SUB_INCi, and serve the action by transmission of TSAC[i] and PSAC[i] to ACT_D.

The action is to be updated for each new STATE event until the event is in the past. In this case, use the time stamp of the last input event instead of the calculated value and use the calculated position stamp of the actual increment as the target position value. Set the corresponding shadow bit in the DPLL_ACT_STA register. Because of the blocking read operation, the ACT_D values can be read only once.

16.7 Signal processing

16.7.1 Time stamp processing

Signal processing, in the DPPL, is the computation of the time stamps in order to calculate at which time the outputs have to appear. For such purposes, the time stamp values have to be stored in the RAM. By calculating the difference between old and new values, the duration of the last time interval is determined. This difference should also be stored in the RAM so that the changes between the intervals by changing the conditions and the speed of the observed process can be monitored.

16.7.2 Count and compare unit

The count and compare unit processes all input signals, taking into account the configuration values. It uses a state machine to provide the output signals as described in Signal processing.

16.7.3 Sub pulse generation for SMC=0

16.7.3.1 Equation DPLL-21 to calculate the number of pulses to be sent in normal mode using the automatic end mode condition

For RMO = 0, SMC = 0 and DMO = 0:

NMB_T = (MLT + 1) * SYN_T + MP + PD_store + MPVAL1;.....Equation DPLL-21

with

PD_store = ADT_T[12:0]; pre-fetched during last increment

SYN_T = ADT_T[18:16]; pre-fetched during last increment

MPVAL1 = pulse correction value for PCM1_shadow_TRIGGER = 1

while the value for PD_store is zero for AMT = 0

and

the value of MP is zero for COA = 0

In order to get a higher resolution for higher speed, a generator for the sub-pulses is chosen using an adder. All missing MP pulses are considered using equation DPLL-21 and they are determined by counting the number of pulses in the last increment. The SYN_T value is stored from the last increment, using NT of the ADT_T[i] value at RAM region 2c.

16.7.3.2 Equations DPLL-22-24 to calculate the number of pulses to be sent in emergency mode using the automatic end mode condition for SMC = 0

For RMO = 1, SMC = 0 and DMO = 0;

the value for PD_S_store is zero for AMS = 0:

NMB_S = MLS1 * SYN_S + MP + PD_S_store;.....Equation DPLL-22

with

MLS1 = (MLT + 1) * (TNU + 1) / (SNU + 1);.....Equation DPLL-23

PD_S_store = ADT_S[15:0] was pre-fetched during last increment

SYN_S = ADT_S[21:16] waspre-fetched during last increment

MPVAL1 = pulse correction value for PCM1_shadow_STATE = 1

while the value for PD_S_store is zero for AMS = 0

and

the value of MP is zero for COA = 0

The calculations for equation DPLL-21 in Equation DPLL-21 to calculate the number of pulses to be sent in normal mode using the automatic end mode condition and equation DPLL-22 are only valid for an automatic end mode, where DMO = 0.

For calculation of the number of generated pulses, a value of 0.5 is added, as shown in equations DPLL-25 or DPLL-26 respectively, to compensate for rounding down errors during the succeeding arithmetic operations. Because in automatic end mode the number of pulses is limited by **INC_CNT1**, it is guaranteed that not more pulses than are needed are generated, and in the same, way missing pulses are caught up for the next increment.

16.7.3.3 Equation DPLL-25 to calculate ADD_IN in normal mode for SMC = 0

In normal mode (for RMO = 0), calculate in the case LOW_RES = TS0_HRT:

ADD_IN_CALN = (NMB_T + 0.5) * RCDT_TX;.....Equation DPLL-25

where

RCDT_TX is the 2^{32} time value of the quotient from equation 16.17b in Equations DPLL-17 to calculate the position stamp forward.

In normal mode (for RMO = 0), calculate in the case LOW_RES = 1 and TS0_HRT = 0:

ADD_IN_CALN = (NMB_T + 0.5) * RCDT_TX;.....Equation DPLL-25

where

RCDT_TX is the 2^{32} time value of the quotient from equation 16.17b in Equations DPLL-17 to calculate the position stamp forward.

For RMO = 0 and SMC = 0:

ADD_IN_CAL1 = ADD_IN_CALN;.....Equation DPLL-25b

LOW_RES = 0 and TS0_HRT = 1 is not a possible configuration because the DPLL_STATUS[RCT, ERR] bits are set together.

In the automatic end mode (DMO=0), missing pulses should be sent to the RPCUx (rapid pulse catch up) input (see Adder for generation of SUB_INCx by the carry c_{out}.) to be caught up on with CMU_CLK0 (for COA=0).

When normal and rapid pulses are generated simultaneously, the of SUB_INCx frequency is doubled at this moment in order to count two pulses at the TBU_CHx_BASE register. In order to make the frequency doubling possible, the CMU_CLK0 should be have a frequency which does not exceed half the frequency of TS_CLK. In addition, the ADD_IN value should never exceed the value 0x800000. This limitation is only necessary for DMO=0 and COA=0 (see DPLL_CTRL_1 register).

For the normal mode, replace ADD_IN of the ADDER (see Adder for generation of SUB_INCx by the carry c_{out}.) by ADD_IN_CAL1 (when calculated with DLM = 0) or ADD_IN_LD1 (when provided by the CPU with DLM = 1).

Sub-pulse generation, in this case, is accomplished by the following methods, where a 24-bit adder with a c_{out} carry out is used with these inputs:

- ADD_IN
- the second input is the output of the adder, stored one time stamp clock before.

In order not to complicate the calculation procedure, use a Multiplier with a sufficient bit width at the output and use the corresponding shifted output bits.

16.7.3.4 Enabling of the compensated output for pulses

The c_{out} of the adder directly influences the *SUB_INC1* output of the DPLL (see Adder for generation of SUB_INCx by the carry c_{out}.). The compensated SUB_INCxc output is in automatic end mode only when enabled by EN_Cxc when **INC_CNTx** > 0.

16.7.3.5 Equation DPLL-26 to calculate ADD_IN in emergency mode for SMC=0

In emergency mode (RMO=1), calculate in the case LOW_RES = TS0_HRS:

ADD_IN_CALE = (NMB_S + 0.5) * RCDT_SX;.....Equation DPLL-26

where

RCDT_SX is the 2^{32} time value of the quotient from equation 16.20b in Equations DPLL-20 to calculate the position stamp forward.

In emergency mode (RMO = 1), calculate in the case LOW_RES = 1 and TS0_HRS = 0:

ADD_IN_CALE = (NMB_S + 0.5) * RCDT_SX / 8;.....Equation DPLL-26a

where

RCDT_SX is the 2^{32} time value of the quotient from equation 16.20b in Equations DPLL-20 to calculate the position stamp forward.

In emergency mode (RMO = 1), calculate for RMO = 1 and SMC = 0:

ADD_IN_CALE = (NMB_S + 0.5) * RCDT_SX / 8;.....Equation DPLL-26a

where

RCDT_SX is the 2^{32} time value of the quotient in equation 16.20b in Equations DPLL-20 to calculate the position stamp forward.

For RMO = 1 and SMC = 0:

ADD_IN_CAL1 = ADD_IN_CALE;.....Equation DPLL-26b

LOW_RES = 0 and TS0_HRS = 1 is not a possible configuration because the DPLL_STATUS[RCS, ERR] bits are set together.

In the automatic end mode (DMO=0) , missing pulses should be sent to the RPCUx (rapid pulse catch up) input in Adder for generation of SUB_INCx by the carry c_{out}. , to be caught up on with CMU_CLK0 (for COA=0).

When normal and rapid pulses are generated simultaneously, the of SUB_INCx frequency is doubled at this moment in order to count two pulses at the TBU_CHx_BASE register. In order to make the frequency doubling possible, the CMU_CLK0 should be have a frequency that does not exceed half the frequency of TS_CLK. In addition, the ADD_IN value should never exceed the value 0x800000. This limitation is only necessary for DMO=0 and COA=0 (see DPLL_CTRL_1 register).

For emergency mode, replace the ADD_IN input to the ADDER (see Adder for generation of SUB_INCx by the carry c_{out}.) with ADD_IN_CAL1 (when calculated with DLM = 0) or with ADD_IN_LD1 (when provided by the CPU with DLM = 1).

The sub-pulse generation, in this case, is done by the following methods, where the 24-bit adder with a c_{out} carry out is used with these inputs:

- ADD_IN
- the second input is the output of the adder, stored one time stamp clock before.

In order not to complicate the calculation procedure, use a Multiplier with a sufficient bit width at the output and use the corresponding shifted output bits.

16.7.3.6 Adder for generation of SUB_INCx by the carry c_{out}.

Generation of SUB_INCx is shown in Figure 16-7.

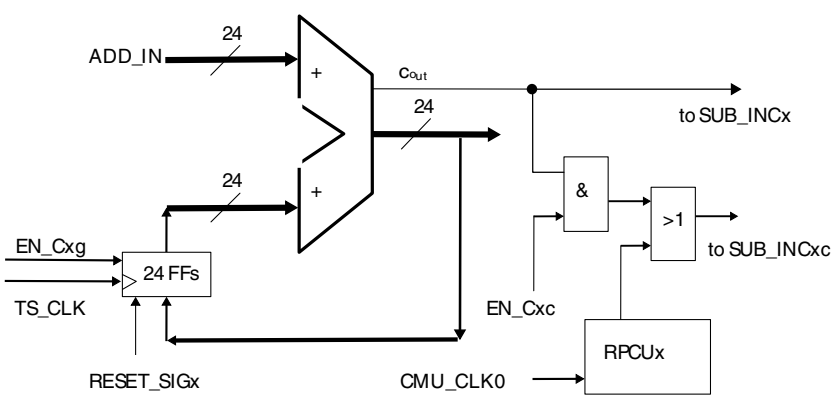


Figure 16-7. Adder for generation of SUB_INCx by the carry c_{out}

Note

SUB_INCx generation by the circuit shown in Figure 16-7 has the advantage that the resolution for higher speed values is better than that of a simple down counter.

After RESET and after EN_Cxc = 0 (after stopping in automatic end mode), the flip-flops (FFs) shown in Figure 16-7 should be reset (have a zero value). EN_Cxg has to be zero until reliable ADD_IN values are available and pulse generation starts. This is controlled by the DPLL_CONTROL_1[SGE1, SGE2] configuration bits. The calculated values for the increment prediction using equations DPLL-2c, DPLL-2c1, DPLL-7c (see Equations DPLL-10 to calculate the current increment value) or DPLL-7c1 respectively, are valid only when at least NUTE > 1 *TRIGGER* values or at least NUSE > 1 *STATE* values are available. For NUTE = 1 or NUSE = 1 respectively, the equations DPLL-25 (see Equation DPLL-25 to calculate ADD_IN in normal mode for SMC = 0) and DPLL-26 (see Equation DPLL-26 to calculate ADD_IN in emergency mode for SMC=0) use the actual increment value subtracted by the weighted average error.

The generation of SUB_INC1 pulses depends on the configuration of the DPLL. In automatic end mode, the INC_CNT1 counter resets the EN_C1 enable signal when the desired number of pulses is reached. In this case only, the uncompensated SUB_INC1 output remains active so that pulses for the input filter unit are provided. For acceleration, missing pulses can be determined at the next TRIGGER/STATE event when in normal/emergency mode. For the correction strategy when COA = 0, missing pulses are sent out at the CMU_CLK0 frequency as soon they are determined. During this time period, EN_Cxg remains cleared. After calculation or after providing of a new ADD_IN value, the FFs are enabled by EN_Cxg. In this way, no pulse is lost. The new pulses are sent out afterward, when INC_CNT1 is set to the desired value, perhaps by adding MLT + 1 or MLS1 respectively for the new TRIGGER/STATE event.

Because the DIV procedure that is used for the algorithms results in only integer values, a systematic failure could appear. The pulse generation at *SUB_INC1* will stop in automatic end mode when the **INC_CNT1** register reaches zero, or all remaining pulses at a new increment will be considered in the next calculation. In this way, the loss of pulses can be avoided.

When a new *TRIGGER/STATE* appears, the value of

SYN_T * (MLT + 1) or SYN_S * MLS1 respectively

is added to **INC_CNT1** when **SGE1 = 1.** Therefore, for FULL_SCALE,

2 * (TNU + 1) * (MLT + 1)

SUB_INC1 pulses are generated when **INC_CNT1** reaches zero. The generation of *SUB_INC1* pulses has to be done as fast as possible. The calculations for the ADD_IN value must be done first. Therefore, all values needed for the calculation must be pre-fetched.

16.7.4 Sub pulse generation for SMC = 1

16.7.4.1 Necessity for two pulse generators

Two Adders in Figure 16-7 must be implemented for SMC = 1:

- one for when SUB_INC1 is controlled by the *TRIGGER* input, and while RMO = 1,
- one for when SUB_INC2 is controlled by the *STATE* input.

For SMC = 0 (see Sub pulse generation for SMC=0), only one Adder is used to generate SUB_INC1controlled by the:

- *TRIGGER* in normal mode, or
- *STATE* in emergency mode.

16.7.4.2 Equation DPLL-27 to calculate the number of pulses to be sent for the first device using the automatic end mode condition

For SMC = 1 and DMO = 0:

NMB_T = MLS1 * SYN_T + MP + PD_store + MPVAL1;.....Equation DPLL-27

where

PD_store = ADT_T[12:0] is pre-fetched during last increment

SYN_T = ADT_T[18:16] is pre-fetched during last increment

MPVAL1 = pulse correction value for PCM1_shadow_TRIGGER = 1

while the value for PD_store is zero for AMT = 0

and,

for COA = 0, use zero instead of the value of MP

16.7.4.3 Equation DPLL-28 to calculate the number of pulses to be sent for the second device using the automatic end mode condition

For RMO = 1, SMC = 1 and DMO = 0:

NMB_S = MLS2 * SYN_S + MP + PD_S_store + MPVAL2;.....Equation DPLL-28

where

PD_S_store = ADT_S[15:0] is pre-fetched during last increment

SYN_S = ADT_S[21:16] is pre-fetched during last increment

MPVAL2 = pulse correction value for PCM2_shadow_STATE = 1

while the value for PD_S_store is zero for AMS = 0

and,

for COA = 0, use zero instead of the value of MP

The calculations in equations DPLL-27 (see Equation DPLL-27 to calculate the number of pulses to be sent for the first device using the automatic end mode condition) and DPLL-28 are only valid for an automatic end mode (DMO = 0). In addition, 0.5 is added to the number of generated pulses (as shown in equation 16.30 in Equation DPLL-30 to calculate ADD_IN for the first device for SMC = 1 or equation DPLL-31 in Equation DPLL-31 to calculate ADD_IN for the second device for SMC = 1 respectively) to compensate for rounding down errors at the subsequent division operation. Because in automatic end mode, the number of pulses is limited by **INC_CNTx**, it is guaranteed that not more pulses than needed are generated. In the same way, missing pulses are made up in the next increment.

16.7.4.4 Equation DPLL-30 to calculate ADD_IN for the first device for SMC = 1

Sub-pulse generation, for SMC = 1, is done by a 24-bit adder with a c_{out} carry out (see Figure 16-7) and these inputs:

- ADD_IN
- the second input is the delayed output of the adder, which is stored with each time stamp clock.

Replace ADD_IN by:

- ADD_IN_CAL1 (when calculated with DLM1 = 0), or
- ADD_IN_LD1 (when provided by the CPU with DLM1 = 1) respectively

while

For SMC = 1 and LOW_RES = TS0_HRT:

ADD_IN_CAL1 = (NMB_T + 0.5) * RCDT_TX;.....Equation DPLL-30

when

RCDT_TX is the 2^{32} time value of the quotient from equation DPLL-17b (see Equations DPLL-17 to calculate the position stamp forward).

For SMC = 1, LOW_RES = 1, and TS0_HRT = 0:

ADD_IN_CAL1 = (NMB_T + 0.5) * RCDT_TX / 8;.....Equation DPLL-30a

when

RCDT_TX is the 2^{32} time value of the quotient in equation DPLL-17b (see Equations DPLL-17 to calculate the position stamp forward).

In order not to complicate the calculation procedure, use a Multiplier with a sufficient bit width at the its output and use the corresponding shifted output bits.

ADD_IN_CAL1 is a 24-bit integer value. CDT_TX is the expected duration of current *TRIGGER* increment.

The c_{out} of the adder directly influences the *SUB_INC1* output of the DPLL (see Figure 16-7). The SUB_INC1 output is in automatic end mode when:

- enabled by EN_C1, and
- **INC_CNT1** >0.

16.7.4.5 Equation DPLL-31 to calculate ADD_IN for the second device for SMC = 1

Replace ADD_IN by ADD_IN_CAL2 (when calculated with DLM2 = 0) or ADD_IN_LD2 (when provided by the CPU with DLM2 = 1) respectively, while:

SMC = 1, RMO = 1 and LOW_RES = TS0_HRS:

ADD_IN_CAL2 = (NMB_S + 0.5) *
RCDT_SX;.....Equation DPLL-31

When RCDT_SX is the 2^{32} time value of the quotient in equation 16.20b (see Equations DPLL-20 to calculate the position stamp forward).

For SMC = 1, RMO = 1, LOW_RES = 1 and TS0_HRS = 0:

ADD_IN_CAL2 = (NMB_S + 0.5) * RCDT_SX /
8;.....Equation DPLL-31

When RCDT_SX is the 2^{32} time value of the quotient in equation 16.20b (see Equations DPLL-20 to calculate the position stamp forward) .

In order to not complicate the calculation procedure, use a Multiplier with a sufficient bit width at the output and use the corresponding shifted output bits.

The c_{out} of the adder2 directly influences the *SUB_INC2* output of the DPLL (see Adder for generation of SUB_INCx by the carry c_{out}.) .

The SUB_INC2 output is in automatic end mode only when enabled by EN_C2 and when **INC_CNT2** > 0.

Note

After RESET and after EN_Cxc =0 (after stopping in automatic end mode), the flip-flops (FFs) have a zero value and EN_Cxg has to be zero until reliable ADD_IN values are available and pulse generation starts. The calculated values for the increment prediction, using equations 16.2c, DPLL-2c1, DPLL-7c (see Equations DPLL-10 to calculate the current increment (nominal value)) or DPLL-7c1 respectively, are valid only when NUTE > 1 or NUSE > 1 respectively. For NUTE = 1 or NUSE = 1 respectively, the equations DPLL-30 (see Equation DPLL-30 to calculate ADD_IN for the first device for SMC = 1) and 16.31 use the actual increment value subtracted by the weighted average error.

The generation of *SUB_INCx* pulses depends on the configuration of the DPLL.

In automatic end mode, the **INC_CNTx** counter resets the EN_Cxcu enable signal when the desired number of pulses is reached. In this case, only the uncompensated SUB_INCx outputs remain active and provide pulses to the input filter units. A new *TRIGGER* or *STATE* input can reset the FFs and ADD_IN, especially when EN_Cxc was zero before. In the case of acceleration, missing pulses can be determined at the next *TRIGGER/ STATE* event. For the correction strategy, COA = 0, those missing pulses are sent out with CMU_CLK0 frequency as soon they are determined. After that, the **INC_CNTx**

pulse counter should always be zero and new pulses are sent out afterward (when **INC_CNTx** is set to the desired value by adding MLS1 or MLS2 for the new *TRIGGER* or *STATE* event respectively).

Because the DIV procedure that is used for the algorithms results in integer values only, a systematic failure can occur. Pulse generation will stop when the **INC_CNTx** register reaches zero or all remaining pulses at a new increment will be considered in the next calculation. In this way, the loss of pulses can be avoided.

When a new *TRIGGER* appears, the value of SYN_T * MLS1 is added to **INC_CNT1**. Therefore, for FULL_SCALE,

2 * (TNU+1) * MLS1

SUB_INC1 pulses are generated when **INC_CNT1** reaches the zero value. The generation of SUB_INC1 pulses has to be accomplished as fast as possible.

When a new *STATE* appears, the value of SYN_S * MLS2 is added to **INC_CNT2**. Therefore, for

FULL_SCALE,

2*(SNU + 1) * MLS2

SUB_INC2 pulses are generated when **INC_CNT2** reaches the zero value. The generation of SUB_INC2 pulses has to be accomplished as fast as possible.

16.7.5 Calculation of Accurate Position Values

After the input filter procedure all appearing TRIGGER and STATE signals have a time stamp and a position stamp assigned. For calculation of the exact time stamp, the filter values are considered in the calculations of equations DPLL-1a (see Equations DPLL-1a to calculate TRIGGER time stamps) or DPLL-6a respectively. A corresponding calculation must be performed for the calculation of position values.

The PSTC and PSSC values can be corrected by the CPU, when needed.

After reset and while FTD = 0 and no active TRIGGER slope is detected:

PSTC = 0;.....Equation DPLL-32a

Calculate the new Postion value for each valid TRIGGER event:

PSTC = PSTC_old + NMB_T_TAR_old;.....Equation DPLL-32b

when

FTD = 1 and SGE1 = 1,

where

PSTC_old is the last PSTC value and NMB_T_old is the number of pulses which are calculated and provided for sending out in the last increment.

After reset and while FSD = 0:

PSSC = 0;.....Equation DPLL-33a

Calculate the new Postion value for each STATE event:

PSSC = PSSC_old + NMB_S_TAR_old;.....Equation DPLL-32b

when

FSD = 1 and SGE1 = 1 (SMC = 0) or SGE2 = 1 (SMC = 1), respectively

where

PSSC_old is the last PSSC value and NMB_S_old is the number of pulses which are calculated and provided for sending out in the last increment.

16.7.6 Scheduling of the Calculation

After enabling the DPLL with each valid TRIGGER or *STATE* event respectively, a cycle of operations is performed to calculate all of the results shown in table of Requirements and demarcation. Two state machines, one for TRIGGER and one for STATE (see State machine partitioning for normal and emergency mode) control this procedure:

- The first state machine is triggered by a valid slope of the *TRIGGER*signal. It begins at step 1 and ends at step 20 (in normal mode and for SMC = 1).
- The second state machine is controlled by a valid slope of the signal *STATE*. It begins at step 21 and ends at step 40 (in emergency mode and also for SMC = RMO = 1).

Depending on the mode used, all 20 steps are executed. After two steps, the jump into the initial state is performed. For each new extended cycle (without this jump) all prediction values for actions (for SMC = 0) are calculated once more (with perhaps improved accuracy because of better parameters) and all pending decisions are made using these new values when they are transmitted to the decision device.

In State description of the State Machine., the steps of the state machine are described. The elaboration of the steps depends on the configuration bits described in the comments. Steps four through 17 are only calculated in normal mode (marked with color yellow in State machine partitioning for normal and emergency mode, but steps 24 through 37 are only calculated in emergency mode (marked with color cyan in State machine partitioning for normal and emergency mode) when SMC = 0.

16.7.6.1 State machine partitioning for normal and emergency mode

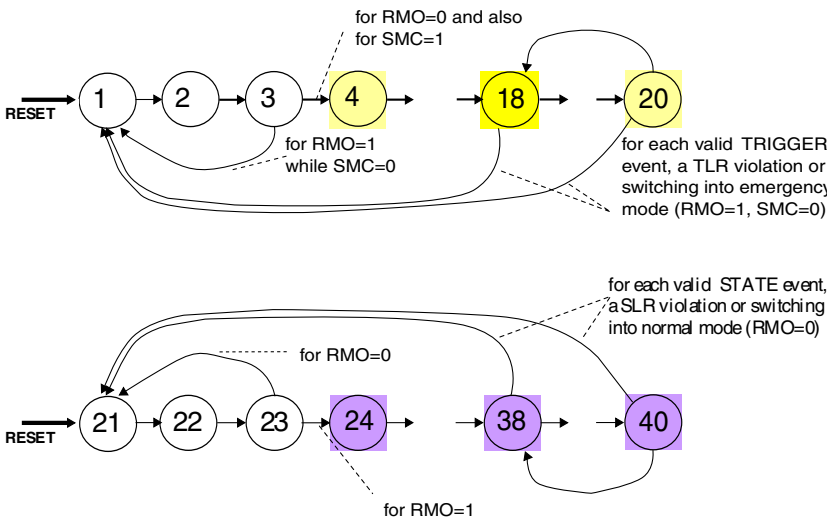


Figure 16-8. State machine partitioning for normal and emergency mode.

16.7.6.2 Synchronization description

TRIGGER:

The APT (address pointer for duration and reciprocal duration values of *TRIGGER* increments) is initially set to zero and it incremented with each valid *TRIGGER* event. Therefore, data is stored in the RAM beginning from the first available value. The actual duration of the last increment is stored at DT_T_actual. For the prediction of the next increment, it is assumed that the same value is valid as long as NUTE = 1.

A missing *TRIGGER* is assumed when (at least) after TOV* DT_T_actual, no valid *TRIGGER* event appears.

The data of equations DPLL-1b1 and DPLL-1c2 (see Equations DPLL-6b1, DPLL-6c1, and DPLL-1c2 to update RAM after calculation) are written in the corresponding RAM regions and the APT pointer is incremented accordingly up to 2 * TNU – 2 * SYN_NT + 1.

The APT_2b (address pointer for the time stamp field of *TRIGGER*) is initially set to zero and it is incremented with each valid *TRIGGER* event. When no gap is detected because of the incomplete synchronization process at the beginning, the time stamp values for all *TRIGGER* events are written in the RAM (up to 2 * (TNU + 1) entries), although only 2 * (TNU + 1 – SYN_NT) events occur in FULL_SCALE. When the current position is detected, the synchronization procedure can be performed as described below:

Before the CPU sets the APT_2c address pointer in order to synchronize to the profile, it:

- writes the corresponding APT_2b_ext increment value, for the necessary extension of the RAM region 2b, into the APT_2b_sync register, and
- sets the APT_2c_status bit.

This value can be:

- 2 * SYN_N (for example) when all gaps in FULL_SCALE already have passed the input data stream of *TRIGGER*, or
- < 2 * SYN_N when (for example) up to now, only a single gap is to be considered in the data stream already stored in RAM region 2b.

The number of virtual increments to be considered depends on the number of inputs that were already received. After the CPU sets the APT_2c pointer for the next *TRIGGER* event, the APT_2b address pointer is:

- incremented (as usual), and then
- the additional APT_2b_ext offset value is added to it once (while APT_2b_status = 1 and for the forward direction).

Then, the APT_2b_status bit is reset. The old APT_2b value, before adding the offset, is stored in the APT_2b_old register for CPU use to determine where to start the extension procedure. In the following, the CPU fills in the time stamp field around the APT_2b_old position by taking into account:

- the corresponding number of virtual entries stored in the APT_2b_ext value, and
- the corresponding NT values in the profile.

The extension procedure ends when all gaps that are considered in the APT_2b_ext value are treated once. Therefore, all RAM region 2b storage locations (up to now) are filled with the corresponding entries. Future gaps are treated by the DPLL.

For a backward direction, the APT_2c_ext value is subtracted accordingly.

When the CPU writes a value to the APT_2c address pointer, the SYT bit is set simultaneously. For SYT = 1 in normal mode with SMC = 0, the LOCK1 bit is set synchronously with the system clock when the correct number of increments between two synchronization gaps is detected by the DPLL. An unexpected missing *TRIGGER* or

an additional *TRIGGER* between two synchronization gaps resets the LOCK1 bit when in normal mode. In that case, the CPU must correct the SUB_INC pulse number and perhaps correct the APT_2c pointer. The LL1I interrupt request can be used for this purpose.

When SYT is set, the calculations of equations DPLL-1a through DPLL-5 (see Increment prediction in Normal Mode forwards (DIR1=0)) are performed accordingly and the values are stored in (and distributed to) the correct RAM positions.

This includes the multiple time stamp storage by the DPLL for a gap according to equations DPLL-1a5 through DPLL-1a7 forward (see Equation DPLL-1a5-7 to extend the time stamp values for TRIGGER in forward direction) or backward (see Equation DPLL-1a5-7 to extend the time stamp values for TRIGGER in forward direction). The APT_2b pointer, for that reason, is incremented or decremented before this operation, considering the virtual increments in the addition.

For the APT and APT_2c pointers, the gap is considered as a single increment.

STATE:

The APS (address pointer for duration and reciprocal duration values of *STATE*) is initially set to zero and incremented with each valid *STATE* event. Therefore, data is stored in the RAM field, beginning at the first location. The actual duration of the last increment is stored at DT_S_actual. For the prediction of the next increment, it is assumed that the same value is valid as long as NUSE = 1.

A missing *STATE* is assumed when no valid *STATE* event appears (at least) after TOV_S * DT_S_actual.

The data of equations DPLL-6b1 and DPLL-6c2 (see Equations DPLL-6b1, DPLL-6c1, and DPLL-1c2 to update RAM after calculation) is written in the corresponding RAM regions and the APS is incremented accordingly up to 2 * SNU - 2 * SYN_NS + 1 (for SYSF = 0).

The APS_1c2 (address pointer for the time stamp field of *STATE*) is initially set to zero and incremented with each valid *STATE* event. When no gap is detected because of the incomplete synchronization process at the beginning, the time stamp values for all *STATE* events are written in the RAM (up to 2*(SNU+1) entries), although (when SYSF = 0 for example) only 2 * (SNU + 1 – SYN_NS) events in FULL_SCALE appear. When the current position is detected, the synchronization procedure can be performed as described below:

Before the CPU sets the APS_1c3 address pointer in order to synchronize to the profile, it writes the corresponding APS_1c2_ext increment value for the necessary extension of the RAM region 1c2 into the DPLL_APS_SYNC register and sets the DPLL_APS_SYNC[APS_1c2_status] bit. The APS_1c2_ext increment value value can

be (for example) 2 * SYN_NS (for SYSF = 0) or SYN_NS (for SYSF = 1) when all gaps in FULL_SCALE have already passed the input data stream of *STATE*. Also, < this value can be considered when up to now, only a single gap is to be considered in the data stream that is already stored in RAM region 1c2. The number of increments to be considered depends on the number of inputs already received. After the CPU configures the APS_1c3 pointer, and with the next valid *STATE* slope, the APS_1c2 address pointer is incremented (as usual) and then the additional APS_1c2_ext offset value is added to it once (while APS_1c2_status = 1 and forward direction). For that reason, the APS_1c2_status bit is then reset. The old APS_1c2 value is stored in the APS_1c2_OLD register for use by the CPU to determine where to start the extension procedure.

The CPU extends the time stamp field beginning from the APS_1c2_old position, taking into account the corresponding number of virtual entries according to the APS_1c2_ext value and the correspondent NS values in the profile. The extension procedure ends when all gaps that are considered in the APS_1c2_ext value are treated once. As a result, all storage locations of RAM region 1c2 (up to now) have the corresponding entries. Future gaps are treated by the DPLL.

For a backward direction, the APS_1c2_ext value is subtracted accordingly.

When the CPU sets the APS_1c3 address pointer, the SYS bit is set simultaneously. For SYS = 1 in emergency mode (SMC = 0 and DMO = 1), the LOCK1 bit is set synchronously with the system clock when the correct number of increments between two synchronization gaps is detected by the DPLL. In emergency mode, an unexpected missing *STATE* or an additional *STATE* between two synchronization gaps results the LOCK1 bit being reset. In that case, the CPU must correct the SUB_INC1 pulse number and perhaps correct the APS_1c3 pointer. The LL1I interrupt can be used for this purpose.

When SYS is set, the calculations of equations DPLL-5 to DPLL-10 are performed accordingly and the values are stored in (and distributed to) the corresponding RAM positions. This includes multiple time stamp storage by the DPLL for a gap according to equations DPLL-6a5 through DPLL-6a7 forward (see Equations DPLL-6a5-7 to extend the time stamp values for STATE) or backward (see Equations DPLL-6a5-7 for backward direction). For that reason, the APS_1c2 pointer is incremented or decremented before this operation, considering the virtual increments in the addition.

NOTE

For the APS and APS_2c pointers, the gap is considered as a single increment.

SMC = 1:

For SMC = 1, it is assumed, that the starting position is known by measuring the characteristic of the device. In this way, the APT and APT_2c as well the APS and APS_1c3 values are set properly, but possibly with an unknown repetition rate. When no gap is to be considered for *TRIGGER* or *STATE* signals, the APT_2b and APS_1c2 address pointers are set equal to APT or APS, respectively. It is assumed that all missing *TRIGGER*s and missing *STATE*s can be considered from the beginning, when a valid profile with the corresponding adapt values is written to RAM regions 1c3 and 2c, respectively. In that case, the TSF_T[i] and TSF_S[i] must be extended by the DPLL according to the profile. Therefore, the SYT and SYS bits can be set from the beginning and the LOCK1 and LOCK2 bits are set accordingly after recognition of the corresponding gaps. When no gap exists (SYN_NT = 0 or SYN_NS = 0), the LOCK bits are set immediately. Later, and without the loss of Lock1 and Lock2, the CPU can correct the APT_2c and APS_1c3 pointer (once more) according to the recognized repetition rate.

16.7.6.3 Operation for direction change in normal and emergency mode (SMC = 0)

When SMC = 0 in normal mode and a backward condition is detected for the TRIGGER input signal (e.g. when THMI is not violated), the DPLL_STATUS[LOCK1] bit is reset to 0, the NUTC[NUTE] bit field value is set to 1 (the same for the NUSC[NUSE] bit). The APT_2c address pointers are updated as described below (and after that, it is decremented for each following valid slope of *TRIGGER* as long as the DIR1 bit shows the backward direction).

In the case of a change of direction, the DPLL_STATUS[ITN, ISN] bits are each reset to 0.

For this transition to the backward direction, no change to the APT and APT_2b address pointer is necessary.

Profile update for TRIGGER when changing direction

The APT_2c profile address pointer is changed, step by step, in order to update the profile information in SYN_T, SYN_T old and PD_store:

- decrement APT_2c, load SYN_T
- decrement APT_2c, load SYN_T
- decrement APT_2c, load SYN_T, PD_store, update SYN_T_old
- decrement APT_2c, **make calculations**, load SYN_T and PD_store, update SYN_T_old and PD_store_old and wait for a new *TRIGGER* event.

Note

Update of SYN_T_old and the loading of PD_store can be performed in all steps above. The value of the APT_2b address pointer does not need to be corrected. For a direction change from backward to forward, make the same corrections by incrementing the APT_2c profile address pointer.

NOTE

"Make calculations" means the operation of the state machine starts with the calculations of NMB_T and INC_CNT1 using the actual APT_2c address pointer value (see Requirements and demarcation).

The TBU_TB1 value must be corrected for the number of pulses sent out in the wrong direction mode during the last and current increments. This correction is accomplished by sending out SUB_INC1 pulses for decrementing the TBU_TB1 (while DIR1 = 1).

At a change of direction, save the inc_cnt1 value to inc_cnt1_save.

Calculate the new inc_cnt1 value:

1. Stop sending pulses and save inc_cnt1 as inc_cnt1_save at the moment that the direction changes.
2. Set inc_cnt1 to the target value of the last increment

   nmb_t_tar_old.
3. Add the target number of triggers, which were calculated for the current increment, when this value was already added to inc_cnt1 (before the direction change is detected)

   + nmb_t_tar.
4. Subtract the value of still not sent pulses (remaining value at inc_cnt1_save)

   - inc_cnt1_save.
5. Calculate the new target pulses to be sent, considering the new values of SYN_T and PD_store, and add them:

   + nmb_t_tar_new.

Use this equation:

inc_cnt1 = nmb_t_tar_old + nmb_t_tar – inc_cnt1_save + nmb_t_tar_new

All pulses summarized at inc_cnt1 are sent out at the maximum possible frequency because no speed information is available for the first increment after a change of direction. During a change of direction, the DPLL_CTRL1[PCM1] bit cannot be used for pulse correction.

When PSTC was incremented/decremented at the active slope, and after that, a direction change was detected at the same input event, correct **PSTC** once by:

- nmb_t_tar_old when changed to backward, or

+ nmb_t_tar_old when changed to forward,

in order to compensate the former operation. If direction information is known before an intended change of PSTC, do not change nmb_t_tar_old.

Store the new calculated nmb_t_tar_new value at nmb_t_tar for the correct calculation of PSTC at the next input event.

With the next valid *STATE* event, direction information is already given. The profile pointer APS_1c3 profile pointer must be corrected by two decrements in order to point to the profile of the next following increment. In the following, it is decremented with each *STATE* event while DIR1 = 1. The SYN_S and PD_S_store values must be updated accordingly, including SYN_S_old and PD_S_store_old.

Because the correct direction is already known when an input event appears, make these corrections:

1. decrement APS_1c3, load SYN_S and PD_S_store, update SYN_S_old and PD_S_store_old, and
2. decrement APS_1c3, **make calculations**, load SYN_S and PD_S_store, update SYN_S_old and PD_S_store_old, and wait for a new *STATE* event.

Note

The update of SYN_S_old and the loading of PD_S_store can be performed in all of the steps above. The value of the APS_1c2 pointer does not need to be corrected.

When a new *STATE* event occurs, all address pointers are decremented accordingly as long as DIR1 = 1.

Emergency mode pulse correction

Calculate the new inc_cnt1 value:

1. Stop sending pulses and save inc_cnt1 to inc_cnt1_save at the moment of direction change.
2. Set inc_cnt1 to the target value of the current increment

   nmb_s_tar.

   NOTE

   Different from the normal mode, nmb_s_tar is to be used instead of nmb_s_tar_old because direction information in emergency mode is only given from the TRIGGER input and occurs independent of a STATE event. That means the calculations at the last STATE event were done for the correct former direction. In addition, still no pulse calculations are performed for the current increment because the direction change is known at the moment of the recent STATE event. Later direction changes are considered at the next STATE event.
3. Do not add the calculated number of state pulses because no new STATE event has occurred.
4. Subtract the value of still not sent target pulses (remaining value at inc_cnt1_save)

   - inc_cnt1_save.
5. Add the new calculated target pulses for the current increment

   + nmb_s_tar_new

when, for the calculation, all new conditions of PD_S_store and SYN_S are considered.

inc_cnt1 = nmb_s_tar_old – inc_cnt1_save + nmb_s_tar_new

All pulses summarized at inc_cnt1 are sent out at the maximum possible frequency because no speed information is available for the first increment after a change of direction. During a direction change, the DPLL_CTRL1[PCM1] bit cannot be used for pulse correction.

Do not change PSSC (suppress incrementing/decrementing PSSC) at the event directly following the direction change.

Store the new calculated nmb_s_tar_new value to nmb_s_tar for the correct calculation of PSTC at the next input event.

Repeated change to forward direction for TRIGGER

The DIR1 bit remains set to 1 as long as the RAM region 1b THMI[THMI] bit field value remains none violated for the *TRIGGER* events that follow, and it is reset to 0 if an invalid TRIGGER slope violates the THMI value.

Resetting DIR1 to 0 results (after repeated reset of the LOCK1, NIT, and NIS bits) in the opposite correction of the corresponding profile address pointer.

This means:

- two increments of the APS_1c3 address pointer including the update of SYN_S and PD_S_store with the automatic update of SYN_S_old and PD_S_store_old for STATE, and
- four increments of the APT_2c address pointer including the update of SYN_T and PD_store with the automatic update of SYN_T_old and PD_store_old for TRIGGER.

Correction of TBU_CH1 is accomplished by sending out SUB_INC1 correction pulses at the highest possible frequency while DIR1 = 0. The number of pulses is calculated as shown above.

Consequences for STATE

Do the corrections above. After that, the address pointers are incremented again with each valid *STATE* event that follows as long as DIR1 = 0.

16.7.6.4 Operation for direction change for TRIGGER (SMC = 1)

When SMC = 1 and a backward condition is detected for the *TRIGGER* input signal (TDIR = 1 resulting in DIR1 = 1), the DPLL_STATUS[LOCK1] bit is reset to 0, and the NUTC[NUTE] bit field value is set to 1. The APT and APT_2c address pointers as well as the APT_2b address pointer are decremented for each valid slope of *TRIGGER* as long as DIR1 = 1(backward direction).

If a change of direction occurs, the DPLL_STATUS[ITN] bit is reset to 0.

Profile update for TRIGGER

Perform the same update steps for the profile address pointer as those shown in Operation for direction change in normal and emergency mode (SMC = 0). That is:

- decrement APT_2c, load SYN_T and PD_store, update SYN_T_old
- decrement APT_2c, **make calculations**, load SYN_T and PD_store, update SYN_T_old and PD_store_old, and wait for a new *TRIGGER* event.

In the normal case, no correction for sending incorrect pulses is necessary because the direction change is immediately detected by the pattern.

Nevertheless a correction is necessary as shown below. In the other case, see treatment of TBU_CH1_BASE pulses in normal mode in Operation for direction change in normal and emergency mode (SMC = 0).

When a direction change occurs, save the inc_cntx value to inc_cnt1_save.

Use the following steps to calculate the new inc_cnt1 value:

1. Clear inc_cnt1.
2. Set inc_cnt1 to the target value of the last increment

   nmb_t_tar.

   NOTE

   Different from normal mode, nmb_t_tar must be used instead of nmb_t_tar_old because the direction information is known before the calculation takes place.
3. Do not add the calculated number of trigger pulses because it is not yet calculated (before the direction change information is known).
4. Subtract the value of pulses that are still not sent (remaining value at inc_cnt1_save)

   – inc_cnt1_save.
5. Add the new calculated target pulses for the current increment

   + nmb_t_tar_new

   when all new conditions of PD_S_store and SYN_S are considered for the calculation.

inc_cnt1 = nmb_t_tar_old – inc_cnt1_save + nmb_t_tar_new

All pulses summarized by inc_cnt1 are sent out at the maximum possible frequency because no speed information is available for the first increment after the change of direction. During a direction change, the DPLL_CTRL1[PCM1] bit cannot be used for pulse correction.

If a direction change is detected, suppress changing the PSTC[PSTC] bit field value for the TRIGGER event .

Store the newly calculated nmb_t_tar_new value to nmb_t_tar for the correct calculation of PSTC at the next input event.

Repeated change to forward direction for TRIGGER

The DIR1 bit remains set to 1 as long as the TDIR bit is set to 1 for the *TRIGGER* events that follow; it is reset to 0 when a valid *TRIGGER* slope occurs and TDIR = 0.

Resetting DIR1 to 0 results (after repeated reset of the LOCK1 and ITN bits) in the opposite correction of the corresponding address pointer. That is, two increments of the address pointer including the update of SYN_T and PD_store.

A complex correction of TBU_CH1_BASE and INC_CNT1 is (in the normal case) not necessary when all increments are equal (SYN_NT = 0) and no adapt information is used. In this case, only the MLS1 value is added to INC_CNT1 in order to back count the value for the last increment. In the other case, see treatment of TBU_CH1_BASE and ICN_CNT1 pulses in normal mode in Operation for direction change in normal and emergency mode (SMC = 0)

16.7.6.5 Operation in backward direction for STATE (SMC = 1)

When SMC = 1 and a backward condition is detected for the *STATE* input signal (SDIR = 1 resulting in DIR2 = 1), the DPLL_STATUS[LOCK2] bit is reset to 0, the NUSC[NUSE] bit field value is set to 1, and the APS, APS_1c3_f, and APS_1c2 address pointers are decremented for each valid slope of *STATE* as long as DIR2 = 1 (backward direction).

For a change of direction, the DPLL_STATUS[ISN] bit is reset to 0.

For this transition to the backward direction, changing the APS and APS_1c2 address pointer is not necessary.

Profile update for STATE

Perform the same update steps for the profile address pointer as those shown in Operation for direction change in normal and emergency mode (SMC = 0). That is:

- decrement APT_1c3, load SYN_S and PD_S_store, update SYN_S_old, and
- decrement APT_1c3, **make calculations**, load SYN_S and PD_S_store, update SYN_S_old and PD_S_store_old, and wait for a new *STATE* event.

A complex correction of TBU_CH2_BASE and INC_CNT2 is (in the normal case) not necessary when all increments are equal (SYN_NS = 0) and no adapt information is used. In this case, only the MLS2 value is added to INC_CNT2 in order to back count the value for the last increment. In the other case, see treatment of TBU_CH2_BASE and ICN_CNT1 pulses in normal mode in Operation for direction change in normal and emergency mode (SMC = 0).

For the second PMSM, the pulses are corrected as follows:

Save inc_cnt2 value at direction change to inc_cnt2_save.

Calculate the new inc_cnt2 value:

1. Clear inc_cnt2.
2. Set inc_cnt2 to the target value of the last increment

   nmb_s_tar.

   NOTE

   Different from normal mode, nmb_s_tar must be used instead of nmb_s_tar_old because no new calculation is performed (so far).
3. Do not add the calculated number of state pulses because it is not yet calculated before the direction change information is known.
4. Subtract the amount of pulses that are still to be sent (remaining value at inc_cnt2_save)

   - inc_cnt2_save.
5. Add the new calculated target pulses for the current increment

   + nmb_s_tar_new.

when all new conditions of PD_S_store and SYN_S are considered for the calculation.

inc_cnt2 = nmb_s_tar_old – inc_cnt2_save + nmb_s_tar_new

All pulses summarized at inc_cnt2 are sent out at the maximum possible frequency because no speed information is available for the first increment after changing direction. During a direction change, the DPLL_CTRL1[PCM2] bit cannot be used for pulse correction .

When a direction change is detected do not change PSSC for a STATE event.

Store the new calculated nmb_s_tar_new value to nmb_s_tar for the correct calculation of PSTC at the next input event.

Repeated change to forward direction for STATE

The DIR2 bit remains set to 1 as long as the SDIR bit is set to 1 for the *STATE* events that follow; DIR2 is reset to 0 when a valid *STATE* slope occurs and SDIR = 0.

Resetting DIR2 to 0 results (after repeated reset of the LOCK2 and FSD bits) in the opposite correction of the corresponding address pointer.

After a last decrementing of all address pointers, the APS_1c3 pointer is incremented two times along with a repeated update of SYN_S, SYN_S_old and PD_S_store after each increment.

16.7.6.6 DPLL reaction in the case of non plausible input signals

When the DPLL is synchronized to the *TRIGGER* signal by setting the DPLL_STATUS[FTD, SYT, LOCK1] bits to 1, the number of valid *TRIGGER* events between the gaps must be continuously checked.

When additional events appear while a gap is expected, the DPLL_STATUS[LOCK1] bit is reset to 0 and the DPLL_STATUS[ITN] bit is set to 1.

When an unexpected gap appears (missing *TRIGGER*S), the NUTC[NUTE] bit field value is set to 1, the LOCK1 bit is reset to 0, and the ITN bit is set to 1. Accordingly, the address pointers are incremented with the next valid *TRIGGER* slope.

When the TLR *TRIGGER* locking range is violated, state machine 1 will remain in state 1 and the APT, APT_2b and APT_2c address pointers will remain unchanged until the CPU sets the APT_2c pointer accordingly. In this case, the NUTC[NUTE] bit field value is set to 1. The DPLL stops generation of SUB_INC1 pulses and will perform no other actions. State machine 1 will remain in step 1 (see Requirements and demarcation).

The DPLL_STATUS[TOR] bit is set to 1 when the time to the next active TRIGGER slope exceeds the value of the last nominal TRIGGER duration multiplied by the value in the TLR register. In this case, a TORI interrupt request is generated (when enabled).

When (in the events that follow) the DIR1 direction changes (as described in the sections above), the DPLL_STATUS[ITN] bit is reset to 0, the use of the APT_2c address pointers is switched, and pulse correction takes place as described above.

In all other cases, the CPU can interact to force departure from the instable state. This can be accomplished by setting the APT_2c address pointer, which results in a reset of the ITN bit. In the following, NUTE can be set to higher values.

When the DPLL is synchronized to the *STATE* signal:

- by setting (for SMC = 0) the DPLL_STATUS[FSD, SYS, LOCK1] bits to 1 or
- by setting (for SMC=1) the DPLL_STATUS[LOCK2] bit to 1,

the amount of valid *STATE* events between the gaps must be continuously checked.

When additional events appear while a gap is expected or while an unexpected missing *STATE* event appears, the LOCK1,2 bit is reset to 0 and the ISN bit is set to 1.

When an unexpected gap appears for RMO = SMC = 1 (missing *STATE*s for synchronous motor control), the NUSC[NUSE] bit field value is set to 1, the LOCK2 bit is reset to 0, and the ISN bit is set to 1. Accordingly, the address pointers are incremented with the next valid *STATE* slope.

When the SLR *STATE* locking range is violated, state machine 2 will remain in state 21 and the APS, APS_1c2, and APS_1c3 address pointers will remain unchanged until the CPU sets the APS_1c3 pointer, accordingly. In this case, the NUSC[NUSE] bit field value is set to 1. The DPLL stops generation of SUB_INC1,2 pulses, respectively, and it will perform no other actions – remaining in step 21 of state machine 2 (see Requirements and demarcation).

The DPLL_STATUS[SOR] bit is set to 1 when the time to the next active STATE slope exceeds the value of the last nominal STATE duration multiplied by the value in the SLR register.

In this case, a SORI interrupt request is generated (when enabled).

When (in the events that follow) the DIR2 direction changes as described in the sections above, the ISN bit is reset to 0, the use of the APS_1c3 address pointers is switched, and pulse correction takes place as described above. In all other cases, the CPU must interact to force departure from the instable state. This can be accomplished by setting the APS_1c3 address pointers, which results in a reset of the ISN bit. For the events that follow, the NUSE bit field value can be set to higher values.

16.7.6.7 State description of the State Machine.

Table 16-88. State descriptions of the state machine

Step	Description	Comments

Always for DEN=1	**For each inactive TRIGGER slope:**
	Generate the TISI interrupt; calculate the time stamp difference ΔT to the last valid event, store this value at THVAL;
	when THMI >0 is violated (ΔT < THMI):
	generate TINI interrupt,
	set DIR1=0 (forward)
	set BWD1=0 (see DPLL_STATUS register)
	else (only for THMI >0):
	set DIR1= 1 (backward);
	set BWD1=1 (see DPLL_STATUS register)
	after changing the direction correct the pulses WP sent with wrong direction information and send the pulses for the actual increment in addition with highest possible frequency: WP=NMB_T-DPLL_INC_CNT1;
		For SMC=0;
		Set DIR1 always after inc./ decr. the address pointers APT, APT_x;
		go to step 1;
		stop output of SUB_INC1 and correct pulses after changing DIR1 after increment/decrement of APS_x
		set DIR2 always after incr./decr. the address pointers APS, APS_x;
		go to step 1

Table continues on the next page...

Table 16-88. State descriptions of the state machine (continued)

Step	Description	Comments
	correct INC_CNT1 by addition of 2*WP before sending the correction pulses; check THMA, when THMA is violated, generate the TAXI interrupt; go to step 1 **for each inactive *STATE* slope:** set DIR2=DIR1	
Always for DEN=1	Set DIR1=BWD1=TDIR , set DIR2=BWD2=SDIR; for each change of TDIR go to step 1 after performing the following calculations: correct INC_CNT1 correct the pulses (WP, see above) sent with wrong direction information and send the pulses for the actual increment in addition with highest possible frequency. For each change of SDIR go to step 21 after performing the following calculations: update of SYN_S, PD_S_store according to Operation for direction change in normal and emergency mode (SMC = 0) correct INC_CNT1,2 correct the pulses sent with wrong direction information and send the pulses for the actual increment in addition with highest possible frequency.	**For SMC=1**; Set the direction bits always after increment/decrement The corresponding address pointer.
1	When DEN = 0 or TEN=0: stay in step 1 until DEN=1, TEN=1 and at least one valid *TRIGGER* has been detected (FTD=1); the following steps are performed always (not necessarily in step 1, but also in steps 18 to 20 (when waiting for new PMTR values to be calculated): compare TRIGGER_S with TSL (valid slope); **When no valid TRIGGER appears** and when TS_T_CHECK time is reached: • send missing *TRIGGER* INT, also when a Gap is expected according to the profile; set MT=1 (missing *TRIGGER* bit) in the DPLL_STATUS register; do not leave the active step, until a valid *TRIGGER* appears. **When a valid TRIGGER appears check PVT** • **when the PVT value is violated:** generate the PWI interrupt, ignore the *TRIGGER* input and wait for the next valid TRIGGER slope (ignore each invalid slope); do not store any value • **When the PVT value is fulfilled:** store the actual position stamp at PSTM (value at the TRIGGER event)	Depending on TSL, TEN, DEN the leaving of step one is done with the next *TRIGGER* input; **Note:** Step 1 is also left in emergency mode when a valid *TRIGGER* event appears in order to make a switch back to normal mode possible; _old - values are values valid at the last but one valid *TRIGGER* event ; for the whole table: use always MLS1 instead of (MLT+1) for the case SMC=1 ; **dir_crement** does mean: increment for DIR1=0 decrement for DIR1=1 *)replace (MLT+1) by MLS1 for SMC=1 **) NMB_T_TAR is the target value of NMB_T of the last increment (see step 5 ff.)

Table continues on the next page...

Table 16-88. State descriptions of the state machine (continued)

Step	Description	Comments
	update the RAM region 2 by equations 16.a through 16.c (see Update of RAM in Normal and Emergency Mode) **Store the actual INC_CNT1 value at MP1 as missing pulses***(instead of calculation in step 5)* Store all relevant configuration bits **X** of the DPLL_CTRL(0,1) registers in **shadow** registers and consider them for all corresponding calculations of steps 2 to 20 accordingly; the relevant bits are explained in the registers itself generate the TASI interrupt; for FTD=0: • set PSTC=PSTM • set FTD (first *TRIGGER* detected) • do not change PSTC,APT, APT_2b • for (RMO=0 or SMC=1) **and SGE1=1**: increment INC_CNT1 by (MLT+1)*) +MPVAL1***) • send SUB_INC1 pulses with highest possible frequency **when SGE1=1** for SYT=0 and FTD =1: • **dir_crement** APT and APT_2b by one; • dir_crement for SGE1_delay****)=1 and TOR=0 PSTC by NMB_T_TAR**) • for (RMO=0 or SMC=1) **and SGE1=1, TOR=0**: increment INC_CNT1 by (MLT+1)*) +MPVAL1***) for SYT=1 and TOR=0 : • **dir_crement** APT, APT_2c**, dir_crement** APT_2b by SYN_T_old • **dir_crement** for **SGE1_delay****)=1** PSTC by NMB_T_TAR**) • for (RMO=0 or SMC=1) **and SGE1=1**: increment INC_CNT1 by SYN_T*(MLT+1)*)+ PD_store*****) + MPVAL1***) PD_store is 0 for AMT=0 within the DPLL_STATUS register: • set LOCK1 bit accordingly;	***) add MPVAL1 once to INC_CNT1, that means only when PCM1=1 ****)**SGE1_delay** is the value of SGE1 delayed by one valid TRIGGER event *****) PD_store = 0 for AMT=0 (see DPLL_CTRL_0 register)
2	Calculate TS_T according to equation 16.1a (see Update of RAM in Normal and Emergency Mode); calculate DT_T_actual = TS_T - TS_T_old calculate RDT_T_actual	

Table continues on the next page...

Table 16-88. State descriptions of the state machine (continued)

Step	Description	Comments
	calculate QDT_TX according to equation 16.2 (see Update of RAM in Normal and Emergency Mode)	
3	Send CDTI interrupt when NTI_CNT is zero or decrement NTI_CNT when not zero; Calculate EDT_T and MEDT_T according to Equation DPLL-3 to calculate the error of last prediction and Equation DPLL-4 to calculate the weighted average error for (RMO=1 and SMC=0): update SYN_T, PD_store and go back to step 1	**Note:** There are different behaviors of RM and HW-IP: For the HW-IP the values of SYN_T and PD_store are not updated until a new valid TRIGGER slope occurs.
4	Calculate CDT_TX according to Equation DPLL-5a and Equation DPLL-5b in Equation DPLL-5 to calculate the current increment value ;	For RMO=0 or SMC=1.
5	Calculate missing pulses: MP1 = INC_CNT1(at the moment of a valid TRIGGER slope) calculate target pulses: NMB_T_TAR = (MLT+1)*)*SYN_T + PD_store + MPVAL1 (instead of PD_store use zero in the case AMT=0)	For RMO=0 or SMC=1; *)replace (MLT+1) by MLS1 for SMC=1; add MPVAL1 only for PCM=1 and reset PCM1 after that.
6	Send MP with highest possible frequency and set NMB_T = NMB_T_TAR	For RMO=0 or SMC=1, DMO=0 and COA=0.
7	Calculate the number of pulses to be sent NMB_T = NMB_T_TAR + MP (see Equation DPLL-21 to calculate the number of pulses to be sent in normal mode using the automatic end mode condition or Equation DPLL-27 to calculate the number of pulses to be sent for the first device using the automatic end mode condition respectively)	For RMO=0 or SMC=1, DMO=0 and COA=1
8	NMB_T = SYN_T * CNT_NUM1	For RMO=0 or SMC=1, DMO=1.
9	Update SYN_T and PD_store;	**Note:** There are different behaviors of RM and HW-IP: For the HW-IP the values of SYN_T and PD_store are not updated until a new valid TRIGGER slope occurs.
10	Calculate ADD_IN_CAL1 according to Equation DPLL-25 to calculate ADD_IN in normal mode for SMC = 0 and Equation DPLL-25 to calculate ADD_IN in normal mode for SMC = 0 or Equation DPLL-31 to calculate ADD_IN for the second device for SMC = 1 and store this value in RAM use ADD_IN_CAL1 as ADD_IN value for the case DLM=0 use ADD_IN_LD1 as ADD_IN for the case DLM=1, but do this update immediately (without waiting for this step 10); for DMO=DLM=0 and EN_C1u=0: reset the FlipFlops in the SUB_INC1 generator; start sending SUB_INC1;	for RMO=0 or SMC=1 for DLM=0 for DLM=1

Table continues on the next page...

Table 16-88. State descriptions of the state machine (continued)

Step	Description	Comments
11	Calculate: TS_T_CHECK = TS_T + DT_T_actual *(TOV)	For RMO=0 or SMC=1;
12	Automatic setting of actions masking bits in the DPLL_STATUS register: for SMC=0: set CAIP1=CAIP2=1 for SMC=1: set only CAIP1=1	Steps 12 to 16 are not valid for the combination: (SMC=0 and RMO=1)
13	For all correspondent actions with ACT_N[i]=1 calculate: NA[i] = (PSA[i] - PSTC)/(MLT+1)*)for forward direction with w= integer part and b = remainder of the division (fractional part); for backward direction use NA[i] = (PSTC - PSA[i])/(MLT+1)*) and consider in both cases the time base overflow in order to get a positive difference	Actions 0…11 for SMC=1 Actions 0…23 for SMC=0 Depending on ACT_N[i] in **DPLL_ACT_STA** register; replace MLT+1 by MLS1 for SMC=1
14	Calculate PDT_T[i] and DTA[i] for up to 24 action values according to Equation DPLL-11a1 to calculate the time prediction for an action and Equation DPLL-12 to calculate the duration value until action;	Actions 0…11 for SMC=1 Actions 0…23 for SMC=0
15	Calculate TSAC[i] according to Equation DPLL-15 to calculate the action time stamp and PSAC[i] according to Equations DPLL-17 to calculate the position stamp forward	Actions 0…11 for SMC=1 Actions 0…23 for SMC=0
16	Automatic resetting of actions masking bits in the DPLL_STATUS register: for SMC=0: set CAIP1=CAIP2=0 for SMC=1: set only CAIP1=0; set the corresponding ACT_N[i] bits in the DPLL_ACT_STA register	Set ACT_N[i] for all enabled actions concerned: 0…11 for SMC=1 0…23 for SMC=0
17	Check the relation of the last increment to its predecessor according to the profile and taking into account TOV: set the ITN status bit and reset the corresponding LOCK bit, when not plausible; go to step 18, when no valid *TRIGGER* appears **for all following steps 18 to 20:** **go immediately back to step 1, when a valid TRIGGER event occurs, interrupt all calculations there and reset all CAIP in that case;** **when going back to step 1:** store TS_T in RAM 2b according to APT_2b; update RAM 2a and RAM 2d	For all conditions
18	Wait for a new PMTR value; set the corresponding CAIPx values and go to step 19 in that case	Go immediately to step 1 and update the RAM according to step 17 when a valid *TRIGGER* event occurs

Table continues on the next page...

Table 16-88. State descriptions of the state machine (continued)

Step	Description	Comments
19	Make the requested action calculation according to new PMTR values	Go immediately to step 1 and update the RAM according to step 17 when a valid *TRIGGER* event occurs
20	Reset CAIPx and go back to step 18	Go immediately to step 1 and update the RAM according to step 17 when a valid *TRIGGER* event occurs
21	When DEN = 0 or SEN=0: stay in step 1 until DEN=1, SEN=1 and at least one valid *STATE* has been detected (FSD=1); the following steps are performed alsways (not necessarily in step 21, but also in steps 38 to 40 (when waiting for new PMTR values to be calculated): compare STATE_S with SSL (valid slope); for each invalid slope: generate a SISI interrupt; • send missing *STATE* INT when TS_S_CHECK time is reached and set MS=1 (missing *STATE* bits) in that case; do not leave step 21 while no valid *STATE* appears. **When a valid *STATE* appears:** store the actual position stamp at PSSM (value at the STATE event) update RAM by Equations DPLL-10 to calculate the current increment (nominal value) (see Update of RAM in Normal and Emergency Mode); **store the actual INC_CNT1/2 at MP1/MP2 respectively as missing pulses** *(instead of calculations in step 25)* store all relevant configuration bits **X** of the DPLL_CTRL(0,1) Registers in **shadow** registers and consider them for all corresponding calculations of steps 22 to 37 accordingly; the relevant bits are explained in the registers itself for FSD=0: • set PSSC=PSSM • set FSD (first *STATE* detected) • do not increment PSSC • for (RMO=1 and SMC=0) **and SGE1=1**: increment INC_CNT1 by MLS1+MPVAL1**) • for (RMO=1 and SMC=1) **and SGE2=1**: increment INC_CNT2 by MLS2+MPVAL2**) • for SYS=0, FSD =1 and SOR=0: • **dir_crement** PSSC by NMB_S_TAR*) for (SMC=0 **and SGE1_delay*****)**=1**) or (SMC=1 and **SGE2_delay******)**=1**)	Depending on SSL, SEN, DEN the leaving of step 21 one is done with the next *STATE* input; for the steps 22-37: for SMC=1 replace: MLS1 by MLS2, LOCK1 by LOCK2; SUB_INC1 by SUB_INC2; CNT_NUM1 by CNT_NUM2; MPVAL1 by MPVAL2; EN_C1u by EN_C2u; **dir_crement** does mean: increment for DIR2=0 decrement for DIR2=1 or DIR1 respectively *) target number of pulses of the last increment (see step 25 ff.) **) add MPVAL1 or MPVAL2 only once, that means as long as PCM1 or PCM2 is set respectively ***)**SGE1_delay** is the value of SGE1 delayed by one valid STATE event ****)**SGE2_delay** is the value of SGE2 delayed by one valid STATE event *****) PD_S_store = 0 for AMS=0 (see DPLL_CTRL_0 register)

Table continues on the next page...

Table 16-88. State descriptions of the state machine (continued)

Step	Description	Comments
	• increment INC_CNT1 by MLS1+MPVAL1**) (for SMC=0, **SGE1=1** and RMO=1); • increment INC_CNT2 by MLS2+MPVAL2**) (for SMC=1, **SGE2=1** and RMO=1); • **dir_crement** APS and APS_1c3 for SYS=1 and SOR=0: • **dir_crement** APS and APS_1c3 • **dir_crement** APS_1c2 by SYN_S_old • for RMO=1 and SMC=0: for **SGE1_delay*****)**=1dir_crement** PSSC by NMB_S_TAR*); for **SGE1=1** increment INC_CNT1 by SYN_S*MLS1 + PD_S_store + MPVAL1**) • for RMO=1 and SMC=1: for **SGE2_delay******)**=1dir_crement** PSSC by NMB_S_TAR*); for **SGE2=1** increment INC_CNT2 by SYN_S*MLS2 + PD_S_store*****) +MPVAL2**) • within the DPLL_STATUS register: set LOCK1 or 2 bit accordingly;	
22	Calculate TS_S according to Equations DPLL-6a to calculate STATE time stamps; calculate DT_S_actual = TS_S - TS_S_old calculate RDT_S_actual calculate QDT_SX	
23	Send CDSI interrupt; calculate EDT_S and MEDT_S according to Equation DPLL-8 to calculate the error of last prediction and Equation DPLL-8 to calculate the error of last prediction for RMO=0: go back to step 21 for RMO=0 and update SYN_S and PD_S_store using the current ADT_S[i] values in that case	**Note:** There are different behaviours of RM and HW-IP: For the HW-IP the values of SYN_S and PD_S_store are not updated until a new valid STATE slope occurs.
24	Calculate CDT_SX according to Equations DPLL-10 to calculate the current increment (nominal value) and Equations DPLL-10 to calculate the current increment (nominal value);	Only for RMO=1
25	Calculate missing pulses: • for TBU_CH1: MP1 = INC_CNT1(valid STATE slope) • for TBU_CH2: MP2 = INC_CNT2(valid STATE slope) calculate target number of pulses: NMB_S_TAR = MLS1*SYN_S + PD_S_store +MPVAL1 (for SMC=0)	Only for RMO=1 for SMC=0 instead of MPVAL1 use zero for PCM1=0 for SMC=1 instead of MPVAL2 use zero for PCM2=0; add MPVAL1/2 once to INC_CNT1/2 and reset PCM1/2 after that

Table continues on the next page...

Table 16-88. State descriptions of the state machine (continued)

Step	Description	Comments
	NMB_S_TAR = MLS2*SYN_S + PD_S_store + MPVAL2 (for SMC=1) (instead of PD_S_store use zero in the case AMS=0)	
26	Send MPx with highest possible frequency and set NMB_S = NMB_S_TAR	Only for RMO=1, DMO=0 and COA=0
27	Calculate number of pulses to be sent according to Equations DPLL-22-24 to calculate the number of pulses to be sent in emergency mode using the automatic end mode condition for SMC = 0 or NMB_S = NMB_S_TAR + MPx	Only for RMO=1, DMO=0 and COA=1
28	NMB_S = SYN_S*CNT_NUM1 (SMC=0) NMB_S = SYN_S*CNT_NUM2 (SMC=1)	Only for RMO=1, DMO=1
29	Update SYN_S and PD_S_store	Note: There are different behaviors of RM and HW-IP: For the HW-IP the values of SYN_S and PD_S_store are not updated until a new valid STATE slope occurs.
30	Calculate ADD_IN_CAL2 according to Equation DPLL-26 to calculate ADD_IN in emergency mode for SMC=0 and Equation DPLL-26 to calculate ADD_IN in emergency mode for SMC=0 or Equation DPLL-31 to calculate ADD_IN for the second device for SMC = 1 respectively and store this value in RAM use ADD_IN_CAL2 as ADD_IN value for the case DLM=0 use ADD_IN_LD2 as ADD_IN for the case DLM=1, but do this update immediately (without waiting for this step 30); for RMO=1, DMO=DLM=0 and EN_C1u=0 (EN_C1u=0): reset the FlipFlops in the SUB_INC1 or SUB_INC2 generator respectively; start sending SUB_INC1 / SUB_INC2	Only for RMO=1 for DLM=0 for DLM=1
31	Calculate: TS_S_CHECK = TS_S + DT_S _actual * (TOV_S);	Only for RMO=1;
32	Automatic setting of actions masking bits in the DPLL_STATUS register: CAIP1 and CAIP2 for SMC=0 only CAIP2 for SMC=1	For RMO=1
33	For all actions with ACT_N[i]=0 calculate: NA[i] = (PSA[i] - PSSC)/MLS1 for forward direction with w = integer part and b = remainder of the division (fractional part) for backward direction use	For SMC=0: 24 actions, for SMC=1: 12 actions; depending on ACT_N[i] in **DPLL_ACT_STA** register

Table continues on the next page...

Table 16-88. State descriptions of the state machine (continued)

Step	Description	Comments
	NA[i] = (PSSC - PSA[i])/(MLS1) and consider in both cases the time base overflow in order to get a positive difference use MLS2 as divider in the case of SMC=1	
34	Calculate PDT_S[i] and DTA[i] for up to 24 action values according to Equation DPLL-13a1 to calculate the time prediction for an action and Equation DPLL-14 to calculate the duration value for an action;	Only for RMO=1; for SMC=0 actions 0…23 for SMC=1 actions 12…23
35	Calculate TSAC[i] according to Equation DPLL-18 to calculate the action time stamp and PSAC[i] according to Equations DPLL-20 to calculate the position stamp forward	For the relevant actions (see above) and RMO=1
36	Automatic reset of the actions masking bit CAIP in the DPLL_STATUS register: CAIP1=CAIP2=0 for SMC=0 and only CAIP2=0 for SMC=1 set the corresponding ACT_N[i] bits in the DPLL_ACT_STA register	For the relevant actions (see above) and RMO=1 Set ACT_N[i] and reset ACT_WRi for all enabled actions
37	Check the duration of the last increment to its predecessor according to the profile and taking into account TOV_S: set the ISN status bit and reset the corresponding LOCK bit, when not plausible; go to step 38, when no valid *STATE* appears for all following steps 38 to 40: **go immediately back to step 21, when a valid*STATE* event occurs, interrupt all calculations there and reset all CAIPx in that case;** **when going back to step 21:** store TS_S in RAM 1c2 according to APS_1c2; update RAM 1c1 and RAM 1c4	For all conditions
38	Wait for a new PMTR value; set the corresponding CAIPx values and go to step 39 in that case	Go immediately to step 21 and update the RAM according to step 37 when a valid *STATE* event occurs
39	Make the requested action calculation according to new PMTR values	Go immediately to step 21 and update the RAM according to step 37 when a valid *STATE* event occurs
40	Reset CAIP and go back to step 38	Go immediately to step 21 and update the RAM according to step 37 when a valid *STATE* event occurs

16.8 DPLL Interrupt signals

The DPLL provides 27 interrupt request signals as shown in Table 16-89.

Table 16-89. DPLL interrupt request signals

Signal	Description
DPLL_SORI_IRQ	*STATE* is out of range
DPLL_TORI_IRQ	*TRIGGER* is out of range
DPLL_CDSI_IRQ	*STATE* duration calculated for last increment
DPLL_CDTI_IRQ	*TRIGGER* duration calculated for last increment
DPLL_TE4_IRQ	*TRIGGER* event interrupt 4 request[3)]
DPLL_TE3_IRQ	*TRIGGER* event interrupt 3 request[3)]
DPLL_TE2_IRQ	*TRIGGER* event interrupt 2 request[3)]
DPLL_TE1_IRQ	*TRIGGER* event interrupt 1 request[3)]
DPLL_TE0_IRQ	*TRIGGER* event interrupt 0 request[3)]
DPLL_LL2_IRQ	Lost of lock interrupt for SUB_INC2 request
DPLL_GL2_IRQ	Get of lock interrupt for SUB_INC2 request
DPLL_E_IRQ	Error interrupt request
DPLL_LL1_IRQ	Lost of lock interrupt for SUB_INC1 request
DPLL_GL1_IRQ	Get of lock interrupt for SUB_INC1 request
DPLL_W1_IRQ	Write access to RAM region 1b or 1c interrupt request
DPLL_W2_IRQ	Write access to RAM region 2 interrupt request
DPLL_PW_IRQ	Plausibility window violation interrupt of *TRIGGER* request
DPLL_TAS_IRQ	*TRIGGER* active slope while NTI_CNT is zero interrupt request
DPLL_SAS_IRQ	*STATE* active slope interrupt request
DPLL_MT_IRQ	Missing *TRIGGER* interrupt request
DPLL_MS_IRQ	Missing *STATE* interrupt request
DPLL_TIS_IRQ	*TRIGGER* inactive slope interrupt request
DPLL_SIS_IRQ	*STATE* inactive slope interrupt request
DPLL_TAX_IRQ	*TRIGGER* maximum hold time violation interrupt request
DPLL_TIN_IRQ	*TRIGGER* minimum hold time violation interrupt request
DPLL_PE_IRQ	DPLL enable interrupt request
DPLL_PD_IRQ	DPLL disable interrupt request

3). see TINT value in the corresponding ADT_T[i] section of RAM region 2

Note

TEi_IRQ depends on the TINT value in ADT_T[i][1)] and is only active when SYT, [2)] =1.

1). see RAM region 2 explanation
2). see DPLL STATUS register

16.9 DPLL Memory Map and Registers

The DPLL module registers are described as follows:

DPLL memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Control Register 0 (DPLL_CTRL_0)	32	R/W	003B_BA57h	16.9.1/576
4	Control Register 1 (DPLL_CTRL_1)	32	R/W	B000_0000h	16.9.2/578
8	Action Enable Register (DPLL_CTRL_2)	32	R/W	0000_0000h	16.9.3/583
C	Action Enable Register (DPLL_CTRL_3)	32	R/W	0000_0000h	16.9.4/586
10	Action Enable Register (DPLL_CTRL_4)	32	R/W	0000_0000h	16.9.5/589
14	Action Enable Register (DPLL_CTRL_5)	32	R/W	0000_0000h	16.9.6/593
18	Action Status Register including Shadow Register (DPLL_ACT_STA)	32	R/W	0000_0000h	16.9.7/596
1C	Offset and Switch Old/New Address Register (DPLL_OSW)	32	R/W	0000_0200h	16.9.8/597
20	Address Offset Register of RAM2 Regions (DPLL_AOSV_2)	32	R	1810_0800h	16.9.9/598
24	Actual RAM Pointer Address for Trigger (DPLL_APT)	32	R/W	0000_0000h	16.9.10/ 599
28	Actual RAM Pointer Address for STATE (DPLL_APS)	32	R/W	0000_0000h	16.9.11/ 601
2C	Actual RAM Pointer Address for Region 2C (DPLL_APT_2C)	32	R/W	0000_0000h	16.9.12/ 602
30	Actual RAM Pointer Address for Region 1C3 (DPLL_APS_1C3)	32	R/W	0000_0000h	16.9.13/ 603
34	Number of Recent TRIGGER Events used for Calculations (DPLL_NUTC)	32	R/W	0001_2001h	16.9.14/ 604
38	Number of Recent STATE Events used for Calculations (DPLL_NUSC)	32	R/W	0000_2081h	16.9.15/ 606
3C	Number of Active TRIGGER Events to Interrupt (DPLL_NTI_CNT)	32	R/W	0000_0000h	16.9.16/ 607
40	Interrupt Register (DPLL_IRQ_NOTIFY)	32	w1c	0000_0000h	16.9.17/ 608
44	Interrupt Enable Register (DPLL_IRQ_EN)	32	R/W	0000_0000h	16.9.18/ 611
48	Force Interrupt Register (DPLL_IRQ_FORCINT)	32	R/W	0000_0000h	16.9.19/ 614
4C	Interrupt Request Mode (DPLL_IRQ_MODE)	32	R/W	See section	16.9.20/ 618
50	Error Interrupt Enable Register (DPLL_EIRQ_EN)	32	R/W	0000_0000h	16.9.21/ 619
B0	Counter Value of Sent SUB_INC1 Pulses (DPLL_INC_CNT1)	32	R/W	0000_0000h	16.9.22/ 622

Table continues on the next page...

DPLL memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
B4	Counter Value of sent SUB_INC2 values (for SMC=1 and RMO=1) (DPLL_INC_CNT2)	32	R/W	0000_0000h	16.9.23/ 623
B8	TRIGGER Time Stamp Field Offset at Synchronization Time (DPLL_APT_SYNC)	32	R/W	0000_0000h	16.9.24/ 624
BC	STATE Time Stamp Field Offset at Synchronization Time (DPLL_APS_SYNC)	32	R/W	0000_0000h	16.9.25/ 626
C0	Time Stamp Value for the last valid TRIGGER (DPLL_TBU_TS0_T)	32	R/W	0000_0000h	16.9.26/ 627
C4	Time Stamp Value for the last valid STATE (DPLL_TBU_TS0_S)	32	R/W	0000_0000h	16.9.27/ 628
C8	ADD_IN Value in Direct Load Mode for TRIGGER (DPLL_ADD_IN_LD1)	32	R/W	0000_0000h	16.9.28/ 628
CC	ADD_IN Value in Direct Load Mode for STATE (DPLL_ADD_IN_LD2)	32	R/W	0000_0000h	16.9.29/ 629
FC	Status Register (DPLL_STATUS)	32	R	0000_0000h	16.9.30/ 629
100	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_0)	32	R/W	0000_01FEh	16.9.31/ 635
104	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_1)	32	R/W	0000_01FEh	16.9.31/ 635
108	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_2)	32	R/W	0000_01FEh	16.9.31/ 635
10C	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_3)	32	R/W	0000_01FEh	16.9.31/ 635
110	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_4)	32	R/W	0000_01FEh	16.9.31/ 635
114	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_5)	32	R/W	0000_01FEh	16.9.31/ 635
118	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_6)	32	R/W	0000_01FEh	16.9.31/ 635
11C	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_7)	32	R/W	0000_01FEh	16.9.31/ 635
120	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_8)	32	R/W	0000_01FEh	16.9.31/ 635
124	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_9)	32	R/W	0000_01FEh	16.9.31/ 635
128	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_10)	32	R/W	0000_01FEh	16.9.31/ 635
12C	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_11)	32	R/W	0000_01FEh	16.9.31/ 635
130	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_12)	32	R/W	0000_01FEh	16.9.31/ 635
134	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_13)	32	R/W	0000_01FEh	16.9.31/ 635

Table continues on the next page...

DPLL memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
138	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_14)	32	R/W	0000_01FEh	16.9.31/ 635
13C	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_15)	32	R/W	0000_01FEh	16.9.31/ 635
140	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_16)	32	R/W	0000_01FEh	16.9.31/ 635
144	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_17)	32	R/W	0000_01FEh	16.9.31/ 635
148	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_18)	32	R/W	0000_01FEh	16.9.31/ 635
14C	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_19)	32	R/W	0000_01FEh	16.9.31/ 635
150	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_20)	32	R/W	0000_01FEh	16.9.31/ 635
154	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_21)	32	R/W	0000_01FEh	16.9.31/ 635
158	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_22)	32	R/W	0000_01FEh	16.9.31/ 635
15C	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_23)	32	R/W	0000_01FEh	16.9.31/ 635
160	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_24)	32	R/W	0000_01FEh	16.9.31/ 635
164	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_25)	32	R/W	0000_01FEh	16.9.31/ 635
168	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_26)	32	R/W	0000_01FEh	16.9.31/ 635
16C	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_27)	32	R/W	0000_01FEh	16.9.31/ 635
170	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_28)	32	R/W	0000_01FEh	16.9.31/ 635
174	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_29)	32	R/W	0000_01FEh	16.9.31/ 635
178	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_30)	32	R/W	0000_01FEh	16.9.31/ 635
17C	ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_31)	32	R/W	0000_01FEh	16.9.31/ 635
1E0	Shadow Register of DPLL_CTRL_0 controlled by a valid TRIGGER Slope (DPLL_CTRL_0_SHADOW_TRIGGER)	32	R	0000_0257h	16.9.32/ 636
1E4	Shadow Register of DPLL_CTRL_0 controlled by a valid STATE Slope (DPLL_CTRL_0_SHADOW_STATE)	32	R	0000_0000h	16.9.33/ 637
1E8	Shadow Register of DPLL_CTRL_1 controlled by a valid TRIGGER Slope (DPLL_CTRL_1_SHADOW_TRIGGER)	32	R	0000_0000h	16.9.34/ 638
1EC	Shadow Register of DPLL_CTRL_1 controlled by a valid STATE Slope (DPLL_CTRL_1_SHADOW_STATE)	32	R	0000_0000h	16.9.35/ 640

Table continues on the next page...

DPLL memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1FC	Register to control the RAM Initialization (DPLL_RAM_INI)	32	R/W	0000_0000h	16.9.36/ 642
E00	Calculated Time Value to start Action i (DPLL_TSAC_0)	32	R/W	007F_FFFFh	16.9.37/ 643
E04	Calculated Time Value to start Action i (DPLL_TSAC_1)	32	R/W	007F_FFFFh	16.9.37/ 643
E08	Calculated Time Value to start Action i (DPLL_TSAC_2)	32	R/W	007F_FFFFh	16.9.37/ 643
E0C	Calculated Time Value to start Action i (DPLL_TSAC_3)	32	R/W	007F_FFFFh	16.9.37/ 643
E10	Calculated Time Value to start Action i (DPLL_TSAC_4)	32	R/W	007F_FFFFh	16.9.37/ 643
E14	Calculated Time Value to start Action i (DPLL_TSAC_5)	32	R/W	007F_FFFFh	16.9.37/ 643
E18	Calculated Time Value to start Action i (DPLL_TSAC_6)	32	R/W	007F_FFFFh	16.9.37/ 643
E1C	Calculated Time Value to start Action i (DPLL_TSAC_7)	32	R/W	007F_FFFFh	16.9.37/ 643
E20	Calculated Time Value to start Action i (DPLL_TSAC_8)	32	R/W	007F_FFFFh	16.9.37/ 643
E24	Calculated Time Value to start Action i (DPLL_TSAC_9)	32	R/W	007F_FFFFh	16.9.37/ 643
E28	Calculated Time Value to start Action i (DPLL_TSAC_10)	32	R/W	007F_FFFFh	16.9.37/ 643
E2C	Calculated Time Value to start Action i (DPLL_TSAC_11)	32	R/W	007F_FFFFh	16.9.37/ 643
E30	Calculated Time Value to start Action i (DPLL_TSAC_12)	32	R/W	007F_FFFFh	16.9.37/ 643
E34	Calculated Time Value to start Action i (DPLL_TSAC_13)	32	R/W	007F_FFFFh	16.9.37/ 643
E38	Calculated Time Value to start Action i (DPLL_TSAC_14)	32	R/W	007F_FFFFh	16.9.37/ 643
E3C	Calculated Time Value to start Action i (DPLL_TSAC_15)	32	R/W	007F_FFFFh	16.9.37/ 643
E40	Calculated Time Value to start Action i (DPLL_TSAC_16)	32	R/W	007F_FFFFh	16.9.37/ 643
E44	Calculated Time Value to start Action i (DPLL_TSAC_17)	32	R/W	007F_FFFFh	16.9.37/ 643
E48	Calculated Time Value to start Action i (DPLL_TSAC_18)	32	R/W	007F_FFFFh	16.9.37/ 643
E4C	Calculated Time Value to start Action i (DPLL_TSAC_19)	32	R/W	007F_FFFFh	16.9.37/ 643
E50	Calculated Time Value to start Action i (DPLL_TSAC_20)	32	R/W	007F_FFFFh	16.9.37/ 643

Table continues on the next page...

DPLL memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
E54	Calculated Time Value to start Action i (DPLL_TSAC_21)	32	R/W	007F_FFFFh	16.9.37/ 643
E58	Calculated Time Value to start Action i (DPLL_TSAC_22)	32	R/W	007F_FFFFh	16.9.37/ 643
E5C	Calculated Time Value to start Action i (DPLL_TSAC_23)	32	R/W	007F_FFFFh	16.9.37/ 643
E60	Calculated Time Value to start Action i (DPLL_TSAC_24)	32	R/W	007F_FFFFh	16.9.37/ 643
E64	Calculated Time Value to start Action i (DPLL_TSAC_25)	32	R/W	007F_FFFFh	16.9.37/ 643
E68	Calculated Time Value to start Action i (DPLL_TSAC_26)	32	R/W	007F_FFFFh	16.9.37/ 643
E6C	Calculated Time Value to start Action i (DPLL_TSAC_27)	32	R/W	007F_FFFFh	16.9.37/ 643
E70	Calculated Time Value to start Action i (DPLL_TSAC_28)	32	R/W	007F_FFFFh	16.9.37/ 643
E74	Calculated Time Value to start Action i (DPLL_TSAC_29)	32	R/W	007F_FFFFh	16.9.37/ 643
E78	Calculated Time Value to start Action i (DPLL_TSAC_30)	32	R/W	007F_FFFFh	16.9.37/ 643
E7C	Calculated Time Value to start Action i (DPLL_TSAC_31)	32	R/W	007F_FFFFh	16.9.37/ 643
E80	DPLL Calculated Position Value ACTION i (DPLL_PSAC_0)	32	R/W	007F_FFFFh	16.9.38/ 644
E84	DPLL Calculated Position Value ACTION i (DPLL_PSAC_1)	32	R/W	007F_FFFFh	16.9.38/ 644
E88	DPLL Calculated Position Value ACTION i (DPLL_PSAC_2)	32	R/W	007F_FFFFh	16.9.38/ 644
E8C	DPLL Calculated Position Value ACTION i (DPLL_PSAC_3)	32	R/W	007F_FFFFh	16.9.38/ 644
E90	DPLL Calculated Position Value ACTION i (DPLL_PSAC_4)	32	R/W	007F_FFFFh	16.9.38/ 644
E94	DPLL Calculated Position Value ACTION i (DPLL_PSAC_5)	32	R/W	007F_FFFFh	16.9.38/ 644
E98	DPLL Calculated Position Value ACTION i (DPLL_PSAC_6)	32	R/W	007F_FFFFh	16.9.38/ 644
E9C	DPLL Calculated Position Value ACTION i (DPLL_PSAC_7)	32	R/W	007F_FFFFh	16.9.38/ 644
EA0	DPLL Calculated Position Value ACTION i (DPLL_PSAC_8)	32	R/W	007F_FFFFh	16.9.38/ 644
EA4	DPLL Calculated Position Value ACTION i (DPLL_PSAC_9)	32	R/W	007F_FFFFh	16.9.38/ 644
EA8	DPLL Calculated Position Value ACTION i (DPLL_PSAC_10)	32	R/W	007F_FFFFh	16.9.38/ 644

Table continues on the next page...

DPLL memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
EAC	DPLL Calculated Position Value ACTION i (DPLL_PSAC_11)	32	R/W	007F_FFFFh	16.9.38/ 644
EB0	DPLL Calculated Position Value ACTION i (DPLL_PSAC_12)	32	R/W	007F_FFFFh	16.9.38/ 644
EB4	DPLL Calculated Position Value ACTION i (DPLL_PSAC_13)	32	R/W	007F_FFFFh	16.9.38/ 644
EB8	DPLL Calculated Position Value ACTION i (DPLL_PSAC_14)	32	R/W	007F_FFFFh	16.9.38/ 644
EBC	DPLL Calculated Position Value ACTION i (DPLL_PSAC_15)	32	R/W	007F_FFFFh	16.9.38/ 644
EC0	DPLL Calculated Position Value ACTION i (DPLL_PSAC_16)	32	R/W	007F_FFFFh	16.9.38/ 644
EC4	DPLL Calculated Position Value ACTION i (DPLL_PSAC_17)	32	R/W	007F_FFFFh	16.9.38/ 644
EC8	DPLL Calculated Position Value ACTION i (DPLL_PSAC_18)	32	R/W	007F_FFFFh	16.9.38/ 644
ECC	DPLL Calculated Position Value ACTION i (DPLL_PSAC_19)	32	R/W	007F_FFFFh	16.9.38/ 644
ED0	DPLL Calculated Position Value ACTION i (DPLL_PSAC_20)	32	R/W	007F_FFFFh	16.9.38/ 644
ED4	DPLL Calculated Position Value ACTION i (DPLL_PSAC_21)	32	R/W	007F_FFFFh	16.9.38/ 644
ED8	DPLL Calculated Position Value ACTION i (DPLL_PSAC_22)	32	R/W	007F_FFFFh	16.9.38/ 644
EDC	DPLL Calculated Position Value ACTION i (DPLL_PSAC_23)	32	R/W	007F_FFFFh	16.9.38/ 644
EE0	DPLL Calculated Position Value ACTION i (DPLL_PSAC_24)	32	R/W	007F_FFFFh	16.9.38/ 644
EE4	DPLL Calculated Position Value ACTION i (DPLL_PSAC_25)	32	R/W	007F_FFFFh	16.9.38/ 644
EE8	DPLL Calculated Position Value ACTION i (DPLL_PSAC_26)	32	R/W	007F_FFFFh	16.9.38/ 644
EEC	DPLL Calculated Position Value ACTION i (DPLL_PSAC_27)	32	R/W	007F_FFFFh	16.9.38/ 644
EF0	DPLL Calculated Position Value ACTION i (DPLL_PSAC_28)	32	R/W	007F_FFFFh	16.9.38/ 644
EF4	DPLL Calculated Position Value ACTION i (DPLL_PSAC_29)	32	R/W	007F_FFFFh	16.9.38/ 644
EF8	DPLL Calculated Position Value ACTION i (DPLL_PSAC_30)	32	R/W	007F_FFFFh	16.9.38/ 644
EFC	DPLL Calculated Position Value ACTION i (DPLL_PSAC_31)	32	R/W	007F_FFFFh	16.9.38/ 644
F00	Control Bits for up to 32 Actions (DPLL_ACB_0)	32	R/W	0000_0000h	16.9.39/ 644

Table continues on the next page...

DPLL memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F04	Control Bits for up to 32 Actions (DPLL_ACB_1)	32	R/W	0000_0000h	16.9.39/ 644
F08	Control Bits for up to 32 Actions (DPLL_ACB_2)	32	R/W	0000_0000h	16.9.39/ 644
F0C	Control Bits for up to 32 Actions (DPLL_ACB_3)	32	R/W	0000_0000h	16.9.39/ 644
F10	Control Bits for up to 32 Actions (DPLL_ACB_4)	32	R/W	0000_0000h	16.9.39/ 644
F14	Control Bits for up to 32 Actions (DPLL_ACB_5)	32	R/W	0000_0000h	16.9.39/ 644
F18	Control Bits for up to 32 Actions (DPLL_ACB_6)	32	R/W	0000_0000h	16.9.39/ 644
F1C	Control Bits for up to 32 Actions (DPLL_ACB_7)	32	R/W	0000_0000h	16.9.39/ 644

16.9.1 Control Register 0 (DPLL_CTRL_0)

Footnotes:

1 Stored in an independent shadow register for a valid TRIGGER event and for DEN = 1.

2 Stored in an independent shadow register for a valid STATE event and for DEN = 1.

3 The time between two valid STATE or TRIGGER events must be always greater then 23.4 µs; in addition the TS_CLK and the resolution must be chosen such that for each nominal increment the time stamps at the beginning and the end of the increment differ at least by the value of 257.

4 For IFP=1 the time between two valid TRIGGER or STATE events must be always greater then 2.34 ms and the value x of MLT, MLS1 or MLS2 must be chosen such that the number of time stamp pulses between two SUB_INC events must be less then 65536. This is fulfilled when x is greater then 256.

Address: 2_8000h base + 0h offset = 2_8000h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	RMO	TEN	SEN	IDT	IDS	AMT	AMS	TNU								
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	SNU					IFP	MLT									
Reset	1	0	1	1	1	0	1	0	0	1	0	1	0	1	1	1

DPLL_CTRL_0 field descriptions

Field	Description
0 RMO	Reference mode. Selection of the relevant the input signal for generation of SUB_INC1. See footnotes 1 and 2. **NOTE:** For SMC=0: TRIGGER and STATE are prepared to calculate SUB_INC1. The RMO bit gives a decision only, which of them is used. For changing from normal mode to emergency mode at the following TRIGGER slope (according to the RMO value in the shadow register)1) the PSSC value is calculated by PSSC = PSSM + correction_value (forward direction) or PSSC = PSSM minus correction value (backward direction) with the correction_value = inc_cnt1 – nmb_t. For changing from emergency mode to normal mode at the following STATE slope (according to the RMO value in the shadow register 2) the PSTC value is calculated by PSTC = PSTM + correction_value (forward direction) or PSTC = PSTM minus correction value (backward direction) with the correction_value = inc_cnt1 – nmb_s. In the case of no further TRIGGER or STATE events appearing, the CPU has to perform the corrections above accordingly. 0 Normal mode: The signal TRIGGER is used to generate the SUB_INC1 signals. 1 Emergency mode for SMC=0; signal STATE is used to generate the SUB_INC1 signals. Double synchronous mode for SMC=1: Signal TRIGGER is used to generate the SUB_INC1 signals and STATE is used to generate the SUB_INC2 signals.
1 TEN	TRIGGER enable. 0 TRIGGER signal is not enabled (no signal considered). 1 TRIGGER signal is enabled.
2 SEN	STATE enable. 0 STATE signal is not enabled (no signal considered). 1 STATE signal is enabled.
3 IDT	Input delay TRIGGER. Use of input delay information transmitted in FT part of the TRIGGER signal. See footnote 1. 0 Delay information is not used. 1 Up to 24 bits of the FT part contain the delay value of the input signal, concerning the corresponding edge.
4 IDS	Input delay STATE. Use of input delay information transmitted in FT part of the STATE signal. See footnote 2. 0 Delay information is not used. 1 Up to 24 bits of the FT part contain the delay value of the input signal, concerning the corresponding edge.
5 AMT	Adapt mode TRIGGER Use of adaptation information of TRIGGER. See footnote 1.

Table continues on the next page...

DPLL_CTRL_0 field descriptions (continued)

Field	Description
	0 No adaptation information for TRIGGER is used. 1 Immediate adapting mode; the values ADT_T[i] are considered to calculate the SUB_INC1 pulses in normal mode and for SMC=1.
6 AMS	Addapt mode STATE. Use of adaptation information of STATE. See footnote 2 0 No adaptation information is used for STATE. 1 Immediate adapting mode; the values ADT_S[i] are considered to calculate SUB_INC1 pulses in emergency mode (SMC=0) or SUB_INC2 pulses for SMC=1.
7–15 TNU	TRIGGER number. TNU+1 is number of TRIGGER events in HALF_SCALE (1…512). See footnote 3. **NOTE:** The number of nominal TRIGGER events is the binary value plus 1. This value can only be written when the DPLL is disabled.
16–20 SNU	STATE number. SNU+1 is number of STATE events in HALF_SCALE (1…32). See footnote 3. **NOTE:** The number of nominal STATE events is the binary value plus 1. This value can only be written when the DPLL is disabled.
21 IFP	Input filter position. Value contains position or time related information. See footnotes 1,2 and 4. 0 TRIGGER_FT and STATE_FT mean time related values, that means the number of time stamp clocks. 1 TRIGGER_FT and STATE_FT mean position related values, that means the number of SUB_INC1 (or SUB_INC2 in the case SMC=1) pulses respectively.
22–31 MLT	Multiplier for TRIGGER. MLT+1 is number of SUB_INC1 pulses between two TRIGGER events in normal mode (1…1024). For emergency mode the number of SUB_INC1 pulses between two STATE events is calculated by the CPU using the formula MLS1=(MLT+1)* (TNU+1) / (SNU+1) in order to get the same number of SUB_INC1 pulses for FULL_SCALE. This value is stored in RAM at 0x05C0. Change of MLT by the CPU must result in the corresponding change of MLS1 by the CPU for SMC=0. See footnote 1. **NOTE:** The number of MLT events is the binary value plus 1. The value MLT+1 is replaced by MLS1 in the case of SMC=1 (see DPLL_CTRL_1 register) for all relevant calculations.

16.9.2 Control Register 1 (DPLL_CTRL_1)

Footnotes:

1 Stored in an independent shadow register for a valid TRIGGER event and for DEN = 1.

2 Stored in an independent shadow register for a valid STATE event and for DEN = 1.

3 Bit is cleared, when transmitted to shadow register.

Address: 2_8000h base + 4h offset = 2_8004h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	TSL		SSL		SMC	TS0_ HRT	TS0_ HRS	SYSF	SWR	LCD	SYN_NT					
Reset	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	SYN_NS					PCM2	DLM2	SGE2	PCM1	DLM1	SGE1	PIT	COA	IDDS	DEN	DMO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_CTRL_1 field descriptions

Field	Description
0–1 TSL	TRIGGER slope select. Definition of active slope for signal TRIGGER each active slope is an event defined by TNU. Set by DEN=0 only. **NOTE:** These bits can only be written when the DPLL is disabled. 00 No slope of TRIGGER will be used; this value makes only sense in emergency mode. 01 Low high slope will be used as active slope, only inputs with a signal value of “1” will be considered. 10 High low slope will be used as active slope, only inputs with a signal value of “0” will be considered. 11 Both slopes will be used as active slopes.
2–3 SSL	STATE slope select. Definition of active slope for signal STATE each active slope is an event defined by SNU. Set by DEN=0 only. **NOTE:** These bits can only be written when the DPLL is disabled. 00 No slope of STATE will be used; this value makes only sense in normal mode. 01 Low high slope will be used as active slope, only inputs with a signal value of “1” will be considered. 10 High low slope will be used as active slope, only inputs with a signal value of “0” will be considered. 11 Both slopes will be used as active slopes.
4 SMC	Synchronous Motor Control **NOTE:** This bit can only be written when the DPLL is disabled. 0 The TRIGGER input is not used for SMC. 1 The TRIGGER input time stamps have a 8 times higher resolution as the TBU_TS0 input.

Table continues on the next page...

DPLL_CTRL_1 field descriptions (continued)

Field	Description
5 TS0_HRT	Time stamp high resolution TRIGGER **NOTE:** This bit can only be written when the DPLL is disabled. 0 The resolution of the used DPLL input TBU_TS0 bits is equal to the TRIGGER input time stamp resolution. 1 The TRIGGER input time stamps have a 8 times higher resolution as the TBU_TS0 input.
6 TS0_HRS	Time stamp high resolution STATE. **NOTE:** This bit can only be written when the DPLL is disabled. 0 The resolution of the used DPLL input TBU_TS0 bits is equal to the STATE input time stamp resolution. 1 The STATE input time stamps have a 8 times higher resolution as the TBU_TS0 DPLL input.
7 SYSF	SYN_NS for FULL_SCALE. The value SYN_NS is sum of all systematic missing STATE events in HALF_SCALE (for SYSF=0) or FULL SCALE (for SYSF=1). **NOTE:** This bit can only be written when the DPLL is disabled. 0 The SYN_NS value is valid for HALF_SCALE. 1 The SYN_NS value is valid for FULL_SCALE.
8 SWR	Software reset Resets all register and internal states of the DPLL. **NOTE:** Setting the SWR bit results in a software reset only when the DPLL is not enabled (DEN=0). **NOTE:** This bit is cleared automatically after a write to 1. 0 No software reset enabled. 1 Software reset enabled.
9 LCD	Locking Condition Definition. **NOTE:** This bit can only be written when the DPLL is disabled 0 Locking condition definition is one times missing TRIGGERs as expected by the profile in HALF_SCALE (one gap) 1 Locking condition definition is n-1 times missing TRIGGERs as expected by the profile in HALF_SCALE (one additional tooth)
10–15 SYN_NT	Synchronization number of TRIGGER. Summarized number of virtual increments in HALF_SCALE. sum of all systematic missing TRIGGER events in HALF_SCALE; the SYN_NT missing TRIGGER can be divided up to an arbitrary number of blocks. The pattern of events and missing events in FULL_SCALE is shown in RAM region 2c as value NT in addition to the adapt values. The number of stored increments in FULL_SCALE must be equal to 2*(TNU–SYN_NT). This pattern is written by the CPU beginning from a fixed reference point (maybe beginning of the FULL_SCALE region). The relation to the actual increment is established by setting of the profile RAM pointer APT_2c in an appropriate relation to the RAM pointer APT of the actual increment by the CPU.

Table continues on the next page...

DPLL_CTRL_1 field descriptions (continued)

Field	Description
	NOTE: This value can be written only when the DPLL is disabled.
16–20 SYN_NS	Synchronization number of STATE. Summarized number of virtual increments in HALF_SCALE. Sum of all systematic missing STATE events in HALF_SCALE (for SYSF=0) or FULL SCALE (for SYSF=1) ; the SYN_NS missing STATES can be divided up to an arbitrary number of blocks. The pattern of events and missing events in FULL_SCALE is shown in RAM region 1c3 as value NS in addition to the adapt values. The number of stored increments in FULL_SCALE must be equal to 2*(SNU+1–SYN_NS) for SYSF=0 or 2*(SNU+1)–SYN_NS for SYSF=1 . This pattern is written by the CPU beginning from a fixed reference point (maybe beginning of the FULL_SCALE region). The relation to the actual increment is established by setting of the profile RAM pointer APS_1c3 in an appropriate relation to the RAM pointer APS of the actual increment by the CPU. **NOTE:** This value can only be written when the DPLL is disabled.
21 PCM2	Pulse Correction Mode. For SUB_INC2 generation. See footnote 2 and 3. 0 The DPLL does not use the correction value stored in MPVAL2. 1 The DPLL uses the correction value stored in MPVAL2.
22 DLM2	Direct Load Mode. For SUB_INC2 generation. See footnote 2. 0 The DPLL uses the calculated ADD_IN_CAL value for the SUB_INC2 generation. 1 The ADD_IN_LD value is used for the SUB_INC2 generation and is provided by the CPU; the value remains valid until the CPU writes a new one; the calculated ADD_IN values are stored as ADD_IN_CAL in the RAM at different locations for normal and emergency mode.
23 SGE2	SUB_INC2 generator enable. See footnote 2 0 The SUB_INC2 generator is not enabled. 1 The SUB_INC2 generator is enabled.
24 PCM1	Pulse Correction Mode. For SUB_INC1 generation. See footnotes 1, 2 and 3. 0 The DPLL does not use the correction value stored in MPVAL1. 1 The DPLL uses the correction value stored in MPVAL1 in normal and emergency mode.
25 DLM1	Direct Load Mode. For SUB_INC1 generation. See footnotes 1 and 2. 0 The DPLL uses the calculated ADD_IN_CAL value for the SUB_INC1 generation. 1 The ADD_IN_LD value is used for the SUB_INC1 generation and is provided by the CPU; the value remains valid until the CPU writes a new one; the calculated ADD_IN values are stored as ADD_IN_CAL in the RAM at different locations for normal and emergency mode.

Table continues on the next page...

DPLL_CTRL_1 field descriptions (continued)

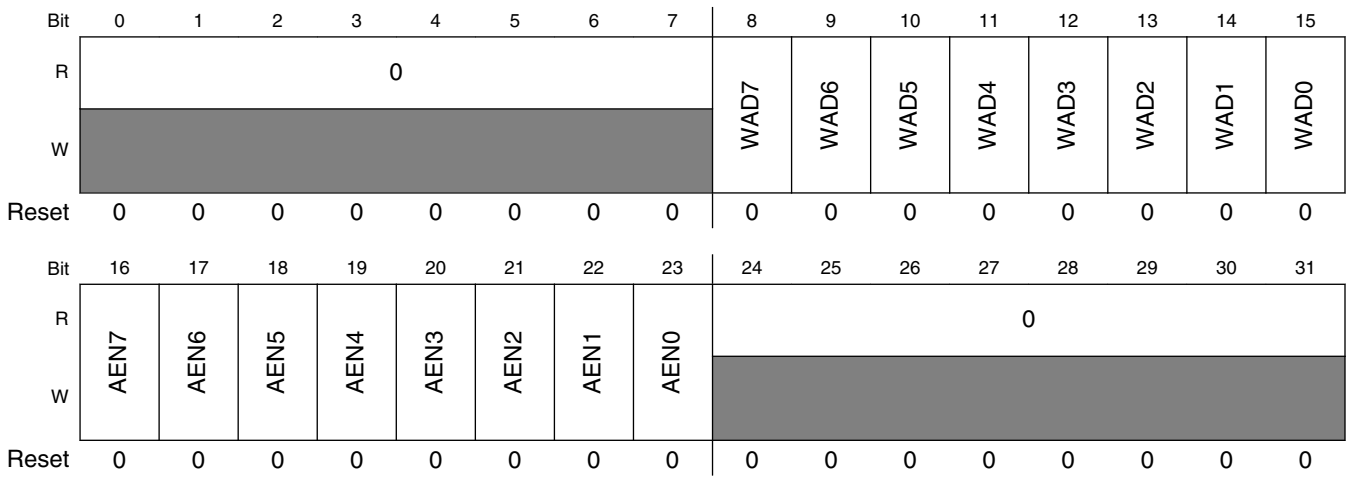
Field	Description
26 SGE1	SUB_INC1 generator enable. See footnotes 1 and 2. 0 The SUB_INC1 generator is not enabled. 1 The SUB_INC1 generator is enabled.
27 PIT	Plausibility. Value PVT to next valid TRIGGER is time related. See footnote 1. 0 The plausibility value is position related (PVT contains the number of SUB_INC1 pulses). 1 The plausibility value is time related (the PVT value is to be multiplied with the duration of the last DT_T_ACT increment and divided by 1024).
28 COA	Correction strategy in automatic end mode (DMO=0). See footnotes 1 and 2. 0 The pulse frequency of the CMU_CLK0 will be used to make up for missing pulses from last increment; the output of the calculated new pulses will start after resetting the FFs in the pulse generation unit. The frequency of CMU_CLK0 should not exceed half the frequency of the system clock, TSCLK, (see 16.8.3.6). 1 Missing pulses of the last increment are distributed evenly to the next increment, calculations are done when the next valid input event appears. The number of missing sub-pulses will be determined by the pulse counter difference between the last two valid TRIGGER/STATE events respectively; the FFs in the pulse generation unit are not reset before sending new pulses.
29 IDDS	Input Direction Detection Strategy in the case of SMC=0. **NOTE:** This bit can only be written when the DPLL is disabled . Independent of the value of IDDS is the direction information for TRIGGER in the case SMC=0, which is always considered at the moment when the invalid slope appears. 0 The input direction is detected comparing the THMI value with the duration between the valid and invalid slope of TRIGGER. 1 In the case SMC = 0, the input direction is detected using the TDIR input signal.
30 DEN	DPLL enable. 0 Bits 31 through 0 of the DPLL_STATUS register are cleared, when the DPLL is disabled. Some bits of the control registers can be set only when DEN=0. The protected bits in the DPLL_CTRL_1 register cannot be written when simultaneously DEN is set to 1. 1 The DPLL is enabled.
31 DMO	DPLL mode select. See footnotes 1 and 2. 0 Automatic end mode: If the number of pulses for a increment is reached, no further pulse is generated until the next valid TRIGGER/STATE is received; in the case of getting a new valid TRIGGER/STATE before the defined number of pulses is reached, the pulse frequency is changed according to the conditions described below (COA) 1 Continuous mode: In this mode, a difference between the predefined number of pulses and the actual number of generated pulses can influence the pulse frequency by writing a corresponding pulse number into CNT_NUM1 or CNT_NUM2 respectively in RAM region 1b.

16.9.3 Action Enable Register (DPLL_CTRL_2)

NOTE

WAD[7:0] are cleared automatically after writing a 1.

Address: 2_8000h base + 8h offset = 2_8008h



DPLL_CTRL_2 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 WAD7	Write control bit of Action_7. **NOTE:** For writing WAD7 =1, only the corresponding the AENx bits are written. The AENi bits remain unchanged when the corresponding WADi=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
9 WAD6	Write control bit of Action_6. **NOTE:** For writing WAD6 =1, only the corresponding the AENx bits are written. The AENi bits remain unchanged when the corresponding WADi=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
10 WAD5	Write control bit of Action_5. **NOTE:** For writing WAD5 =1, only the corresponding the AENx bits are written. The AENi bits remain unchanged when the corresponding WADi=0.

Table continues on the next page...

DPLL_CTRL_2 field descriptions (continued)

Field	Description
	0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
11 WAD4	Write control bit of Action_4. **NOTE:** For writing WAD4 =1, only the corresponding the AENx bits are written. The AENi bits remain unchanged when the corresponding WADi=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
12 WAD3	Write control bit of Action_3. **NOTE:** For writing WAD3 =1, only the corresponding the AENx bits are written. The AENi bits remain unchanged when the corresponding WADi=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
13 WAD2	Write control bit of Action_2. **NOTE:** For writing WAD2 =1, only the corresponding the AENx bits are written. The AENi bits remain unchanged when the corresponding WADi=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
14 WAD1	Write control bit of Action_1. **NOTE:** For writing WAD1 =1, only the corresponding the AENx bits are written. The AENi bits remain unchanged when the corresponding WADi=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
15 WAD0	Write control bit of Action_0. **NOTE:** For writing WAD0 =1, only the corresponding the AENx bits are written. The AENi bits remain unchanged when the corresponding WADi=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
16 AEN7	ACTION_7 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.

Table continues on the next page...

DPLL_CTRL_2 field descriptions (continued)

Field	Description
17 AEN6	ACTION_6 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
18 AEN5	ACTION_5 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
19 AEN4	ACTION_4 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
20 AEN3	ACTION_3 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
21 AEN2	ACTION_2 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
22 AEN1	ACTION_1 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1).

Table continues on the next page...

DPLL_CTRL_2 field descriptions (continued)

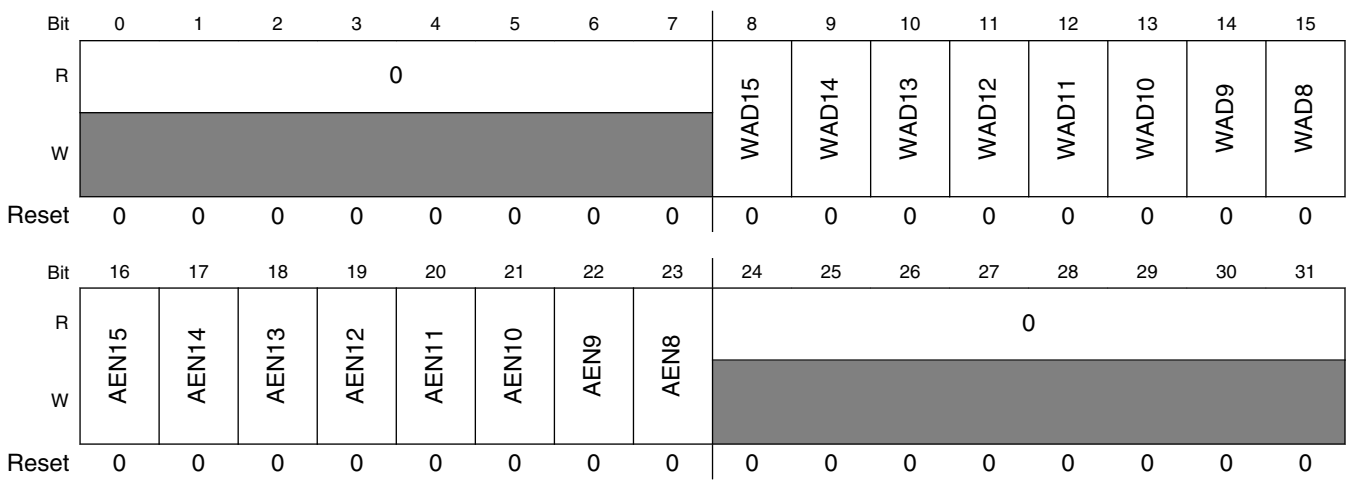
Field	Description
	0 The corresponding action is not enabled. 1 The corresponding action is enabled.
23 AEN0	ACTION_0 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
24–31 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.

16.9.4 Action Enable Register (DPLL_CTRL_3)

NOTE

WAD[15:8] are cleared automatically after writing a 1.

Address: 2_8000h base + Ch offset = 2_800Ch



DPLL_CTRL_3 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 WAD15	Write control bit of Action_15.

Table continues on the next page...

DPLL_CTRL_3 field descriptions (continued)

Field	Description
	NOTE: For writing WAD15 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
9 WAD14	Write control bit of Action_14. **NOTE:** For writing WAD14 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
10 WAD13	Write control bit of Action_13. **NOTE:** For writing WAD13 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
11 WAD12	Write control bit of Action_12. **NOTE:** For writing WAD12 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
12 WAD11	Write control bit of Action_11. **NOTE:** For writing WAD11 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
13 WAD10	Write control bit of Action_10. **NOTE:** For writing WAD10 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
14 WAD9	Write control bit of Action_9. **NOTE:** For writing WAD9 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0.

Table continues on the next page...

DPLL_CTRL_3 field descriptions (continued)

Field	Description
	0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
15 WAD8	Write control bit of Action_8. **NOTE:** For writing WAD8 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
16 AEN15	ACTION_15 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
17 AEN14	ACTION_14 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
18 AEN13	ACTION_13 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. The corresponding action is enabled.
19 AEN12	ACTION_12 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
20 AEN11	ACTION_11 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1).

Table continues on the next page...

DPLL_CTRL_3 field descriptions (continued)

Field	Description
	0 The corresponding action is not enabled. 1 The corresponding action is enabled.
21 AEN10	ACTION_10 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
22 AEN9	ACTION_9 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
23 AEN8	ACTION_8 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
24–31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

16.9.5 Action Enable Register (DPLL_CTRL_4)

Footnote:

1 This bit can only be written when the correspondent WADi bit is set. It can also be set by CPU for debug purposes when DPLL is disabled.

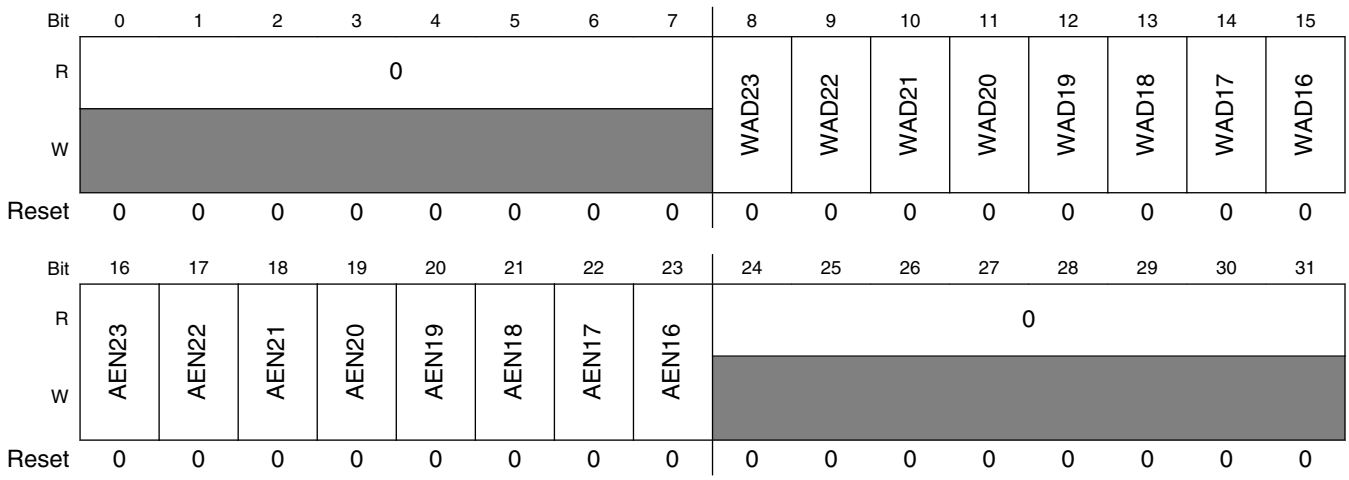
NOTE

For writing WADx =1, only the corresponding AENx bits are written. The AENx bit remains unchanged when the corresponding WADx=0.

NOTE

WAD[23:16] are cleared automatically after writing a 1.

Address: 2_8000h base + 10h offset = 2_8010h



DPLL_CTRL_4 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 WAD23	Write control bit of Action_23. **NOTE:** For writing WAD23 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
9 WAD22	Write control bit of Action_22. **NOTE:** For writing WAD22 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
10 WAD21	Write control bit of Action_21. **NOTE:** For writing WAD21 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
11 WAD20	Write control bit of Action_20. **NOTE:** For writing WAD20 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.

Table continues on the next page...

DPLL_CTRL_4 field descriptions (continued)

Field	Description
12 WAD19	Write control bit of Action_19. **NOTE:** For writing WAD19 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
13 WAD18	Write control bit of Action_18. **NOTE:** For writing WAD18 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
14 WAD17	Write control bit of Action_17. **NOTE:** For writing WAD17, =1 only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
15 WAD16	Write control bit of Action_16. **NOTE:** For writing WAD16 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
16 AEN23	ACTION_23 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
17 AEN22	ACTION_22 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
18 AEN21	ACTION_21 enable.

Table continues on the next page...

DPLL_CTRL_4 field descriptions (continued)

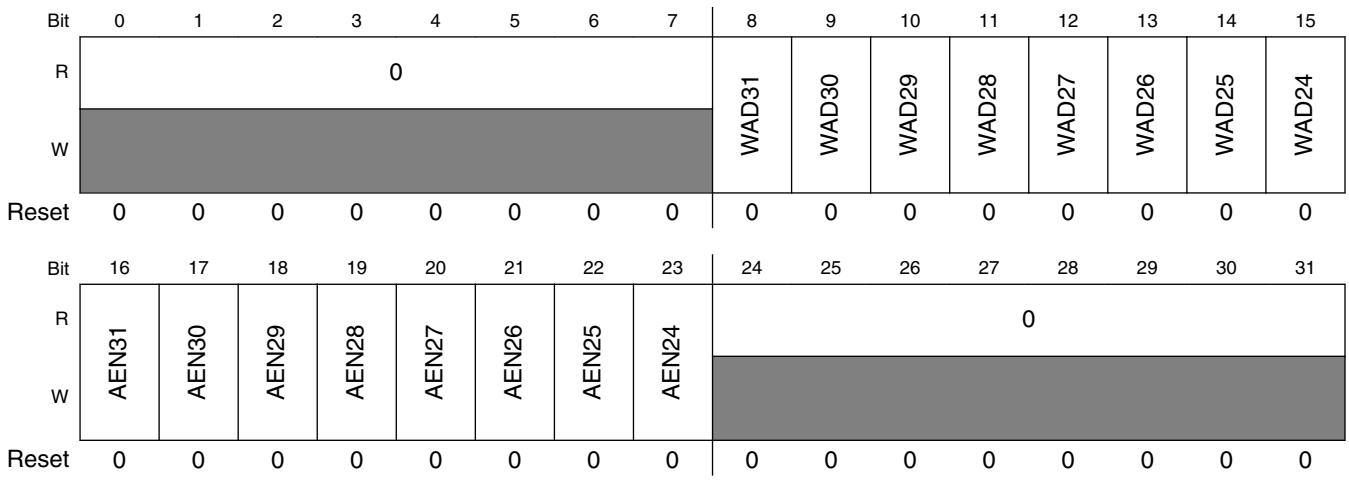
Field	Description
	NOTE: This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
19 AEN20	ACTION_20 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
20 AEN19	ACTION_19 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
21 AEN18	ACTION_18 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
22 AEN17	ACTION_17 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
23 AEN16	ACTION_16 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
24–31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

16.9.6 Action Enable Register (DPLL_CTRL_5)

NOTE

WAD[24:31] are cleared automatically after a writing a 1.

Address: 2_8000h base + 14h offset = 2_8014h



DPLL_CTRL_5 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 WAD31	Write control bit of Action_31. **NOTE:** For writing WAD31 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
9 WAD30	Write control bit of Action_30. **NOTE:** For writing WAD30 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
10 WAD29	Write control bit of Action_29. **NOTE:** For writing WAD29 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0.

Table continues on the next page...

DPLL_CTRL_5 field descriptions (continued)

Field	Description
	0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
11 WAD28	Write control bit of Action_28. **NOTE:** For writing WAD28 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
12 WAD27	Write control bit of Action_27. **NOTE:** For writing WAD27 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
13 WAD26	Write control bit of Action_26. **NOTE:** For writing WAD26 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
14 WAD25	Write control bit of Action_25. **NOTE:** For writing WAD25 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
15 WAD24	Write control bit of Action_24. **NOTE:** For writing WAD23 =1, only the corresponding the AENx bits are written. The AENx bits remain unchanged when the corresponding WADx=0. 0 The corresponding AENi bit is not writeable. 1 The corresponding AENi bit is writeable.
16 AEN31	ACTION_31 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.

Table continues on the next page...

DPLL_CTRL_5 field descriptions (continued)

Field	Description
17 AEN30	ACTION_30 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
18 AEN29	ACTION_29 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
19 AEN28	ACTION_28 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
20 AEN27	ACTION_27 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
21 AEN26	ACTION_26 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
22 AEN25	ACTION_25 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1).

Table continues on the next page...

DPLL_CTRL_5 field descriptions (continued)

Field	Description
	NOTE: This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
23 AEN24	ACTION_24 enable. **NOTE:** This bit can only be written when the correspondent WADi bit is set. It can also be set for debug purposes by the CPU when the DPLL is disabled .The enable bit becomes active only when the DPLL is in operation (DEN=1). 0 The corresponding action is not enabled. 1 The corresponding action is enabled.
24–31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

16.9.7 Action Status Register including Shadow Register (DPLL_ACT_STA)

Address: 2_8000h base + 18h offset = 2_8018h

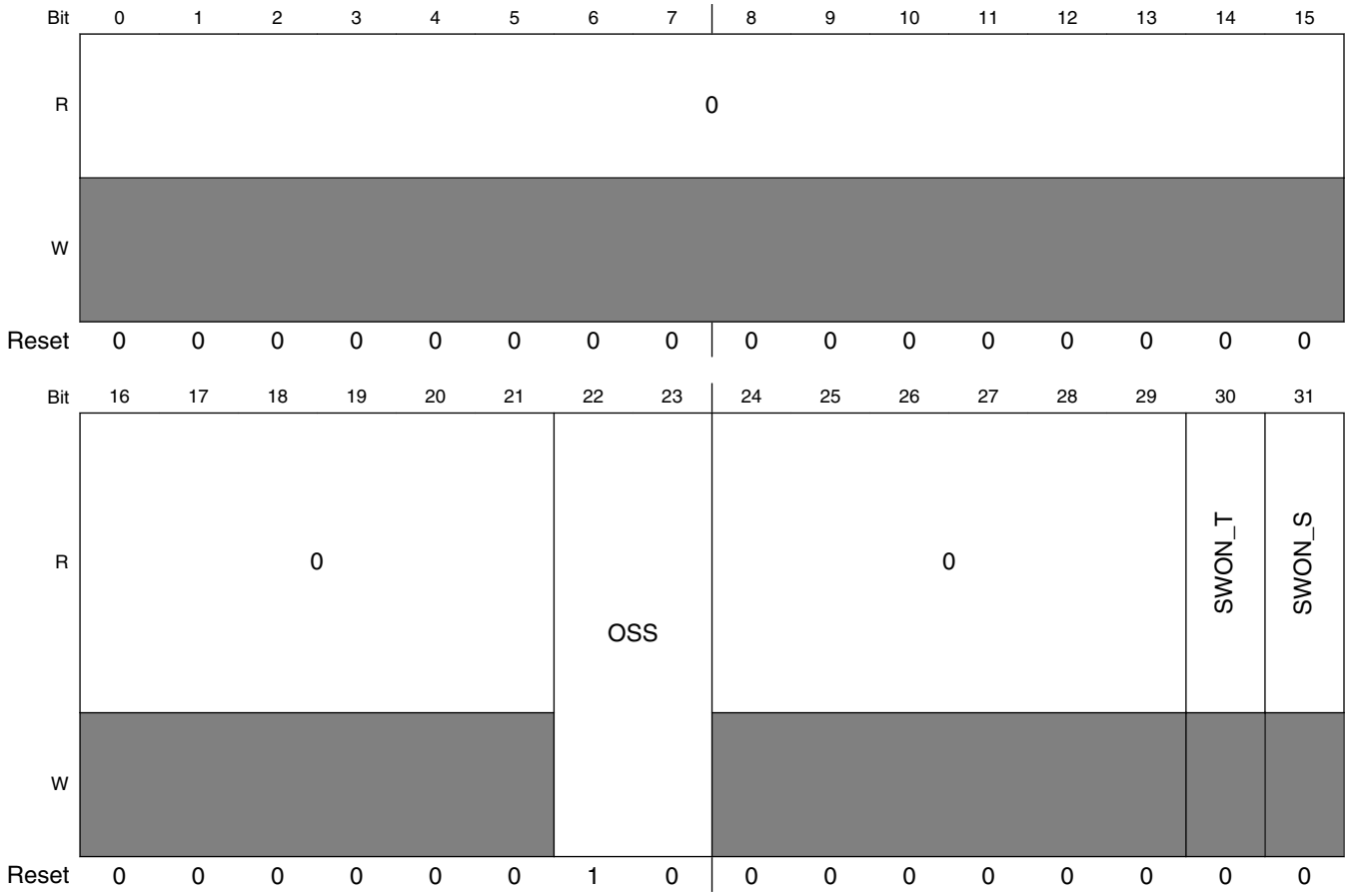
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	ACT_N																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_ACT_STA field descriptions

Field	Description
0–31 ACT_N	New output data values concerning to action i provided. ACT_N[i] is: • Set (for AENi=1 and a new valid PMTR), that means when new action data are to be calculated for the correspondent action. After each calculation of the new actions values the ACT_N[i] bit updates the corresponding bit in the connected shadow register. The status of the ACT_N[i] bits in the shadow register is reflected by the corresponding DPLL output signal ACT_V (valid bit). • Reset together with the corresponding shadow register bit for AENi=0. • Reset without the corresponding shadow register bit when the calculated event is in the past (the shadow register bit is set, when it was not set before in that case) • The corresponding shadow register bit is reset, when new PMTR data are written or when the provided action data are read (blocking read). • Writeable for debugging purposes together with the corresponding shadow register when DEN=0. 0 No new output data available after a recent PMT request or actual event value is in the past or invalid. 1 New PMTR data received or calculation is to be precised by taking into account new TRIGGER or STATE values.

16.9.8 Offset and Switch Old/New Address Register (DPLL_OSW)

Address: 2_8000h base + 1Ch offset = 2_801Ch



DPLL_OSW field descriptions

Field	Description
0–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22–23 OSS	Offset size of RAM region 2. **NOTE:** At least 128 and at most 1024 values can be stored in each of the RAM 2 regions a to d accordingly. The value can be set only when DEN=0. The change of the OSS value results in an automatic change of the offset values in the DPLL_AOSV_2 register. 00 Offset size 128 of RAM region 2. 01 Offset size 256 of RAM region 2. 10 Offset size 512 of RAM region 2. 11 Offset size 1024 of RAM region 2.
24–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DPLL_OSW field descriptions (continued)

Field	Description
30 SWON_T	Switch of new TRIGGER. Switch bit for LSB address of TRIGGER. This bit is changed for each write access to TS_T or TS_T_OLD. Using this unchanged address bit SWON_T for any access to TS_T results always in an access to TS_T_OLD. For writing to this address the former old (TS_T_OLD_old) value is overwritten by the new one while the SWON_T bit changes. Thus the former new one is now the old one and the next access is after changing SWON_T directed to this place. Therefore write to TS_T first and after that immediately to FTV_T and PSTM, always before a new TS_T value is to be written. **NOTE:** After writing TS_T, FTV_T and PSTM in this order the address pointer AP with LSB(AP)=SWON_T shows for the corresponding address to TS_T_OLD, FTV_T and PSTM while LSB(AP)=/SWON_T results in an access to TS_T, FTV_T_OLD and PSTM_OLD. respectively. The value can be read only. This bit is reset when disabling the DPLL (DEN=0).
31 SWON_S	Switch of new STATE. Switch bit for LSB address of STATE. This bit is changed for each write access to TS_S or TS_S_OLD. Using this unchanged address bit SWON_S for any access to TS_S results always in an access to TS_S_OLD. For writing to this address the former old (TS_S_OLD_old) value is overwritten by the new one while the SWON_S bit changes. Thus the former new one is now the old one and the next access is after changing SWON_S directed to this place. Therefore write to TS_S first and after that immediately to FTV_S and PSSM, always before a new TS_S value is to be written. **NOTE:** After writing TS_S, FTV_S and PSSM in this order the address pointer AP with LSB(AP)=SWON_S shows for the corresponding address to TS_S_OLD, FTV_S and PSSM while LSB(AP)=/SWON_S results in an access to TS_S, FTV_S_OLD and PSSM_OLD, respectively. The value can be read only. This bit is reset when disabling the DPLL (DEN=0).

16.9.9 Address Offset Register of RAM2 Regions (DPLL_AOSV_2)

NOTE

The offset values are needed to support a scalable RAM size of region 2 from 1.5 Kbytes to 12 Kbytes. The values above must be in correlation with the offset size defined in the OSW register. All offset values are set automatically in accordance to the OSS value in the DPLL_OSW register. This value can be set only for DEN=0.

Address: 2_8000h base + 20h offset = 2_8020h

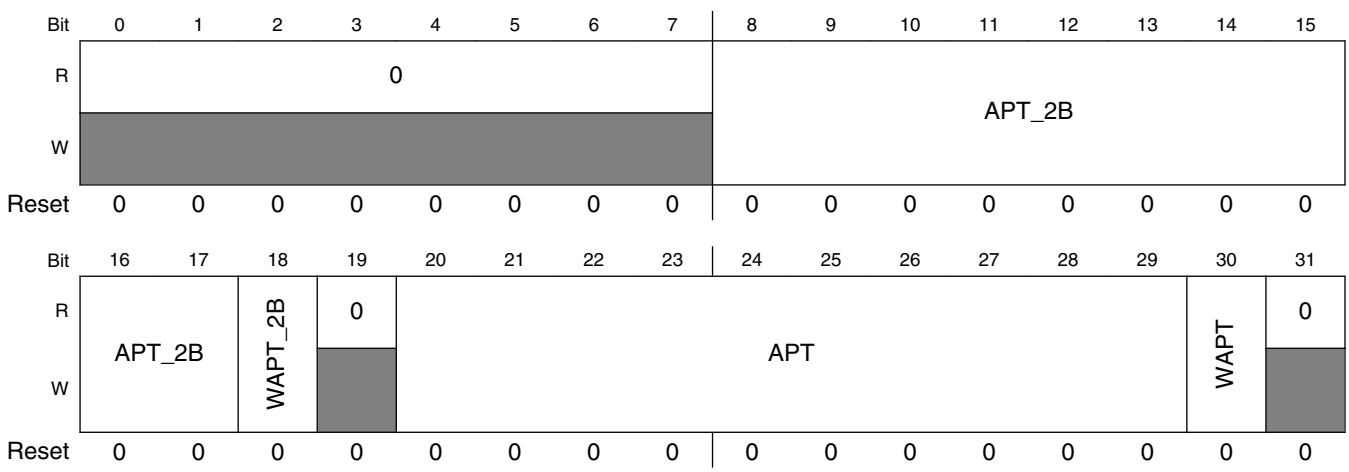
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	AOSV_2D								AOSV_2C								AOSV_2B								AOSV_2A							
W																																
Reset	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

DPLL_AOSV_2 field descriptions

Field	Description
0–7 AOSV_2D	Address offset value of the RAM 2d region. The value in this field is to be multiplied by 256 (shift left 8 Bits) and added with the start address of the RAM in order to get the start address of RAM region 2D. When the APT value is added to this start address, the current RAM cell DT_Tx is addressed.
8–15 AOSV_2C	Address offset value of the RAM 2c region. The value in this field is to be multiplied by 256 (shift left 8 Bits) and added with the start address of the RAM in order to get the start address of RAM region 2C. When the APT value is added to this start address, the current RAM cell ADT_Tx is addressed.
16–23 AOSV_2B	Address offset value of the RAM 2b region. The value in this field is to be multiplied by 256 (shift left 8 Bits) and added with the start address of the RAM in order to get the start address of RAM region 2B. When the APT value is added to this start address, the current RAM cell TSF_Tx is addressed.
24–31 AOSV_2A	Address offset value of the RAM 2a region. The value in this field is to be multiplied by 256 (shift left 8 Bits) and added with the start address of the RAM in order to get the start address of RAM region 2A. When the APT value is added to this start address, the current RAM cell RDT_Tx is addressed.

16.9.10 Actual RAM Pointer Address for Trigger (DPLL_APT)

Address: 2_8000h base + 24h offset = 2_8024h



DPLL_APT field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–17 APT_2B	Address pointer TRIGGER for RAM region 2b. Actual RAM pointer address value for TSF_T[i]

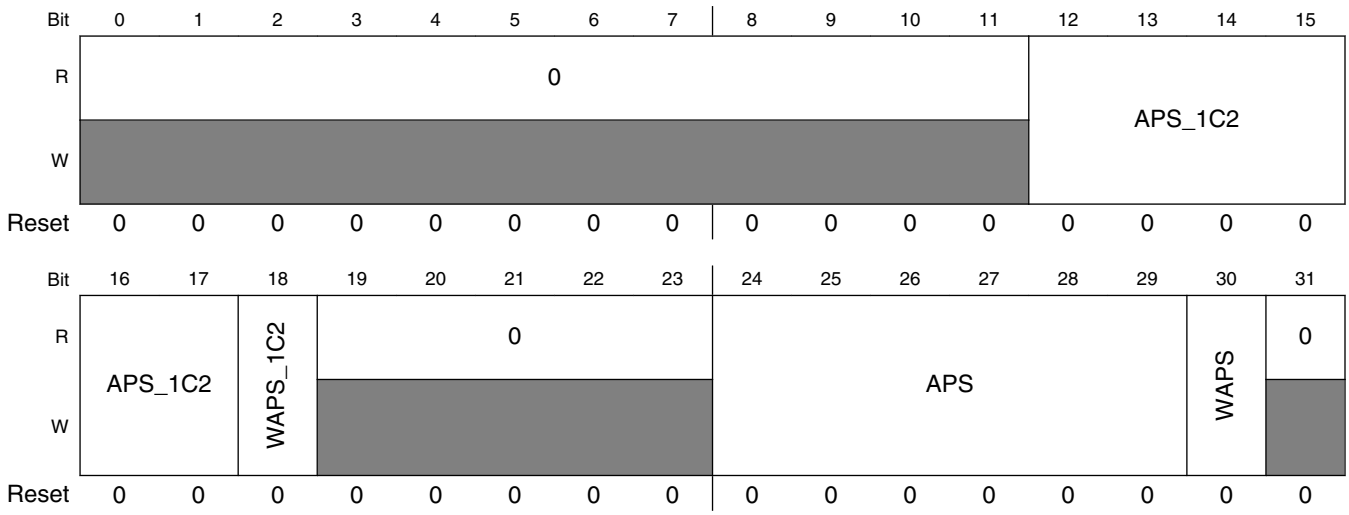
Table continues on the next page...

DPLL_APT field descriptions (continued)

Field	Description
	Actual RAM pointer address of TRIGGER events in FULL_SCALE for 2*(TNU+1) TRIGGER periods; this pointer is used for the RAM region 2B. The RAM pointer is initially set to zero. For SYT=1: The pointer APT_2B is incremented by SYN_T_old for each valid TRIGGER event (simultaneously with APT and APT_2C) for DIR1=0 when a valid TRIGGER input appears. For DIR1=1 (backwards) the APT is decremented by SYN_T_old. For SYT=0: APT_2B is incremented or decremented by 1. In addition, when the APT_2C value is written by the CPU - in order to synchronize the DPLL - with the next valid TRIGGER event, the APT_2B_ext value is added/substracted (while APT_2B_status is one; see DPLL_APT_sync register at chapter ?16.11.24). **NOTE:** This value can only be written when the WAPT_2B bit is set.
18 WAPT_2B	Write bit for address pointer APT_2b, read as zero. **NOTE:** This bit is cleared automatically after a write to 1. 0 The APT_2B is not writeable. 1 The APT_2B is writeable.
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–29 APT	Address pointer TRIGGER. Actual RAM pointer address value offset for DT_T[i] and RDT_T[i] in FULL_SCALE for 2*(TNU+1-SYN_NT) TRIGGER events. This pointer is used for the RAM region 2 subsections 2a and 2d. The pointer APT is incremented for each valid TRIGGER event (simultaneously with APT_2b, APT_2c) for DIR1=0. For DIR1=1 the APT is decremented. The APT offset value is added in the above shown bit position with the subsection address offset of the corresponding RAM region. **NOTE:** The APT pointer value is directed to the RAM position, in which the data values are to be written, which corresponds to the last increment. The APT value is not to be changed, when the direction (shown by DIR1) changes, because it points always to a storage place after the considered increment. Changing of DIR1 takes place always after a valid TRIGGER event and the resulting increment/decrement. **NOTE:** This value can only be written when the WAPT bit is set.
30 WAPT	Write bit for address pointer APT, reads as zero. **NOTE:** This bit is cleared automatically after a write to 1. 0 The APT is not writeable. 1 The APT is writeable.
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

16.9.11 Actual RAM Pointer Address for STATE (DPLL_APS)

Address: 2_8000h base + 28h offset = 2_8028h



DPLL_APS field descriptions

Field	Description
0–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–17 APS_1C2	Address pointer STATE for RAM regoin 1c2. Actual RAM pointer address value for TSF_S[i]. Initial value: zero (0x00). Actual RAM pointer and synchronization position/value of STATE events in FULL_SCALE for up to 64 STATE events but limited to 2*(SNU+1) in normal and emergency mode; this pointer is used for the RAM region 1C2. For SYS=1: APS_1C2 is incremented (decremented) by SYN_S_old for each valid STATE event and DIR2=0 (DIR2=1). For SYS=0: APT_1C2 is incremented or decremented by 1 respectively. The APS_1C2 offset value is added in the above shown bit position with the subsection offset of the RAM region. In addition, when the APS_1C3 value is written by the CPU - in order to synchronize the DPLL- with the next valid STATE event, the APS_1C2_ext value is added/substracted (while APS_1C2_status is one; see DPLL_APT_sync register at chapter ?16.11.25). **NOTE:** This value can only be written when the WAPS_1C2 bit is set.
18 WAPS_1C2	Write bit for address pointer APS_1c2, reads as zero. **NOTE:** This bit is cleared automatically after a write to 1. 0 The APS_1C2 is not writeable. 1 The APS_1C2 is writeable.
19–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DPLL_APS field descriptions (continued)

Field	Description
24–29 APS	Address pointer STATE. Actual RAM pointer address value for DT_S[i] and RDT_S[i]. Actual RAM pointer and synchronization position/value of STATE events in FULL_SCALE for up to 64 STATE events but limited to 2*(SNU+1-SYN_NS) in normal and emergency mode for SYSF=0 or to 2*(SNU+1)-SYN_NS for SYSF=1 respectively; this pointer is used for the RAM region 1c1 and 1c4. APS is incremented (decremented) by one for each valid STATE event and DIR2=0 DIR2=1). The APS offset value is added in the above shown bit position with the subsection offset of the RAM region. **NOTE:** The APS pointer value is directed to the RAM position, in which the data values are to be written, which correspond to the last increment. The APS value is not to be changed, when the direction (shown by DIR2) changes, because it points always to a storage place after the considered increment. Changing of DIR2 takes place always after a valid STATE event and the resulting increment/decrement. **NOTE:** This value can only be written when the WAPS bit is set.
30 WAPS	Write bit for address pointer APS, reads as zero. **NOTE:** This bit is cleared automatically after a write to 1. 0 The APS is not writeable. 1 The APS is writeable.
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

16.9.12 Actual RAM Pointer Address for Region 2C (DPLL_APT_2C)

Address: 2_8000h base + 2Ch offset = 2_802Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																				APT_2C										0	
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_APT_2C field descriptions

Field	Description
0–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–29 APT_2C	Address pointer TRIGGER for RAM region 2c. Actual RAM pointer address value for ADT_T[i]. Actual RAM pointer address value of TRIGGER adapt events in FULL_SCALE for 2*(TNU+1-SYN_NT) TRIGGER periods depending on the size of the used RAM 2; this pointer is used for the RAM region 2 for the subsection 2c only. The RAM pointer is initially set to zero. The APT_2C value is set by the CPU when the synchronization condition was detected. Within the RAM region 2C initially the conditions for synchronization gaps and adapt values are stored by the CPU.

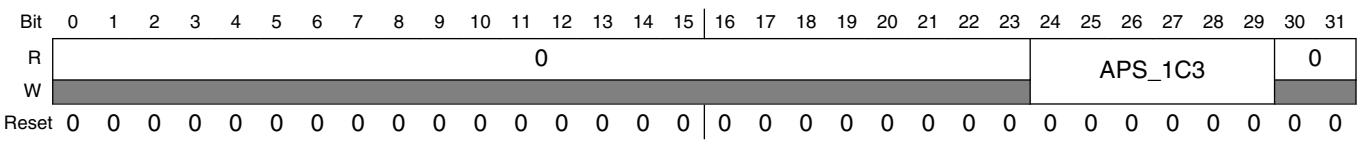
Table continues on the next page...

DPLL_APT_2C field descriptions (continued)

Field	Description
	NOTE: The APT_2c pointer values are directed to the RAM position of the profile element in RAM region 2C, which correspond to the current increment. For DIR1=0 (DIR1=1) the pointers APT_2c_x are incremented (decremented) by one simultaneously with APT. For SMC=0 the change of DIR1 takes place always after a valid TRIGGER event (by evaluation of the invalid slope) and the resulting increment/decrement. In the case SMC=1 the direction change is known before the input event is processed. The correction of the APT_2C pointer differs: for SMC=0 correct 4 times and for SMC=1 correct only 2 times. The APT_2C_x offset value is added in the above shown bit position with the subsection address offset of the corresponding RAM region.
30–31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

16.9.13 Actual RAM Pointer Address for Region 1C3 (DPLL_APS_1C3)

Address: 2_8000h base + 30h offset = 2_8030h

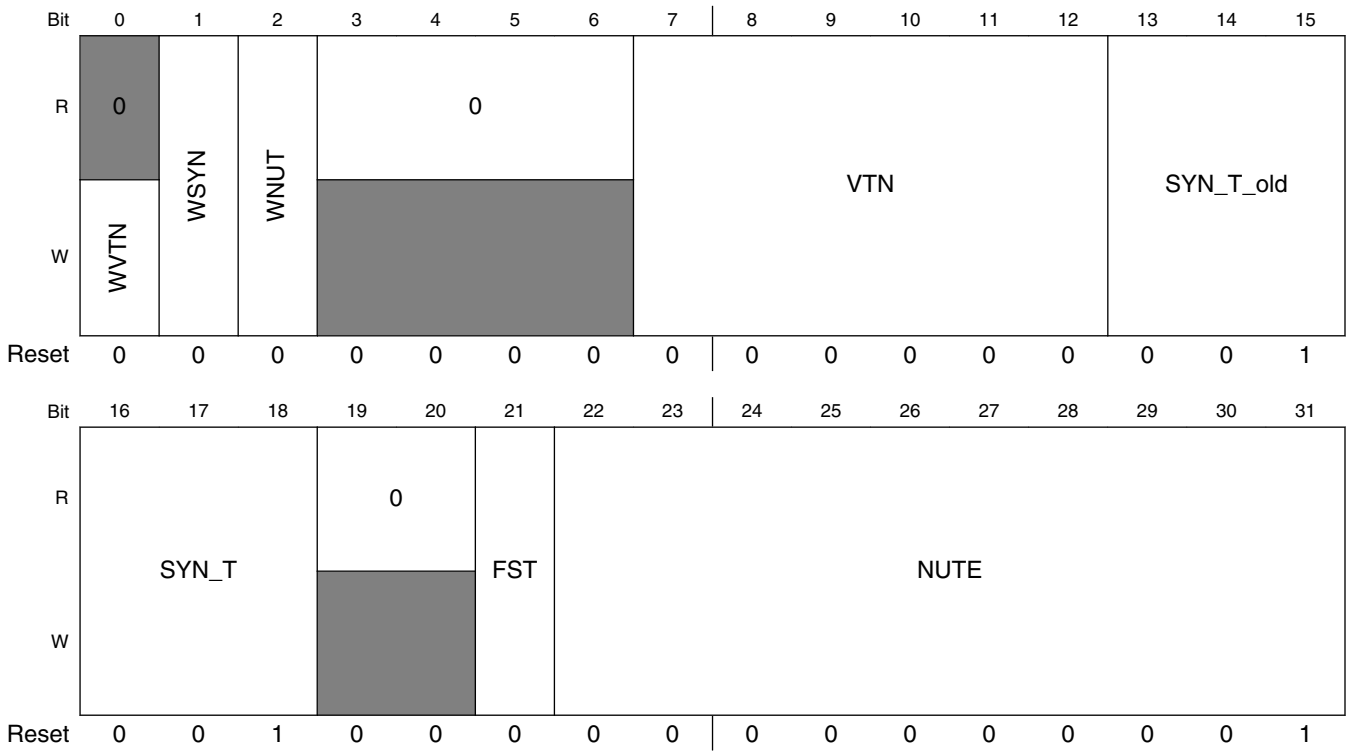


DPLL_APS_1C3 field descriptions

Field	Description
0–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24–29 APS_1C3	Address pointer STATE for RAM region 1c3. Actual RAM pointer address value for ADT_S[i]. Initial value: zero (0x00). Actual RAM pointer and synchronization position/value of STATE events in FULL_SCALE for up to 64 STATE events but limited to 2*(SNU+1-SYN_NS) in normal and emergency mode for SYSF=0 or to 2*(SNU+1)-SYN_NS for SYSF=1 respectively; this pointer is used for the RAM region 1C3. The RAM pointer is set by the CPU accordingly, when the synchronization condition was detected. **NOTE:** The APS_1C3 pointer value is directed to the RAM position of the profile element in RAM region 1C2, which corresponds to the current increment. When changing the direction DIR1 or DIR2 respectively, this is always known before a valid STATE event is processed. This is because of the pattern recognition in SPE (for PMSM) or because of the direction chnga recognition by TRIGGER. This directioin change results in an automatic increment (forwards) or decrement (backwards) when the input event occurs in addition with a 2 times correction. The APS_1C3_x offset value is added in the above shown bit position with the subsection address offset of the corresponding RAM region.
30–31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

16.9.14 Number of Recent TRIGGER Events used for Calculations (DPLL_NUTC)

Address: 2_8000h base + 34h offset = 2_8034h



DPLL_NUTC field descriptions

Field	Description
0 WVTN	Write control bit for VTN; reads as zero. **NOTE:** This bit is cleared automatically after a write to 1. 0 The VTN value is not writeable. 1 The VTN value is writeable.
1 WSYN	Write control bit for SYN_T and SYN_T_old; reads as zero. **NOTE:** This bit is cleared automatically after a write to 1. 0 The SYN_T value is not writeable. 1 The SYN_T value is writeable.
2 WNUT	Write control bit for NUTE and FST; reads as zero. **NOTE:** This bit is cleared automatically after a write to 1.

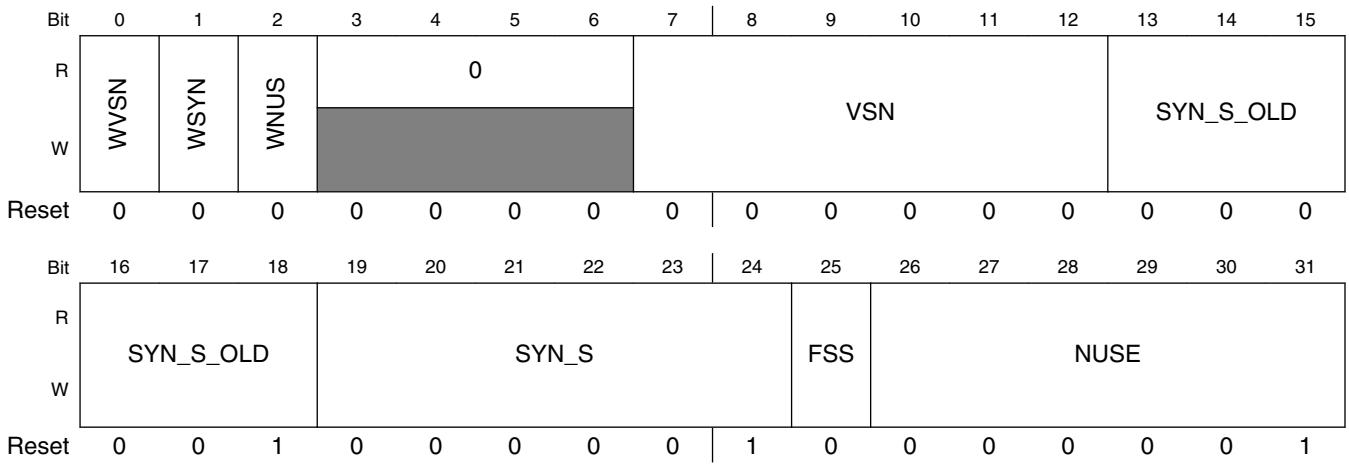
Table continues on the next page...

DPLL_NUTC field descriptions (continued)

Field	Description
	0 The NUTE value is not writeable. 1 The NUTE value is writeable.
3–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–12 VTN	Virtual TRIGGER number. Number of vitrtual increments in the current NUTE region. This value reflects the number of virtual increments in the current NUTE region; for NUTE=1 this value is zero, when the CPU sets NUTE to a value > 1 , it must also set VTN to the correspondent value; for NUTE is set to FULL_SCALE including NUTE=zero (211 modulo 211) the VTN is to be set to 2* SYN_NT. The VTN value is subtracted from the NUTE value in order to get the corresponding APT value for the past; the VTN value is not used for the APT_2b pointer. the VTN value is subtracted from the NUTE value in order to get the corresponding APT value for the past; the VTN value is not used for the APT_2b pointer. **NOTE:** This value can only be written when the WVTN bit is set..
13–15 SYN_T_old	Number of real and virtual events to be considered for the last increment. number of real and virtual events to be considered for the last increment. **NOTE:** This value is updated by the SYN_T value when the WSYN bit in this register is set.
16–18 SYN_T	Number of real and virtual events to be considered for the current increment. This value reflects the NT value of the last valid increment, stored in ADT_T[i]; to be updated after all calculations in step 17 of Table ?16.8.6.7. **NOTE:** This value is updated by the SYN_T value when the WSYN bit in this register is set.
19–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 FST	FULL_SCALE of TRIGGER. This value is to be set, when NUTE is set to FULL_SCALE. **NOTE:** This value can only be written when the WNUT bit is set. 0 The NUTE value is less then FULL_SCALE. 1 The NUTE value is equal to FULL_SCALE.
22–31 NUTE	Number of recent TRIGGER events used for SUB_INC1 and action calculations modulo 2 * (TNU_{max} +1). No gap is considered in that case for this value, but in the VTN value (see below). This register is set by the CPU, but reset automatically to “1” by a change of direction or lost of LOCK. Each other value can be set by the CPU, maybe Full_SCALE, HALF_SCALE or parts of them. For FULL_SCALE set NUTE = 2*(TNU +1) and for HALF_SCALE NUTE = TNU +1. The relation values QDT_Tx are calculated using NUTE values in the past with its maximum value of 2 * (TNU + 1). The value zero(in combination with the value FST = 1) means 2^{11} values in the past. **NOTE:** This value can only be written when the WNUT bit is set.

16.9.15 Number of Recent STATE Events used for Calculations (DPLL_NUSC)

Address: 2_8000h base + 38h offset = 2_8038h



DPLL_NUSC field descriptions

Field	Description
0 WVSN	Write control bit for VSN; reads as zero. **NOTE:** This bit is cleared automatically after a write to 1. 0 The VSN value is not writeable. 1 The VSN value is writeable.
1 WSYN	Write control bit for SYN_S and SYN_S_old; reads as zero. **NOTE:** This bit is cleared automatically after a write to 1. 0 The SYN_S value is not writeable. 1 The SYN_S value is writeable.
2 WNUS	Write control bit for NUSE; reads as zero. **NOTE:** This bit is cleared automatically after a write to 1. 0 The NUSE value is not writeable. 1 The NUSE value is writeable.
3–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–12 VSN	Virtual STATE number. Number of virtual state increments in the current NUSE region. This value reflects the number of virtual increments in the current NUSE region; for NUSE=1 this value is zero, when the CPU sets NUSE to a value > 1 or zero(27 modulo 27) , it must also set VSN to the correspondent value.

Table continues on the next page...

DPLL_NUSC field descriptions (continued)

Field	Description
	The VSN value is subtracted from the NUSE value in order to get the corresponding APS value for the past; the VSN value is not used for the APS_1c2 pointer. VSN is to be updated by the CPU when a new gap is to be considered for NUSE or a gap is leaving the NUSE region; for this purpose the SASI interrupt can be used; no further update of VSN is necessary when NUSE is set to FULL_SCALE. **NOTE:** This value can only be written when the WVSN bit is set.
13–18 SYN_S_OLD	Number of real and virtual events to be considered for the last increment. This value reflects the NS value of the last but one valid increment, stored in ADT_S[i]; is updated automatically when writing SYN_S. **NOTE:** This value is updated by the SYN_S value when the WSYN bit in this register is set.
19–24 SYN_S	Number of real and virtual events to be considered for the current increment. This value reflects the NS value of the last valid increment, stored in ADT_S[i]; to be updated after all calculations in step 37 of Table ?16.8.6.7. **NOTE:** This value can only be written when the WSYN bit in this register is set.
25 FSS	FULL_SCALE of STATE. This value is to be set, when NUSE is set to FULL_SCALE. **NOTE:** This value can only be written when the WNUT bit is set. 0 The NUSE value is less then FULL_SCALE. 1 The NUSE value is equal to FULL_SCALE.
26–31 NUSE	Number of recent STATE events used for SUB_INCx calculations modulo 2*(SNU $_{max}$ +1). No gap is considered in that case for this value, but in the VSN value (see below). This register is set by the CPU but reset automatically to “1” by a change of direction or lost of LOCK. Each other value can be set by the CPU, maybe Full_SCALE, HALF_SCALE or parts of them. The relation values QDT_Sx are calculated using NUSE values in the past with its maximum value of 2*SNU +1. **NOTE:** This value can only be written when the WNUS bit is set.

16.9.16 Number of Active TRIGGER Events to Interrupt (DPLL_NTI_CNT)

Address: 2_8000h base + 3Ch offset = 2_803Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																						NTI_CNT									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_NTI_CNT field descriptions

Field	Description
0–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22–31 NTI_CNT	Number of TRIGGERs to interrupt. Number of active TRIGGER events to the next DPLL_CDTI interrupt. This value shows the remaining TRIGGER events until an active TRIGGER slope results in an DPLL_CDTI interrupt. The value is to be counted down for each valid TRIGGER event.

16.9.17 Interrupt Register (DPLL_IRQ_NOTIFY)

NOTE

All bits in the DPLL_IRQ_NOTIFY register are set permanently until writing a one value to the corresponding bit.

Address: 2_8000h base + 40h offset = 2_8040h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0				DCGI	SORI	TORI	CDSI	CDTI	TE4I	TE3I	TE2I	TE1I	TE0I	LL2I	GL2I
W					w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	EI	LL1I	GL1I	W1I	W2I	PWI	TASI	SASI	MTI	MSI	TISI	SISI	TAXI	TINI	PEI	PDI
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_IRQ_NOTIFY field descriptions

Field	Description
0–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 DCGI	Direction change interrupt. 0 No direction change of TRIGGER is detected. 1 Direction change of TRIGGER is detected.
5 SORI	STATE out of range. 0 STATE is not out of range. 1 STATE is out of range, the SOR bit in the DPLL_STATUS register is set to 1.
6 TORI	TRIGGER out of range interrupt.

Table continues on the next page...

DPLL_IRQ_NOTIFY field descriptions (continued)

Field	Description
	0 TRIGGER is not out of range. 1 TRIGGER is out of range, the TOR bit in the DPLL_STATUS register is set to 1.
7 CDSI	Calculation of STATE duration done. 0 No Interrupt on calculated STATE duration requested. 1 Interrupt on calculated STATE duration requested.
8 CDTI	Calculation of TRIGGER duration done, only while NTI_CNT is zero. 0 No Interrupt on calculated TRIGGER duration requested or NTI_CNT is not zero. 1 Interrupt on calculated TRIGGER duration requested while NTI_CNT is zero.
9 TE4I	TRIGGER event interrupt 4. 0 No Interrupt on TRIGGER event 4 requested. 1 Interrupt on TRIGGER event 4 requested.
10 TE3I	TRIGGER event interrupt 3. 0 No Interrupt on TRIGGER event 3 requested. 1 Interrupt on TRIGGER event 3 requested.
11 TE2I	TRIGGER event interrupt 2. 0 No Interrupt on TRIGGER event 2 requested. 1 Interrupt on TRIGGER event 2 requested.
12 TE1I	TRIGGER event interrupt 1. 0 No Interrupt on TRIGGER event 1 requested. 1 Interrupt on TRIGGER event 0 requested.
13 TE0I	TRIGGER event interrupt 0. 0 No Interrupt on TRIGGER event 0 requested. 1 Interrupt on TRIGGER event 0 requested.
14 LL2I	Loss of lock interrupt for SUB_INC2. 0 The lock loss interrupt is not requested. 1 The lock loss interrupt is requested.
15 GL2I	Get of lock interrupt, for SUB_INC2. 0 The lock getting interrupt is not requested. 1 The lock getting interrupt is requested.
16 EI	Error interrupt (see status register bit 31). 0 The error interrupt is not requested. 1 The error interrupt is requested.
17 LL1I	Loss of lock interrupt for SUB_INC1. 0 The lock loss interrupt is not requested. 1 The lock loss interrupt is requested.
18 GL1I	Get of lock interrupt, for SUB_INC1.

Table continues on the next page...

DPLL_IRQ_NOTIFY field descriptions (continued)

Field	Description
	0 The lock getting interrupt is not requested. 1 The lock getting interrupt is requested.
19 W1I	Write access to RAM region 1b or 1c interrupt. 0 The RAM write access interrupt is not requested. 1 The RAM write access interrupt is requested.
20 W2I	RAM write access to RAM region 2 interrupt. 0 The RAM write access interrupt is not requested. 1 The RAM write access interrupt is requested.
21 PWI	Plausibility window (PVT) violation interrupt of TRIGGER. 0 The plausibility window is not violated. 1 The plausibility window is violated.
22 TASI	TRIGGER active slope interrupt. 0 No active slope of TRIGGER is detected. 1 An active slope of TRIGGER is detected.
23 SASI	STATE active slope interrupt. 0 No active slope of STATE is detected. 1 An active slope of STATE is detected.
24 MTI	Missing TRIGGER interrupt. 0 The missing TRIGGER interrupt is not requested. 1 The missing TRIGGER interrupt is requested.
25 MSI	Missing STATE interrupt. 0 The missing STATE interrupt is not requested. 1 The missing STATE interrupt is requested.
26 TISI	TRIGGER inactive slope interrupt. 0 No inactive slope of TRIGGER is detected. 1 An inactive slope of TRIGGER is detected
27 SISI	STATE inactive slope interrupt. 0 No inactive slope of STATE is detected. 1 An inactive slope of STATE is detected.
28 TAXI	TRIGGER maximum hold time violation interrupt (dt > THMA > 0). 0 No violation of maximum hold time of TRIGGER is detected. 1 A violation of maximum hold time of TRIGGER is detected.
29 TINI	TINI 0 No violation of minimum hold time of TRIGGER is detected. 1 A violation of minimum hold time of TRIGGER is detected.
30 PEI	DPLL enable interrupt; announces the switch on of the DEN bit.

Table continues on the next page...

DPLL_IRQ_NOTIFY field descriptions (continued)

Field	Description
	NOTE: This event is combined with the PDI interrupt to the common PDI + PEI interrupt line number 1. 0 The DPLL enable interrupt is not requested. 1 The DPLL enable interrupt is requested.
31 PDI	DPLL disable interrupt; announces the switch off of the DEN bit. **NOTE:** This event is combined with the PEI interrupt to the common PDI + PEI interrupt line number 1. 0 The DPLL disable interrupt is not requested. 1 The DPLL disable interrupt is requested.

16.9.18 Interrupt Enable Register (DPLL_IRQ_EN)

Address: 2_8000h base + 44h offset = 2_8044h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0				DCGI_IRQ_EN	SORI_IRQ_EN	TORI_IRQ_EN	CDSI_IRQ_EN	CDTI_IRQ_EN	TE4I_IRQ_EN	TE3I_IRQ_EN	TE2I_IRQ_EN	TE1I_IRQ_EN	TE0I_IRQ_EN	LL2I_IRQ_EN	GL2I_IRQ_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	EI_IRQ_EN	LL1I_IRQ_EN	GL1I_IRQ_EN	W1I_IRQ_EN	W2I_IRQ_EN	PWI_IRQ_EN	TASI_IRQ_EN	SASI_IRQ_EN	MTI_IRQ_EN	MSI_IRQ_EN	TISI_IRQ_EN	SISI_IRQ_EN	TAXI_IRQ_EN	TINI_IRQ_EN	PEI_IRQ_EN	PDI_IRQ_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_IRQ_EN field descriptions

Field	Description
0–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 DCGI_IRQ_EN	Direction change interrupt. 0 No Interrupt when a direction change of TRIGGER is detected. 1 Interrupt when a direction change of TRIGGER is detected.
5 SORI_IRQ_EN	STATE out of range. 0 No Interrupt when STATE is out of range enabled. 1 Interrupt when STATE is out of range enabled.
6 TORI_IRQ_EN	TRIGGER out of range interrupt.

Table continues on the next page...

DPLL_IRQ_EN field descriptions (continued)

Field	Description
	0 No Interrupt whenTRIGGER is out of range enabled. 1 Interrupt whenTRIGGER is out of range enabled.
7 CDSI_IRQ_EN	Enable interrupt when calculation of TRIGGER duration done. 0 No Interrupt on calculated STATE duration enabled. 1 Interrupt on calculated STATE duration enabled.
8 CDTI_IRQ_EN	Enable interrupt when calculation of TRIGGER duration done. 0 No Interrupt on calculated TRIGGER duration enabled. 1 Interrupt on calculated TRIGGER duration enabled.
9 TE4I_IRQ_EN	TRIGGER event interrupt 4 enable. 0 No Interrupt on TRIGGER event 4 enabled. 1 Interrupt on TRIGGER event 4 enabled.
10 TE3I_IRQ_EN	TRIGGER event interrupt 3 enable. 0 No Interrupt on TRIGGER event 3 enabled. 1 Interrupt on TRIGGER event 3 enabled.
11 TE2I_IRQ_EN	TRIGGER event interrupt 2 enable. 0 No Interrupt on TRIGGER event 2 enabled. 1 Interrupt on TRIGGER event 2 enabled.
12 TE1I_IRQ_EN	TRIGGER event interrupt 1 enable. 0 No Interrupt on TRIGGER event 1 enabled. 1 Interrupt on TRIGGER event 1 enabled.
13 TE0I_IRQ_EN	TRIGGER event interrupt 0 enable. 0 No Interrupt on TRIGGER event 0 enabled. 1 Interrupt on TRIGGER event 0 enabled.
14 LL2I_IRQ_EN	Loss of lock interrupt enable for SUB_INC2. 0 The lock loss interrupt is not requested. 1 The lock loss interrupt is requested.
15 GL2I_IRQ_EN	Get of lock interrupt enable for SUB_INC2. 0 The lock getting interrupt is not requested. 1 The lock getting interrupt is requested.
16 EI_IRQ_EN	Error interrupt enable (see status register). 0 The error interrupt is not enabled. 1 The error interrupt is enabled.
17 LL1I_IRQ_EN	Loss of lock interrupt enable. 0 The lock loss interrupt is not enabled. 1 The lock loss interrupt is enabled.
18 GL1I_IRQ_EN	Get of lock interrupt enable, when lock arises.

Table continues on the next page...

DPLL_IRQ_EN field descriptions (continued)

Field	Description
	0 The lock getting interrupt is not enabled. 1 The lock getting interrupt is enabled.
19 W1I_IRQ_EN	Write access to RAM region 1b or 1c interrupt. 0 The RAM write access interrupt is not enabled. 1 The RAM write access interrupt is enabled.
20 W2I_IRQ_EN	RAM write access to RAM region 2 interrupt enable. 0 The RAM write access interrupt is not enabled. 1 The RAM write access interrupt is enabled.
21 PWI_IRQ_EN	Plausibility window (PVT) violation interrupt of TRIGGER enable. 0 The plausibility violation interrupt is not enabled. 1 The plausibility violation interrupt is enabled.
22 TASI_IRQ_EN	TRIGGER active slope interrupt enable. 0 The active slope TRIGGER interrupt is not enabled. 1 The active slope TRIGGER interrupt is enabled.
23 SASI_IRQ_EN	STATE active slope interrupt enable. 0 The active slope STATE interrupt is not enabled. 1 The active slope STATE interrupt is enabled.
24 MTI_IRQ_EN	Missing TRIGGER interrupt enable. 0 The missing TRIGGER interrupt is not enabled. 1 The missing TRIGGER interrupt is enabled.
25 MSI_IRQ_EN	Missing STATE interrupt enable. 0 The missing STATE interrupt is not enabled. 1 The missing STATE interrupt is enabled.
26 TISI_IRQ_EN	TRIGGER inactive slope interrupt enable bit. 0 The interrupt at the inactive slope of TRIGGER is not enabled. 1 The interrupt at the inactive slope of TRIGGER is enabled.
27 SISI_IRQ_EN	STATE inactive slope interrupt enable bit. 0 The interrupt at the inactive slope of STATE is not enabled. 1 The interrupt at the inactive slope of STATE is enabled.
28 TAXI_IRQ_EN	TRIGGER maximum hold time violation interrupt enable bit. 0 Maximum hold time violation of TRIGGER interrupt is not enabled. 1 The maximum hold time violation of TRIGGER interrupt is enabled.
29 TINI_IRQ_EN	TRIGGER minimum hold time violation interrupt enable bit. 0 Minimum hold time violation of TRIGGER interrupt is not enabled. 1 The minimum hold time violation of TRIGGER interrupt is enabled.
30 PEI_IRQ_EN	DPLL enable interrupt enable, when switch on of the DEN bit.

Table continues on the next page...

DPLL_IRQ_EN field descriptions (continued)

Field	Description
	0 The DPLL enable interrupt is not enabled. 1 The DPLL enable interrupt is enabled.
31 PDI_IRQ_EN	DPLL disable interrupt enable, when switch off of the DEN bit. 0 The DPLL disable interrupt is not enabled. 1 The DPLL disable interrupt is enabled.

16.9.19 Force Interrupt Register (DPLL_IRQ_FORCINT)

NOTE

Bit[4:31] is write protected by bit GTM_CTRL[RF_PROT] and is cleared automatically after a write.

Address: 2_8000h base + 48h offset = 2_8048h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0				TRG_DCGI	TRG_SORI	TRG_TORI	TRG_CDSI	TRG_CDTI	TRG_TE4I	TRG_TE3I	TRG_TE2I	TRG_TE1I	TRG_TE0I	TRG_LL2I	TRG_GL2I
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TRG_EI	TRG_LL1I	TRG_GL1I	TRG_W1I	TRG_W2I	TRG_PWI	TRG_TASI	TRG_SASI	TRG_MTI	TRG_MSI	TRG_TISI	TRG_SISI	TRG_TAXI	TRG_TINI	TRG_PEI	TRG_PDI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_IRQ_FORCINT field descriptions

Field	Description
0–3 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
4 TRG_DCGI	Force interrupt DCGI. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.

Table continues on the next page...

DPLL_IRQ_FORCINT field descriptions (continued)

Field	Description
5 TRG_SORI	Force Interrupt SORI. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
6 TRG_TORI	Force Interrupt TORI. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
7 TRG_CDSI	Force Interrupt CDSI. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
8 TRG_CDTI	Force Interrupt CDTI. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
9 TRG_TE4I	Force Interrupt TE4I. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
10 TRG_TE3I	Force Interrupt TE3I. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
11 TRG_TE2I	Force Interrupt TE2I. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
12 TRG_TE1I	Force Interrupt TE1I.

Table continues on the next page...

DPLL_IRQ_FORCINT field descriptions (continued)

Field	Description
	NOTE: This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
13 TRG_TE0I	Force Interrupt TE0I. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
14 TRG_LL2I	Force Interrupt LL2I. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
15 TRG_GL2I	Force Interrupt GL2I. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
16 TRG_EI	Force Interrupt EI. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
17 TRG_LL1I	Force Interrupt LL1I. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
18 TRG_GL1I	Force Interrupt GL1I. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
19 TRG_W1I	Force Interrupt W1I. **NOTE:** This bit is cleared automatically after a write to 1.

Table continues on the next page...

DPLL_IRQ_FORCINT field descriptions (continued)

Field	Description
	0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
20 TRG_W2I	Force Interrupt W2I. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
21 TRG_PWI	Force Interrupt PWI. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
22 TRG_TASI	Force Interrupt TASI. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
23 TRG_SASI	Force Interrupt SASI. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
24 TRG_MTI	**NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
25 TRG_MSI	Force Interrupt MSI. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
26 TRG_TISI	Force Interrupt TISI. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
27 TRG_SISI	Force Interrupt SISI.

Table continues on the next page...

DPLL_IRQ_FORCINT field descriptions (continued)

Field	Description
	NOTE: This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
28 TRG_TAXI	Force Interrupt TAXI. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
29 TRG_TINI	Force Interrupt TINI. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
30 TRG_PEI	Force Interrupt PEI. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.
31 TRG_PDI	Force Interrupt PDI. **NOTE:** This bit is cleared automatically after a write to 1. 0 The corresponding interrupt is not forced. 1 The corresponding interrupt is forced for one clock.

16.9.20 Interrupt Request Mode (DPLL_IRQ_MODE)

Address: 2_8000h base + 4Ch offset = 2_804Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0														IRQ_MODE	
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- Bits 28-31 do not reset to a given value.

DPLL_IRQ_MODE field descriptions

Field	Description
0–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–31 IRQ_MODE	IRQ mode selection. **NOTE:** The interrupt modes are described in XXXXX. 00 Level mode. 01 Pulse mode. 10 Pulse-Notify mode. 11 Single-Pulse mode.

16.9.21 Error Interrupt Enable Register (DPLL_EIRQ_EN)

Footnotes:

7 The SOR Bit is set, when the time to the next active STATE slope exceeds the value of the last nominal STATE duration multiplied with the value of the SLR register (see chapter ?16.13.37) and is reset, when at the current or last valid input event a direction change was detected.

8 The TOR Bit is set, when the time to the next active TRIGGER slope exceeds the value of the last nominal TRIGGER duration multiplied with the value of the TLR register (see chapter ?16.13.36) and is reset, when at the current or last valid input event a direction change was detected.

NOTE

The DPLL_STATUS register is reset when the DPLL is disabled (switching DEN from 1 to 0).

Address: 2_8000h base + 50h offset = 2_8050h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0				DCGI_EIRQ_EN	SORI_EIRQ_EN	TORI_EIRQ_EN	CDSI_EIRQ_EN	CDTI_EIRQ_EN	TE4I_EIRQ_EN	TE3I_EIRQ_EN	TE2I_EIRQ_EN	TE1I_EIRQ_EN	TE0I_EIRQ_EN	LL2I_EIRQ_EN	GL2I_EIRQ_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	EI_EIRQ_EN	LL1I_EIRQ_EN	GL1I_EIRQ_EN	W1I_EIRQ_EN	W2I_EIRQ_EN	PWI_EIRQ_EN	TASI_EIRQ_EN	SASI_EIRQ_EN	MTI_EIRQ_EN	MSI_EIRQ_EN	TISI_EIRQ_EN	SISI_EIRQ_EN	TAXI_EIRQ_EN	TINI_EIRQ_EN	PEI_EIRQ_EN	PDI_EIRQ_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_EIRQ_EN field descriptions

Field	Description
0–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 DCGI_EIRQ_EN	Direction Change Interrupt. 0 No Interrupt when a direction change of TRIGGER is detected 1 Interrupt when a direction change of TRIGGER is detected
5 SORI_EIRQ_EN	STATE Out of Range Interrupt. 0 No Interrupt when STATE is out of range enabled 1 Interrupt when STATE is out of range enabled
6 TORI_EIRQ_EN	TRIGGER Out of Range Interrupt 0 No Interrupt whenTRIGGER is out of range enabled 1 Interrupt whenTRIGGER is out of range enabled
7 CDSI_EIRQ_EN	Enable interrupt when calculation of STATE duration done. 0 No Interrupt on calculated STATE duration enabled 1 Interrupt on calculated STATE duration enabled
8 CDTI_EIRQ_EN	Enable interrupt when calculation of TRIGGER duration done 0 No Interrupt on calculated TRIGGER duration enabled 1 Interrupt on calculated TRIGGER duration enabled
9 TE4I_EIRQ_EN	TRIGGER event interrupt 4 enable. 0 No Interrupt on TRIGGER event 4 enabled 1 Interrupt on TRIGGER event 4 enabled
10 TE3I_EIRQ_EN	TRIGGER event interrupt 3 enable. 0 No Interrupt on TRIGGER event 3 enabled 1 Interrupt on TRIGGER event 3 enabled
11 TE2I_EIRQ_EN	TRIGGER event interrupt 2 enable. 0 No Interrupt on TRIGGER event 2 enabled 1 Interrupt on TRIGGER event 2 enabled
12 TE1I_EIRQ_EN	TRIGGER event interrupt 1 enable.

Table continues on the next page...

DPLL_EIRQ_EN field descriptions (continued)

Field	Description
	0 No Interrupt on TRIGGER event 1 enabled 1 Interrupt on TRIGGER event 1 enabled
13 TE0I_EIRQ_EN	TRIGGER event interrupt 0 enable. 0 No Interrupt on TRIGGER event 0 enabled 1 Interrupt on TRIGGER event 0 enabled
14 LL2I_EIRQ_EN	Loss of lock interrupt enable for SUB_INC2. 0 The lock loss interrupt is not requested 1 The lock loss interrupt is requested.
15 GL2I_EIRQ_EN	Get of lock interrupt enable for SUB_INC2. 0 The lock getting interrupt is not requested 1 The lock getting interrupt is requested
16 EI_EIRQ_EN	Error interrupt enable (see status register). 0 The error interrupt is not enabled 1 The error interrupt is enabled
17 LL1I_EIRQ_EN	Loss of lock interrupt enable. 0 The lock loss interrupt is not enabled 1 The lock loss interrupt is enabled
18 GL1I_EIRQ_EN	Get of lock interrupt enable, when lock arises. 0 The lock getting interrupt is not enabled 1 The lock getting interrupt is enabled
19 W1I_EIRQ_EN	Write access to RAM region 1b or 1c interrupt. 0 The RAM write access interrupt is not enabled 1 The RAM write access interrupt is enabled.
20 W2I_EIRQ_EN	RAM write access to RAM region 2 interrupt enable. 0 The RAM write access interrupt is not enabled 1 The RAM write access interrupt is enabled
21 PWI_EIRQ_EN	Plausibility window (PVT) violation interrupt of TRIGGER enable. 0 The plausibility violation interrupt is not enabled 1 The plausibility violation interrupt is enabled
22 TASI_EIRQ_EN	TRIGGER active slope interrupt enable. 0 The active slope TRIGGER interrupt is not enabled 1 The active slope TRIGGER interrupt is enabled
23 SASI_EIRQ_EN	STATE active slope interrupt enable. 0 The active slope STATE interrupt is not enabled. 1 The active slope STATE interrupt is enabled
24 MTI_EIRQ_EN	Missing TRIGGER interrupt enable.

Table continues on the next page...

DPLL_EIRQ_EN field descriptions (continued)

Field	Description
	0 The missing TRIGGER interrupt is not enabled 1 The missing TRIGGER interrupt is enabled
25 MSI_EIRQ_EN	Missing STATE interrupt enable. 0 The missing STATE interrupt is not enabled 1 The missing STATE interrupt is enabled
26 TISI_EIRQ_EN	TRIGGER inactive slope interrupt enable bit. 0 The interrupt at the inactive slope of TRIGGER is not enabled 1 The interrupt at the inactive slope of TRIGGER is enabled
27 SISI_EIRQ_EN	STATE inactive slope interrupt enable bit. 0 The interrupt at the inactive slope of STATE is not enabled 1 The interrupt at the inactive slope of STATE is enabled
28 TAXI_EIRQ_EN	TRIGGER maximum hold time violation interrupt enable bit. 0 Maximum hold time violation of TRIGGER interrupt is not enabled 1 The maximum hold time violation of TRIGGER interrupt is enabled
29 TINI_EIRQ_EN	TRIGGER minimum hold time violation interrupt enable bit. 0 Minimum hold time violation of TRIGGER interrupt is not enabled 1 The minimum hold time violation of TRIGGER interrupt is enabled
30 PEI_EIRQ_EN	DPLL enable interrupt enable, when switch on of the DEN bit. 0 The DPLL enable interrupt is not enabled 1 The DPLL enable interrupt is enabled
31 PDI_EIRQ_EN	DPLL disable interrupt enable, when switch off of the DEN bit. 0 The DPLL disable interrupt is not enabled 1 The DPLL disable interrupt is enabled

16.9.22 Counter Value of Sent SUB_INC1 Pulses (DPLL_INC_CNT1)

Address: 2_8000h base + B0h offset = 2_80B0h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								INC_CNT1																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_INC_CNT1 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 INC_CNT1	Actual number of pulses to be still sent out at the current increment until the next valid input signal in automatic end mode.

Table continues on the next page...

DPLL_INC_CNT1 field descriptions (continued)

Field	Description
	Automatic addition of the number of demanded pulses MLT/MLS1 when getting a valid TRIGGER/STATE input in normal or emergency mode respectively when SGE1=1. In the case of a change of the direction, the wrong number of pulses are corrected twice. So, add the difference between NMB_T and INC_CNT1 twice to INC_CNT1 before sending out the correction pulses. **NOTE:** For test purposes only, this value can be written when the DPLL is disabled.

16.9.23 Counter Value of sent SUB_INC2 values (for SMC=1 and RMO=1) (DPLL_INC_CNT2)

Address: 2_8000h base + B4h offset = 2_80B4h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								INC_CNT2																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

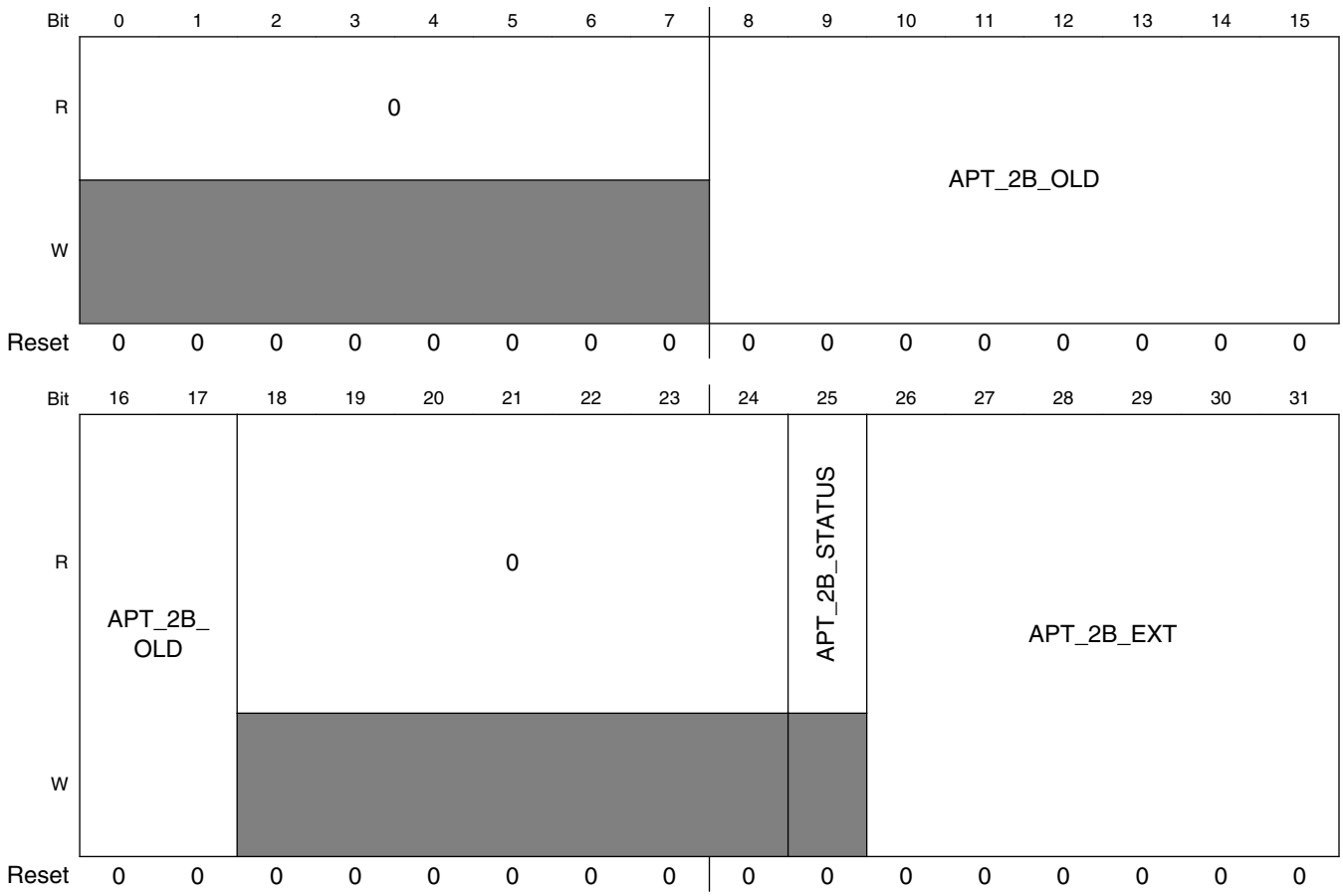
DPLL_INC_CNT2 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 INC_CNT2	Actual number of pulses to be still sent out at the current increment until the next valid input signal in automatic end mode. Automatic addition of the number of demanded pulses (MLT+1)/MLS1 when getting a valid TRIGGER/STATE input in normal or emergency mode respectively when SGE2=1. In the case of a change of the direction, the wrong number of pulses are corrected twice. So, add the difference between NMB_S and INC_CNT2 twice to INC_CNT2 before sending out the correction pulses. **NOTE:** For test purposes only, this value can be written when the DPLL is disabled.

16.9.24 TRIGGER Time Stamp Field Offset at Synchronization Time (DPLL_APT_SYNC)

TSF offset at synchronization time

Address: 2_8000h base + B8h offset = 2_80B8h



DPLL_APT_SYNC field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–17 APT_2B_OLD	Address pointer TRIGGER for RAM region 2B at synchronization time. This value is set by the current APT_2B value when the synchronization takes place for the first valid TRIGGER event after writing APT_2C but before adding the offset value APT_2B_ext (that means: when APT_2B_status=1). Address pointer APT_2B value at the moment of synchronization, before the offset value is added, means the pointer with this value points to the last value before the additional inserted gap

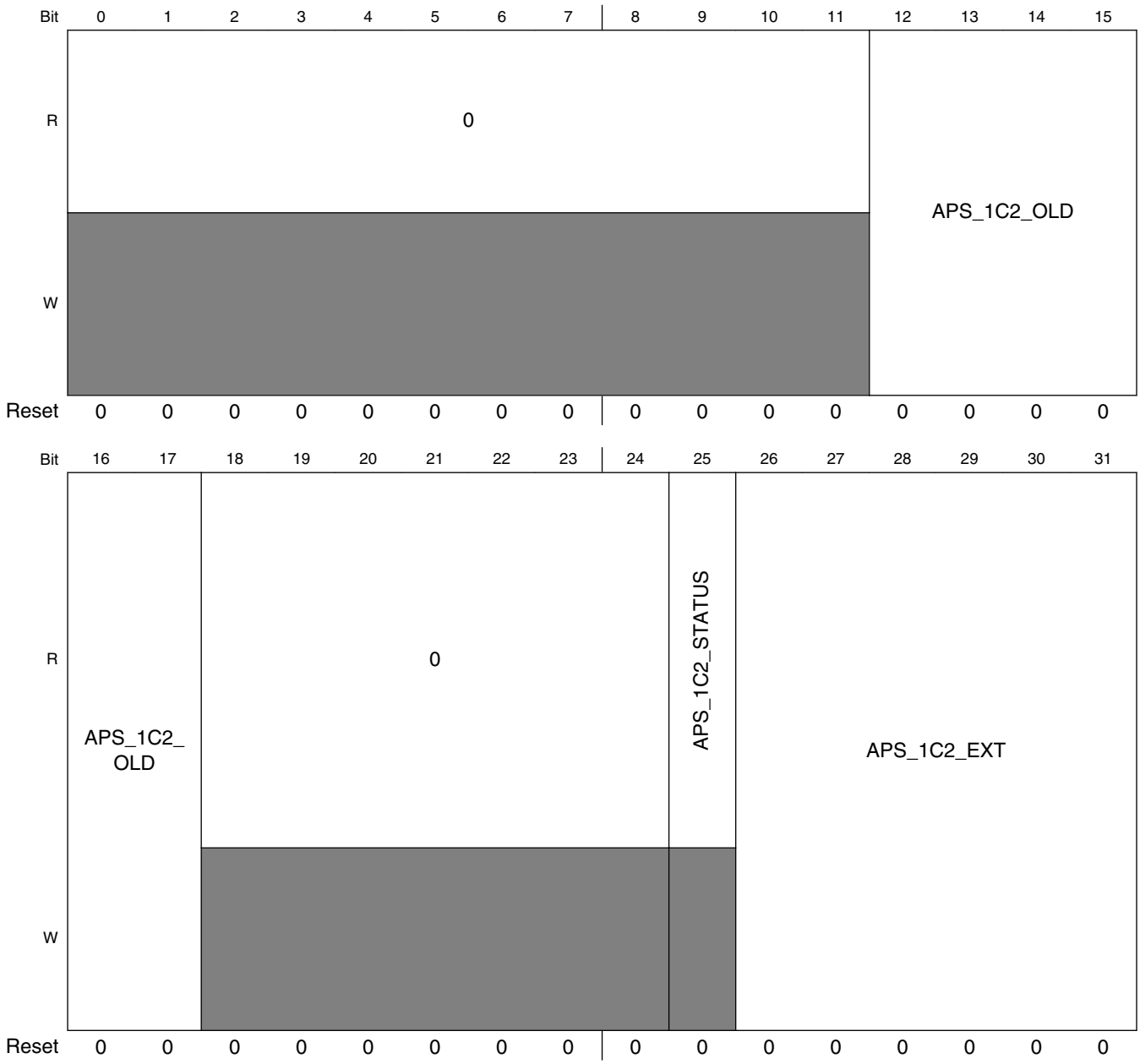
Table continues on the next page...

DPLL_APT_SYNC field descriptions (continued)

Field	Description
18–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 APT_2B_ STATUS	Address pointer 2B status. Set by CPU before the synchronization is performed. The value is cleared when the APT_2B_OLD value is written. 0 APT_2B_EXT is not to be considered. 1 APT_2B_EXT has to be considered for time stamp field extension.
26–31 APT_2B_EXT	Address pointer 2B extension. This offset value determines the value by which the APT_2B is changed at the synchronization time; set by the CPU before the synchronization is performed. This offset value is the number of virtual increments to be inserted into the TSF for an imminent intended synchronization; the CPU sets its value dependent on the gaps until the synchronization time taking into account the considered NUTE value to be set and including the next future increment (when SYN_T_OLD is still 1). When the synchronization takes place, this value is to be added to the APT_2B address pointer (for forward direction, DIR1=0) and the APT_2B_status bit is cleared after it. For backward direction subtract APT_2B_EXT accordingly. This correction is done after updating the RAM TSF with the last TS_T value. **NOTE:** When the synchronization is intended and the NUTE value is to be set to FULL_SCALE after it, the APT_2B_EXT value must be set to 2*SYN_NT in order to be able to fill all gaps in the extended TSF_T with the corresponding values by the CPU. When all of the values for FULL_SCALE are not all available, the APT_2B_EXT value considers only a share according to the corresponding NUTE value to be set after the synchronization.

16.9.25 STATE Time Stamp Field Offset at Synchronization Time (DPLL_APS_SYNC)

Address: 2_8000h base + BCh offset = 2_80BCh



DPLL_APS_SYNC field descriptions

Field	Description
0–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

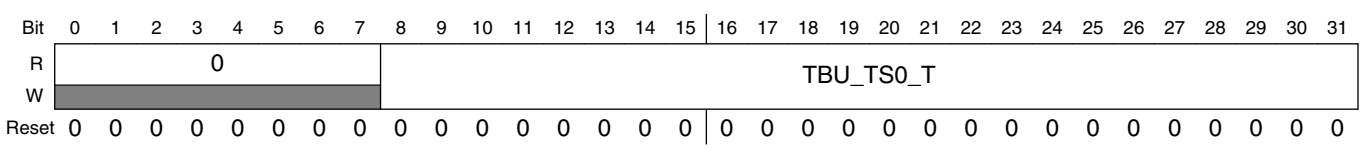
Table continues on the next page...

DPLL_APS_SYNC field descriptions (continued)

Field	Description
12–17 APS_1C2_OLD	Address pointer STATE for RAM regoin 1C2 at synchronization time. This value is set by the current APS_1C2 value when the synchronization takes place for the first valid STATE event after writing APS_1C3 but before adding the offset value APS_1C2_EXT (that means: when APS_1C2_status = 1). Address pointer APS_1C2 value at the moment of synchronization, before the offset value is added, that means the pointer with this value points to the last value before the additional inserted gap.
18–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 APS_1C2_ STATUS	Address pointer 1C2 status. Set by CPU before the synchronization is performed. The value is cleared automatically when the APS_1C2_OLD value is written. 0 APS_1C2_EXT is not to be considered. 1 APS_1C2_EXT has to be considered for time stamp field extension.
26–31 APS_1C2_EXT	Address pointer 1C2 extension. This offset value determines the value by which the APS_1C2 is changed at the synchronization time; set by the CPU before the synchronization is performed.. This offset value is the number of virtual increments to be inserted in the TSF for an imminent intended synchronization; the CPU sets its value dependent on the gaps until the synchronization time taking into account the considered NUSE value to be set and including the next future increment (when SYN_S_OLD is still 1). When the synchronization takes place, this value is to be added to the APS_1C2 address pointer (for forward direction, DIR2=0) and the APT_1C2_STATUS bit is cleared after it. For backward direction subtract APS_1C2_EXT accordingly. **NOTE:** When the synchronization is intended and the NUSE value is to be set to FULL_SCALE after it, the APS_1C2_EXT value must be set to SYN_NS (for SYSF=1) or 2*SYN_NS (for SYSF=0) in order to be able to fill all gaps in the extended TSF_S with the corresponding values by the CPU. When all of the values for FULL_SCALE are not available, the APS_1C2_EXT value considers only a share according to the NUSE value to be set after the synchronization.

16.9.26 Time Stamp Value for the last valid TRIGGER (DPLL_TBU_TS0_T)

Address: 2_8000h base + C0h offset = 2_80C0h



DPLL_TBU_TS0_T field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DPLL_TBU_TS0_T field descriptions (continued)

Field	Description
8–31 TBU_TS0_T	Value of TBU_TS0 at the last TRIGGER event. For each T_VALID, the value of TBU_TS0 is stored in this register **NOTE:** For test purposes only, this value can be written when the DPLL is disabled.

16.9.27 Time Stamp Value for the last valid STATE (DPLL_TBU_TS0_S)

Address: 2_8000h base + C4h offset = 2_80C4h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								TBU_TS0_S																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_TBU_TS0_S field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 TBU_TS0_S	Value of TBU_TS0 at the last STATE event. For each S_VALID, the value of TBU_TS0 is stored in this register. **NOTE:** For test purposes only, this value can be written when the DPLL is disabled.

16.9.28 ADD_IN Value in Direct Load Mode for TRIGGER (DPLL_ADD_IN_LD1)

Address: 2_8000h base + C8h offset = 2_80C8h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								ADD_IN_LD1																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_ADD_IN_LD1 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 ADD_IN_LD1	Input value for SUB_INC1 generation. Given by the CPU, this value can be used in normal und emergency mode (SMC=0) as well as for SMC=1.

Table continues on the next page...

DPLL_ADD_IN_LD1 field descriptions (continued)

Field	Description
	The value is loaded by the CPU but used by the DPLL only for DLM1=1 (see DPLL_CTRL_1 register). When switching DLM1 to 1, the value in the register is used for the SUB_INC1 generation beginning from the next valid TRIGGER or STATE event respectively independently if new values are written by the CPU or not. When a new value is written, the output frequency changes according to the given value beginning immediately from the moment of writing. Do not wait for performing step 10 in the state machine for ADD_IN calculations. If the ADD_IN_LD1 value is zero all pulses are sent with the highest possible frequency.

16.9.29 ADD_IN Value in Direct Load Mode for STATE (DPLL_ADD_IN_LD2)

Address: 2_8000h base + CCh offset = 2_80CCh

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								ADD_IN_LD2																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_ADD_IN_LD2 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 ADD_IN_LD2	Input value for SUB_INC2 generation. Given by the CPU, this value can be used in normal und emergency mode (SMC=0) as well as for SMC=1. The value is loaded by the CPU but used by the DPLL only for DLM1=2 (see DPLL_CTRL_1 register). When switching DLM2 to 1, the value in the register is used for the SUB_INC2 generation beginning from the next valid TRIGGER or STATE event respectively independently if new values are written by the CPU or not. When a new value is written, the output frequency changes according to the given value beginning immediately from the moment of writing. Do not wait for performing step 30 in the state machine for ADD_IN calculations. If the ADD_IN_LD2 value is zero all pulses are sent with the highest possible frequency.

16.9.30 Status Register (DPLL_STATUS)

Footnotes:

7 The SOR Bit is set, when the time to the next active STATE slope exceeds the value of the last nominal STATE duration multiplied with the value of the SLR register (see chapter ?16.13.37) and is reset, when at the current or last valid input event a direction change was detected.

8 The TOR Bit is set, when the time to the next active TRIGGER slope exceeds the value of the last nominal TRIGGER duration multiplied with the value of the TLR register (see chapter ?16.13.36) and is reset, when at the current or last valid input event a direction change was detected.

NOTE

The DPLL_STATUS register is reset when the DPLL is disabled (switching DEN from 1 to 0).

Address: 2_8000h base + FCh offset = 2_80FCh

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	ERR	LOCK1	FTD	FSD	SYT	SYS	LOCK2	0	BWD1	BWD2	ITN	ISN	CAIP1	CAIP2	CSVT	CSVS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	LOW_RES	0		RAM2_ERR	MT	TOR	MS	SOR	PSE	RCT	RCS	CRO	CTO	0	CSO	0
W				w1c	w1c	w1c	w1c	w1c				w1c	w1c		w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_STATUS field descriptions

Field	Description
0 ERR	Error. Error during configuration or operation resulting in unexpected values. 0 When all bits in positions 8 through 0 and 10 are zero. 1 When at least one bit in positions 8 through 0 or 10 is one.
1 LOCK1	DPLL Lock status concerning SUB_INC1. DPLL Lock status concerning SUB_INC1 = 0. The DPLL is not locked for TRIGGER (while SMC=RMO=0 or SMC=1) or for STATE (while SMC=0 and RMO=1). LOCK1 is set : • In normal mode (for RMO=SMC=0): Bit is set for a valid TRIGGER event when SYT is set and the number of events between two gaps is as expected by the profile (NT values in the ADT_T[i] field) or when SYN_NT=0 and SYT=1. • In normal mode (for RMO=SMC=0, LCD=1): Bit is set for a valid TRIGGER event when SYT is set and the number of events between two increments without missing TRIGGER (no gap) is as expected by the profile (NT values in the ADT_T[i] field). • In emergency mode (for RMO=1 and SMC=0): Bit is set for a valid STATE event, when SYS is set and the received events are in correspondence to the profile (NS values in the ADT_S[i] field) for at least two expected missing STATE events or when SYN_NS=0. • For SMC=1: Bit is set for a valid TRIGGER even when SYT is set and SYN_NT=0 or when SYT is set and the profile stored in the ADT_T[i] field matches once between two gaps. LOCK1 is reset for RMO=SMC=0 when: • A corresponding missing TRIGGER event occurs while SYN_T=1. This does mean an unexpected missing TRIGGER. • The corresponding input signal TRIGGER is out of locking range TLR. • A corresponding direction change is detected. LOCK1 is reset for for RMO=1 and SMC=0 when:

Table continues on the next page...

DPLL_STATUS field descriptions (continued)

Field	Description
	• A corresponding missing STATE event occurs while SYN_S=1. This means an unexpected missing STATE. • The corresponding input signal STATE is out of locking range TLR LOCK1 is reset for SMC=1 when: • A corresponding missing TRIGGER event occurs while SYN_T = 1. This means an unexpected missing TRIGGER. • The corresponding input signal TRIGGER is out of locking range TLR. • A corresponding direction change is detected. 0 The DPLL is not locked for TRIGGER (while SMC=RMO=0 or SMC=1) or for STATE (while SMC=0 and RMO=1). 1 The DPLL is locked for TRIGGER (while SMC=RMO=0 or SMC=1) or for STATE (while SMC=0 and RMO=1).
2 FTD	First TRIGGER detected. **NOTE:** No change of FTD for switching from normal to emergency mode or vice versa. 0 No valid TRIGGER event was detected after enabling DPLL. 1 At least one valid TRIGGER event was detected after enabling DPLL.
3 FSD	First STATE detected. **NOTE:** No change of FSD for switching from normal to emergency mode or vice versa. 0 Still no valid STATE event was detected after enabling DPLL. 1 At least one valid STATE event was detected after enabling DPLL.
4 SYT	Synchronization condition of TRIGGER fixed. **NOTE:** This bit is set when the CPU writes to the APT_2c address pointer.
5 SYS	Synchronization condition of STATE fixed. **NOTE:** This bit is set when the CPU writes to the APS_1c3 address pointer.
6 LOCK2	DPLL Lock status concerning SUB_INC2. **NOTE:** Locking of SUB_INC2 appears for RMO=SMC=1: Bit is set, when SYS is set and the number of events between two missing STATEs is as expected by the SYN_S values. Note: LOCK2 is set for SMC=RMO=1: • For a valid STATE event when SYS is set and SYN_NS=0, or • When SYS is set and the profile stored in the ADS_Ti field matches once between two gaps. LOCK2 is reset for SMC=RMO=1: • When a missing STATE event occurs while SYN_S = 1. This means an unexpected missing STATE. • When the corresponding input signal STATE is out of locking range SLR. 0 The DPLL is not locked concerning STATE for SMC=1. 1 The DPLL is locked concerning STATE for SMC=1.

Table continues on the next page...

DPLL_STATUS field descriptions (continued)

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 BWD1	Backwards drive of SUB_INC1. 0 Forward direction. 1 Backward direction.
9 BWD2	Backwards drive of SUB_INC2. 0 Forward direction. 1 Backward direction.
10 ITN	Increment number of TRIGGER is not plausible. **NOTE:** Bit is set when the number of TRIGGERS is different to profile. 0 The number of TRIGGER events between synchronization gaps is plausible, a direction change is detected or the address pointer APT_2c is written. 1 After setting LOCK1 in normal mode (for SMC=0 or SMC=1) or in emergency mode (only for SMC=0) for missing or additional TRIGGER signals detected; bit is cleared when a direction change is detected or the APT_2c is written.
11 ISN	Increment number of STATE is not plausible. **NOTE:** Bit is set when the number of STATES is different to profile 0 The number of STATE events between synchronization gaps is plausible, a direction change is detected or the APS_1c3 pointer is written. 1 After setting LOCK1 in emergency mode (SMC=0 and RMO=1) or LOCK2 for SMC=RMO=1 missing or additional STATE signals detected; bit is cleared when a direction change is detected or the APS_1c3 is written.
12 CAIP1	Calculation of actions 0 to 11 in progress (1st part). 0 Currently no action calculation, new data requests possible. 1 Action calculation in progress, no new data requests possible.
13 CAIP2	Calculation of actions 12 to 23 in progress (2nd part). 0 Currently no action calculation, new data requests possible. 1 Action calculation in progress, no new data requests possible.
14 CSVT	Current signal value TRIGGER. 0 The last TRIGGER_S value was 0. 1 The last TRIGGER_S value was 1.
15 CSVS	Current signal value STATE. 0 The last STATE_S value was 0. 1 The last STATE_S value was 1.
16 LOW_RES	Low resolution of TBU_TS0. Used for DPLL input, this value reflects the input signal LOW_RES.

Table continues on the next page...

DPLL_STATUS field descriptions (continued)

Field	Description
	0 The lower 24 bits of TBU_TS0 are used as input for the DPLL. 1 The higher 24 bits of TBU_TS0 are used as input for the DPLL.
17–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 RAM2_ERR	DPLL internal access to not configured RAM2 memory space. 0 No access to not configured RAM2 memory space. 1 Access to not configured RAM2 memory space.
20 MT	Missing TRIGGER detected according to TOV. 0 No missing TRIGGER detected or a new valid TRIGGER slope occured. 1 At least one missing TRIGGER detected after the last valid slope.
21 TOR	TRIGGER out of range. See footnote 8. 0 All TRIGGER signal events appear within TLR interval or a direction change was detected. 1 At least one TRIGGER signal event is out of TLR; address pointers APT, APT_2B and APT_2C are frozen and the generation of pulses SUB_INC1 is stopped.
22 MS	Missing STATE detected according to TOV_S. 0 No missing STATE detected or a new valid STATE slope occured. 1 At least one missing STATE detected after the last valid slope.
23 SOR	STATE out of range. See footnote 7. 0 All STATE signal events appear within SLR interval or a direction change was detected. 1 At least one STATE signal event is out of SLR; address pointers APS, APS_1C2 and APS_1C3 are frozen and the generation of pulses SUB_INC1,2 respectively is stopped.
24 PSE	Prediction space configuration error. 0 No prediction space error detected. 1 Configured offset value of RAM2 is too small in order to store all TNU+1 values twice in FULL_SCALE.
25 RCT	Resolution conflict TRIGGER. 0 No resolution conflict detected. 1 The TS0_HRS value is set to 1 while LOW_RES=0.
26 RCS	Resolution conflict STATE. 0 No resolution conflict detected. 1 The TS0_HRT value is set to 1 while LOW_RES=0.
27 CRO	Calculated Reciprocal value overflow. Bit is set when the calculation of RDT_T_actual or RDT_S_actual leads to an overflow. **NOTE:** An overflow in calculation of reciprocal values can occur, when the condition of Footnote 3) to the DPLL_CTRL_0 register is violated (see chapter ?16.11.1). Such an overflow can occur according to the calculations in equations (16.1c) or (16.6c).

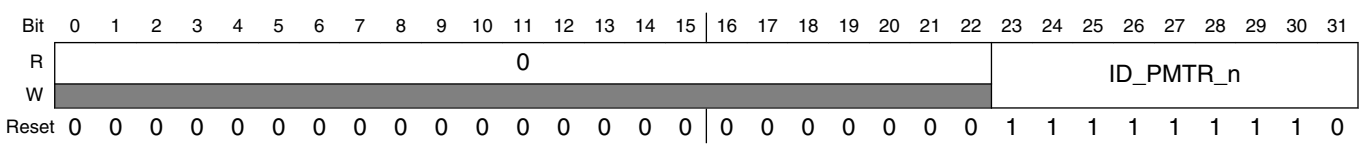
Table continues on the next page...

DPLL_STATUS field descriptions (continued)

Field	Description
	The overflow is detected when after the calculation and shifting left 32 bits at least one of the bits 31 to 24 is not zero. In that case the corresponding register is set to 0xFFFFFF. 0 No overflow at any reciprocal calculation. 1 Overflow for at least one reciprocal calculation.
28 CTO	Calculated TRIGGER duration overflow. Bit is set when equation DPLL-5a or DPLL-5b leads to an overflow. **NOTE:** When one of bits 3:0 is set, the corresponding register contains the maximum value 0xFFFFFF. 0 No overflow at equation 16.5b. 1 Overflow at equation 16.5b.
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0. 0 No overflow at equation 16.5a. 1 Overflow at equation 16.5a.
30 CSO	Calculated STATE duration overflow. Bit is set when equation DPLL-10a or DPLL-10b leads to an overflow. **NOTE:** When one of bits 3:0 is set, the corresponding register contains the maximum value 0xFFFFFF. 0 No overflow at equation 16.10b. 1 Overflow at equation 16.10b.
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0. 0 No overflow at equation 16.10a. 1 Overflow at equation 16.10a.

16.9.31 ID Information for In ut Signal PTR x (Position minus Time Request) (DPLL_ID_PMTR_*n*)

Address: 2_8000h base + 100h offset + (4d × i), where i=0d to 31d



DPLL_ID_PMTR_*n* field descriptions

Field	Description
0–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–31 ID_PMTR_n	ID information to the input signal PMTR_n from the ARU.

Table continues on the next page...

DPLL_ID_PMTR_*n* field descriptions (continued)

Field	Description
	NOTE: This value can only be written when the DPLL is disabled.

16.9.32 Shadow Register of DPLL_CTRL_0 controlled by a valid TRIGGER Slope (DPLL_CTRL_0_SHADOW_TRIGGER)

Footnote:

1 For the HW-IP only, the values characterized are stored for a valid TRIGGER slope. All other values remain 0. In the reference model also, the other values may be readable.

Address: 2_8000h base + 1E0h offset = 2_81E0h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	RMO	0			IDT	0	AMT	0								
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0					IFP	MLT									
W																
Reset	0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	1

DPLL_CTRL_0_SHADOW_TRIGGER field descriptions

Field	Description
0 RMO	Reference mode. Selection of the relevant input signal for generation of SUB_INC1. **NOTE:** See footnote 1.
1–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0. 0 Do not use information. 1 Use information.
4 IDT	Input delay TRIGGER; Use of input delay information transmitted in FT part of the TRIGGER signal. **NOTE:** See footnote 1. 0 Do not use information. 1 Use information.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DPLL_CTRL_0_SHADOW_TRIGGER field descriptions (continued)

Field	Description
	0 Do not use information. 1 Use information.
6 AMT	Adapt mode TRIGGER Use of adaptation information of TRIGGER. **NOTE:** See footnote 1. 0 Do not use information. 1 Use information.
7–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 IFP	Input filter position. Value contains position or time related information. **NOTE:** See footnote 1. 0 TRIGGER_FT and STATE_FT mean time related values, that means the number of time stamp clocks 1 TRIGGER_FT and STATE_FT mean position related values, that means the number of SUB_INC1 (or SUB_INC2 in the case SM=1) pulses respectively
22–31 MLT	Multiplier for TRIGGER. MLT+1 is number of SUB_INC1 pulses between two TRIGGER events in normal mode (1…1024). **NOTE:** See footnote 1.

16.9.33 Shadow Register of DPLL_CTRL_0 controlled by a valid STATE Slope (DPLL_CTRL_0_SHADOW_STATE)

Footnote:

2 For the HW-IP only, the values characterized are stored for a valid STATE slope. All other values remain 0. In the reference model also the other values may be readable.

Address: 2_8000h base + 1E4h offset = 2_81E4h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	RMO	0			IDS	0	AMS	0								
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0					IFP	0									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_CTRL_0_SHADOW_STATE field descriptions

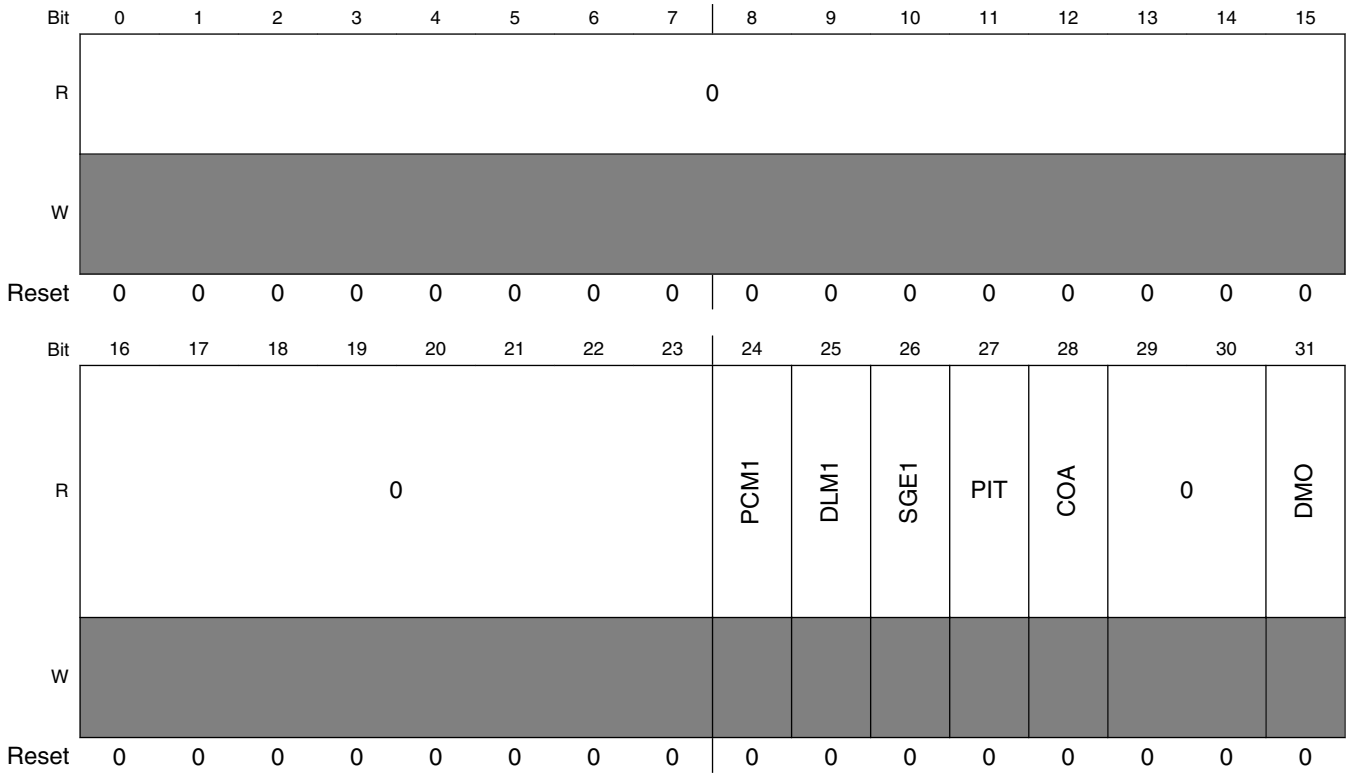
Field	Description
0 RMO	Reference mode. Selection of the relevant the input signal for generation of SUB_INC1. **NOTE:** See footnote 2.
1–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0. 0 Do not use information. 1 Use information.
4 IDS	Input delay STATE. Use of input delay information transmitted in FT part of the STATE signal. **NOTE:** See footnote 2. 0 Do not use information. 1 Use information.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0. 0 Do not use information. 1 Use information.
6 AMS	Addapt mode STATE. Use of adaptation information of STATE. **NOTE:** See footnote 2. 0 Do not use information. 1 Use information.
7–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 IFP	Input filter position. Value contains position or time related information. **NOTE:** See footnote 2. 0 TRIGGER_FT and STATE_FT mean time related values, that means the number of time stamp clocks 1 TRIGGER_FT and STATE_FT mean position related values, that means the number of SUB_INC1 (or SUB_INC2 in the case SMC=1) pulses respectively
22–31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

16.9.34 Shadow Register of DPLL_CTRL_1 controlled by a valid TRIGGER Slope (DPLL_CTRL_1_SHADOW_TRIGGER)

Footnote:

1 For the HW-IP only the values characterized are stored for a valid TRIGGER slope. All other values remain 0. In the reference model also the other values may be readable.

Address: 2_8000h base + 1E8h offset = 2_81E8h



DPLL_CTRL_1_SHADOW_TRIGGER field descriptions

Field	Description
0–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0. 0 Disabled. 1 Enabled.
24 PCM1	Pulse Correction Mode. For SUB_INC1 generation. **NOTE:** See footnote 2. 0 The DPLL does not use the correction value stored in MPVAL1 1 The DPLL uses the correction value stored in MPVAL1 in normal and emergency mode
25 DLM1	Direct Load Mode. For SUB_INC1 generation. **NOTE:** See footnote 2. 0 the DPLL uses the calculated ADD_IN_CAL value for the SUB_INC1 generation 1 the ADD_IN_LD value is used for the SUB_INC1 generation and is provided by the CPU

Table continues on the next page...

DPLL_CTRL_1_SHADOW_TRIGGER field descriptions (continued)

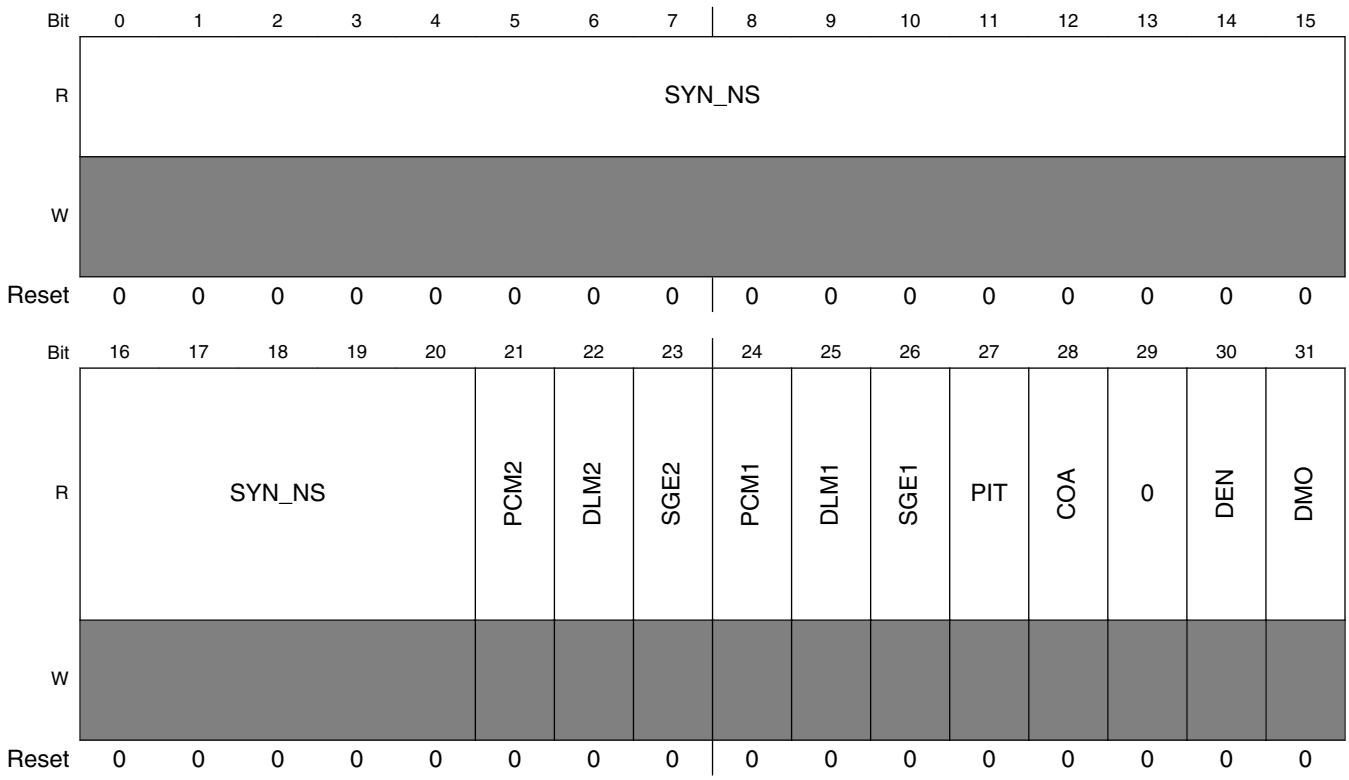
Field	Description
26 SGE1	SUB_INC1 generator enable. **NOTE:** See footnote 2. 0 Disabled. 1 Enabled.
27 PIT	Plausibility. Value PVT to next valid TRIGGER is time related 0 The plausibility value is position related 1 The plausibility value is time related
28 COA	Correction strategy in automatic end mode (DMO=0). **NOTE:** See footnote 2. 0 The pulse frequency of the CMU_CLK0 will be used to make up for missing pulses from last increment 1 missing pulses of the last increment are distributed evenly to the next increment
29–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
31 DMO	DPLL mode select.. **NOTE:** See footnote 2. 0 Automatic end mode 1 Continuous mode

16.9.35 Shadow Register of DPLL_CTRL_1 controlled by a valid STATE Slope (DPLL_CTRL_1_SHADOW_STATE)

Footnote:

2 For the HW-IP, only the values characterized are stored for a valid STATE slope. All other values remain 0. When DEN=0, the relevant bit values of the original register DPLL_CTRL_1 are transferred at the next system clock without any input event.

Address: 2_8000h base + 1ECh offset = 2_81ECh



DPLL_CTRL_1_SHADOW_STATE field descriptions

Field	Description
0–20 SYN_NS	Synchronization number of STATE. Summarized number of virtual increments in HALF_SCALE.
21 PCM2	Pulse Correction Mode for SUB_INC2 generation. See Footnote 2.
22 DLM2	Direct Load Mode for SUB_INC2 generation. See Footnote 2.
23 SGE2	SUB_INC2 generator enable. See Footnote 2.
24 PCM1	Pulse Correction Mode for SUB_INC1 generation. See Footnote 2.
25 DLM1	Direct Load Mode for SUB_INC1 generation. See Footnote 2.
26 SGE1	SUB_INC1 generator enable. See Footnote 2.
27 PIT	Plausibility value PVT to next valid TRIGGER is time related.
28 COA	Correction strategy in automatic end mode (DMO=0).

Table continues on the next page...

DPLL_CTRL_1_SHADOW_STATE field descriptions (continued)

Field	Description
	See Footnote 2.
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 DEN	DPLL enable.
31 DMO	DPLL mode select. See Footnote 2.

16.9.36 Register to control the RAM Initialization (DPLL_RAM_INI)

NOTE

Depending on the vendor configuration, the connected RAM regions are initialized to zero in the case of a module HW reset or for setting the RST bit in the GTM_RST register.

Address: 2_8000h base + 1FCh offset = 2_81FCh

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0											INIT_RAM	0	INIT_2	INIT_1B	INIT_1A
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_RAM_INI field descriptions

Field	Description
0–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 INIT_RAM	RAM regions 1A, 1B and 2 are to be initialized. **NOTE:** Setting the INIT_RAM bit results in a RAM reset only when the DPLL is not enabled (DEN=0). **NOTE:** This bit is cleared automatically after a write to 1. 0 Do not start initialization of all RAM regions. 1 Start initialization of all RAM regions.
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 INIT_2	RAM region 2 initialization in progress. 0 No initialization of considered RAM region in progress. 1 Initialization of considered RAM region in progress.
30 INIT_1B	RAM region 1B initialization in progress. 0 No initialization of considered RAM region in progress. 1 Initialization of considered RAM region in progress.
31 INIT_1A	RAM region 1A initialization in progress. 0 No initialization of considered RAM region in progress. 1 Initialization of considered RAM region in progress.

16.9.37 Calculated Time Value to start Action i (DPLL_TSAC_*n*)

Address: 2_8000h base + E00h offset + (4d × i), where i=0d to 31d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	TSAC																															
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

DPLL_TSAC_*n* field descriptions

Field	Description
0–31 TSAC	Calculated time stamp for ACTION_i. **NOTE:** This value can only be written when the DPLL is disabled.

16.9.38 DPLL Calculated Position Value ACTION i (DPLL_PSAC_*n*)

Address: 2_8000h base + E80h offset + (4d × i), where i=0d to 31d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	PSAC																															
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

DPLL_PSAC_*n* field descriptions

Field	Description
0–31 PSAC	Calculated position value for the start of ACTION_i . In normal or emergency mode, according to equations DPLL-16.17 or DPLL-16.20, respectively. **NOTE:** This value can only be written when the DPLL is disabled.

16.9.39 Control Bits for up to 32 Actions (DPLL_ACB_*n*)

Address: 2_8000h base + F00h offset + (4d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0			ACB_3					0			ACB_2					0			ACB_1					0			ACB_0			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLL_ACB_*n* field descriptions

Field	Description
0–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–7 ACB_3	Action Control Bits of ACTION_(i + 3), reflects ACT_D[i+3](52:48), i=4*j. **NOTE:** This value can only be written when the DPLL is disabled.
8–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–15 ACB_2	Action Control Bits of ACTION_(i + 2), reflects ACT_D[i+2](52:48), i=4*j. **NOTE:** This value can only be written when the DPLL is disabled.
16–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–23 ACB_1	Action Control Bits of ACTION_(i + 1) , reflects ACT_D[i+1](52:48), i=4*j. **NOTE:** This value can only be written when the DPLL is disabled.
24–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DPLL_ACB_*n* field descriptions (continued)

Field	Description
27–31 ACB_0	Control Bits of ACTION_i, reflects ACT_D[i](52:48), i=4*j. **NOTE:** This value can only be written when the DPLL is disabled.

16.10 RAM Region 1A Memory Map and Registers

Bits 31 down to 24 in each RAM region are not implemented and therefore always read as zero (reserved). Other bits which are declared as reserved are not protected against writing. Reserved address regions are not protected against writing.

RAM Region 1a is writeable only for DEN=0 (see DPLL_CTRL_1 register).

The contents of the DPLL RAM Region 1a memory locations are described as follows:

memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
200	Position Action Request for ACTION n (PSA0)	32	R/W	0000_0000h	16.10.1/645
204	Position Action Request for ACTION n (PSA1)	32	R/W	0000_0000h	16.10.1/645
208	Position Action Request for ACTION n (PSA2)	32	R/W	0000_0000h	16.10.1/645
20C	Position Action Request for ACTION n (PSA3)	32	R/W	0000_0000h	16.10.1/645
210	Position Action Request for ACTION n (PSA4)	32	R/W	0000_0000h	16.10.1/645
214	Position Action Request for ACTION n (PSA5)	32	R/W	0000_0000h	16.10.1/645
218	Position Action Request for ACTION n (PSA6)	32	R/W	0000_0000h	16.10.1/645
21C	Position Action Request for ACTION n (PSA7)	32	R/W	0000_0000h	16.10.1/645
220	Position Action Request for ACTION n (PSA8)	32	R/W	0000_0000h	16.10.1/645
224	Position Action Request for ACTION n (PSA9)	32	R/W	0000_0000h	16.10.1/645
228	Position Action Request for ACTION n (PSA10)	32	R/W	0000_0000h	16.10.1/645
22C	Position Action Request for ACTION n (PSA11)	32	R/W	0000_0000h	16.10.1/645
230	Position Action Request for ACTION n (PSA12)	32	R/W	0000_0000h	16.10.1/645
234	Position Action Request for ACTION n (PSA13)	32	R/W	0000_0000h	16.10.1/645
238	Position Action Request for ACTION n (PSA14)	32	R/W	0000_0000h	16.10.1/645
23C	Position Action Request for ACTION n (PSA15)	32	R/W	0000_0000h	16.10.1/645
240	Position Action Request for ACTION n (PSA16)	32	R/W	0000_0000h	16.10.1/645
244	Position Action Request for ACTION n (PSA17)	32	R/W	0000_0000h	16.10.1/645
248	Position Action Request for ACTION n (PSA18)	32	R/W	0000_0000h	16.10.1/645
24C	Position Action Request for ACTION n (PSA19)	32	R/W	0000_0000h	16.10.1/645

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
250	Position Action Request for ACTION n (PSA20)	32	R/W	0000_0000h	16.10.1/645
254	Position Action Request for ACTION n (PSA21)	32	R/W	0000_0000h	16.10.1/645
258	Position Action Request for ACTION n (PSA22)	32	R/W	0000_0000h	16.10.1/645
25C	Position Action Request for ACTION n (PSA23)	32	R/W	0000_0000h	16.10.1/645
260	Position Action Request for ACTION n (PSA24)	32	R/W	0000_0000h	16.10.1/645
264	Position Action Request for ACTION n (PSA25)	32	R/W	0000_0000h	16.10.1/645
268	Position Action Request for ACTION n (PSA26)	32	R/W	0000_0000h	16.10.1/645
26C	Position Action Request for ACTION n (PSA27)	32	R/W	0000_0000h	16.10.1/645
270	Position Action Request for ACTION n (PSA28)	32	R/W	0000_0000h	16.10.1/645
274	Position Action Request for ACTION n (PSA29)	32	R/W	0000_0000h	16.10.1/645
278	Position Action Request for ACTION n (PSA30)	32	R/W	0000_0000h	16.10.1/645
27C	Position Action Request for ACTION n (PSA31)	32	R/W	0000_0000h	16.10.1/645
280	Time to React for Action n (DLA0)	32	R/W	0000_0000h	16.10.2/ 651
284	Time to React for Action n (DLA1)	32	R/W	0000_0000h	16.10.2/ 651
288	Time to React for Action n (DLA2)	32	R/W	0000_0000h	16.10.2/ 651
28C	Time to React for Action n (DLA3)	32	R/W	0000_0000h	16.10.2/ 651
290	Time to React for Action n (DLA4)	32	R/W	0000_0000h	16.10.2/ 651
294	Time to React for Action n (DLA5)	32	R/W	0000_0000h	16.10.2/ 651
298	Time to React for Action n (DLA6)	32	R/W	0000_0000h	16.10.2/ 651
29C	Time to React for Action n (DLA7)	32	R/W	0000_0000h	16.10.2/ 651
2A0	Time to React for Action n (DLA8)	32	R/W	0000_0000h	16.10.2/ 651
2A4	Time to React for Action n (DLA9)	32	R/W	0000_0000h	16.10.2/ 651
2A8	Time to React for Action n (DLA10)	32	R/W	0000_0000h	16.10.2/ 651
2AC	Time to React for Action n (DLA11)	32	R/W	0000_0000h	16.10.2/ 651
2B0	Time to React for Action n (DLA12)	32	R/W	0000_0000h	16.10.2/ 651
2B4	Time to React for Action n (DLA13)	32	R/W	0000_0000h	16.10.2/ 651
2B8	Time to React for Action n (DLA14)	32	R/W	0000_0000h	16.10.2/ 651

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
2BC	Time to React for Action n (DLA15)	32	R/W	0000_0000h	16.10.2/ 651
2C0	Time to React for Action n (DLA16)	32	R/W	0000_0000h	16.10.2/ 651
2C4	Time to React for Action n (DLA17)	32	R/W	0000_0000h	16.10.2/ 651
2C8	Time to React for Action n (DLA18)	32	R/W	0000_0000h	16.10.2/ 651
2CC	Time to React for Action n (DLA19)	32	R/W	0000_0000h	16.10.2/ 651
2D0	Time to React for Action n (DLA20)	32	R/W	0000_0000h	16.10.2/ 651
2D4	Time to React for Action n (DLA21)	32	R/W	0000_0000h	16.10.2/ 651
2D8	Time to React for Action n (DLA22)	32	R/W	0000_0000h	16.10.2/ 651
2DC	Time to React for Action n (DLA23)	32	R/W	0000_0000h	16.10.2/ 651
2E0	Time to React for Action n (DLA24)	32	R/W	0000_0000h	16.10.2/ 651
2E4	Time to React for Action n (DLA25)	32	R/W	0000_0000h	16.10.2/ 651
2E8	Time to React for Action n (DLA26)	32	R/W	0000_0000h	16.10.2/ 651
2EC	Time to React for Action n (DLA27)	32	R/W	0000_0000h	16.10.2/ 651
2F0	Time to React for Action n (DLA28)	32	R/W	0000_0000h	16.10.2/ 651
2F4	Time to React for Action n (DLA29)	32	R/W	0000_0000h	16.10.2/ 651
2F8	Time to React for Action n (DLA30)	32	R/W	0000_0000h	16.10.2/ 651
2FC	Time to React for Action n (DLA31)	32	R/W	0000_0000h	16.10.2/ 651
300	Calculated Number of TRIGGER/STATE Increments to Action n (NA0)	32	R/W	0000_0000h	16.10.3/ 651
304	Calculated Number of TRIGGER/STATE Increments to Action n (NA1)	32	R/W	0000_0000h	16.10.3/ 651
308	Calculated Number of TRIGGER/STATE Increments to Action n (NA2)	32	R/W	0000_0000h	16.10.3/ 651
30C	Calculated Number of TRIGGER/STATE Increments to Action n (NA3)	32	R/W	0000_0000h	16.10.3/ 651
310	Calculated Number of TRIGGER/STATE Increments to Action n (NA4)	32	R/W	0000_0000h	16.10.3/ 651

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
314	Calculated Number of TRIGGER/STATE Increments to Action n (NA5)	32	R/W	0000_0000h	16.10.3/ 651
318	Calculated Number of TRIGGER/STATE Increments to Action n (NA6)	32	R/W	0000_0000h	16.10.3/ 651
31C	Calculated Number of TRIGGER/STATE Increments to Action n (NA7)	32	R/W	0000_0000h	16.10.3/ 651
320	Calculated Number of TRIGGER/STATE Increments to Action n (NA8)	32	R/W	0000_0000h	16.10.3/ 651
324	Calculated Number of TRIGGER/STATE Increments to Action n (NA9)	32	R/W	0000_0000h	16.10.3/ 651
328	Calculated Number of TRIGGER/STATE Increments to Action n (NA10)	32	R/W	0000_0000h	16.10.3/ 651
32C	Calculated Number of TRIGGER/STATE Increments to Action n (NA11)	32	R/W	0000_0000h	16.10.3/ 651
330	Calculated Number of TRIGGER/STATE Increments to Action n (NA12)	32	R/W	0000_0000h	16.10.3/ 651
334	Calculated Number of TRIGGER/STATE Increments to Action n (NA13)	32	R/W	0000_0000h	16.10.3/ 651
338	Calculated Number of TRIGGER/STATE Increments to Action n (NA14)	32	R/W	0000_0000h	16.10.3/ 651
33C	Calculated Number of TRIGGER/STATE Increments to Action n (NA15)	32	R/W	0000_0000h	16.10.3/ 651
340	Calculated Number of TRIGGER/STATE Increments to Action n (NA16)	32	R/W	0000_0000h	16.10.3/ 651
344	Calculated Number of TRIGGER/STATE Increments to Action n (NA17)	32	R/W	0000_0000h	16.10.3/ 651
348	Calculated Number of TRIGGER/STATE Increments to Action n (NA18)	32	R/W	0000_0000h	16.10.3/ 651
34C	Calculated Number of TRIGGER/STATE Increments to Action n (NA19)	32	R/W	0000_0000h	16.10.3/ 651
350	Calculated Number of TRIGGER/STATE Increments to Action n (NA20)	32	R/W	0000_0000h	16.10.3/ 651
354	Calculated Number of TRIGGER/STATE Increments to Action n (NA21)	32	R/W	0000_0000h	16.10.3/ 651
358	Calculated Number of TRIGGER/STATE Increments to Action n (NA22)	32	R/W	0000_0000h	16.10.3/ 651
35C	Calculated Number of TRIGGER/STATE Increments to Action n (NA23)	32	R/W	0000_0000h	16.10.3/ 651
360	Calculated Number of TRIGGER/STATE Increments to Action n (NA24)	32	R/W	0000_0000h	16.10.3/ 651
364	Calculated Number of TRIGGER/STATE Increments to Action n (NA25)	32	R/W	0000_0000h	16.10.3/ 651
368	Calculated Number of TRIGGER/STATE Increments to Action n (NA26)	32	R/W	0000_0000h	16.10.3/ 651

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
36C	Calculated Number of TRIGGER/STATE Increments to Action n (NA27)	32	R/W	0000_0000h	16.10.3/ 651
370	Calculated Number of TRIGGER/STATE Increments to Action n (NA28)	32	R/W	0000_0000h	16.10.3/ 651
374	Calculated Number of TRIGGER/STATE Increments to Action n (NA29)	32	R/W	0000_0000h	16.10.3/ 651
378	Calculated Number of TRIGGER/STATE Increments to Action n (NA30)	32	R/W	0000_0000h	16.10.3/ 651
37C	Calculated Number of TRIGGER/STATE Increments to Action n (NA31)	32	R/W	0000_0000h	16.10.3/ 651
380	Calculated Relative Time to Action n (DTA0)	32	R/W	0000_0000h	16.10.4/ 652
384	Calculated Relative Time to Action n (DTA1)	32	R/W	0000_0000h	16.10.4/ 652
388	Calculated Relative Time to Action n (DTA2)	32	R/W	0000_0000h	16.10.4/ 652
38C	Calculated Relative Time to Action n (DTA3)	32	R/W	0000_0000h	16.10.4/ 652
390	Calculated Relative Time to Action n (DTA4)	32	R/W	0000_0000h	16.10.4/ 652
394	Calculated Relative Time to Action n (DTA5)	32	R/W	0000_0000h	16.10.4/ 652
398	Calculated Relative Time to Action n (DTA6)	32	R/W	0000_0000h	16.10.4/ 652
39C	Calculated Relative Time to Action n (DTA7)	32	R/W	0000_0000h	16.10.4/ 652
3A0	Calculated Relative Time to Action n (DTA8)	32	R/W	0000_0000h	16.10.4/ 652
3A4	Calculated Relative Time to Action n (DTA9)	32	R/W	0000_0000h	16.10.4/ 652
3A8	Calculated Relative Time to Action n (DTA10)	32	R/W	0000_0000h	16.10.4/ 652
3AC	Calculated Relative Time to Action n (DTA11)	32	R/W	0000_0000h	16.10.4/ 652
3B0	Calculated Relative Time to Action n (DTA12)	32	R/W	0000_0000h	16.10.4/ 652
3B4	Calculated Relative Time to Action n (DTA13)	32	R/W	0000_0000h	16.10.4/ 652
3B8	Calculated Relative Time to Action n (DTA14)	32	R/W	0000_0000h	16.10.4/ 652
3BC	Calculated Relative Time to Action n (DTA15)	32	R/W	0000_0000h	16.10.4/ 652
3C0	Calculated Relative Time to Action n (DTA16)	32	R/W	0000_0000h	16.10.4/ 652

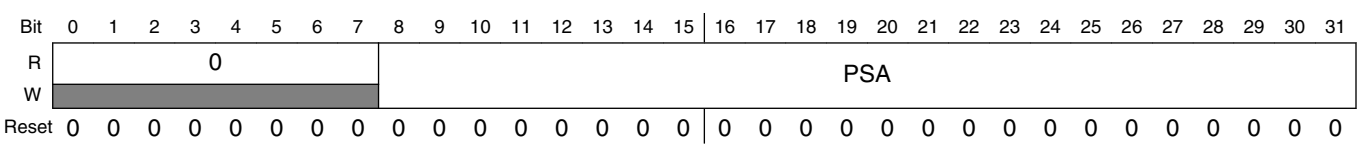
Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3C4	Calculated Relative Time to Action n (DTA17)	32	R/W	0000_0000h	16.10.4/ 652
3C8	Calculated Relative Time to Action n (DTA18)	32	R/W	0000_0000h	16.10.4/ 652
3CC	Calculated Relative Time to Action n (DTA19)	32	R/W	0000_0000h	16.10.4/ 652
3D0	Calculated Relative Time to Action n (DTA20)	32	R/W	0000_0000h	16.10.4/ 652
3D4	Calculated Relative Time to Action n (DTA21)	32	R/W	0000_0000h	16.10.4/ 652
3D8	Calculated Relative Time to Action n (DTA22)	32	R/W	0000_0000h	16.10.4/ 652
3DC	Calculated Relative Time to Action n (DTA23)	32	R/W	0000_0000h	16.10.4/ 652
3E0	Calculated Relative Time to Action n (DTA24)	32	R/W	0000_0000h	16.10.4/ 652
3E4	Calculated Relative Time to Action n (DTA25)	32	R/W	0000_0000h	16.10.4/ 652
3E8	Calculated Relative Time to Action n (DTA26)	32	R/W	0000_0000h	16.10.4/ 652
3EC	Calculated Relative Time to Action n (DTA27)	32	R/W	0000_0000h	16.10.4/ 652
3F0	Calculated Relative Time to Action n (DTA28)	32	R/W	0000_0000h	16.10.4/ 652
3F4	Calculated Relative Time to Action n (DTA29)	32	R/W	0000_0000h	16.10.4/ 652
3F8	Calculated Relative Time to Action n (DTA30)	32	R/W	0000_0000h	16.10.4/ 652
3FC	Calculated Relative Time to Action n (DTA31)	32	R/W	0000_0000h	16.10.4/ 652

16.10.1 Position Action Request for ACTION n (PSA*n*)

Address: 0h base + 200h offset + (4d × i), where i=0d to 31d



PSA*n* field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

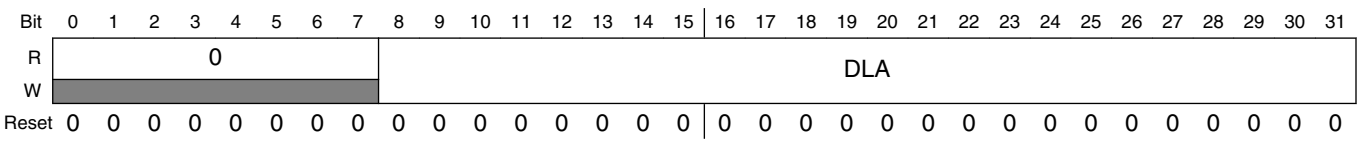
Table continues on the next page...

PSA*n* field descriptions (continued)

Field	Description
8–31 PSA	Position information of a desired action (i=0…23). **NOTE:** This value can be written only when the DPLL is disabled.

16.10.2 Time to React for Action n (DLA*n*)

Address: 0h base + 280h offset + (4d × i), where i=0d to 31d

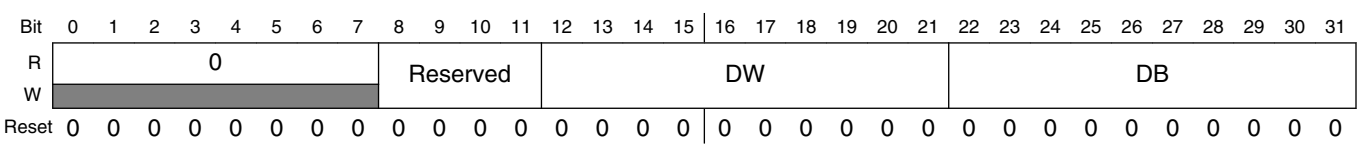


DLA*n* field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 DLA	Time to react before the corresponding position value of a desired action is reached. In the case of LOW_RES=1 (see chapter {REF:DPLL_6906}), this delay value must be also given as low resolution value. **NOTE:** This value can be written only when the DPLL is disabled.

16.10.3 Calculated Number of TRIGGER/STATE Increments to Action n (NA*n*)

Address: 0h base + 300h offset + (4d × i), where i=0d to 31d



NA*n* field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–11 Reserved	Reserved. This field is reserved.

Table continues on the next page...

NA*n* field descriptions (continued)

Field	Description
12–21 DW	Number of events to Action_i (integer part). **NOTE:** Use the maximum value for DW=0x3FF in the case of a calculated value which exceeds the representable value. **NOTE:** This value can only be written when the DPLL is disabled.
22–31 DB	Number of events to Action_i (fractional part). **NOTE:** This value can only be written when the DPLL is disabled.

16.10.4 Calculated Relative Time to Action n (DTA*n*)

Address: 0h base + 380h offset + (4d × i), where i=0d to 31d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								DTA																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DTA*n* field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 DTA	Calculated relative time to ACTION_i. **NOTE:** This value can only be written when the DPLL is disabled.

16.11 RAM Region 1B Memory Map and Registers

Bits 31 down to 24 in each RAM region are not implemented and therefore always read as zero (reserved). Other bits which are declared as reserved are not protected against writing. Reserved address regions are not protected against writing.

The contents of the DPLL RAM Region 1b memory locations are described as follows:

memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400	Actual TRIGGER Time Stamp Value (TS_T)	32	R/W	0000_0000h	16.11.1/653
404	Previous TRIGGER Time Stamp Value (TS_T_OLD)	32	R/W	0000_0000h	16.11.2/ 657
408	Actual TRIGGER Filter Value (FTV_T)	32	R/W	0000_0000h	16.11.3/ 657
410	Actual STATE Time Stamp Register (TS_S)	32	R/W	0000_0000h	16.11.4/ 658
414	Previous STATE Time Stamp Register (TS_S_OLD)	32	R/W	0000_0000h	16.11.5/ 658
418	Actual STATE Filter Value (FTV_S)	32	R/W	0000_0000h	16.11.6/ 659
41C	TRIGGER Hold Time Minimum Value (THMI)	32	R/W	0000_0000h	16.11.7/ 659
420	TRIGGER Hold Time Maximum Value (THMA)	32	R/W	0000_0000h	16.11.8/ 660
424	Measured TRIGGER Hold Time Value (THVAL)	32	R/W	0000_0000h	16.11.9/ 660
428	Time Out Value of Active TRIGGER Slope (for missing TRIGGER generation) (TOV)	32	R/W	0000_0000h	16.11.10/ 661
434	Time Out Value of Actvie STATE Slope (for missing STATE generation) (TOV_S)	32	R/W	0000_0000h	16.11.11/ 662
438	Calculated ADD_IN Value for SUB_INC1 Generation (ADD_IN_CAL1)	32	R/W	0000_0438h	16.11.12/ 663
43C	Calculated ADD_IN Value for SUB_INC2 Generation (ADD_IN_CAL2)	32	R/W	0000_0000h	16.11.13/ 663
440	Missing Pulses to be Added or Subtracted Directly (MPVAL1)	32	R/W	0000_0000h	16.11.14/ 664
444	Missing Pulses to be Added or Subtracted Directly (MPVAL2)	32	R/W	0000_0000h	16.11.15/ 664
448	Target Number of Pulses to be Sent in Normal Mode (NMB_T_TAR)	32	R/W	0000_0000h	16.11.16/ 665
44C	Last but one Target Number of Pulses to be sent in Normal Mode (NMB_T_TAR_OLD)	32	R/W	0000_0000h	16.11.17/ 666
450	Target Number Of Pulses to be Sent In Emergency Mode (NMB_S_TAR)	32	R/W	0000_0000h	16.11.18/ 666
454	Last but one Target Number Of Pulses to be Sent In Emergency Mode (NMB_S_TAR_OLD)	32	R/W	0000_0000h	16.11.19/ 667
460	Reciprocal Value of the Expected Increment Duration of TRIGGER (RCDT_TX)	32	R/W	0000_0000h	16.11.20/ 667
464	Reciprocal Value of the Expected Increment Duration of STATE (RCDT_SX)	32	R/W	0000_0000h	16.11.21/ 668
468	Reciprocal Value of the Expected Nominal Increment Duration of TRIGGER (RCDT_TX_NOM)	32	R/W	0000_0000h	16.11.22/ 668
46C	Reciprocal Value of the Expected Nominal Increment Duration of STATE (RCDT_SX_NOM)	32	R/W	0000_0000h	16.11.23/ 669

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
470	Reciprocal Value of last Increment of TRIGGER (RDT_T_ACT)	32	R/W	0000_0000h	16.11.24/ 669
474	Reciprocal Value of the Last Increment of STATE (RDT_S_ACT)	32	R/W	0000_0000h	16.11.25/ 670
478	Duration of the Last TRIGGER Increment (DT_T_ACT)	32	R/W	0000_0000h	16.11.26/ 670
47C	Duration of the Last STATE Increment (DT_S_ACT)	32	R/W	0000_0000h	16.11.27/ 671
480	Difference of Prediction to Actual Value of the Last TRIGGER Increment (EDT_T)	32	R/W	0000_0000h	16.11.28/ 671
484	Weighted Difference of Prediction Errors of TRIGGER (MEDT_T)	32	R/W	0000_0000h	16.11.29/ 671
488	Difference of Prediction to Actual Value of the Last STATE Increment (EDT_S)	32	R/W	0000_0000h	16.11.30/ 672
48C	Weighted Difference of Prediction Errors of STATE (MEDT_S)	32	R/W	0000_0000h	16.11.31/ 672
490	Prediction of the Actual TRIGGER Increment Duration (CDT_TX)	32	R/W	0000_0000h	16.11.32/ 673
494	Prediction of the Actual STATE Increment Duration (CDT_SX)	32	R/W	0000_0000h	16.11.33/ 673
498	Prediction of the Nominal TRIGGER Increment Duration (CDT_TX_NOM)	32	R/W	0000_0000h	16.11.34/ 674
49C	Prediction of the Nominal STATE Increment Duration (CDT_SX_NOM)	32	R/W	0000_0000h	16.11.35/ 674
4A0	TRIGGER Locking Range (TLR)	32	R/W	0000_0000h	16.11.36/ 674
4A4	STATE Locking Range (SLR)	32	R/W	0000_0000h	16.11.37/ 675
500	Projected Increment Sum Relations for Action n (PDT0)	32	R/W	0000_0000h	16.11.38/ 675
504	Projected Increment Sum Relations for Action n (PDT1)	32	R/W	0000_0000h	16.11.38/ 675
508	Projected Increment Sum Relations for Action n (PDT2)	32	R/W	0000_0000h	16.11.38/ 675
50C	Projected Increment Sum Relations for Action n (PDT3)	32	R/W	0000_0000h	16.11.38/ 675
510	Projected Increment Sum Relations for Action n (PDT4)	32	R/W	0000_0000h	16.11.38/ 675
514	Projected Increment Sum Relations for Action n (PDT5)	32	R/W	0000_0000h	16.11.38/ 675
518	Projected Increment Sum Relations for Action n (PDT6)	32	R/W	0000_0000h	16.11.38/ 675
51C	Projected Increment Sum Relations for Action n (PDT7)	32	R/W	0000_0000h	16.11.38/ 675

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
520	Projected Increment Sum Relations for Action n (PDT8)	32	R/W	0000_0000h	16.11.38/ 675
524	Projected Increment Sum Relations for Action n (PDT9)	32	R/W	0000_0000h	16.11.38/ 675
528	Projected Increment Sum Relations for Action n (PDT10)	32	R/W	0000_0000h	16.11.38/ 675
52C	Projected Increment Sum Relations for Action n (PDT11)	32	R/W	0000_0000h	16.11.38/ 675
530	Projected Increment Sum Relations for Action n (PDT12)	32	R/W	0000_0000h	16.11.38/ 675
534	Projected Increment Sum Relations for Action n (PDT13)	32	R/W	0000_0000h	16.11.38/ 675
538	Projected Increment Sum Relations for Action n (PDT14)	32	R/W	0000_0000h	16.11.38/ 675
53C	Projected Increment Sum Relations for Action n (PDT15)	32	R/W	0000_0000h	16.11.38/ 675
540	Projected Increment Sum Relations for Action n (PDT16)	32	R/W	0000_0000h	16.11.38/ 675
544	Projected Increment Sum Relations for Action n (PDT17)	32	R/W	0000_0000h	16.11.38/ 675
548	Projected Increment Sum Relations for Action n (PDT18)	32	R/W	0000_0000h	16.11.38/ 675
54C	Projected Increment Sum Relations for Action n (PDT19)	32	R/W	0000_0000h	16.11.38/ 675
550	Projected Increment Sum Relations for Action n (PDT20)	32	R/W	0000_0000h	16.11.38/ 675
554	Projected Increment Sum Relations for Action n (PDT21)	32	R/W	0000_0000h	16.11.38/ 675
558	Projected Increment Sum Relations for Action n (PDT22)	32	R/W	0000_0000h	16.11.38/ 675
55C	Projected Increment Sum Relations for Action n (PDT23)	32	R/W	0000_0000h	16.11.38/ 675
560	Projected Increment Sum Relations for Action n (PDT24)	32	R/W	0000_0000h	16.11.38/ 675
564	Projected Increment Sum Relations for Action n (PDT25)	32	R/W	0000_0000h	16.11.38/ 675
568	Projected Increment Sum Relations for Action n (PDT26)	32	R/W	0000_0000h	16.11.38/ 675
56C	Projected Increment Sum Relations for Action n (PDT27)	32	R/W	0000_0000h	16.11.38/ 675
570	Projected Increment Sum Relations for Action n (PDT28)	32	R/W	0000_0000h	16.11.38/ 675
574	Projected Increment Sum Relations for Action n (PDT29)	32	R/W	0000_0000h	16.11.38/ 675

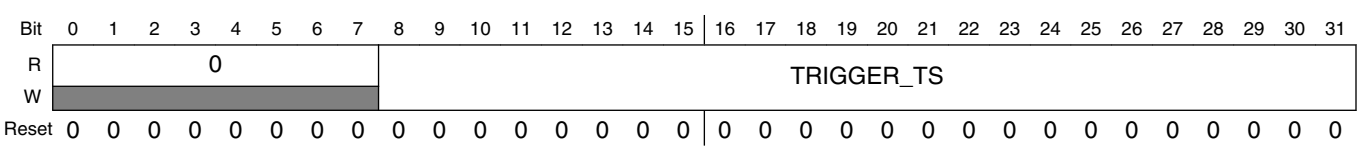
Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
578	Projected Increment Sum Relations for Action n (PDT30)	32	R/W	0000_0000h	16.11.38/ 675
57C	Projected Increment Sum Relations for Action n (PDT31)	32	R/W	0000_0000h	16.11.38/ 675
5C0	Calculated Number of Sub-Pulses between two STATE Events for SMC=0 (MLS1)	32	R/W	0000_0000h	16.11.39/ 676
5C4	Calculated Number of Sub-pulses between two STATE Events for SMC=1 and RMO=1 (MLS2)	32	R/W	0000_0000h	16.11.40/ 676
5C8	Counter for number of SUB_INC1 pulses (CNT_NUM1)	32	R/W	0000_0000h	16.11.41/ 677
5CC	Counter for number of SUB_INC2 pulses (CNT_NUM2)	32	R/W	0000_0000h	16.11.42/ 677
5D0	Plausibility Value of Next TRIGGER Slope (PVT)	32	R/W	0000_0000h	16.11.43/ 678
5E0	Actual Calculated Position Stamp of TRIGGER (PSTC)	32	R/W	0000_0000h	16.11.44/ 678
5E4	Actual Calculated Position Stamp of STATE (PSSC)	32	R/W	0000_0000h	16.11.45/ 679
5E8	Measured Position Stamp at Last TRIGGER Input (PSTM)	32	R/W	0000_0000h	16.11.46/ 679
5EC	Measured Position Stamp at Last But One TRIGGER Input (PSTM_OLD)	32	R/W	0000_0000h	16.11.47/ 680
5F0	Measured Position Stamp at Last STATE Input (PSSM)	32	R/W	0000_0000h	16.11.48/ 680
5F4	Measured Position Stamp at Last but One STATE Input (PSSM_OLD)	32	R/W	0000_0000h	16.11.49/ 681
5F8	Number of Pulses to be Sent in Normal Mode (NMB_T)	32	R/W	0000_0000h	16.11.50/ 681
5FC	Number of Pulses to be Sent in Emergency Mode (NMB_S)	32	R/W	0000_0000h	16.11.51/ 682

16.11.1 Actual TRIGGER Time Stamp Value (TS_T)

Address: 0h base + 400h offset = 400h



TS_T field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

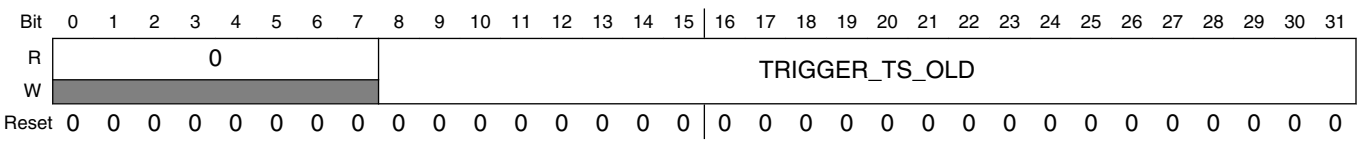
Table continues on the next page...

TS_T field descriptions (continued)

Field	Description
8–31 TRIGGER_TS	Time stamp value of the last valid TRIGGER input. measured TRIGGER time stamp. **NOTE:** The LSB address is determined using the SWON_T value in the OSW register (see chapter 16.11.8).

16.11.2 Previous TRIGGER Time Stamp Value (TS_T_OLD)

Address: 0h base + 404h offset = 404h

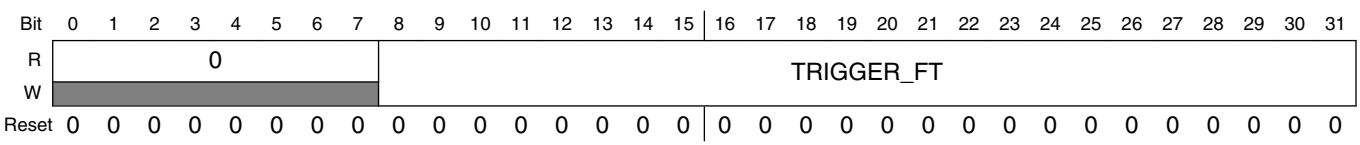


TS_T_OLD field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 TRIGGER_TS_ OLD	Time stamp value of the last but one valid TRIGGER input (previous measured TRIGGER time stamp). **NOTE:** The LSB address is determined using the SWON_T value in the OSW register (see chapter 16.11.8).

16.11.3 Actual TRIGGER Filter Value (FTV_T)

Address: 0h base + 408h offset = 408h



FTV_T field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 TRIGGER_FT	Filter value of the last valid TRIGGER input (transmitted filter value). **NOTE:** The LSB address is determined using the SWON_T value in the OSW register (see chapter 16.11.8).

16.11.4 Actual STATE Time Stamp Register (TS_S)

Address: 0h base + 410h offset = 410h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								STATE_TS																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TS_S field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 STATE_TS	Time stamp value of the last valid STATE input. **NOTE:** The LSB address is determined using the SWON_S value in the OSW register (see chapter 16.11.8).

16.11.5 Previous STATE Time Stamp Register (TS_S_OLD)

Address: 0h base + 414h offset = 414h

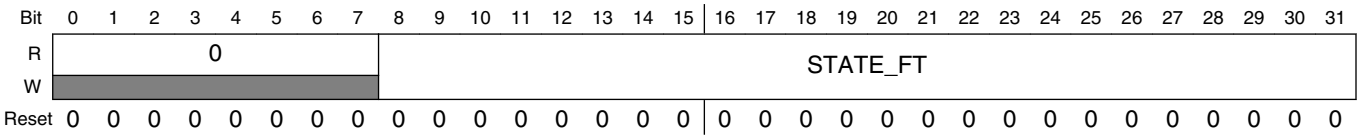
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								TS_S_OLD																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TS_S_OLD field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 TS_S_OLD	Time stamp value of the last valid STATE input. **NOTE:** The LSB address is determined using the SWON_S value in the OSW register (see chapter 16.11.8).

16.11.6 Actual STATE Filter Value (FTV_S)

Address: 0h base + 418h offset = 418h

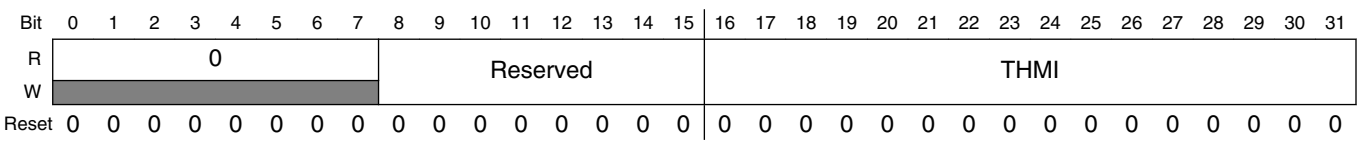


FTV_S field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 STATE_FT	Filter value of the last valid STATE input transmitted filter value. **NOTE:** The LSB address is determined using the SWON_S value in the OSW register (see chapter 16.11.8).

16.11.7 TRIGGER Hold Time Minimum Value (THMI)

Address: 0h base + 41Ch offset = 41Ch

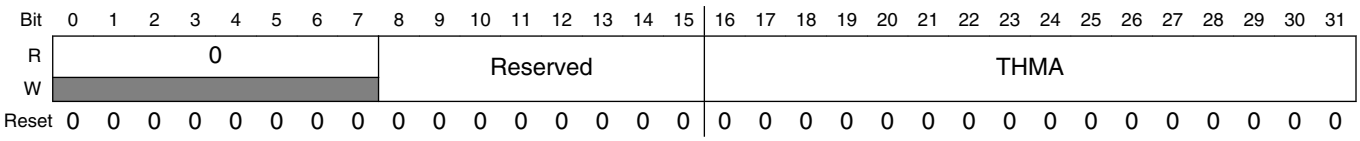


THMI field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–15 Reserved	Reserved. This field is reserved. Must be written to zero, then reads as zero.
16–31 THMI	Minimum time between active and inactive TRIGGER slope (uint16). The time value corresponds to the time stamp clock counts, that is, the clock selected for the TBU_CH0_BASE (see TBU_CH0_CTRL register). Set min value, then generate the TINI interrupt in the case of a violation for THMI>0. **NOTE:** Typical retention time values after a valid slope can be e.g. between 45 µs (forwards) and 90 µs (backwards). When THMI is zero, consider always a THMI violation (forwards).

16.11.8 TRIGGER Hold Time Maximum Value (THMA)

Address: 0h base + 420h offset = 420h

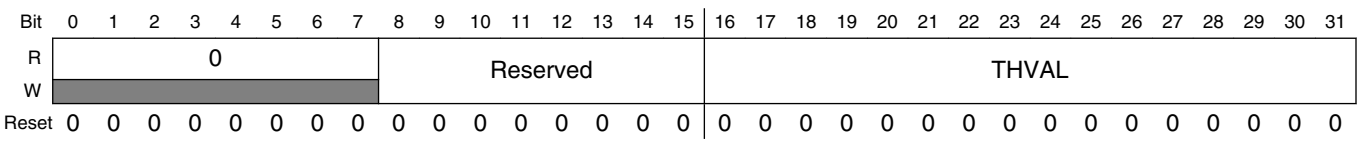


THMA field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–15 Reserved	Reserved. This field is reserved. Must be written to zero, then reads as zero.
16–31 THMA	Maximum time tbeween active and inactive TRIGGER slope (uint16). The time value corresponds to the time stamp clock counts, that is, the clock selected for THMA>0. The TBU_CH0_BASE (see TBU_CH0_CTRL register) maximum value to be set; generate the TAX interrupt in the case of a violation for THMA>0.

16.11.9 Measured TRIGGER Hold Time Value (THVAL)

Address: 0h base + 424h offset = 424h



THVAL field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–15 Reserved	Reserved. This field is reserved. Must be written to zero, then reads as zero.
16–31 THVAL	Measured time value from the last valid slope to the next inactive TRIGGER slope in time stamp clock counts (uint16). That is, the clock selected for the TBU_CH0_BASE (uint16); measured value.

16.11.10 Time Out Value of Active TRIGGER Slope (for missing TRIGGER generation) (TOV)

TOV[16:31] is to be multiplied by the duration of the last increment and divided by 1024 in order to get the timeout time value.

NOTE

For the case of LOW_RES=1 (see DPLL_STATUS register), consider for the calculation of the time out value the following cases:

- For LOW_RES=1 and DPLL_CTRL_1/TS0_HRT=1, multiply the TBU_TS0 value by 8.
- For LOW_RES=1 and DPLL_CTRL_1/TS0_HRT=0, multiply the TBU_TS0 value by 8 and multiply the estimated time point value (using TS_T, dt_t_actual and TOV) by 8.
- LOW_RES=0 and DPLL_CTRL_1/TS0_HRT=0, use TBU_TS0 and the estimated time point value unchanged.

Address: 0h base + 428h offset = 428h

Bit	0–7	8–15	16–21	22–31
R	0	Reserved	DW	DB
W				
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0

TOV field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–15 Reserved	Reserved. This field is reserved. Must be written to zero, then reads as zero.
16–21 DW	Decision value (integer part) for missing TRIGGER interrupt. TOV(15:0) is to be multiplied with the duration of the last increment and divided by 1024 in order to get the timeout time value for a missing TRIGGER event. **NOTE:** For the case of LOW_RES=1 (see DPLL_STATUS register) consider for the calculation of the time out value the following cases for: • LOW_RES=1 and DPLL_CTRL_1/TS0_HRT=1, multiply the TBU_TS0 value by 8. • LOW_RES=1 and DPLL_CTRL_1/TS0_HRT=0, multiply the TBU_TS0 value by 8 and multiply the estimated time point value (using TS_T, dt_t_ACT and TOV) by 8. • LOW_RES=0 and DPLL_CTRL_1/TS0_HRT=0, use TBU_TS0 and the estimated time point value unchanged.
22–31 DB	Decision value (fractional part) for missing TRIGGER interrupt.

16.11.11 Time Out Value of Actvie STATE Slope (for missing STATE generation) (TOV_S)

TOV_S[16:31] is to be multiplied with the duration of the last increment and divided by 1024 in order to get the timeout time value.

NOTE

For the case of LOW_RES=1 (see DPLL_STATUS register), consider for the calculation of the time out value the following cases:

- For LOW_RES=1 and DPLL_CTRL_1/TS0_HRT=1, multiply the TBU_TS0 value by 8.
- For LOW_RES=1 and DPLL_CTRL_1/TS0_HRT=0, multiply the TBU_TS0 value by 8 and multiply the estimated time point value (using TS_T, dt_t_actual and TOV) by 8.
- LOW_RES=0 and DPLL_CTRL_1/TS0_HRT=0, use TBU_TS0 and the estimated time point value unchanged.

Address: 0h base + 434h offset = 434h

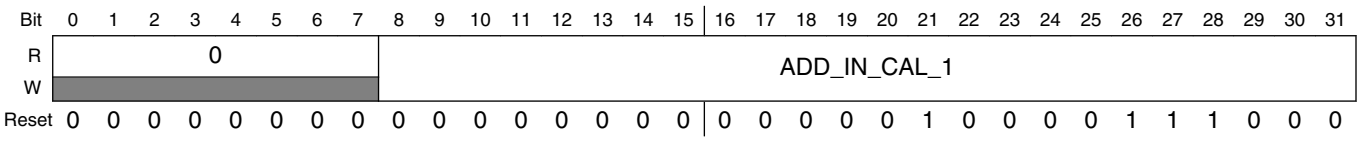
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								Reserved								DW						DB									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOV_S field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–15 Reserved	Reserved. This field is reserved. Must be written to zero, then reads as zero.
16–21 DW	Decision value (integer part) for missing STATE interrupt. TOV_S (15:0) is to be multiplied with the duration of the last increment and divided by 1024 in order to get the timeout time value for a missing STATE event. **NOTE:** For the case of LOW_RES=1 (see DPLL_STATUS register) consider for the calculation of the time out value the following cases: • LOW_RES=1 and DPLL_CTRL_1/TS0_HRS=1, multiply the TBU_TS0 value by 8. • LOW_RES=1 and DPLL_CTRL_1/TS0_HRS=0, multiply the TBU_TS0 value by 8 and multiply the estimated time point value (using TS_T, dt_s_ACT and TOV_S) by 8. • LOW_RES=0 and DPLL_CTRL_1/TS0_HRS=0, use TBU_TS0 and the estimated time point value unchanged.
22–31 DB	Decision value (fractional part) for missing STATE interrupt.

16.11.12 Calculated ADD_IN Value for SUB_INC1 Generation (ADD_IN_CAL1)

Address: 0h base + 438h offset = 438h

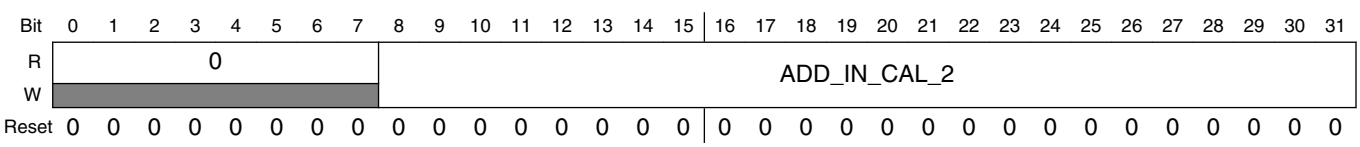


ADD_IN_CAL1 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 ADD_IN_CAL_1	Calculated input value for SUB_INC1 generation, calculated by the DPLL. The update of the ADD_IN value by the new calculated value ADD_IN_CAL1 is suppressed for one increment when an unexpected missing TRIGGER (SMC=1 or RMO=0) or an unexpected STATE (RMO=1 and SMC=0) is detected.

16.11.13 Calculated ADD_IN Value for SUB_INC2 Generation (ADD_IN_CAL2)

Address: 0h base + 43Ch offset = 43Ch



ADD_IN_CAL2 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 ADD_IN_CAL_2	Input value for SUB_INC2 generation, calculated by the DPLL for SMC=RMO=1. The update of the ADD_IN value by the calculated value ADD_IN_CAL2 is suppressed for one increment when an unexpected missing STATE (RMO=SMC=1) is detected.

16.11.14 Missing Pulses to be Added or Subtracted Directly (MPVAL1)

NOTE

Do not provide negative values which exceed the amount of NT *(MLT + 1) or MLS1 respectively. When considering negative PD values, the sum of both should not exceed the amount of NT * (MLT + 1) or MLS1, respectively.

Address: 0h base + 440h offset = 440h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								SIX1								MPVAL1															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MPVAL1 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–15 SIX1	Sign extention for MPVAL1. 0x00 MPVAL1 is a positive number. 0xFF MPVAL1 is a negative number. **NOTE:** All bits must be written to either all zeros or all ones.
16–31 MPVAL1	Missing pulses for direct correction of SUB_INC1 pulses by the CPU (sint16). Used only for RMO=0 or SMC=1 for the case PCM1=1. Add MPVAL1 once to INC_CNT1 and reset PCM1 after applying once. **NOTE:** Do not provide negative values which exceed the amount of MLT or MLS1, respectively.

16.11.15 Missing Pulses to be Added or Subtracted Directly (MPVAL2)

Address: 0h base + 444h offset = 444h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								SIX2								MPVAL2															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MPVAL2 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–15 SIX2	Sign extention for MPVAL2. 0x00 MPVAL2 is a positive number. 0xFF MPVAL2 is a negative number. **NOTE:** All bits must be written to either all zeros or all ones.
16–31 MPVAL2	Missing pulses for direct correction of SUB_INC2 pulses by the CPU (sint16). Used only for SMC=RMO=1 for the case PCM2=1. Add MPVAL2 once to INC_CNT2 and reset PCM2 after applying once. **NOTE:** Do not provide negative values which exceed the amount of MLS2. When considering negative PD_S values, the sum of both should not exceed the amount of MLS2.

16.11.16 Target Number of Pulses to be Sent in Normal Mode (NMB_T_TAR)

Address: 0h base + 448h offset = 448h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								Reserved								NMB_T_TAR															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NMB_T_TAR field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–15 Reserved	Reserved. This field is reserved. Must be written to zero, then reads as zero.
16–31 NMB_T_TAR	Target Number of pulses for TRIGGER. Calculated target number of pulses in normal mode for the current TRIGGER increment without missing pulses. **NOTE:** The LSB address is determined using the SWON_S value in the OSW register (see chapter 16.11.8).

16.11.17 Last but one Target Number of Pulses to be sent in Normal Mode (NMB_T_TAR_OLD)

Address: 0h base + 44Ch offset = 44Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								Reserved								NMB_T_TAR_OLD															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NMB_T_TAR_OLD field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–15 Reserved	Reserved. This field is reserved. Must be written to zero, then reads as zero.
16–31 NMB_T_TAR_OLD	Target Number of pulses for TRIGGER. Calculated number of pulses in normal mode for the current TRIGGER increment without missing pulses. **NOTE:** The LSB address is determined using the SWON_S value in the OSW register (see chapter 16.11.8).

16.11.18 Target Number Of Pulses to be Sent In Emergency Mode (NMB_S_TAR)

Must be written to zero, then reads as zero.

Address: 0h base + 450h offset = 450h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								Reserved				NMB_S_TAR																			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NMB_S_TAR field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–11 Reserved	Reserved. This field is reserved. Must be written to zero, then reads as zero.

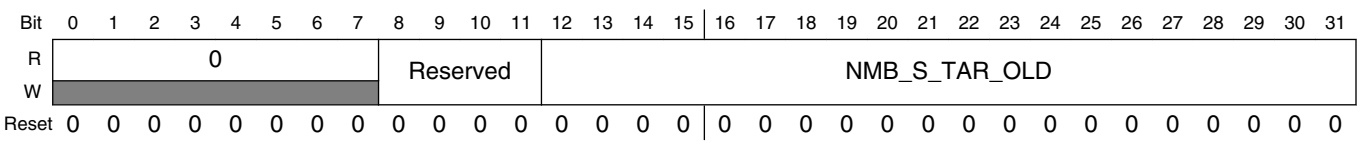
Table continues on the next page...

NMB_S_TAR field descriptions (continued)

Field	Description
12–31 NMB_S_TAR	Target Number of pulses for STATE. Calculated target number of pulses in emergency mode for the current STATE increment without missing pulses. **NOTE:** The LSB address is determined using the SWON_S value in the OSW register (see chapter 16.11.8).

16.11.19 Last but one Target Number Of Pulses to be Sent In Emergency Mode (NMB_S_TAR_OLD)

Address: 0h base + 454h offset = 454h

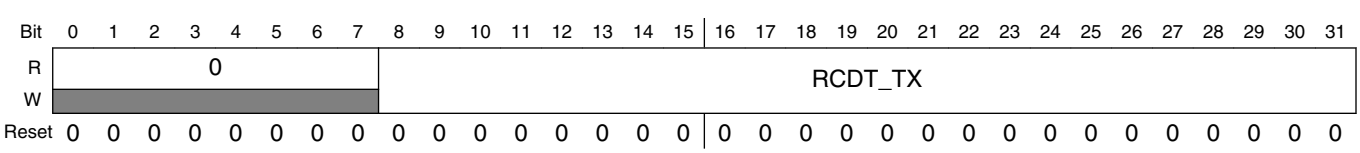


NMB_S_TAR_OLD field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–11 Reserved	Reserved. This field is reserved. Must be written to zero, then reads as zero.
12–31 NMB_S_TAR_ OLD	Target Number of pulses for STATE. Calculated target number of pulses in emergency mode for the current STATE increment without missing pulses. **NOTE:** The LSB address is determined using the SWON_S value in the OSW register (see chapter 16.11.8).

16.11.20 Reciprocal Value of the Expected Increment Duration of TRIGGER (RCDT_TX)

Address: 0h base + 460h offset = 460h

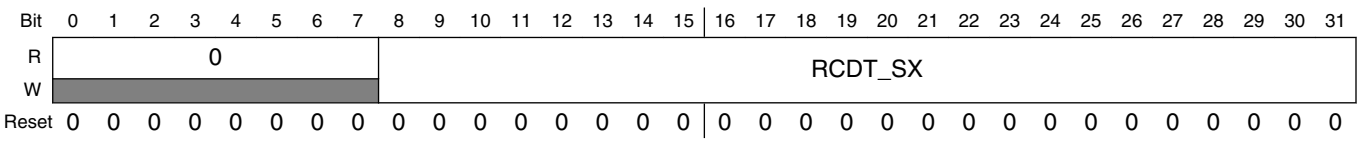


RCDT_TX field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 RCDT_TX	Reciprocal value of expected increment duration * 2^{32} while only the lower 24 bits are used. Calculated value. When an overflow occurs in calculation, the value is set to 0xFFFFFF.

16.11.21 Reciprocal Value of the Expected Increment Duration of STATE (RCDT_SX)

Address: 0h base + 464h offset = 464h

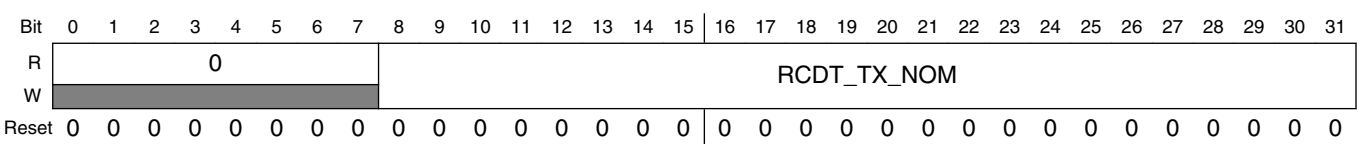


RCDT_SX field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 RCDT_SX	Reciprocal value of expected increment duration * 2^{32} while only the lower 24 bits are used. Calculated value. When an overflow occurs in calculation, the value is set to 0xFFFFFF.

16.11.22 Reciprocal Value of the Expected Nominal Increment Duration of TRIGGER (RCDT_TX_NOM)

Address: 0h base + 468h offset = 468h



RCDT_TX_NOM field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 RCDT_TX_NOM	Reciprocal value of normal increment duration * 2^{32} while only the lower 24 bits are used. Calculated value. When an overflow occurs in calculation, the value is set to 0xFFFFFF. **NOTE:** RCDT_TX_nom and RCDT_SX_nom are calculated by the values RCDT_TX and RCDT_SX to be multiplied with SYN_T or SYN_S, respectively.

16.11.23 Reciprocal Value of the Expected Nominal Increment Duration of STATE (RCDT_SX_NOM)

Address: 0h base + 46Ch offset = 46Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								RCDT_SX_NOM																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RCDT_SX_NOM field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 RCDT_SX_NOM	Reciprocal value of normal increment duration * 2^{32} while only the lower 24 bits are used. Calculated value. When an overflow occurs in calculation, the value is set to 0xFFFFFF. **NOTE:** RCDT_TX_nom and RCDT_SX_nom are calculated by the values RCDT_TX and RCDT_SX to be multiplied with SYN_T or SYN_S, respectively.

16.11.24 Reciprocal Value of last Increment of TRIGGER (RDT_T_ACT)

Address: 0h base + 470h offset = 470h

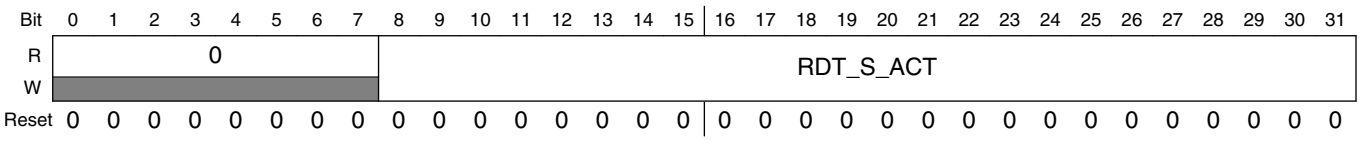
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								RDT_T_ACT																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDT_T_ACT field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 RDT_T_ACT	Reciprocal value of last TRIGGER increment * 2^{32} . Only the lower 24 bits are used; the LSB is rounded up when the next truncated bit is 1. Calculated value. When an overflow occurs in calculation, the value is set to 0xFFFFFF and the CRO bit in the DPLL_STATUS register is set (see chapter 16.11.6).

16.11.25 Reciprocal Value of the Last Increment of STATE (RDT_S_ACT)

Address: 0h base + 474h offset = 474h

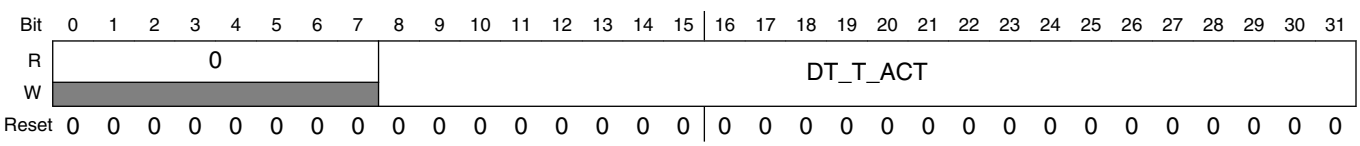


RDT_S_ACT field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 RDT_S_ACT	Reciprocal value of last STATE increment * 2^{32} . Only the lower 24 bits are used; the LSB is rounded up when the next truncated bit is 1. Calculated value. When an overflow occurs in calculation, the value is set to 0xFFFFFF and the CRO bit in the DPLL_STATUS register is set (see chapter 16.11.6).

16.11.26 Duration of the Last TRIGGER Increment (DT_T_ACT)

Address: 0h base + 478h offset = 478h



DT_T_ACT field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 DT_T_ACT	Calculated duration of the last TRIGGER increment. Value will be written into the corresponding RAM field, when all calculations for the considered increment are done and APT is valid.

16.11.27 Duration of the Last STATE Increment (DT_S_ACT)

Address: 0h base + 47Ch offset = 47Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								DT_S_ACT																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DT_S_ACT field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 DT_S_ACT	Calculated duration of the last STATE increment. Value will be written into the corresponding RAM field when all calculations for the considered increment are done and APS is valid.

16.11.28 Difference of Prediction to Actual Value of the Last TRIGGER Increment (EDT_T)

Address: 0h base + 480h offset = 480h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								EDT_T																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EDT_T field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 EDT_T	Signed difference between actual value and a simple prediction of the last TRIGGER increment: sint24 calculated error value..

16.11.29 Weighted Difference of Prediction Errors of TRIGGER (MEDT_T)

Address: 0h base + 484h offset = 484h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								MEDT_T																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MEDT_T field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 MEDT_T	Signed middle weighted difference between actual value and prediction of the last TRIGGER increments: sint24. Calculated medium error value. The value is calculated only after synchronization (SYT=1) and the update is suppressed for one increment when an unexpected missing TRIGGER is detected.

16.11.30 Difference of Prediction to Actual Value of the Last STATE Increment (EDT_S)

Address: 0h base + 488h offset = 488h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								EDT_S																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EDT_S field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 EDT_S	Signed difference between actual value and prediction of the last STATE increment: sint24. Calculated error value.

16.11.31 Weighted Difference of Prediction Errors of STATE (MEDT_S)

Address: 0h base + 48Ch offset = 48Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								MEDT_S																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MEDT_S field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

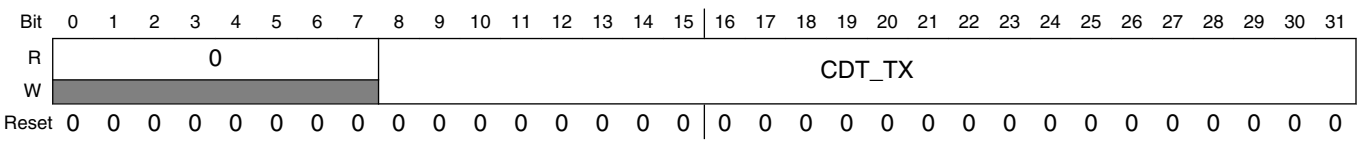
Table continues on the next page...

MEDT_S field descriptions (continued)

Field	Description
8–31 MEDT_S	Signed middle weighted difference between actual value and prediction of the last STATE increments: sint24. Calculated medium error value. The value is calculated only after synchronization (SYS=1) and the update is suppressed for one increment when an unexpected missing STATE is detected.

16.11.32 Prediction of the Actual TRIGGER Increment Duration (CDT_TX)

Address: 0h base + 490h offset = 490h

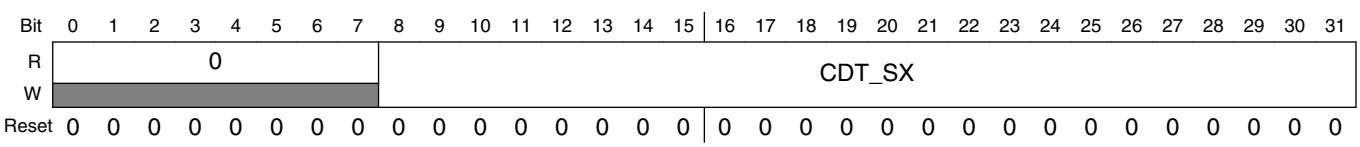


CDT_TX field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 CDT_TX	Calculated duration value of the current TRIGGER increment.

16.11.33 Prediction of the Actual STATE Increment Duration (CDT_SX)

Address: 0h base + 494h offset = 494h

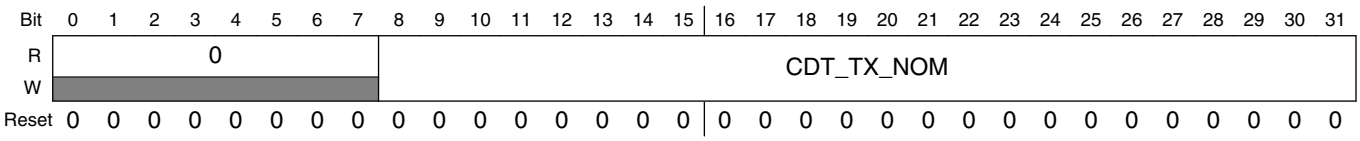


CDT_SX field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 CDT_SX	Calculated duration value of the current STATE increment.

16.11.34 Prediction of the Nominal TRIGGER Increment Duration (CDT_TX_NOM)

Address: 0h base + 498h offset = 498h

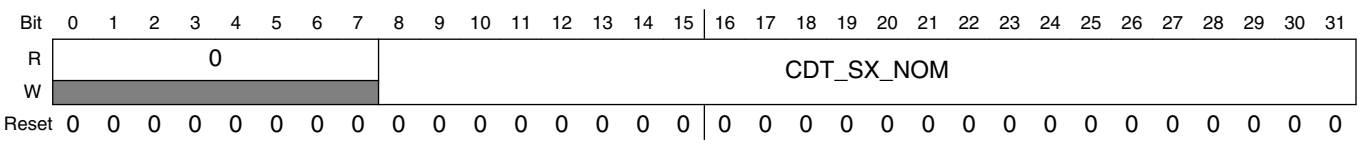


CDT_TX_NOM field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 CDT_TX_NOM	Calculated duration value of the current nominal TRIGGER event.

16.11.35 Prediction of the Nominal STATE Increment Duration (CDT_SX_NOM)

Address: 0h base + 49Ch offset = 49Ch

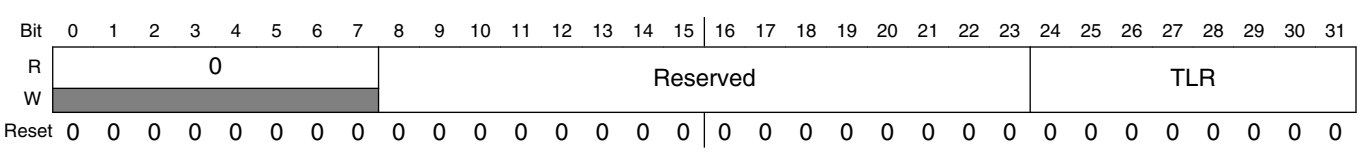


CDT_SX_NOM field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 CDT_SX_NOM	Calculated duration value of the current nominal STATE event.

16.11.36 TRIGGER Locking Range (TLR)

Address: 0h base + 4A0h offset = 4A0h

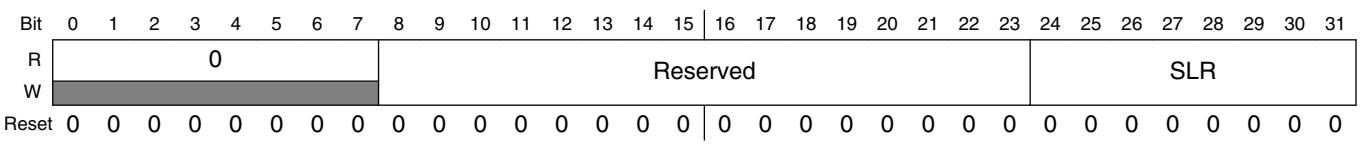


TLR field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–23 Reserved	Reserved. This field is reserved. Must be written to zero, then reads as zero.
24–31 TLR	Value is to be multiplied with the last nominal TRIGGER duration. In order to get the range for the next TRIGGER event without setting TOR in the DPLL_STATUS register, multiply value with the last nominal increment duration and check violation. When TLR=0, don't perform the check.

16.11.37 STATE Locking Range (SLR)

Address: 0h base + 4A4h offset = 4A4h

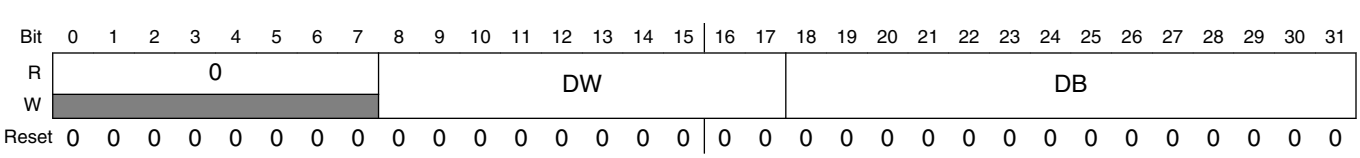


SLR field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–23 Reserved	Reserved. This field is reserved. Must be written to zero, then reads as zero.
24–31 SLR	Value is to be multiplied with the last nominal STATE duration. In order to get the range for the next STATE event without setting SOR in the DPLL_STATUS register, multiply value with the last nominal increment duration and check violation. When SLR=0, don't perform the check.

16.11.38 Projected Increment Sum Relations for Action n (PDT)

Address: 0h base + 500h offset + (4d × i), where i=0d to 31d



PDT*n* field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–17 DW	Integer part of relation between TRIGGER or STATE increments. Definition of relation values between TRIGGER or STATE increments PDT[i] according to Equations 16.11 or 16.13 (i = 0.....31) .
18–31 DB	Fractional part of relation between TRIGGER or STATE increments.

16.11.39 Calculated Number of Sub-Pulses between two STATE Events for SMC=0 (MLS1)

Address: 0h base + 5C0h offset = 5C0h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																MLS1															
W									0																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLS1 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–13 Reserved	This field is reserved.
14–31 MLS1	Number of pulses between two STATE events (to be set and updated by the CPU). For SMC=0, the value of MLS1 is calculated once by the CPU for fixed values in the DPLL_CTRL_0 register by the formula MLS1 = ((MLT+1) * (TNU+1) / (SNU+1)) and set accordingly. FOR SMC=1, the value of MLS1 represents the number of pulses between two TRIGGER events (to be set and updated by the CPU).

16.11.40 Calculated Number of Sub-pulses between two STATE Events for SMC=1 and RMO=1 (MLS2)

Address: 0h base + 5C4h offset = 5C4h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																MLS2															
W									0																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLS2 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–13 Reserved	This field is reserved.
14–31 MLS2	Number of pulses between two STATE events (to be set and updated by the CPU). Using adapt information and the missing STATE event information SYN_S, this value can be corrected for each increment automatically.

16.11.41 Counter for number of SUB_INC1 pulses (CNT_NUM1)

Address: 0h base + 5C8h offset = 5C8h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								CNT_NUM_1																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CNT_NUM1 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 CNT_NUM_1	Counter for number of SUB_INC1 pulses. Number of pulses in continuous mode for a nominal increment in normal and emergency mode for SUB_INC1, given and updated by CPU only.

16.11.42 Counter for number of SUB_INC2 pulses (CNT_NUM2)

Address: 0h base + 5CCh offset = 5CCh

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								CNT_NUM_2																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CNT_NUM2 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 CNT_NUM_2	Counter for number of SUB_INC2 pulses. Number of pulses in continuous mode for a nominal increment in normal and emergency mode for SUB_INC2, given and updated by CPU only.

16.11.43 Plausibility Value of Next TRIGGER Slope (PVT)

Address: 0h base + 5D0h offset = 5D0h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								PVT																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PVT field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 PVT	Plausibility value of next valid TRIGGER slope. The meaning of the value depends on the value of the PIT value in the DPLL_CTRL_1 register: • For PIT=0, the number of SUB_INC1 pulses to be waited for until a next valid TRIGGER event is accepted. • For PIT=1: PVT is to be multiplied with the last nominal increment time DT_T_actual and divided by 1024 and reduced to a 24 bit value in order to get the time to be waited for until the next valid TRIGGER event is accepted

16.11.44 Actual Calculated Position Stamp of TRIGGER (PSTC)

Address: 0h base + 5E0h offset = 5E0h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								PSTC																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PSTC field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

8–31
PSTC
	Calculated position stamp of last TRIGGER input.
	The value is set by the DPLL and can be updated by the CPU when filter values are to be considered for the exact position (see DPLL_STATUS and DPLL_CTRL registers for explanation of the status and control bits used).
	For each valid slope of TRIGGER in normal mode:
	• When FTD=0, PSTC is set from actual position value, for the first valid TRIGGER event (no filter delay considered).
	• The CPU must update the value once, taking into account the filter value.
	• When FTD=1: PSTC is incremented at each TRIGGER event by:
	  • SMC=0: (MLT+1)*(SYN_T) +PD; while PD=0 for AMT=0.
	  • SMC=1: (MLS1)*(SYN_T) +PD; while PD=0 for AMT=0

16.11.45 Actual Calculated Position Stamp of STATE (PSSC)

Address: 0h base + 5E4h offset = 5E4h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								PSSC																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PSSC field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

8–31
PSSC
	Calculated position stamp of last STATE input.
	first value is set by the DPLL and can be updated by the CPU when the filter delay is to be considered.
	For each valid slope of STATE in normal mode:
	• When FSD=0, PSSC is set from actual position value(no filter delay considered),
	• The CPU must update the value once, taking into account the filter value.
	• When FSD=1, at each valid slope of STATE (PD_S_store=0 for AMS=0):
	  • SMC=0: Add MLS1*(SYN_S) + PD_S_store.
	  • SMC=1: Add MLS2*(SYN_S) + PD_S_store.

16.11.46 Measured Position Stamp at Last TRIGGER Input (PSTM)

Address: 0h base + 5E8h offset = 5E8h

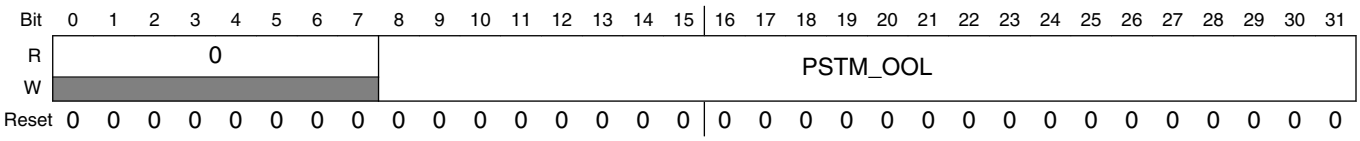
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								PSTM																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PSTM field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 PSTM	Position stamp of TRIGGER, measured. Measured position stamp of last valid TRIGGER input. Store the value TBU_TS1 when a valid TRIGGER event occurs. The value of PSTM is invalid for (RMO=1 and SMC=0). The LSB address is determined using the SWON_T value in the OSW register (see chapter 16.11.8).

16.11.47 Measured Position Stamp at Last But One TRIGGER Input (PSTM_OLD)

Address: 0h base + 5ECh offset = 5ECh

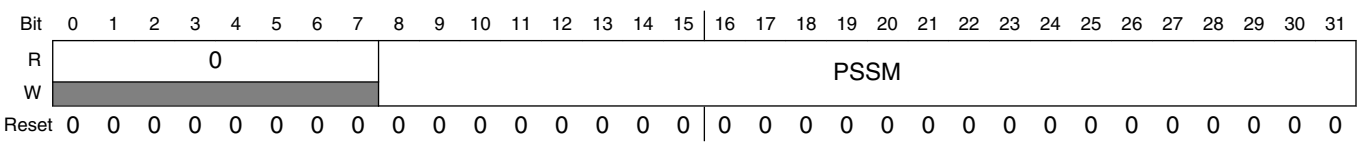


PSTM_OLD field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 PSTM_OOL	Last but one position stamp of TRIGGER, measured. Measured position stamp of last valid TRIGGER input. Last PSTM value: see explanation of PSTM.

16.11.48 Measured Position Stamp at Last STATE Input (PSSM)

Address: 0h base + 5F0h offset = 5F0h

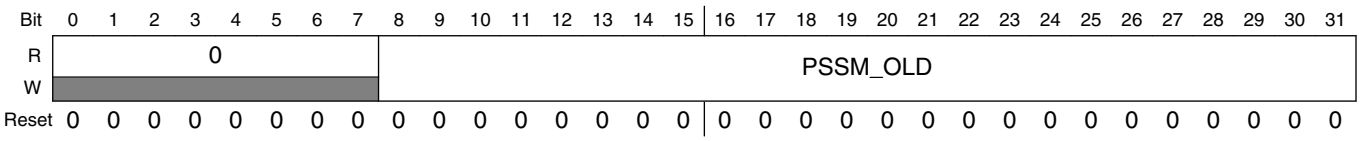


PSSM field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 PSSM	Position stamp of STATE, measured. Measured position stamp of last valid STATE input. Store the value TBU_TS1 or TBU_TS2 respectively at the moment when a valid STATE event occurs. The value of PSSM is invalid for (RMO=0 and SMC=0). The LSB address is determined using the SWON_S value in the OSW register (see chapter 16.11.8).

16.11.49 Measured Position Stamp at Last but One STATE Input (PSSM_OLD)

Address: 0h base + 5F4h offset = 5F4h

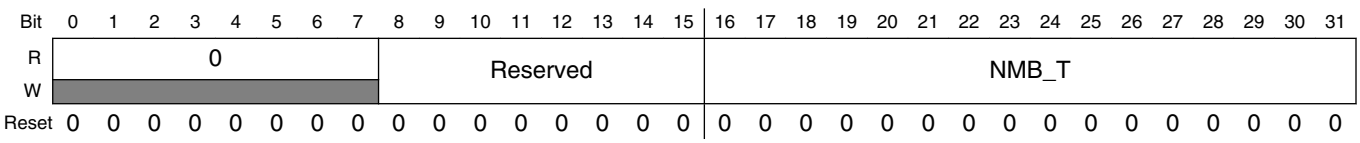


PSSM_OLD field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 PSSM_OLD	Last but one position stamp of STATE, measured. Measured position stamp of last but one valid STATE input. Last PSSM value: see explanation of PSSM The LSB address is determined using the SWON_S value in the OSW register (see chapter 16.11.8).

16.11.50 Number of Pulses to be Sent in Normal Mode (NMB_T)

Address: 0h base + 5F8h offset = 5F8h



NMB_T field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–15 Reserved	Reserved. This field is reserved. Must be written to zero, then reads as zero.
16–31 NMB_T	Number of pulses for TRIGGER. Calculated number of pulses in normal mode for the current TRIGGER increment.

16.11.51 Number of Pulses to be Sent in Emergency Mode (NMB_S)

Address: 0h base + 5FCh offset = 5FCh

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								Reserved				NMB_S																			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NMB_S field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–11 Reserved	Reserved. This field is reserved. Must be written to zero, then reads as zero.
12–31 NMB_S	Number of pulses for STATE. Calculated pulse number. Calculated number of pulses in emergency mode for the current STATE increment.

16.12 RAM Region 1C Memory Map and Registers

Bits 31 down to 24 in each RAM region are not implemented and therefore always read as zero (reserved). Other bits which are declared as reserved are not protected against writing. Reserved address regions are not protected against writing.

The contents of the DPLL RAM Region 1c memory locations are described as follows:

memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
600	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S0)	32	R/W	0000_0000h	16.12.1/682
604	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S1)	32	R/W	0000_0000h	16.12.1/682
608	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S2)	32	R/W	0000_0000h	16.12.1/682
60C	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S3)	32	R/W	0000_0000h	16.12.1/682

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
610	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S4)	32	R/W	0000_0000h	16.12.1/682
614	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S5)	32	R/W	0000_0000h	16.12.1/682
618	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S6)	32	R/W	0000_0000h	16.12.1/682
61C	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S7)	32	R/W	0000_0000h	16.12.1/682
620	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S8)	32	R/W	0000_0000h	16.12.1/682
624	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S9)	32	R/W	0000_0000h	16.12.1/682
628	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S10)	32	R/W	0000_0000h	16.12.1/682
62C	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S11)	32	R/W	0000_0000h	16.12.1/682
630	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S12)	32	R/W	0000_0000h	16.12.1/682
634	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S13)	32	R/W	0000_0000h	16.12.1/682
638	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S14)	32	R/W	0000_0000h	16.12.1/682
63C	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S15)	32	R/W	0000_0000h	16.12.1/682
640	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S16)	32	R/W	0000_0000h	16.12.1/682
644	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S17)	32	R/W	0000_0000h	16.12.1/682
648	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S18)	32	R/W	0000_0000h	16.12.1/682
64C	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S19)	32	R/W	0000_0000h	16.12.1/682
650	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S20)	32	R/W	0000_0000h	16.12.1/682
654	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S21)	32	R/W	0000_0000h	16.12.1/682
658	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S22)	32	R/W	0000_0000h	16.12.1/682
65C	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S23)	32	R/W	0000_0000h	16.12.1/682
660	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S24)	32	R/W	0000_0000h	16.12.1/682
664	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S25)	32	R/W	0000_0000h	16.12.1/682

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
668	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S26)	32	R/W	0000_0000h	16.12.1/682
66C	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S27)	32	R/W	0000_0000h	16.12.1/682
670	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S28)	32	R/W	0000_0000h	16.12.1/682
674	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S29)	32	R/W	0000_0000h	16.12.1/682
678	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S30)	32	R/W	0000_0000h	16.12.1/682
67C	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S31)	32	R/W	0000_0000h	16.12.1/682
680	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S32)	32	R/W	0000_0000h	16.12.1/682
684	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S33)	32	R/W	0000_0000h	16.12.1/682
688	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S34)	32	R/W	0000_0000h	16.12.1/682
68C	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S35)	32	R/W	0000_0000h	16.12.1/682
690	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S36)	32	R/W	0000_0000h	16.12.1/682
694	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S37)	32	R/W	0000_0000h	16.12.1/682
698	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S38)	32	R/W	0000_0000h	16.12.1/682
69C	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S39)	32	R/W	0000_0000h	16.12.1/682
6A0	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S40)	32	R/W	0000_0000h	16.12.1/682
6A4	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S41)	32	R/W	0000_0000h	16.12.1/682
6A8	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S42)	32	R/W	0000_0000h	16.12.1/682
6AC	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S43)	32	R/W	0000_0000h	16.12.1/682
6B0	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S44)	32	R/W	0000_0000h	16.12.1/682
6B4	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S45)	32	R/W	0000_0000h	16.12.1/682
6B8	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S46)	32	R/W	0000_0000h	16.12.1/682
6BC	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S47)	32	R/W	0000_0000h	16.12.1/682

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
6C0	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S48)	32	R/W	0000_0000h	16.12.1/682
6C4	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S49)	32	R/W	0000_0000h	16.12.1/682
6C8	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S50)	32	R/W	0000_0000h	16.12.1/682
6CC	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S51)	32	R/W	0000_0000h	16.12.1/682
6D0	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S52)	32	R/W	0000_0000h	16.12.1/682
6D4	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S53)	32	R/W	0000_0000h	16.12.1/682
6D8	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S54)	32	R/W	0000_0000h	16.12.1/682
6DC	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S55)	32	R/W	0000_0000h	16.12.1/682
6E0	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S56)	32	R/W	0000_0000h	16.12.1/682
6E4	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S57)	32	R/W	0000_0000h	16.12.1/682
6E8	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S58)	32	R/W	0000_0000h	16.12.1/682
6EC	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S59)	32	R/W	0000_0000h	16.12.1/682
6F0	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S60)	32	R/W	0000_0000h	16.12.1/682
6F4	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S61)	32	R/W	0000_0000h	16.12.1/682
6F8	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S62)	32	R/W	0000_0000h	16.12.1/682
6FC	Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S63)	32	R/W	0000_0000h	16.12.1/682
700	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S0)	32	R/W	0000_0000h	16.12.2/ 695
704	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S1)	32	R/W	0000_0000h	16.12.2/ 695
708	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S2)	32	R/W	0000_0000h	16.12.2/ 695
70C	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S3)	32	R/W	0000_0000h	16.12.2/ 695
710	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S4)	32	R/W	0000_0000h	16.12.2/ 695
714	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S5)	32	R/W	0000_0000h	16.12.2/ 695

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
718	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S6)	32	R/W	0000_0000h	16.12.2/ 695
71C	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S7)	32	R/W	0000_0000h	16.12.2/ 695
720	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S8)	32	R/W	0000_0000h	16.12.2/ 695
724	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S9)	32	R/W	0000_0000h	16.12.2/ 695
728	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S10)	32	R/W	0000_0000h	16.12.2/ 695
72C	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S11)	32	R/W	0000_0000h	16.12.2/ 695
730	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S12)	32	R/W	0000_0000h	16.12.2/ 695
734	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S13)	32	R/W	0000_0000h	16.12.2/ 695
738	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S14)	32	R/W	0000_0000h	16.12.2/ 695
73C	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S15)	32	R/W	0000_0000h	16.12.2/ 695
740	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S16)	32	R/W	0000_0000h	16.12.2/ 695
744	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S17)	32	R/W	0000_0000h	16.12.2/ 695
748	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S18)	32	R/W	0000_0000h	16.12.2/ 695
74C	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S19)	32	R/W	0000_0000h	16.12.2/ 695
750	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S20)	32	R/W	0000_0000h	16.12.2/ 695
754	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S21)	32	R/W	0000_0000h	16.12.2/ 695
758	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S22)	32	R/W	0000_0000h	16.12.2/ 695
75C	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S23)	32	R/W	0000_0000h	16.12.2/ 695
760	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S24)	32	R/W	0000_0000h	16.12.2/ 695
764	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S25)	32	R/W	0000_0000h	16.12.2/ 695
768	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S26)	32	R/W	0000_0000h	16.12.2/ 695
76C	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S27)	32	R/W	0000_0000h	16.12.2/ 695

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
770	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S28)	32	R/W	0000_0000h	16.12.2/ 695
774	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S29)	32	R/W	0000_0000h	16.12.2/ 695
778	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S30)	32	R/W	0000_0000h	16.12.2/ 695
77C	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S31)	32	R/W	0000_0000h	16.12.2/ 695
780	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S32)	32	R/W	0000_0000h	16.12.2/ 695
784	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S33)	32	R/W	0000_0000h	16.12.2/ 695
788	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S34)	32	R/W	0000_0000h	16.12.2/ 695
78C	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S35)	32	R/W	0000_0000h	16.12.2/ 695
790	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S36)	32	R/W	0000_0000h	16.12.2/ 695
794	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S37)	32	R/W	0000_0000h	16.12.2/ 695
798	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S38)	32	R/W	0000_0000h	16.12.2/ 695
79C	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S39)	32	R/W	0000_0000h	16.12.2/ 695
7A0	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S40)	32	R/W	0000_0000h	16.12.2/ 695
7A4	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S41)	32	R/W	0000_0000h	16.12.2/ 695
7A8	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S42)	32	R/W	0000_0000h	16.12.2/ 695
7AC	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S43)	32	R/W	0000_0000h	16.12.2/ 695
7B0	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S44)	32	R/W	0000_0000h	16.12.2/ 695
7B4	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S45)	32	R/W	0000_0000h	16.12.2/ 695
7B8	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S46)	32	R/W	0000_0000h	16.12.2/ 695
7BC	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S47)	32	R/W	0000_0000h	16.12.2/ 695
7C0	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S48)	32	R/W	0000_0000h	16.12.2/ 695
7C4	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S49)	32	R/W	0000_0000h	16.12.2/ 695

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
7C8	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S50)	32	R/W	0000_0000h	16.12.2/ 695
7CC	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S51)	32	R/W	0000_0000h	16.12.2/ 695
7D0	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S52)	32	R/W	0000_0000h	16.12.2/ 695
7D4	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S53)	32	R/W	0000_0000h	16.12.2/ 695
7D8	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S54)	32	R/W	0000_0000h	16.12.2/ 695
7DC	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S55)	32	R/W	0000_0000h	16.12.2/ 695
7E0	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S56)	32	R/W	0000_0000h	16.12.2/ 695
7E4	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S57)	32	R/W	0000_0000h	16.12.2/ 695
7E8	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S58)	32	R/W	0000_0000h	16.12.2/ 695
7EC	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S59)	32	R/W	0000_0000h	16.12.2/ 695
7F0	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S60)	32	R/W	0000_0000h	16.12.2/ 695
7F4	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S61)	32	R/W	0000_0000h	16.12.2/ 695
7F8	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S62)	32	R/W	0000_0000h	16.12.2/ 695
7FC	Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S63)	32	R/W	0000_0000h	16.12.2/ 695
800	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S0)	32	R/W	0000_0000h	16.12.3/ 695
804	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S1)	32	R/W	0000_0000h	16.12.3/ 695
808	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S2)	32	R/W	0000_0000h	16.12.3/ 695
80C	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S3)	32	R/W	0000_0000h	16.12.3/ 695
810	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S4)	32	R/W	0000_0000h	16.12.3/ 695
814	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S5)	32	R/W	0000_0000h	16.12.3/ 695
818	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S6)	32	R/W	0000_0000h	16.12.3/ 695
81C	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S7)	32	R/W	0000_0000h	16.12.3/ 695

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
820	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S8)	32	R/W	0000_0000h	16.12.3/ 695
824	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S9)	32	R/W	0000_0000h	16.12.3/ 695
828	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S10)	32	R/W	0000_0000h	16.12.3/ 695
82C	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S11)	32	R/W	0000_0000h	16.12.3/ 695
830	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S12)	32	R/W	0000_0000h	16.12.3/ 695
834	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S13)	32	R/W	0000_0000h	16.12.3/ 695
838	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S14)	32	R/W	0000_0000h	16.12.3/ 695
83C	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S15)	32	R/W	0000_0000h	16.12.3/ 695
840	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S16)	32	R/W	0000_0000h	16.12.3/ 695
844	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S17)	32	R/W	0000_0000h	16.12.3/ 695
848	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S18)	32	R/W	0000_0000h	16.12.3/ 695
84C	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S19)	32	R/W	0000_0000h	16.12.3/ 695
850	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S20)	32	R/W	0000_0000h	16.12.3/ 695
854	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S21)	32	R/W	0000_0000h	16.12.3/ 695
858	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S22)	32	R/W	0000_0000h	16.12.3/ 695
85C	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S23)	32	R/W	0000_0000h	16.12.3/ 695
860	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S24)	32	R/W	0000_0000h	16.12.3/ 695
864	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S25)	32	R/W	0000_0000h	16.12.3/ 695
868	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S26)	32	R/W	0000_0000h	16.12.3/ 695
86C	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S27)	32	R/W	0000_0000h	16.12.3/ 695
870	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S28)	32	R/W	0000_0000h	16.12.3/ 695
874	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S29)	32	R/W	0000_0000h	16.12.3/ 695

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
878	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S30)	32	R/W	0000_0000h	16.12.3/ 695
87C	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S31)	32	R/W	0000_0000h	16.12.3/ 695
880	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S32)	32	R/W	0000_0000h	16.12.3/ 695
884	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S33)	32	R/W	0000_0000h	16.12.3/ 695
888	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S34)	32	R/W	0000_0000h	16.12.3/ 695
88C	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S35)	32	R/W	0000_0000h	16.12.3/ 695
890	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S36)	32	R/W	0000_0000h	16.12.3/ 695
894	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S37)	32	R/W	0000_0000h	16.12.3/ 695
898	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S38)	32	R/W	0000_0000h	16.12.3/ 695
89C	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S39)	32	R/W	0000_0000h	16.12.3/ 695
8A0	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S40)	32	R/W	0000_0000h	16.12.3/ 695
8A4	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S41)	32	R/W	0000_0000h	16.12.3/ 695
8A8	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S42)	32	R/W	0000_0000h	16.12.3/ 695
8AC	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S43)	32	R/W	0000_0000h	16.12.3/ 695
8B0	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S44)	32	R/W	0000_0000h	16.12.3/ 695
8B4	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S45)	32	R/W	0000_0000h	16.12.3/ 695
8B8	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S46)	32	R/W	0000_0000h	16.12.3/ 695
8BC	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S47)	32	R/W	0000_0000h	16.12.3/ 695
8C0	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S48)	32	R/W	0000_0000h	16.12.3/ 695
8C4	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S49)	32	R/W	0000_0000h	16.12.3/ 695
8C8	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S50)	32	R/W	0000_0000h	16.12.3/ 695
8CC	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S51)	32	R/W	0000_0000h	16.12.3/ 695

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
8D0	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S52)	32	R/W	0000_0000h	16.12.3/ 695
8D4	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S53)	32	R/W	0000_0000h	16.12.3/ 695
8D8	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S54)	32	R/W	0000_0000h	16.12.3/ 695
8DC	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S55)	32	R/W	0000_0000h	16.12.3/ 695
8E0	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S56)	32	R/W	0000_0000h	16.12.3/ 695
8E4	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S57)	32	R/W	0000_0000h	16.12.3/ 695
8E8	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S58)	32	R/W	0000_0000h	16.12.3/ 695
8EC	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S59)	32	R/W	0000_0000h	16.12.3/ 695
8F0	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S60)	32	R/W	0000_0000h	16.12.3/ 695
8F4	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S61)	32	R/W	0000_0000h	16.12.3/ 695
8F8	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S62)	32	R/W	0000_0000h	16.12.3/ 695
8FC	Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S63)	32	R/W	0000_0000h	16.12.3/ 695
900	Nominal STATE Increment Durations in FULL SCALE (DT_S0)	32	R/W	0000_0000h	16.12.4/ 696
904	Nominal STATE Increment Durations in FULL SCALE (DT_S1)	32	R/W	0000_0000h	16.12.4/ 696
908	Nominal STATE Increment Durations in FULL SCALE (DT_S2)	32	R/W	0000_0000h	16.12.4/ 696
90C	Nominal STATE Increment Durations in FULL SCALE (DT_S3)	32	R/W	0000_0000h	16.12.4/ 696
910	Nominal STATE Increment Durations in FULL SCALE (DT_S4)	32	R/W	0000_0000h	16.12.4/ 696
914	Nominal STATE Increment Durations in FULL SCALE (DT_S5)	32	R/W	0000_0000h	16.12.4/ 696
918	Nominal STATE Increment Durations in FULL SCALE (DT_S6)	32	R/W	0000_0000h	16.12.4/ 696
91C	Nominal STATE Increment Durations in FULL SCALE (DT_S7)	32	R/W	0000_0000h	16.12.4/ 696
920	Nominal STATE Increment Durations in FULL SCALE (DT_S8)	32	R/W	0000_0000h	16.12.4/ 696
924	Nominal STATE Increment Durations in FULL SCALE (DT_S9)	32	R/W	0000_0000h	16.12.4/ 696

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
928	Nominal STATE Increment Durations in FULL SCALE (DT_S10)	32	R/W	0000_0000h	16.12.4/ 696
92C	Nominal STATE Increment Durations in FULL SCALE (DT_S11)	32	R/W	0000_0000h	16.12.4/ 696
930	Nominal STATE Increment Durations in FULL SCALE (DT_S12)	32	R/W	0000_0000h	16.12.4/ 696
934	Nominal STATE Increment Durations in FULL SCALE (DT_S13)	32	R/W	0000_0000h	16.12.4/ 696
938	Nominal STATE Increment Durations in FULL SCALE (DT_S14)	32	R/W	0000_0000h	16.12.4/ 696
93C	Nominal STATE Increment Durations in FULL SCALE (DT_S15)	32	R/W	0000_0000h	16.12.4/ 696
940	Nominal STATE Increment Durations in FULL SCALE (DT_S16)	32	R/W	0000_0000h	16.12.4/ 696
944	Nominal STATE Increment Durations in FULL SCALE (DT_S17)	32	R/W	0000_0000h	16.12.4/ 696
948	Nominal STATE Increment Durations in FULL SCALE (DT_S18)	32	R/W	0000_0000h	16.12.4/ 696
94C	Nominal STATE Increment Durations in FULL SCALE (DT_S19)	32	R/W	0000_0000h	16.12.4/ 696
950	Nominal STATE Increment Durations in FULL SCALE (DT_S20)	32	R/W	0000_0000h	16.12.4/ 696
954	Nominal STATE Increment Durations in FULL SCALE (DT_S21)	32	R/W	0000_0000h	16.12.4/ 696
958	Nominal STATE Increment Durations in FULL SCALE (DT_S22)	32	R/W	0000_0000h	16.12.4/ 696
95C	Nominal STATE Increment Durations in FULL SCALE (DT_S23)	32	R/W	0000_0000h	16.12.4/ 696
960	Nominal STATE Increment Durations in FULL SCALE (DT_S24)	32	R/W	0000_0000h	16.12.4/ 696
964	Nominal STATE Increment Durations in FULL SCALE (DT_S25)	32	R/W	0000_0000h	16.12.4/ 696
968	Nominal STATE Increment Durations in FULL SCALE (DT_S26)	32	R/W	0000_0000h	16.12.4/ 696
96C	Nominal STATE Increment Durations in FULL SCALE (DT_S27)	32	R/W	0000_0000h	16.12.4/ 696
970	Nominal STATE Increment Durations in FULL SCALE (DT_S28)	32	R/W	0000_0000h	16.12.4/ 696
974	Nominal STATE Increment Durations in FULL SCALE (DT_S29)	32	R/W	0000_0000h	16.12.4/ 696
978	Nominal STATE Increment Durations in FULL SCALE (DT_S30)	32	R/W	0000_0000h	16.12.4/ 696
97C	Nominal STATE Increment Durations in FULL SCALE (DT_S31)	32	R/W	0000_0000h	16.12.4/ 696

Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
980	Nominal STATE Increment Durations in FULL SCALE (DT_S32)	32	R/W	0000_0000h	16.12.4/ 696
984	Nominal STATE Increment Durations in FULL SCALE (DT_S33)	32	R/W	0000_0000h	16.12.4/ 696
988	Nominal STATE Increment Durations in FULL SCALE (DT_S34)	32	R/W	0000_0000h	16.12.4/ 696
98C	Nominal STATE Increment Durations in FULL SCALE (DT_S35)	32	R/W	0000_0000h	16.12.4/ 696
990	Nominal STATE Increment Durations in FULL SCALE (DT_S36)	32	R/W	0000_0000h	16.12.4/ 696
994	Nominal STATE Increment Durations in FULL SCALE (DT_S37)	32	R/W	0000_0000h	16.12.4/ 696
998	Nominal STATE Increment Durations in FULL SCALE (DT_S38)	32	R/W	0000_0000h	16.12.4/ 696
99C	Nominal STATE Increment Durations in FULL SCALE (DT_S39)	32	R/W	0000_0000h	16.12.4/ 696
9A0	Nominal STATE Increment Durations in FULL SCALE (DT_S40)	32	R/W	0000_0000h	16.12.4/ 696
9A4	Nominal STATE Increment Durations in FULL SCALE (DT_S41)	32	R/W	0000_0000h	16.12.4/ 696
9A8	Nominal STATE Increment Durations in FULL SCALE (DT_S42)	32	R/W	0000_0000h	16.12.4/ 696
9AC	Nominal STATE Increment Durations in FULL SCALE (DT_S43)	32	R/W	0000_0000h	16.12.4/ 696
9B0	Nominal STATE Increment Durations in FULL SCALE (DT_S44)	32	R/W	0000_0000h	16.12.4/ 696
9B4	Nominal STATE Increment Durations in FULL SCALE (DT_S45)	32	R/W	0000_0000h	16.12.4/ 696
9B8	Nominal STATE Increment Durations in FULL SCALE (DT_S46)	32	R/W	0000_0000h	16.12.4/ 696
9BC	Nominal STATE Increment Durations in FULL SCALE (DT_S47)	32	R/W	0000_0000h	16.12.4/ 696
9C0	Nominal STATE Increment Durations in FULL SCALE (DT_S48)	32	R/W	0000_0000h	16.12.4/ 696
9C4	Nominal STATE Increment Durations in FULL SCALE (DT_S49)	32	R/W	0000_0000h	16.12.4/ 696
9C8	Nominal STATE Increment Durations in FULL SCALE (DT_S50)	32	R/W	0000_0000h	16.12.4/ 696
9CC	Nominal STATE Increment Durations in FULL SCALE (DT_S51)	32	R/W	0000_0000h	16.12.4/ 696
9D0	Nominal STATE Increment Durations in FULL SCALE (DT_S52)	32	R/W	0000_0000h	16.12.4/ 696
9D4	Nominal STATE Increment Durations in FULL SCALE (DT_S53)	32	R/W	0000_0000h	16.12.4/ 696

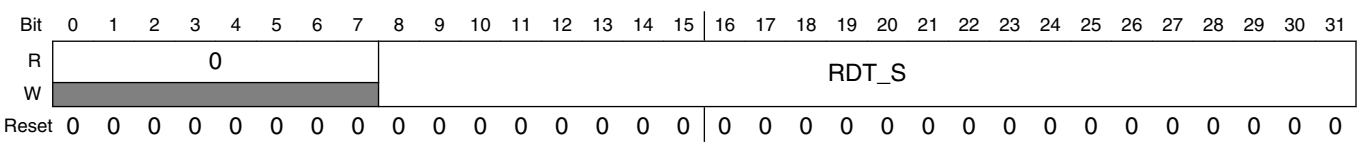
Table continues on the next page...

memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
9D8	Nominal STATE Increment Durations in FULL SCALE (DT_S54)	32	R/W	0000_0000h	16.12.4/ 696
9DC	Nominal STATE Increment Durations in FULL SCALE (DT_S55)	32	R/W	0000_0000h	16.12.4/ 696
9E0	Nominal STATE Increment Durations in FULL SCALE (DT_S56)	32	R/W	0000_0000h	16.12.4/ 696
9E4	Nominal STATE Increment Durations in FULL SCALE (DT_S57)	32	R/W	0000_0000h	16.12.4/ 696
9E8	Nominal STATE Increment Durations in FULL SCALE (DT_S58)	32	R/W	0000_0000h	16.12.4/ 696
9EC	Nominal STATE Increment Durations in FULL SCALE (DT_S59)	32	R/W	0000_0000h	16.12.4/ 696
9F0	Nominal STATE Increment Durations in FULL SCALE (DT_S60)	32	R/W	0000_0000h	16.12.4/ 696
9F4	Nominal STATE Increment Durations in FULL SCALE (DT_S61)	32	R/W	0000_0000h	16.12.4/ 696
9F8	Nominal STATE Increment Durations in FULL SCALE (DT_S62)	32	R/W	0000_0000h	16.12.4/ 696
9FC	Nominal STATE Increment Durations in FULL SCALE (DT_S63)	32	R/W	0000_0000h	16.12.4/ 696

16.12.1 Reciprocal Values of the Nominal STATE Increment Durations in FULL SCALE (RDT_S*n*)

Address: 0h base + 600h offset + (4d × i), where i=0d to 63d

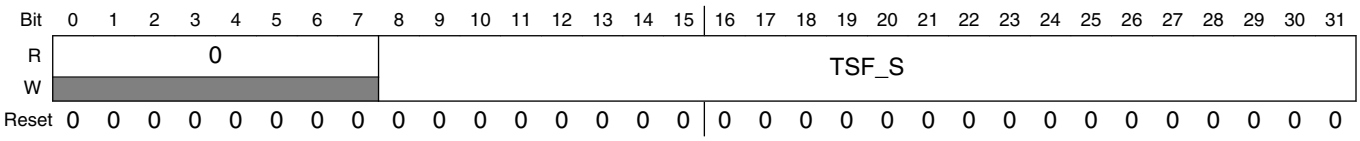


RDT_S*n* field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 RDT_S	Reciprocal difference time of TRIGGER. Nominal reciprocal value of the number of time stamp clocks measured in the corresponding increment *2 ^{32} while only the lower 24 bits are used; no gap considered. The LSB is rounded up when the next truncated bit is 1. **NOTE:** There are 2*(SNU+1-SYN_NS) entries for SYSF=0 or 2*(SNU+1)-SYN_NS entries for SYSF=1 respectively.

16.12.2 Time Stamp Values of the Nominal STATE Events in FULL SCALE (TSF_S*n*)

Address: 0h base + 700h offset + (4d × i), where i=0d to 63d

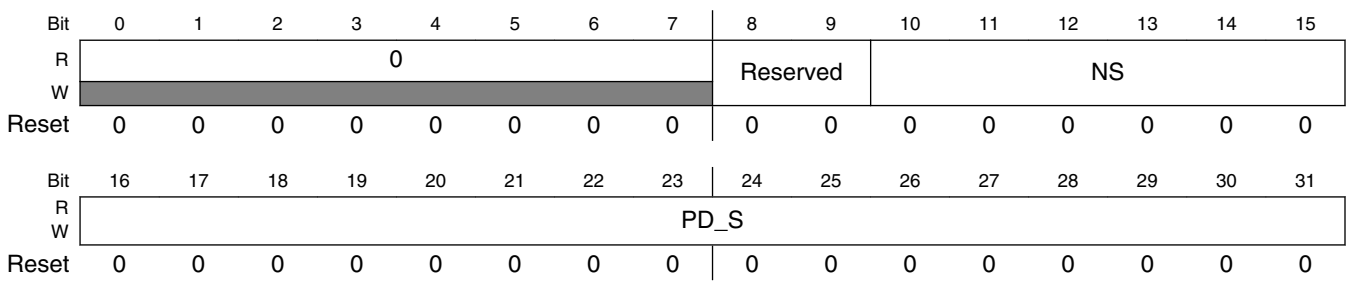


TSF_S*n* field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 TSF_S	Time stamp field of STATE. Time stamp value of each valid STATE event. **NOTE:** There are 2* (SNU+1) entries.

16.12.3 Adapt and Profile Values STATE Incrementsin FULL SCALE (ADT_S*n*)

Address: 0h base + 800h offset + (4d × i), where i=0d to 63d



ADT_S*n* field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–9 Reserved	Reserved. This field is reserved. **NOTE:** Must be written to zero, afterward reads as zero.
10–15 NS	Number of STATEs. Number of nominal STATE parts in the corresponding increment.

Table continues on the next page...

ADT_S*n* field descriptions (continued)

Field	Description
	NOTE: There are 2*(SNU+1-SYN_NS) entries for SYSF=0 or 2*(SNU+1)-SYN_NS entries for SYSF=1, respectively.
16–31 PD_S	Physical deviation of STATE. Adapt values for each STATE increment in FULL_SCALE (sint16). This value represents the number of pulses to be added to the correspondent increment. The absolute value of a negative PD_S must not exceed MLS1 or MLS2, respectively.

16.12.4 Nominal STATE Increment Durations in FULL SCALE (DT_S*n*)

Address: 0h base + 900h offset + (4d × i), where i=0d to 63d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								DT_S																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DT_S*n* field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 DT_S	Difference time of STATE. Nominal increment duration values for each STATE increment in FULL_SCALE (considering no gap). **NOTE:** There are 2*(SNU+1-SYN_NS) entries for SYSF=0 or 2*(SNU+1)-SYN_NS entries for SYSF=1 respectively.

16.13 RAM Region 2 Memory Map and Registers

Bits 31 down to 24 in each RAM region are not implemented and therefore always read as zero (reserved). Other bits which are declared as reserved are not protected against writing. Reserved address regions are not protected against writing.

RDT_T0 through RDT_T1023 are 32-bit (4-byte) memory locations with address offests 0x4000 + 4n, n=0.....1023.

TSF_T0 through TSF_T1023 are 32-bit (4-byte) memory locations with address offsets 0x4200 + 4n, n=0.....1023.

ADT_T0 through ADT_T1023 are 32-bit (4-byte) memory locations with address offsets 0x4400 + 4n, n=0.....1023.

DT_T0 through DT_T1023 are 32-bit (4-byte) memory locations with address offsets 0x4600 + 4n, n=0.....1023.

memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000	Reciprocal Values of the Nominal TRIGGER Increment Durations in FULL_SCALE (RDT_T)	32	R/W	0000_0000h	16.13.1/697
4200	Time Stamp Values of the Nominal TRIGGER Increments in FULL_SCALE (TSF_T)	32	R/W	0000_0000h	16.13.2/ 698
4400	Adapt and Profile Values of the TRIGGER Increments in FULL_SCALE (ADT_T)	32	R/W	0000_0000h	16.13.3/ 698
4600	Nominal TRIGGER Increment Durations in FULL_SCALE (DT_T)	32	R/W	0000_0000h	16.13.4/ 699

16.13.1 Reciprocal Values of the Nominal TRIGGER Increment Durations in FULL_SCALE (RDT_T)

Address: 0h base + 4000h offset = 4000h

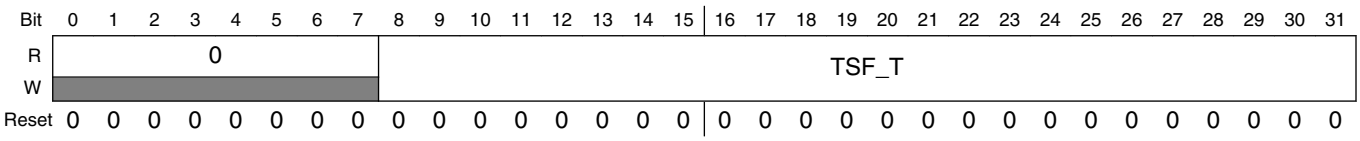
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								RDT_T																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDT_T field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 RDT_T	Reciprocal difference time of TRIGGER. 2* (TNU+1- SYN_NT) stored values nominal reciprocal value of the number of time stamp clocks measured in the corresponding increment (which is divided by the number of nominal increments); multiplied by 2^{32} while only the lower 24 bits are used; the LSB is rounded up, when the next truncated bit is 1. **NOTE:** There are 2* (TNU+1- SYN_NT) entries. The maximum number of entries is restricted to a value corresponding to the OSS value in the DPLL_OSW register.

16.13.2 Time Stamp Values of the Nominal TRIGGER Increments in FULL_SCALE (TSF_T)

Address: 0h base + 4200h offset = 4200h

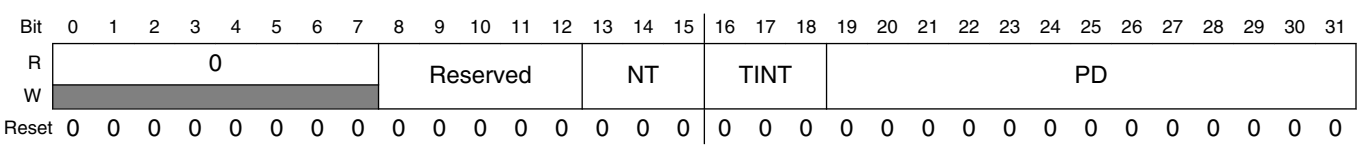


TSF_T field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 TSF_T	Time stamp field of valid TRIGGER slopes. **NOTE:** There are 2* (TNU+1) entries. The maximum number of entries is restricted to a value corresponding to the OSS value in the DPLL_OSW register.

16.13.3 Adapt and Profile Values of the TRIGGER Increments in FULL_SCALE (ADT_T)

Address: 0h base + 4400h offset = 4400h



ADT_T field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–12 Reserved	Reserved. This field is reserved. Must be written to zero, afterward reads as zero.
13–15 NT	Number of TRIGGERs. Number of nominal TRIGGER parts in the corresponding increment. **NOTE:** There are 2* (TNU+1- SYN_NT) entries. The maximum number of entries is restricted to a value corresponding to the OSS value in the DPLL_OSW register.
16–18 TINT	TRIGGER Interrupt information.

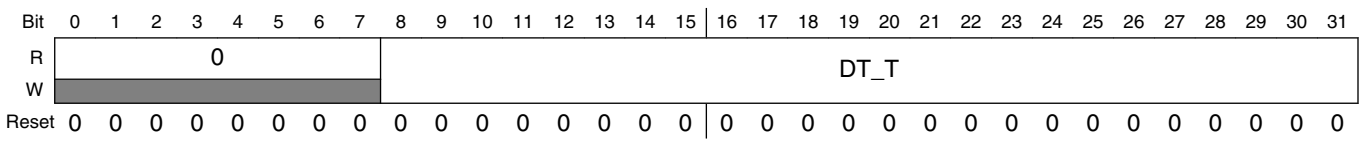
Table continues on the next page...

ADT_T field descriptions (continued)

Field	Description
	Depending on the value, up to 7 different interrupts can be generated. In the current version the 5 interrupts TE0_IRQ … TE4_IRQ are supported by TINT=”001”, “010”, "011", “100”, “101” respectively. For the values "000", "110" and "111" no interrupt is generated and no other reaction is performed. The corresponding interrupt is activated, when the TINT value is read by the DPLL together with the other values (PD, NT) according to the profile.
19–31 PD	Physical deviation. Adapt values for each TRIGGER increment in FULL_SCALE (sint13); the PD value does mean the number of SUB_INC1 pulses to be added to NT*(MLT+1). The absolute value of a negative PD must not exceed NT*(MLT+1) or MLS1 respectively. Systematic missing TRIGGER events must not be considered for the value of PD.

16.13.4 Nominal TRIGGER Increment Durations in FULL_SCALE (DT_T)

Address: 0h base + 4600h offset = 4600h



DT_T field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 DT_T	Difference time of TRIGGER. Increment duration values for each TRIGGER increment in FULL_SCALE divided by the number of nominal increments (nominal value). **NOTE:** There are 2* (TNU+1- SYN_NT) entries. The maximum number of entries is restricted to a value corresponding to the OSS value in the DPLL_OSW register.

Chapter 17
Sensor Pattern Evaluation (SPE)

17.1 SPE Overview

The Sensor Pattern Evaluation (SPE) submodule can be used to evaluate three hall sensor inputs and together with the TOM module, support the drive of BLDC engines. The input signals have been filtered in the connected TIM channels.

The SPE can also be used as an input stage to the MAP submodule if the DPLL is used to calculate the rotation speed of one or two electric engine(s).

The integration of the SPE into the overall GTM architecture concept is shown in Figure 17-1.

17.1.1 SPE Submodule integration concept into GTM

The SPE submodule can determine a rotation direction out of the combined *TIM[i]_CHx(48)*, *TIM[i]_CHy(48)* and *TIM[i]_CHz(48)* signals, as shown in Figure 17-1. A pattern matching algorithm is applied to these input signals to generate the *SPEx_DIR* signal on behalf of the temporal relationship between the input patterns. A possible sample pattern of the three input signals is shown in Figure 17-2. The input pattern is programmable within the SPE submodule.

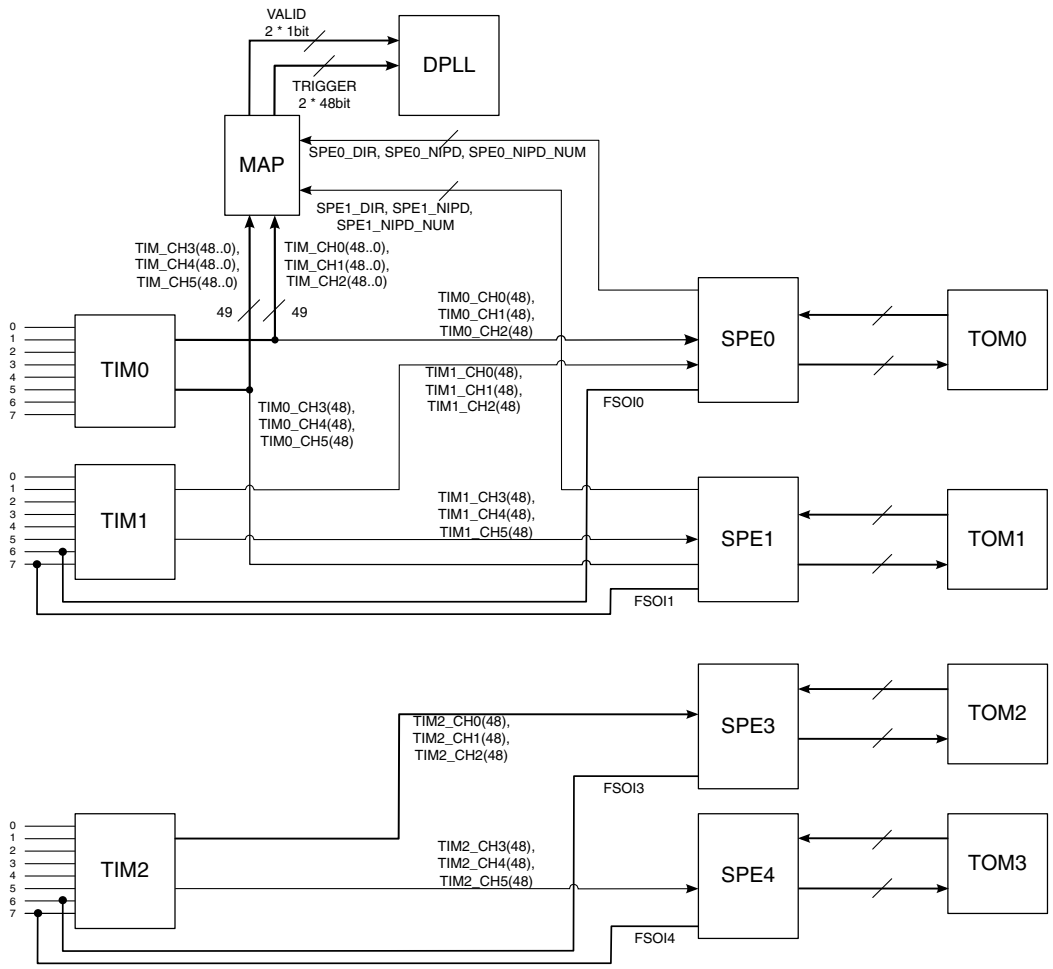


Figure 17-1. SPE architecture

17.1.2 SPE Sample input pattern for *TIM[i]_CH[x,y,z](48)*

The input signals shown in Figure 17-2 define the pattern from the input sensors, which have a 50% high and 50% low phase. The pattern is:

100 – 110 – 010 – 011 – 001 – 101 – 100,

where the first bit (smallest circle) represents the TIM[i]_CH[x](48) channel, the second bit represents the TIM[i]_CH[y](48) channel, and the third bit (greatest circle) represents the TIM[i]_CH[z](48) channel.

The SPE module expects that, with every new pattern, only one of the three input signals has a value change.

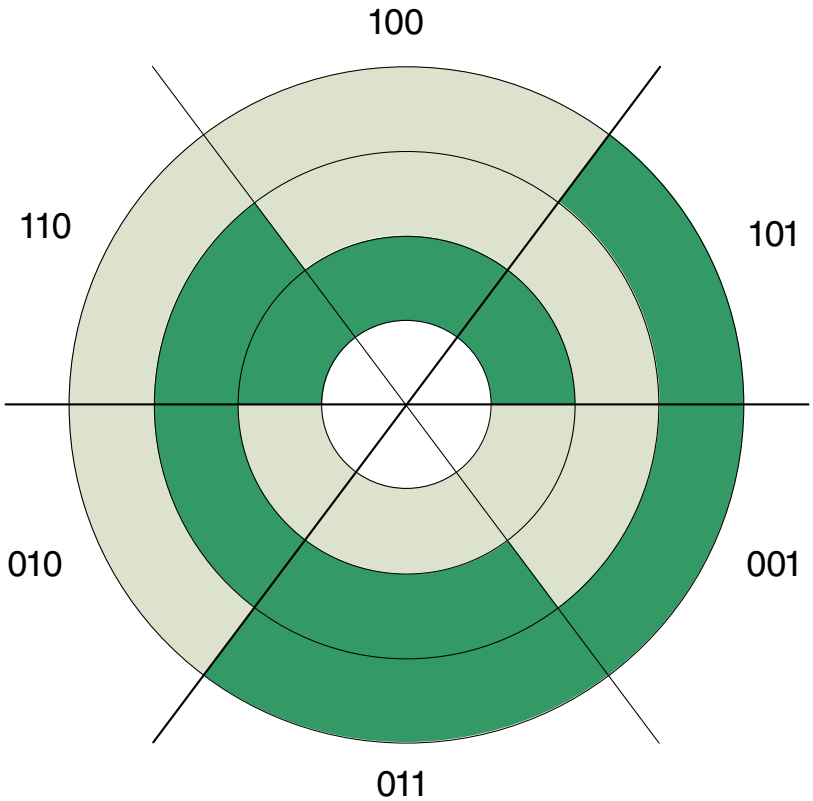


Figure 17-2. SPE sample input pattern for TIM[i]_CH[x,y,z](48)

17.2 SPE Submodule description

The SPE submodule can handle sensor pattern inputs. If one of the *TIM[i]_CH[x](48)*,*TIM[i]_CH[y](48) or TIM[i]_CH[z](48)* input signals changes its value, a sample of all three input signals is taken. The encoded rotation direction and the validity of the input pattern sequence (derived from the sample of the three input signals) can be detected and signalled. When a valid input pattern is detected, the SPE submodule can control the outputs of a dedicated TOM submodule. This connection is shown in Figure 17-3.

17.2.1 SPE to TOM Connections

The TOM[i]_CH0_TRIG_CCU[x] and TOM[i]_CHn_SOUR signal lines, shown in Figure 17-3, are used to evaluate the current state of the TOM outputs, where the SPE[i]_OUT output vector is used to control the TOM output depending on the new input pattern. The SPE[i]_OUT output vector is defined inside the SPE submodule in a pattern definition table SPE[i]_OUT_PAT[x]. The internal SPE submodule architecture is shown in Figure 17-4.

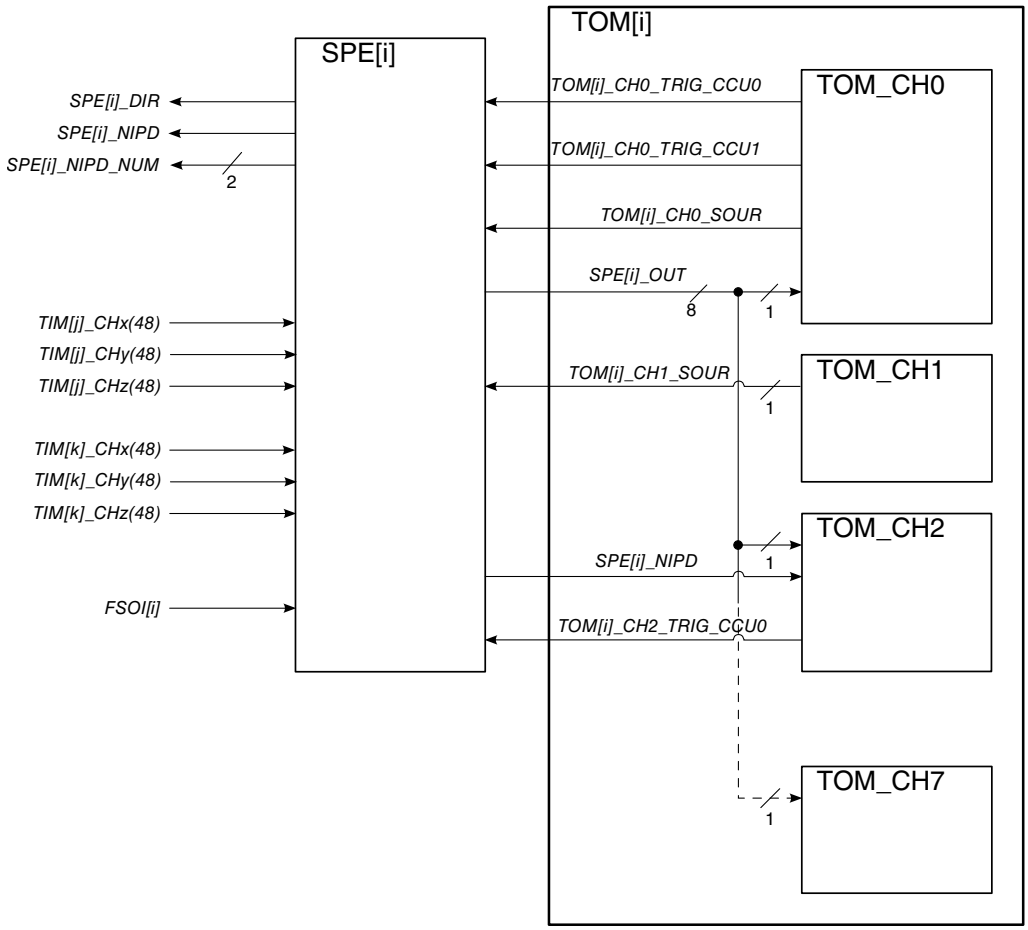


Figure 17-3. SPE-to-TOM connections

17.2.2 SPE Submodule architecture

Figure 17-4 shows the SPE submodule architecture.

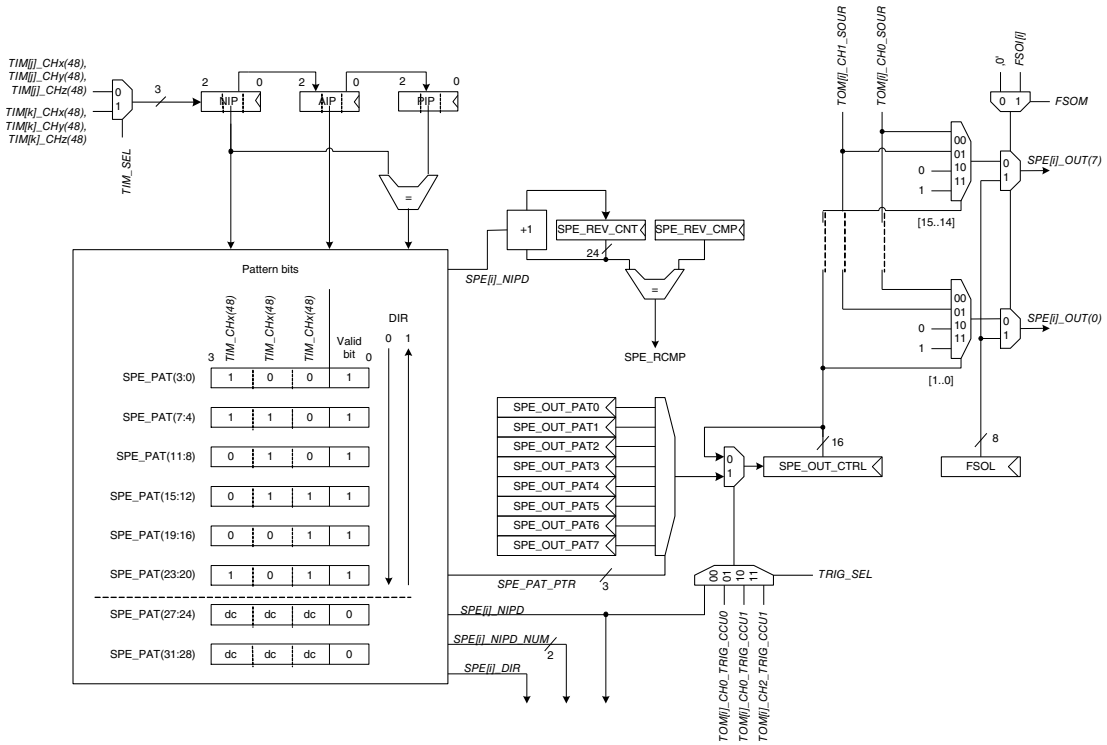


Figure 17-4. SPE submodule architecture

The pattern pointer in **SPE[i]_CTRL_STAT[SPE_PAT_PTR]** bit field holds the valid-input pattern for the three *TIM[i]_CH[x](48)*, *TIM[i]_CH[y](48)* and *TIM[i]_CH[z](48)* input patterns. The valid-input pattern is programmable. The valid bit, shown in Figure 17-4, indicates the validity of the programmed pattern. Figure 17-4 shows the programming of the **SPE[i]_CTRL_STAT[SPE_PAT_PTR]** bit field for the input pattern defined in Figure 17-2.

The rotation direction is determined by the order of the valid input pattern. This rotation direction defines if the***SPE_PAT_PTR*** pointer (shown in Figure 17-4) is incremented (DIR = 0) or decremented (DIR = 1). Whenever a valid-input pattern is detected, the *NIPD* signal is asserted, the *SPE_PAT_PTR* pointer is incremented/decremented and a new *SPE[i]_OUT(x)* output control signal is routed to the corresponding TOM submodule.

After a new input pattern is detected (signalled by *SPE[i]_NIPD*), the TOM[i]_CH2 (with i=0..3) channel can be used with the SPE submodule to trigger a delayed update of the **SPE_OUT_CTRL** register.

To do this:

1. The TOM[i]_CH2 has to be configured to work in one-shot mode by setting the **TOM[i]_CH2_CTRL[OSM]** bit = 1.
2. The SPE mode of this channel has to be enabled by setting the **TOM[i]_CH2_CTRL**[SPEM] bit = 1.

3. The SPE module has to be configured to update the **SPE_OUT_CTRL** register with the*TOM[i]_CH2_TRIG_CCU1* signal by setting the **SPE[i]_CTRL_STAT**[**TRIG_SEL]** bit field to '11'.

Then, when a new input is detected, the *SPE[i]_NIPD* signal triggers TOM channel 2 to generate the first edge of one PWM period (by resetting the **TOM[0]_CH_CN0** counter register to 0). When the second PWM edge is generated by CCU1 of TOM channel 2, the *TOM[i]_CH2_TRIG_CCU1* signal triggers the update of **SPE_OUT_CTRL** register.

According to Figure 17-2, the two input patterns "000" and "111" are invalid combinations that will end in assertion of an *SPE[i]_PERR* interrupt. These two patterns can be used to determine a sensor input error. An *SPE[i]_PERR* interrupt will also be asserted if the input patterns occur in an incorrect order, e.g. if the pattern "010" does not follow the pattern "110" or "011".

The input pattern history is stored in the **SPE[i]_CTRL_STAT**[NIP, AIP, PIP] bit fields, where NIP stores the New Input Pattern, AIP stores the Actual Input Pattern, and PIP stores the Previous Input Pattern (PIP). When the SPE[i]_PERR interrupt is asserted, the CPU can determine if a sensor is broken by analysing the **NIP** bit field. The input pattern in the **NIP** bit field is updated whenever a valid edge is detected on one of the *TIM[i]_CH[x](48)*, *TIM[i]_CH[y](48)* or *TIM[i]_CH[z](48)* inputs.

The SPE module provides a 'Fast Shut-Off' feature for all TOM channels that are controlled by the SPE. Inputs from TIM channel 6 or TIM channel 7, depending on the TIM instance connected to the SPE module as shown in Figure 17-1, are used to trigger 'Fast Shut-Off'. 'Fast Shut-Off' is enabled by setting the **SPE[i]_CTRL_STAT[FSOM]** bit = 1. Channels are configured for 'Fast Shut-Off' by setting the corresponding bits in the SPE[i]_CTRL_STAT[FSOL] bit field to 1.

17.2.3 SPE[i]_IN_PAT register representation

The CPU can disable one of the three input signals (e.g. when a broken input sensor is detected) by writing a zero to the corresponding input enable bit inside the SPE[i]_CTRL_STAT[SIE] bit field.

When at least one of the *TIM[i]_CH[x](48)*, *TIM[i]_CH[y](48)* or *TIM[i]_CH[z](48)* input signals (shown in Figure 17-5) changes, the new bit pattern is stored in the **SPE[i]_CTRL_STAT[NIP]** bit field. If the current input pattern in the NIP is the same that in the **SPE[i]_CTRL_STAT[PIP]** bit field, the direction of the engine has changed and the *SPE[i]_DCHG* interrupt is asserted. The **SPE[i]_CTRL_STAT[PDIR]** bit is toggled and the *SPE[i]_DIR* signal is changed accordingly. The contents of the AIP are moved to the PIP and the contents of the NIP are moved to the AIP.

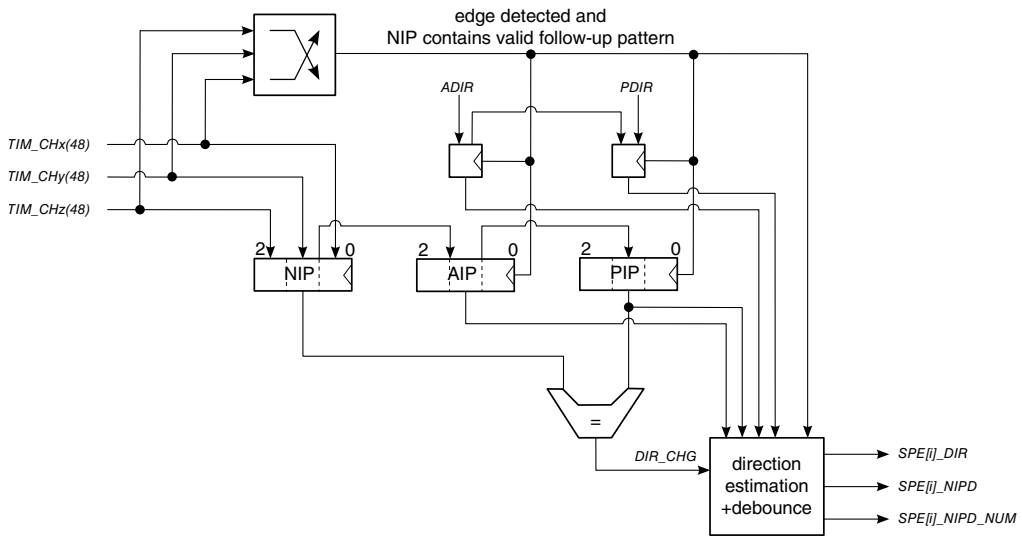


Figure 17-5. Pattern representation

If the next input pattern indicates another rotation change (as described above), the input signal is categorized as bouncing and the 'bouncing input signal' SPE[i]_BIS interrupt is asserted.

Immediately after a new input pattern is detected and stored in the NIP and if the NIP does not match the the PIP (i.e. no direction change was detected), the SPE moves the contents of the AIP to the PIP and moves the contents of the NIP to the AIP. The *SPE[i]_NIPD* interrupt is then asserted.

The number of the channel that has changed (and thereby leads to the new input pattern) is encoded in the two *SPE[i]_NIPD_NUM* signals.

If a sensor error is detected, the CPU can analyze the pattern in the NIP to determine which input line comes from the broken sensor. Then, the CPU should disable the erroneous input line by clearing the corresponding enable bit in the **SPE[i]_CTRL_STAT[SIE]** bit field to 0.The SPE then uses the two remaining *TIM[i]_CH* input lines to determine rotation direction.

The CPU can determine pattern history by reading the SPE[i]_CTRL_STAT[AIP, PIP] bit fields. The AIP holds the actual detected pattern on the *TIM[i]_CH[x](48)*, *TIM[i]_CH[y](48)* and *TIM[i]_CH[z](48)* inputs and the PIP holds the previously detected pattern.

After reset, the SPE[i]_CTRL_STAT[NIP, AIP, PIP, SPE_PAT_PTR] bit fields and the **SPE[i]_OUT_CTRL** register must be initialized before the SPE is enabled. To do this, the microcontroller can read the NIP, and depending on its value, determine initialization values for the AIP, PIP, SPT_PAT_PTR bit fields and the SPE[i]_OUT_CTRL register.

17.2.4

17.3 SPE Interrupt signals

Table 17-1 describes SPE interrupt signals.

Table 17-1. SPE interrpt request signals

Signal	Description
SPE[i]_NIPD	SPE new valid input pattern detected.
SPE[i]_DCHG	SPE rotation direction change detected on behalf of input pattern.
SPE[i]_PERR	SPE invalid input pattern detected.
SPE[i]_BIS	SPE bouncing input signal detected.
SPE[i]_RCMP	SPE Revolution counter compare value reached.

17.4 Memory Map and Registers

The Sensor Pattern Evaluation (SPE0) module registers are described as follows:

SPE_0 memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	SPE0 Control Status Register (SPE_0_CTRL_STAT)	32	R/W	0000_0000h	17.4.1/709
4	SPE0 Input Pattern Register (SPE_0_PAT)	32	R/W	0000_0000h	17.4.2/711
8	SPE0 Output Pattern n Register (SPE_0_OUT_PAT0)	32	R/W	0000_0000h	17.4.3/714
C	SPE0 Output Pattern n Register (SPE_0_OUT_PAT1)	32	R/W	0000_0000h	17.4.3/714
10	SPE0 Output Pattern n Register (SPE_0_OUT_PAT2)	32	R/W	0000_0000h	17.4.3/714
14	SPE0 Output Pattern n Register (SPE_0_OUT_PAT3)	32	R/W	0000_0000h	17.4.3/714
18	SPE0 Output Pattern n Register (SPE_0_OUT_PAT4)	32	R/W	0000_0000h	17.4.3/714
1C	SPE0 Output Pattern n Register (SPE_0_OUT_PAT5)	32	R/W	0000_0000h	17.4.3/714
20	SPE0 Output Pattern n Register (SPE_0_OUT_PAT6)	32	R/W	0000_0000h	17.4.3/714
24	SPE0 Output Pattern n Register (SPE_0_OUT_PAT7)	32	R/W	0000_0000h	17.4.3/714
28	SPE0 Output Control Register (SPE_0_OUT_CTRL)	32	R/W	0000_0000h	17.4.4/714
2C	SPE0 Interrupt Request Notify Register (SPE_0_IRQ_NOTIFY)	32	w1c	0000_0000h	17.4.5/715
30	SPE0 Interrupt Request Enable Register (SPE_0_IRQ_EN)	32	R/W	0000_0000h	17.4.6/716

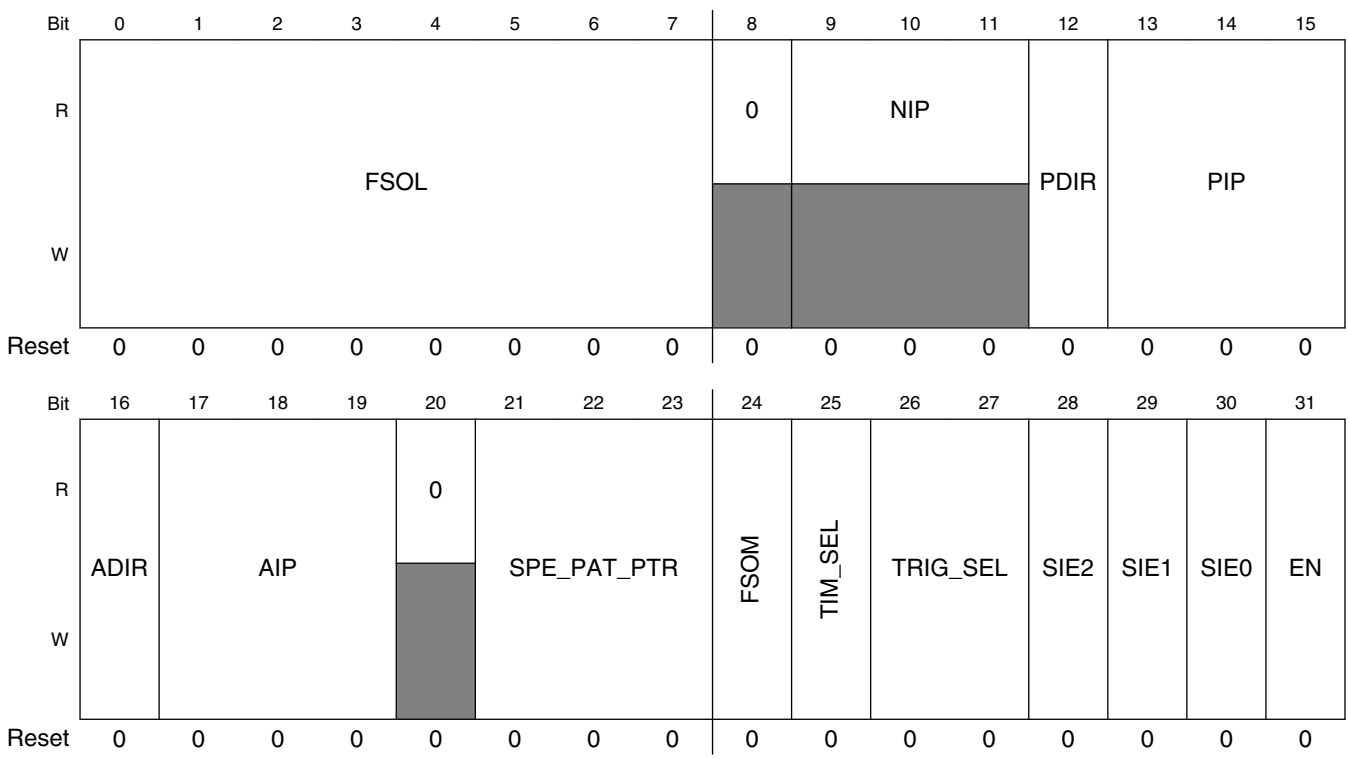
Table continues on the next page...

SPE_0 memory map (continued)

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
34	SPE0 Interrupt Request Force Interrupt Register (SPE_0_IRQ_FORCINT)	32	R/W	0000_0000h	17.4.7/718
38	SPE0 Interrupt Request Mode Register (SPE_0_IRQ_MODE)	32	R/W	See section	17.4.8/719
3C	SPE0 Error Interrupt Request Enable register (SPE_0_EIRQ_EN)	32	R/W	0000_0000h	17.4.9/720
40	SPE0 Revolution Counter register (SPE_0_REV_CNT)	32	R/W	0000_0000h	17.4.10/ 721
44	SPE0 Revolution Counter Compare register (SPE_0_REV_CMP)	32	R/W	0000_0000h	17.4.11/ 721

17.4.1 SPE0 Control Status Register (SPE_0_CTRL_STAT)

Address: 800h base + 0h offset = 800h



SPE_0_CTRL_STAT field descriptions

Field	Description
0–7 FSOL	Fast Shut-Off Level for TOM[i] channel 0 to 7.
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

SPE_0_CTRL_STAT field descriptions (continued)

Field	Description
9–11 NIP	New input pattern that was detected. **NOTE:** This bit field mirrors the new input pattern. SPE internal functionality is triggered on each change of this bit field.
12 PDIR	Previous rotation direction. 0 Rotation direction is 0 according to SPE[i]_PAT register. 1 Rotation direction is 1 according to SPE[i]_PAT register.
13–15 PIP	Previous input pattern that was detected by a regular input pattern change.
16 ADIR	Actual rotation direction. 0 Rotation direction is 0 according to SPE[i]_PAT register. 1 Rotation direction is 1 according to SPE[i]_PAT register.
17–19 AIP	Actual input pattern that was detected by a regular input pattern change.
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–23 SPE_PAT_PTR	Pattern selector for TOM output signals. Actual index into the SPE[i]_OUT_PAT[x] register table. Each register SPE[i]_OUT_PAT[x] is fixed assigned to one bit field IPx_PAT of register SPE[i]_PAT. Thus, the pointer SPE[i]_PAT_PTR represents an index to the selected SPE[i]_OUT_PAT[x] register as well as the actual detected input pattern IPx_PAT.
24 FSOM	Fast Shut-Off Mode. 0 Fast Shut-Off mode disabled. 1 Fast Shut-Off mode enabled.
25 TIM_SEL	Select TIM input signal. SPE0: 0 TIM0_CH0..2 1 TIM1_CH0..2 SPE1: 0 TIM0_CH3..5 1 TIM1_CH3..5 SPE2: 0 TIM2_CH0..2 1 Unused SPE3: 0 TIM2_CH3..5 1 Unused

Table continues on the next page...

SPE_0_CTRL_STAT field descriptions (continued)

Field	Description
26–27 TRIG_SEL	Select trigger input signal. 00 SPE[i]_NIPD selected. 01 TOM_CH0_TRIG_CCU0 selected. 10 TOM_CH0_TRIG_CCU1 selected. 11 Select trigger input signal.
28 SIE2	SPE Input enable for TIM_CHz(48). 0 SPE Input is disabled. 1 SPE Input is enabled..
29 SIE1	SPE Input enable for TIM_CHy(48). 0 SPE Input is disabled. 1 SPE Input is enabled.
30 SIE0	SPE Input enable for TIM_CHx(48). 0 SPE Input is disabled. 1 SPE Input is enabled.
31 EN	SPE Submodule enable. 0 SPE disabled. 1 SPE enabled.

17.4.2 SPE0 Input Pattern Register (SPE_0_PAT)

NOTE

Only the first block of valid input patterns defines the commutator. All input pattern following the first marked invalid input pattern are ignored.

Address: 800h base + 4h offset = 804h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	IP7_PAT			IP7_ VAL	IP6_PAT			IP6_ VAL	IP5_PAT			IP5_ VAL	IP4_PAT			IP4_ VAL
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	IP3_PAT			IP3_ VAL	IP2_PAT			IP2_ VAL	IP1_PAT			IP1_ VAL	IP0_PAT			IP0_ VAL
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPE_0_PAT field descriptions

Field	Description
0–2 IP7_PAT	Input pattern 7. Bit field defines the seventh input pattern of the SPE input signals: Bit 0 defines the TIM[i]_CHz(48) input signal. Bit 1 defines the TIM[i]_CHy(48) input signal. Bit 2 defines the TIM[i]_CHx(48) input signal.
3 IP7_VAL	Input pattern 7 is a valid pattern. 0 Pattern invalid. 1 Pattern valid.
4–6 IP6_PAT	Input pattern 6. Bit field defines the seventh input pattern of the SPE input signals: Bit 4 defines the TIM[i]_CHz(48) input signal. Bit 5 defines the TIM[i]_CHy(48) input signal. Bit 6 defines the TIM[i]_CHx(48) input signal.
7 IP6_VAL	Input pattern 6 is a valid pattern. 0 Pattern invalid. 1 Pattern valid.
8–10 IP5_PAT	Input pattern 5. Bit field defines the seventh input pattern of the SPE input signals: Bit 8 defines the TIM[i]_CHz(48) input signal. Bit 9 defines the TIM[i]_CHy(48) input signal. Bit 10 defines the TIM[i]_CHx(48) input signal.
11 IP5_VAL	Input pattern 5 is a valid pattern. 0 Pattern invalid. 1 Pattern valid.
12–14 IP4_PAT	Input pattern 4. Bit field defines the seventh input pattern of the SPE input signals: Bit 12 defines the TIM[i]_CHz(48) input signal. Bit 13 defines the TIM[i]_CHy(48) input signal. Bit 14 defines the TIM[i]_CHx(48) input signal.
15 IP4_VAL	Input pattern 4 is a valid pattern. 0 Pattern invalid. 1 Pattern valid.
16–18 IP3_PAT	Input pattern 3. Bit field defines the seventh input pattern of the SPE input signals: Bit 16 defines the TIM[i]_CHz(48) input signal. Bit 17 defines the TIM[i]_CHy(48) input signal. Bit 18 defines the TIM[i]_CHx(48) input signal.

Table continues on the next page...

SPE_0_PAT field descriptions (continued)

Field	Description
19 IP3_VAL	Input pattern 3 is a valid pattern. 0 Pattern invalid. 1 Pattern valid.
20–22 IP2_PAT	Input pattern 2. Bit field defines the seventh input pattern of the SPE input signals: Bit 20 defines the TIM[i]_CHz(48) input signal. Bit 21 defines the TIM[i]_CHy(48) input signal. Bit 22 defines the TIM[i]_CHx(48) input signal.
23 IP2_VAL	Input pattern 2 is a valid pattern. 0 Pattern invalid. 1 Pattern valid.
24–26 IP1_PAT	The Input Pattern 1 Bit field defines the seventh input pattern of the SPE input signals: Bit 24 defines the TIM[i]_CHz(48) input signal. Bit 25 defines the TIM[i]_CHy(48) input signal. Bit 26 defines the TIM[i]_CHx(48) input signal.
27 IP1_VAL	Input pattern 1 is a valid pattern. 0 Pattern invalid. 1 Pattern valid.
28–30 IP0_PAT	Input pattern 0. Bit field defines the seventh input pattern of the SPE input signals: Bit 28 defines the TIM[i]_CHz(48) input signal. Bit 29 defines the TIM[i]_CHy(48) input signal. Bit 30 defines the TIM[i]_CHx(48) input signal.
31 IP0_VAL	Input pattern 0 is a valid pattern. 0 Pattern invalid. 1 Pattern valid.

17.4.3 SPE0 Output Pattern n Register (SPE_0_OUT_PAT*n*)

NOTE

Register SPE_OUT_PAT[n] defines the output selection for TOM[i]_CH0 to TOM[i]_CH7 depending on actual input pattern IP[n]_PAT with n:0..7.

Address: 800h base + 8h offset + (4d × i), where i=0d to 7d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																SPE_OUT_PAT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPE_0_OUT_PAT*n* field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–31 SPE_OUT_PAT	SPE output control value for TOM_CH0 to TOM_CH7 SPE_OUT_PAT[n+1:n] defines output select signal of TOM[i]_CH[n]. 00 Set SPE_OUT(n) to TOM_CH0_SOUR. 01 Set SPE_OUT(n) to TOM_CH1_SOUR. 10 Set SPE_OUT(n) to '0'. 11 Set SPE_OUT(n) to '1'.

17.4.4 SPE0 Output Control Register (SPE_0_OUT_CTRL)

Address: 800h base + 28h offset = 828h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																SPE_OUT_CTRL															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPE_0_OUT_CTRL field descriptions

Field	Description
0–15 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
16–31 SPE_OUT_CTRL	SPE output control value for TOM_CH0 to TOM_CH7. SPE_OUT_CTRL[n+1:n] defines output select signal of TOM_CHn.

Table continues on the next page...

SPE_0_OUT_CTRL field descriptions (continued)

Field	Description
	NOTE: Current output control selection for SPE[i]_OUT(0..7). 00 Set SPE_OUT(n) to TOM_CH0_SOUR, with n = 0..7. 01 Set SPE_OUT(n) to TOM_CH1_SOUR, with n = 0..7. 10 Set SPE_OUT(n) to '0', with n = 0..7. 11 Set SPE_OUT(n) to '1', with n = 0..7.

17.4.5 SPE0 Interrupt Request Notify Register (SPE_0_IRQ_NOTIFY)

NOTE

Bits 27:31 will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Address: 800h base + 2Ch offset = 82Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0											SPE_RCMP	SPE_BIS	SPE_PERR	SPE_DCHG	SPE_NIPD
W												w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPE_0_IRQ_NOTIFY field descriptions

Field	Description
0–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 SPE_RCMP	SPE revolution counter match event. 0 No interrupt occurred. 1 New input pattern detected interrupt occurred.
28 SPE_BIS	Bouncing input signal detected. 0 No interrupt occurred. 1 New input pattern detected interrupt occurred.
29 SPE_PERR	Wrong or invalid pattern detected at input. 0 No interrupt occurred. 1 New input pattern detected interrupt occurred.
30 SPE_DCHG	SPE_DIR bit changed on behalf of new input pattern. 0 No interrupt occurred. 1 New input pattern detected interrupt occurred.
31 SPE_NIPD	New input pattern interrupt occurred. 0 No interrupt occurred. 1 New input pattern detected interrupt occurred.

17.4.6 SPE0 Interrupt Request Enable Register (SPE_0_IRQ_EN)

Address: 800h base + 30h offset = 830h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0											SPE_RCMP_IRQ_EN	SPE_BIS_IRQ_EN	SPE_PERR_IRQ_EN	SPE_DCHG_IRQ_EN	SPE_NIPD_IRQ_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPE_0_IRQ_EN field descriptions

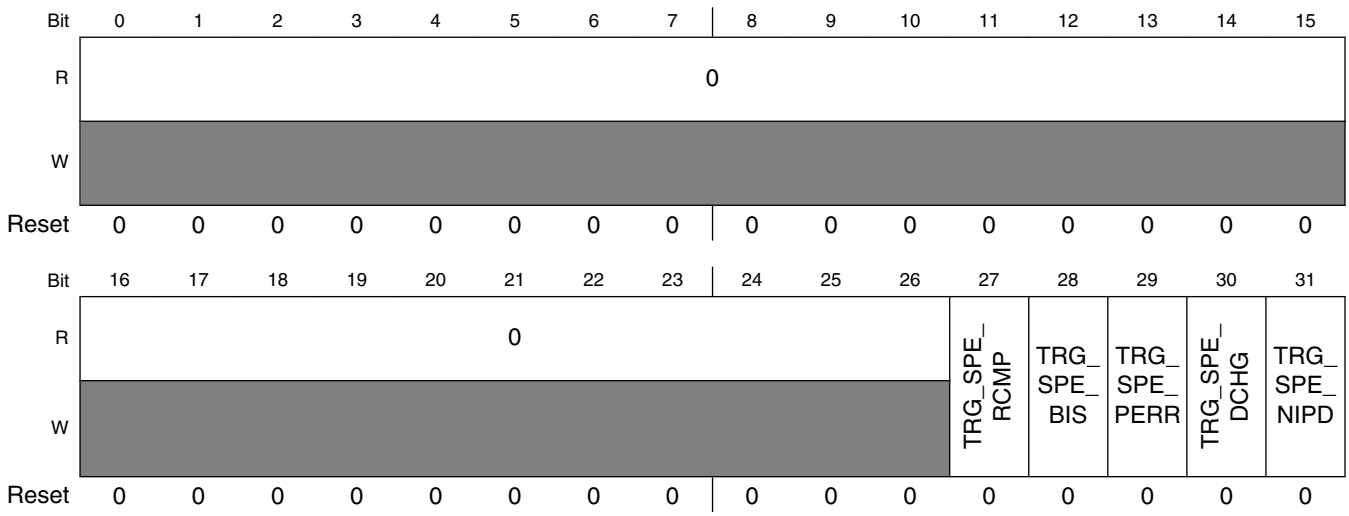
Field	Description
0–26 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
27 SPE_RCMP_IRQ_EN	SPE_RCMP_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
28 SPE_BIS_IRQ_EN	SPE_BIS_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
29 SPE_PERR_IRQ_EN	SPE_PERR_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
30 SPE_DCHG_IRQ_EN	SPE_DCHG_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM. 1 Enable interrupt, interrupt is visible outside GTM.
31 SPE_NIPD_IRQ_EN	SPE_NIPD_IRQ interrupt enable. 0 Disable interrupt, interrupt is not visible outside GTM 1 Enable interrupt, interrupt is visible outside GTM.

17.4.7 SPE0 Interrupt Request Force Interrupt Register (SPE_0_IRQ_FORCINT)

NOTE

Bit[28:31] is cleared automatically after interrupt is released, and is write protected by bit GTM_CTRL[RF_PROT].

Address: 800h base + 34h offset = 834h



SPE_0_IRQ_FORCINT field descriptions

Field	Description
0–26 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
27 TRG_SPE_RCMP	Force interrupt of SPE_RCMP. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in SPE_IRQ_NOTIFY register.
28 TRG_SPE_BIS	Force interrupt of SPE_BIS. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in SPE_IRQ_NOTIFY register.
29 TRG_SPE_PERR	Force interrupt of SPE_PERR. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in SPE_IRQ_NOTIFY register.
30 TRG_SPE_DCHG	Force interrupt of SPE_DCHG. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in SPE_IRQ_NOTIFY register.

Table continues on the next page...

SPE_0_IRQ_FORCINT field descriptions (continued)

Field	Description
31 TRG_SPE_NIPD	Force interrupt of SPE_NIPD. 0 Corresponding bit in status register will not be forced. 1 Assert corresponding field in SPE_IRQ_NOTIFY register.

17.4.8 SPE0 Interrupt Request Mode Register (SPE_0_IRQ_MODE)

Address: 800h base + 38h offset = 838h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0														IRQ_MODE	
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

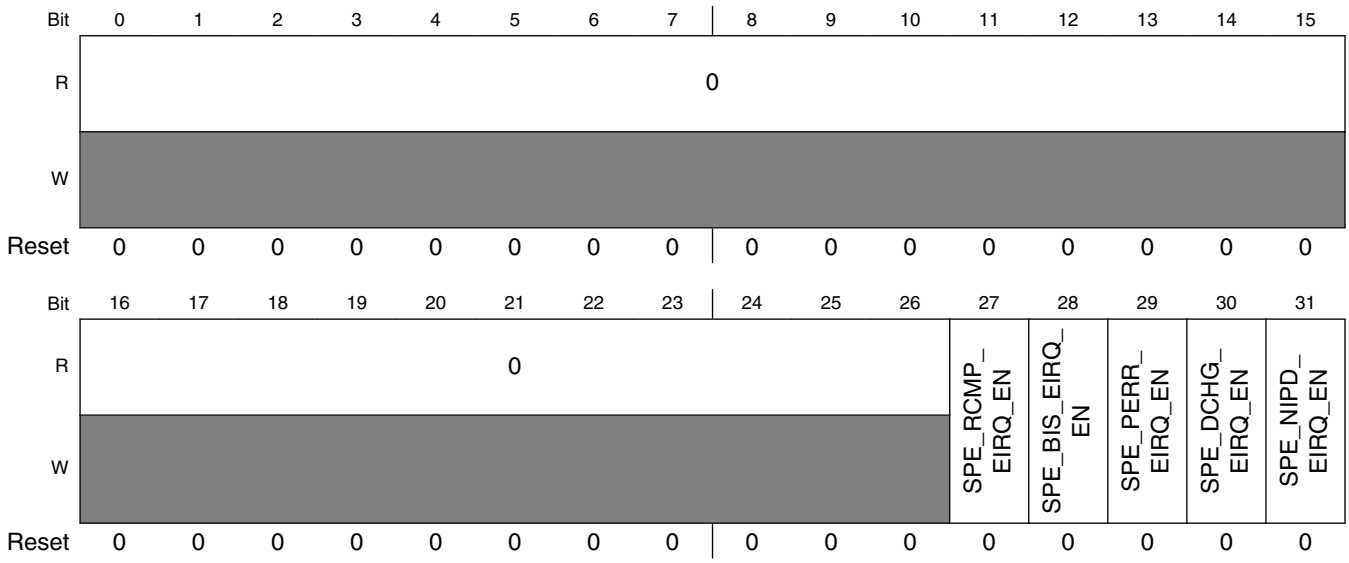
* Notes:
- Bits 28-31 do not reset to a given value.

SPE_0_IRQ_MODE field descriptions

Field	Description
0–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–31 IRQ_MODE	IRQ mode selection. **NOTE:** The interrupt modes are described in section 2.5. 00 Level mode. 01 Pulse mode 10 Pulse-Notify mode. 11 Single-Pulse mode.

17.4.9 SPE0 Error Interrupt Request Enable register (SPE_0_EIRQ_EN)

Address: 800h base + 3Ch offset = 83Ch



SPE_0_EIRQ_EN field descriptions

Field	Description
0–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 SPE_RCMP_EIRQ_EN	SPE_BIS_EIRQ error interrupt enable. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.
28 SPE_BIS_EIRQ_EN	SPE_BIS_EIRQ error interrupt enable. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.
29 SPE_PERR_EIRQ_EN	SPE_PERR_EIRQ interrupt enable. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.
30 SPE_DCHG_EIRQ_EN	SPE_DCHG_EIRQ error interrupt enable. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.
31 SPE_NIPD_EIRQ_EN	SPE_NIPD_EIRQ interrupt enable. 0 Disable error interrupt, error interrupt is not visible outside GTM-IP. 1 Enable error interrupt, error interrupt is visible outside GTM-IP.

17.4.10 SPE0 Revolution Counter register (SPE_0_REV_CNT)

Address: 800h base + 40h offset = 840h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								REV_CNT																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPE_0_REV_CNT field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 REV_CNT	Input signal revolution counter. The counter is running if SPE module is enabled (bit SPE_EN). REV_CNT is incrementing if SPE_PAT_PTR is incrementing. REV_CNT is decrementing if SPE_PAT_PTR is decrementing.

17.4.11 SPE0 Revolution Counter Compare register (SPE_0_REV_CMP)

Address: 800h base + 44h offset = 844h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0								REV_CMP																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPE_0_REV_CMP field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–31 REV_CMP	Input signal revolution counter compare value. The SPE[i]_RCMP interrupt is asserted when the SPE[i]_REV_CNT value equals the SPE[i]_REV_CMP register. **NOTE:** SPE[i]_RCMP is only raised if an incrementation or decremention of SPE[i]_REV_CNT is applied, due to a input signal change. Any update of SPE[i]_REV_CNT or SPE[i]_REV_CMP via AEI does not raise an SPE[i]_RCMP interrupt.

Chapter 18
Interrupt Concentrator Module (ICM)

18.1 ICM Overview

The Interrupt Concentrator Module (ICM):

- Bundles some interrupt lines (from the various submodules/channels) into interrupt line groups. As a result, a smaller amount of group interrupt lines is routed to the outside of the GTM module.
- Directly feeds-through some interrupt lines to the outside of the GTM.

The interrupt sources have to be enabled inside the submodules/channels in order for them to participate in the bundling process.

The feed through architecture of bundled interrupt lines is used for the AEI, ARU, BRC, CMP, SPE, PSM, TIM, DPLL, TOM, ATOM and MCS submodules.

To determine the detailed interrupt source, the CPU has to read the NOTIFY submodule/ channel interrupt notification register and serve the channel individual interrupt.

NOTE

The interrupts are only visible inside the ICM and in consequence outside of the GTM, only when the interrupt is enabled inside the submodules themselves.

18.2 Bundling

The ICM either feeds-through individual interrupt lines to the outside of the GTM or bundles them into groups and and then routes the group interrupt lines to the outside of the GTM.

18.2.1 GTM Infrastructure Interrupt Bundling

The first interrupt group is composed of interrupt lines from the GTM's infrastructure and safety components, which include the AEI, ARU, BRC, PSM, SPE and CMP submodules. Each individual channel from each submodule has its own interrupt line that is feed-through to the outside of the GTM.

As a result, an asserted interrupt request can be sensed by the CPU and used to directly determine which submodule/channel asserted it. The condition of these interrupt sources is stored the **ICM_IRQG_0** interrupt group register. Although this register is readable, it is typically not read by the CPU.

18.2.2 DPLL Interrupt Bundling

Each of interrupt lines from the DPLL submodule is feed-through to the outside of the GTM. As a result, an asserted interrupt request can be sensed by the CPU and used to directly determine which submodule/channel asserted it. The condition of these interrupt sources is stored the ICM_IRQG_1 interrupt group register. Although this register is readable, it is typically not read by the CPU.

18.2.3 TIM Interrupt Bundling

The TIM[i] submodules operate on input signals from the outside of the GTM. Each TIM submodule channel can generate six individual interrupt requests (if they are enabled in the TIM channel). The ICM bundles six interrupt lines into one group interrupt line for each TIM channel that is connected to the ICM. The group interrupt lines are routed to the outside of the GTM. When a group interrupt request is asserted, the CPU must read the interrupt notification registers in the TIM submodule channels to determine the source of the interrupt request.

The ICM_IRQG_2 and ICM_IRQG_3 registers mirror the TIM channel interrupts. Although these registers are readable, they are typically not read by the CPU.

18.2.4 MCS Interrupt Bundling

Each MCS submodule has eight channels, each with one interrupt line. The interrupt line from each channel is routed to the ICM submodule, where it is is directly feed-through to the outside of the GTM.

The interrupt status for each MCS channel is stored in the **ICM_IRQG_4** and **ICM_IRQG_5** registers. Although these registers are readable, they are typically not read by the CPU.

18.2.5 TOM and ATOM Interrupt Bundling

For the TOM and ATOM submodules, the interrupts are bundled within the ICM submodule a second time to reduce external interrupt lines. The interrupts are OR-ed in a manner that one GTM external interrupt line represents two adjacent TOM or ATOM channel interrupts. For TOM[i] and ATOM[i], the bundling is shown in Table 18-1

Table 18-1. TOM and ATOM interrupt bundling within ICM

TOM[i]-input IRQs [i]=0..number of TOM's-1	TOM-output IRQs (OR-ed)		ATOM[i]-input IRQs [i]=0..number of ATOM's-1	ATOM-output IRQs (OR-ed)
TOM[i]_CH0_IRQ	GTM_TOM[i]_IRQ[0]		ATOM[i]_CH0_IRQ	GTM_ATOM[i]_IR Q[0]
TOM[i]_CH1_IRQ		ATOM[i]_CH1_IRQ		
TOM[i]_CH2_IRQ	GTM_TOM[i]_IRQ[1]		ATOM[i]_CH2_IRQ	GTM_ATOM[i]_IR Q[1]
TOM[i]_CH3_IRQ		ATOM[i]_CH3_IRQ		
TOM[i]_CH4_IRQ	GTM_TOM[i]_IRQ[2]		ATOM[i]_CH4_IRQ	GTM_ATOM[i]_IR Q[2]
TOM[i]_CH5_IRQ			ATOM[i]_CH5_IRQ	
TOM[i]_CH6_IRQ	GTM_TOM[i]_IRQ[3]		ATOM[i]_CH6_IRQ	GTM_ATOM[i]_IR Q[3]
TOM[i]_CH7_IRQ			ATOM[i]_CH7_IRQ	
TOM[i]_CH8_IRQ	GTM_TOM[i]_IRQ[4]			
TOM[i]_CH9_IRQ				
TOM[i]_CH10_IRQ	GTM_TOM[i]_IRQ[5]			
TOM[i]_CH11_IRQ				
TOM[i]_CH12_IRQ	GTM_TOM[i]_IRQ[6]			
TOM[i]_CH13_IRQ				
TOM[i]_CH14_IRQ	GTM_TOM[i]_IRQ[7]			
TOM[i]_CH15_IRQ				

The interrupts coming from the TOM[i] submodules are registered in the **ICM_IRQG_6,** ICM_IRQG_7, and ICM_IRQG_8 registers. Always, two TOM's are bundled in one ICM register. For example, TOM0 and TOM1 are bundled in the ICM_IRQG_6 register. To identify the TOM submodule channel where the interrupt occurred, the CPU must read the **ICM_IRQG_6, ICM_IRQG_7, and ICM_IRQG_8** registers first before it goes to the TOM submodule channel itself.

The bits in the **ICM_IRQG_6, ICM_IRQG_7, and ICM_IRQG_8** registers are cleared automatically, when their corresponding interrupt in the submodule channels is cleared.

The interrupts coming from the ATOM[i] submodules are registered in the ICM_IRQG_9, ICM_IRQG_10, and ICM_IRQG_11 registers. Always, four ATOM's are bundled in one ICM register. For example, ATOM0, ATOM1, ATOM2, and ATOM3 are bundled in the ICM_IRQG_9 register. To identify the ATOM submodule channel where the interrupt occurred, the CPU must read the ICM_IRQG_9, ICM_IRQG_10, and ICM_IRQG_11 registers first before it goes to the ATOM submodule channel itself.

The bits in the ICM_IRQG_9, ICM_IRQG_10, and ICM_IRQG_11 registers are cleared automatically, when their corresponding interrupt in the submodule channels is cleared.

18.2.6 Module error interrupt bundling

The Module Error Interrupt group handles the error interrupts coming from the BRC, FIFO, TIM, MCS, SPE, CMP, DPLL submodule of the GTM. The Module Error interrupts are additionally identified in the ICM_IRQG_MEI error interrupt group register. This register is typically not read by the CPU, but it is readable.

The ICM_IRQG_MEI register bits are cleared automatically when their corresponding error interrupt in the submodule is cleared.

18.2.7 FIFO channel error interrupt bundling

The FIFO Channel Error Interrupt group handles the error interrupts coming from the FIFO channel of the GTM. The FIFO Channel Error interrupts are additionally identified in the ICM_IRQG_CEI0 error interrupt group register. This register is typically not read by the CPU, but it is readable.

The ICM_IRQG_CEI0 register bits are cleared automatically when their corresponding error interrupt in the submodule channel is cleared.

18.2.8 TIM channel error interrupt bundling

The TIM Channel Error Interrupt group handles the error interrupts coming from the TIM channel of the GTM. The TIM Channel Error interrupts are additionally identified for the submodules TIM0, TIM1, TIM2 and TIM3 in the ICM_IRQG_CEI1 error interrupt group register and for the submodules TIM4, TIM5 and TIM6 in the ICM_IRQG_CEI2 error interrupt group register. These registers are typically not read by the CPU, but they are readable.

The ICM_IRQG_CEI1 and ICM_IRQG_CEI2 register bits are cleared automatically when their corresponding error interrupt in the submodule channel is cleared.

18.2.9 MCS channel error interrupt bundling

The MCS Channel Error Interrupt group handles the error interrupts coming from the MCS channel of the GTM. The MCS Channel Error interrupts are additionally ICM_531, ICM_1390, ICM_1890, ICM_1391, ICM_1876, ICM_1877, ICM_1880, ICM_1878, ICM_1879, ICM_1881, ICM_1882, ICM_1883, ICM_1884, ICM_1885, and ICM_1886 (see Automotive Electronics GTM-IP Specification Revision 1.5.4 Robert Bosch GmbH 459/505 06.07.2012 Confidential) identified for the submodules MCS0, MCS1, MCS2 and MCS3 in the ICM_IRQG_CEI3 error interrupt group register and for the submodules MCS4, MCS5 and MCS6 in the ICM_IRQG_CEI4 error interrupt group register. These registers are typically not read by the CPU, but they are readable.

The ICM_IRQG_CEI3 and ICM_IRQG_CEI4 register bits are cleared automatically when their corresponding error interrupt in the submodule channel is cleared.

18.3 ICM Interrupt Signals

Table 18-2 shows direct or bundled interrupt signals that are routed to the outside of the GTM.

Table 18-2. ICM Interrupt Signals

Signal	Description
GTM_AEI_IRQ	AEI Shared interrupt
GTM_ARU_IRQ[2:0]	[0]: ARU_NEW_DATA0 Interrupt [1]: ARU_NEW_DATA1 Interrupt [2]: ARU_ACC_ACK Interrupt
GTM_BRC_IRQ	BRC Shared interrupt
GTM_CMP_IRQ	CMP Shared interrupt

Table continues on the next page...

Table 18-2. ICM Interrupt Signals (continued)

Signal	Description
GTM_SPE[i]_IRQ	SPE[i] Shared interrupt (i: 0..number of SPE's –1)
GTM_PSM[i]_IRQx	PSM Shared interrupts (x: 0…7) (i: 0..number of PSM's – 1)
GTM_DPLL_IRQ0	DPLL_DCGI: DPLL direction change interrupt
GTM_DPLL_IRQ1	DPLL_EDI; DPLL enable or disable interrupt
GTM_DPLL_IRQ2	DPLL_TINI: DPLL TRIGGER minimum hold time (THMI) violation detected
GTM_DPLL_IRQ3	DPLL_TAXI: DPLL TRIGGER maximum hold time (THMA) violation detected
GTM_DPLL_IRQ4	DPLL_SISI: DPLL STATE inactive slope detected
GTM_DPLL_IRQ5	DPLL_TISI: DPLL TRIGGER inactive slope detected
GTM_DPLL_IRQ6	DPLL_MSI: DPLL Missing STATE interrupt
GTM_DPLL_IRQ7	DPLL_MTI: DPLL Missing TRIGGER interrupt
GTM_DPLL_IRQ8	DPLL_SASI: DPLL STATE active slope detected
GTM_DPLL_IRQ9	DPLL_TASI: DPLL TRIG. active slope detected while NTI_CNT is 0
GTM_DPLL_IRQ10	DPLL_PWI: DPLL Plausibility window (PVT) violation int. of TRIG.
GTM_DPLL_IRQ11	DPLL_W2I: DPLL Write access to RAM region 2 interrupt
GTM_DPLL_IRQ12	DPLL_W1I: DPLL Write access to RAM region 1b or 1c int.
GTM_DPLL_IRQ13	DPLL_GL1I: DPLL Get of lock interrupt for SUB_INC1
GTM_DPLL_IRQ14	DPLL_LL1I: DPLL Lost of lock interrupt for SUB_INC1
GTM_DPLL_IRQ15	DPLL_EI: DPLL Error interrupt
GTM_DPLL_IRQ16	DPLL_GL2I: DPLL Get of lock interrupt for SUB_INC2
GTM_DPLL_IRQ17	DPLL_LL2I: DPLL Lost of lock interrupt for SUB_INC2
GTM_DPLL_IRQ18	DPLL_TE0I: DPLL TRIGGER event interrupt 0
GTM_DPLL_IRQ19	DPLL_TE1I: DPLL TRIGGER event interrupt 1
GTM_DPLL_IRQ20	DPLL_TE2I: DPLL TRIGGER event interrupt 2
GTM_DPLL_IRQ21	DPLL_TE3I: DPLL TRIGGER event interrupt 3
GTM_DPLL_IRQ22	DPLL_TE4I; DPLL TRIGGER event interrupt 4
GTM_DPLL_IRQ23	DPLL_CDTI; DPLL calculated duration interrupt for
GTM_DPLL_IRQ24	DPLL_CDSI; DPLL calculated duration interrupt for
GTM_DPLL_IRQ25	DPLL_TORI; TRIGGER out of range interrupt
GTM_DPLL_IRQ26	DPLL_SORI; STATE out of range interrupt
GTM_TIM[i]_IRQx	TIM Shared interrupts (i: 0..number of TIM's –1) (x: 0..7)
GTM_MCS[i]_IRQx	MCS Interrupt for channel x (x: 0…7) (i: 0..number of MCS's-1)
GTM_TOM[i]_IRQx	TOM Shared interrupts for x:0..7 = {ch0||ch1,...,ch14||ch15} (i: 0..number of TOM's –1)
GTM_ATOM[i]_IRQx	ATOM Shared interrupts for x:0..3 = {ch0||ch1,...,ch6||ch7} (i: 0..number of ATOM's –1)
GTM_ERR_IRQ	GTM Error Interrupt

18.4 Memory Map and Registers

The Interrupt Concentrator Module (ICM) registers are described as follows:

ICM memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	GTM Infrastructure Interrupt Group Register (ICM_IRQG_0)	32	R	0000_0000h	18.4.1/729
4	DPLL Interrupt Group Register (ICM_IRQG_1)	32	R	0000_0000h	18.4.2/735
8	TIM Interrupt Group 0 Register (ICM_IRQG_2)	32	R	0000_0000h	18.4.3/738
C	TIM Interrupt Group 1 Register (ICM_IRQG_3)	32	R	0000_0000h	18.4.4/742
10	MCS Interrupt Group 0 Register (ICM_IRQG_4)	32	R	0000_0000h	18.4.5/749
14	MCS Interrupt Group 1 Register (ICM_IRQG_5)	32	R	0000_0000h	18.4.6/754
18	TOM Interrupt Group 0 Register (ICM_IRQG_6)	32	R	0000_0000h	18.4.7/758
1C	TOM Interrupt Group 1 Register (ICM_IRQG_7)	32	R	0000_0000h	18.4.8/763
20	TOM Interrupt Group 2 Register (ICM_IRQG_8)	32	R	0000_0000h	18.4.9/768
24	ATOM Interrupt Group 0 Register (ICM_IRQG_9)	32	R	0000_0000h	18.4.10/ 773
28	ATOM Interrupt Group 1 Register (ICM_IRQG_10)	32	R	0000_0000h	18.4.11/ 778
2C	ATOM Interrupt Group 2 Register (ICM_IRQG_11)	32	R	0000_0000h	18.4.12/ 783
30	ICM IRQG Module Error Interrupt register (ICM_IRQG_MEI)	32	R	0000_0000h	18.4.13/ 788
34	ICM Channel Error Interrupt Request Group 0 register (ICM_IRQG_CEI0)	32	R	0000_0000h	18.4.14/ 792
38	Channel Error Interrupt Request Group 1 register (ICM_IRQG_CEI1)	32	R	0000_0000h	18.4.15/ 795
3C	Channel Error Interrupt Request Group 2 register (ICM_IRQG_CEI2)	32	R	0000_0000h	18.4.16/ 800
40	Channel Error Interrupt Request Group 3 register (ICM_IRQG_CEI3)	32	R	0000_0000h	18.4.17/ 804
44	Channel Error Interrupt Request Group 4 register (ICM_IRQG_CEI4)	32	R	0000_0000h	18.4.18/ 809

18.4.1 GTM Infrastructure Interrupt Group Register (ICM_IRQG_0)

NOTE

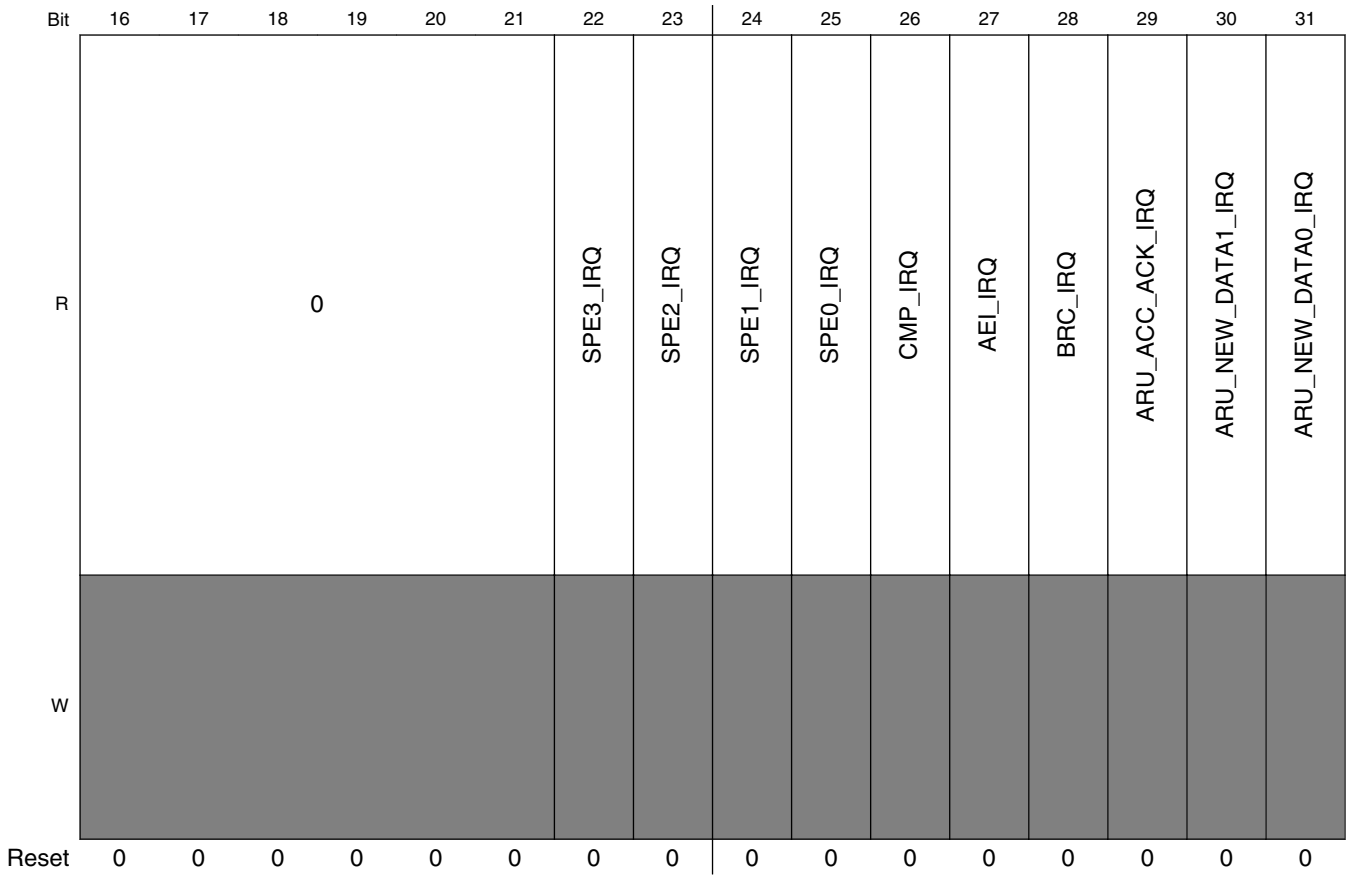
For bit[0:15] and bit[22:31], the bit is only set when the interrupt is enabled in the interrupt enable register of the corresponding submodule.

NOTE

For bit[0:15] when set, this bit represents one of the four interrupt sources FIFO_[n]_EMPTY, FIFO_[n]_FULL, FIFO_[n]_LOWER_WM or FIFO_[n]_UPPER_WM.

Address: 600h base + 0h offset = 600h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	PSM1_CH7_IRQ	PSM1_CH6_IRQ	PSM1_CH5_IRQ	PSM1_CH4_IRQ	PSM1_CH3_IRQ	PSM1_CH2_IRQ	PSM1_CH1_IRQ	PSM1_CH0_IRQ	PSM0_CH7_IRQ	PSM0_CH6_IRQ	PSM0_CH5_IRQ	PSM0_CH4_IRQ	PSM0_CH3_IRQ	PSM0_CH2_IRQ	PSM0_CH1_IRQ	PSM0_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



ICM_IRQG_0 field descriptions

Field	Description
0 PSM1_CH7_IRQ	PSM1 shared submodule channel 7 interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the four interrupt sources FIFO_[x]_EMPTY, FIFO_[x]_FULL, FIFO_[x]_LOWER_WM or FIFO_[x]_UPPER_WM 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
1 PSM1_CH6_IRQ	PSM1 shared submodule channel 6 interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the four interrupt sources FIFO_[x]_EMPTY, FIFO_[x]_FULL, FIFO_[x]_LOWER_WM or FIFO_[x]_UPPER_WM 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
2 PSM1_CH5_IRQ	PSM1 shared submodule channel 5 interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule.

Table continues on the next page...

ICM_IRQG_0 field descriptions (continued)

Field	Description
	When set this bit represents one of the four interrupt sources FIFO_[x]_EMPTY, FIFO_[x]_FULL, FIFO_[x]_LOWER_WM or FIFO_[x]_UPPER_WM 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
3 PSM1_CH4_IRQ	PSM1 shared submodule channel 4 interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the four interrupt sources FIFO_[x]_EMPTY, FIFO_[x]_FULL, FIFO_[x]_LOWER_WM or FIFO_[x]_UPPER_WM 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
4 PSM1_CH3_IRQ	PSM1 shared submodule channel 3 interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the four interrupt sources FIFO_[x]_EMPTY, FIFO_[x]_FULL, FIFO_[x]_LOWER_WM or FIFO_[x]_UPPER_WM 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
5 PSM1_CH2_IRQ	PSM1 shared submodule channel 2 interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the four interrupt sources FIFO_[x]_EMPTY, FIFO_[x]_FULL, FIFO_[x]_LOWER_WM or FIFO_[x]_UPPER_WM 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
6 PSM1_CH1_IRQ	PSM1 shared submodule channel 1 interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the four interrupt sources FIFO_[x]_EMPTY, FIFO_[x]_FULL, FIFO_[x]_LOWER_WM or FIFO_[x]_UPPER_WM 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
7 PSM1_CH0_IRQ	PSM1 shared submodule channel 0 interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set, this bit represents one of the four interrupt sources FIFO_[x]_EMPTY, FIFO_[x]_FULL, FIFO_[x]_LOWER_WM or FIFO_[x]_UPPER_WM 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_0 field descriptions (continued)

Field	Description
8 PSM0_CH7_IRQ	PSM0 shared submodule channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
9 PSM0_CH6_IRQ	PSM0 shared submodule channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
10 PSM0_CH5_IRQ	PSM0 shared submodule channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
11 PSM0_CH4_IRQ	PSM0 shared submodule channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
12 PSM0_CH3_IRQ	PSM0 shared submodule channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
13 PSM0_CH2_IRQ	PSM0 shared submodule channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
14 PSM0_CH1_IRQ	PSM0 shared submodule channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
15 PSM0_CH0_IRQ	PSM0 shared submodule channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
16–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 SPE3_IRQ	SPE3 shared submodule interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
23 SPE2_IRQ	SPE2 shared submodule interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
24 SPE1_IRQ	SPE1 shared submodule interrupt.

Table continues on the next page...

ICM_IRQG_0 field descriptions (continued)

Field	Description
	This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
25 SPE0_IRQ	SPE0 shared submodule interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
26 CMP_IRQ	CMP shared submodule interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
27 AEI_IRQ	AEI_IRQ interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
28 BRC_IRQ	BRC shared submodule interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
29 ARU_ACC_ ACK_IRQ	ARU_ACC_ACK interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
30 ARU_NEW_ DATA1_IRQ	ARU_NEW_DATA1 interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
31 ARU_NEW_ DATA0_IRQ	ARU_NEW_DATA0 interrupt. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

18.4.2 DPLL Interrupt Group Register (ICM_IRQG_1)

NOTE

Bit[5:31] is only set when the interrupt is enabled in the interrupt enable register of the corresponding submodule.

Address: 600h base + 4h offset = 604h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0					DPLL_SORI_IRQ	DPLL_TORI_IRQ	DPLL_CDSI_IRQ	DPLL_CDTI_IRQ	DPLL_TE4I_IRQ	DPLL_TE3I_IRQ	DPLL_TE2I_IRQ	DPLL_TE1I_IRQ	DPLL_TE0I_IRQ	DPLL_LL2I_IRQ	DPLL_GL2I_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	DPLL_EI_IRQ	DPLL_LLI_IRQ	DPLL_GLI_IRQ	DPLL_W1I_IRQ	DPLL_W2I_IRQ	DPLL_PWI_IRQ	DPLL_TASI_IRQ	DPLL_SASI_IRQ	DPLL_MTI_IRQ	DPLL_MSI_IRQ	DPLL_TISI_IRQ	DPLL_SISI_IRQ	DPLL_TAXI_IRQ	DPLL_TINI_IRQ	DPLL_EDI_IRQ	DPLL_DCGI_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICM_IRQG_1 field descriptions

Field	Description
0–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 DPLL_SORI_IRQ	DPLL calculated duration interrupt for STATE. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
6 DPLL_TORI_IRQ	DPLL calculated duration interrupt for TRIGGER. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
7 DPLL_CDSI_IRQ	DPLL calculated duration interrupt for STATE. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
8 DPLL_CDTI_IRQ	DPLL calculated duration interrupt for TRIGGER. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
9 DPLL_TE4I_IRQ	TRIGGER event interrupt 4. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
10 DPLL_TE3I_IRQ	TRIGGER event interrupt 3. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_1 field descriptions (continued)

Field	Description
11 DPLL_TE2I_IRQ	TRIGGER event interrupt 2. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
12 DPLL_TE1I_IRQ	TRIGGER event interrupt 1. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
13 DPLL_TE0I_IRQ	TRIGGER event interrupt 0. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
14 DPLL_LL2I_IRQ	Lost of lock interrupt for SUB_INC2. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
15 DPLL_GL2I_IRQ	Get of lock interrupt for SUB_INC2. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
16 DPLL_EI_IRQ	Error interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
17 DPLL_LLI_IRQ	Lost of lock interrupt for SUB_INC1. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
18 DPLL_GLI_IRQ	Get of lock interrupt for SUB_INC1. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
19 DPLL_W1I_IRQ	Write access to RAM region 1b or 1c interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
20 DPLL_W2I_IRQ	Write access to RAM region 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
21 DPLL_PWI_IRQ	Plausibility window (PVT) violation interrupt of TRIGGER. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
22 DPLL_TASI_IRQ	TRIGGER active slope detected while NTI_CNT is zero. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_1 field descriptions (continued)

Field	Description
23 DPLL_SASI_IRQ	STATE active slope detected. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
24 DPLL_MTI_IRQ	Missing TRIGGER interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
25 DPLL_MSI_IRQ	Missing STATE interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
26 DPLL_TISI_IRQ	TRIGGER inactive slope detected interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
27 DPLL_SISI_IRQ	STATE inactive slope detected interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
28 DPLL_TAXI_IRQ	TRIGGER maximum hold time (THMA) violation detected interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
29 DPLL_TINI_IRQ	TRIGGER minimum hold time (THMI) violation detected interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
30 DPLL_EDI_IRQ	DPLL enable/disable interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
31 DPLL_DCGI_IRQ	TRIGGER direction change detected. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

18.4.3 TIM Interrupt Group 0 Register (ICM_IRQG_2)

NOTE

Bit[0:31] is only set when the interrupt is enabled in the interrupt enable register of the corresponding submodule.

NOTE

When bit[0:31] is set, it represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ.

Address: 600h base + 8h offset = 608h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	TIM3_CH7_IRQ	TIM3_CH6_IRQ	TIM3_CH5_IRQ	TIM3_CH4_IRQ	TIM3_CH3_IRQ	TIM3_CH2_IRQ	TIM3_CH1_IRQ	TIM3_CH0_IRQ	TIM2_CH7_IRQ	TIM2_CH6_IRQ	TIM2_CH5_IRQ	TIM2_CH4_IRQ	TIM2_CH3_IRQ	TIM2_CH2_IRQ	TIM2_CH1_IRQ	TIM2_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TIM1_CH7_IRQ	TIM1_CH6_IRQ	TIM1_CH5_IRQ	TIM1_CH4_IRQ	TIM1_CH3_IRQ	TIM1_CH2_IRQ	TIM1_CH1_IRQ	TIM1_CH0_IRQ	TIM0_CH7_IRQ	TIM0_CH6_IRQ	TIM0_CH5_IRQ	TIM0_CH4_IRQ	TIM0_CH3_IRQ	TIM0_CH2_IRQ	TIM0_CH1_IRQ	TIM0_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICM_IRQG_2 field descriptions

Field	Description
0 TIM3_CH7_IRQ	TIM3 shared interrupt channel 7. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
1 TIM3_CH6_IRQ	TIM3 shared interrupt channel 6. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
2 TIM3_CH5_IRQ	TIM3 shared interrupt channel 5. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
3 TIM3_CH4_IRQ	TIM3 shared interrupt channel 4. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
4 TIM3_CH3_IRQ	TIM3 shared interrupt channel 3. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
5 TIM3_CH2_IRQ	TIM3 shared interrupt channel 2. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
6 TIM3_CH1_IRQ	TIM3 shared interrupt channel 1. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
7 TIM3_CH0_IRQ	TIM3 shared interrupt channel 0. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
8 TIM2_CH7_IRQ	TIM2 shared interrupt channel 7. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
9 TIM2_CH6_IRQ	TIM2 shared interrupt channel 6. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
10 TIM2_CH5_IRQ	TIM2 shared interrupt channel 5. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
11 TIM2_CH4_IRQ	TIM2 shared interrupt channel 4. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_2 field descriptions (continued)

Field	Description
12 TIM2_CH3_IRQ	TIM2 shared interrupt channel 3. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
13 TIM2_CH2_IRQ	TIM2 shared interrupt channel 2. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
14 TIM2_CH1_IRQ	TIM2 shared interrupt channel 1. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
15 TIM2_CH0_IRQ	TIM2 shared interrupt channel 0. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
16 TIM1_CH7_IRQ	TIM1 shared interrupt channel 7. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
17 TIM1_CH6_IRQ	TIM1 shared interrupt channel 6. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
18 TIM1_CH5_IRQ	TIM1 shared interrupt channel 5. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
19 TIM1_CH4_IRQ	TIM1 shared interrupt channel 4. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
20 TIM1_CH3_IRQ	TIM1 shared interrupt channel 3. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
21 TIM1_CH2_IRQ	TIM1 shared interrupt channel 2. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
22 TIM1_CH1_IRQ	TIM1 shared interrupt channel 1. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
23 TIM1_CH0_IRQ	TIM1 shared interrupt channel 0. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_2 field descriptions (continued)

Field	Description
24 TIM0_CH7_IRQ	TIM0 shared interrupt channel 7. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
25 TIM0_CH6_IRQ	TIM0 shared interrupt channel 6. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
26 TIM0_CH5_IRQ	TIM0 shared interrupt channel 5. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
27 TIM0_CH4_IRQ	TIM0 shared interrupt channel 4. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
28 TIM0_CH3_IRQ	TIM0 shared interrupt channel 3. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
29 TIM0_CH2_IRQ	TIM0 shared interrupt channel 2. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
30 TIM0_CH1_IRQ	TIM0 shared interrupt channel 1. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
31 TIM0_CH0_IRQ	TIM0 shared interrupt channel 0. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

18.4.4 TIM Interrupt Group 1 Register (ICM_IRQG_3)

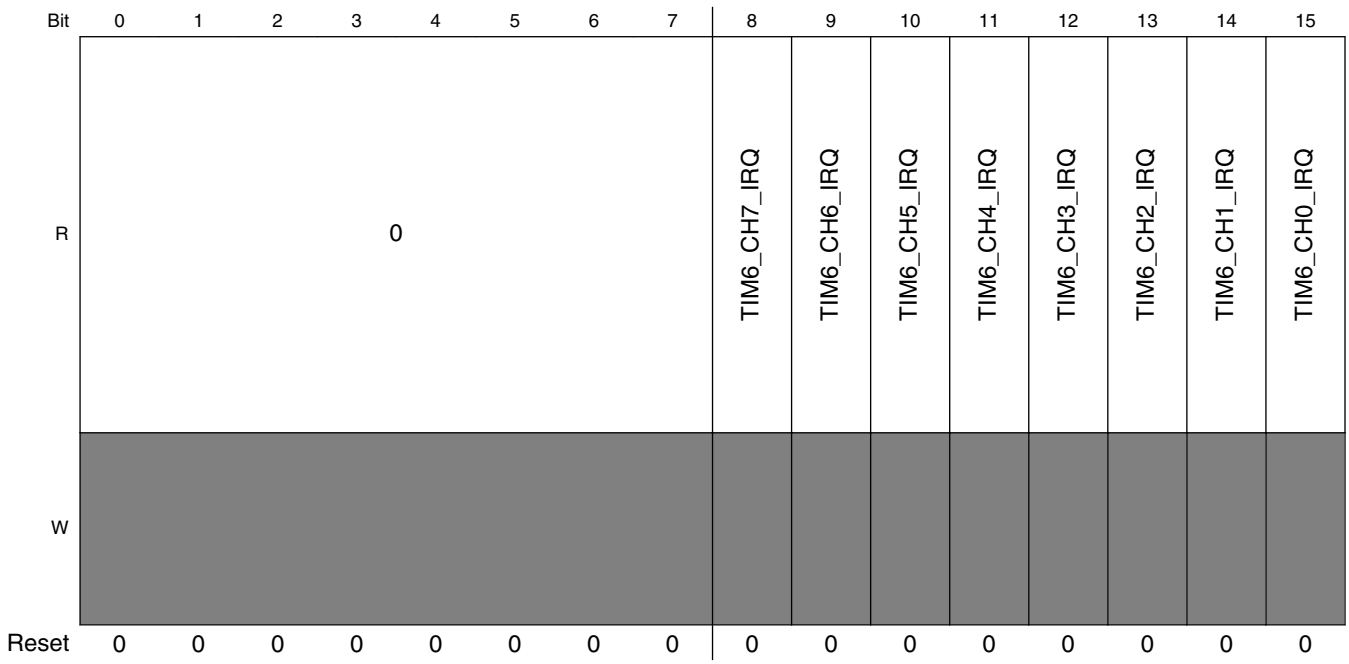
NOTE

Bit[8:31] is only set when the interrupt is enabled in the interrupt enable register of the corresponding submodule.

NOTE

Bit[8:31] when set, this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ.

Address: 600h base + Ch offset = 60Ch



ICM_IRQG_3 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 TIM6_CH7_IRQ	TIM6 shared interrupt channel 7. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule.

Table continues on the next page...

ICM_IRQG_3 field descriptions (continued)

Field	Description
	When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
9 TIM6_CH6_IRQ	TIM6 shared interrupt channel 6. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
10 TIM6_CH5_IRQ	TIM6 shared interrupt channel 5. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
11 TIM6_CH4_IRQ	TIM6 shared interrupt channel 4. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
12 TIM6_CH3_IRQ	TIM6 shared interrupt channel 3. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
13 TIM6_CH2_IRQ	TIM6 shared interrupt channel 2. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_3 field descriptions (continued)

Field	Description
14 TIM6_CH1_IRQ	TIM6 shared interrupt channel 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
15 TIM6_CH0_IRQ	TIM6 shared interrupt channel 0. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
16 TIM5_CH7_IRQ	TIM5 shared interrupt channel 7. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
17 TIM5_CH6_IRQ	TIM5 shared interrupt channel 6. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
18 TIM5_CH5_IRQ	TIM5 shared interrupt channel 5. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
19 TIM5_CH4_IRQ	TIM5 shared interrupt channel 4. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ.

Table continues on the next page...

ICM_IRQG_3 field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
20 TIM5_CH3_IRQ	TIM5 shared interrupt channel 3. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
21 TIM5_CH2_IRQ	TIM5 shared interrupt channel 2. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
22 TIM5_CH1_IRQ	TIM5 shared interrupt channel 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
23 TIM5_CH0_IRQ	TIM5 shared interrupt channel 0. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
24 TIM4_CH7_IRQ	TIM4 shared interrupt channel 7. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
25 TIM4_CH6_IRQ	TIM4 shared interrupt channel 6. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule.

Table continues on the next page...

ICM_IRQG_3 field descriptions (continued)

Field	Description
	When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
26 TIM4_CH5_IRQ	TIM4 shared interrupt channel 5. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
27 TIM4_CH4_IRQ	TIM4 shared interrupt channel 4. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
28 TIM4_CH3_IRQ	TIM4 shared interrupt channel 3. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
29 TIM4_CH2_IRQ	TIM4 shared interrupt channel 2. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
30 TIM4_CH1_IRQ	TIM4 shared interrupt channel 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_3 field descriptions (continued)

Field	Description
31 TIM4_CH0_IRQ	TIM4 shared interrupt channel 0. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding submodule. When set this bit represents one of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TOx_IRQ. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

18.4.5 MCS Interrupt Group 0 Register (ICM_IRQG_4)

NOTE

Bit[0:31] is only set when the interrupt is enabled in the interrupt enable register of the corresponding submodule.

Address: 600h base + 10h offset = 610h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	MCS3_CH7_IRQ	MCS3_CH6_IRQ	MCS3_CH5_IRQ	MCS3_CH4_IRQ	MCS3_CH3_IRQ	MCS3_CH2_IRQ	MCS3_CH1_IRQ	MCS3_CH0_IRQ	MCS2_CH7_IRQ	MCS2_CH6_IRQ	MCS2_CH5_IRQ	MCS2_CH4_IRQ	MCS2_CH3_IRQ	MCS2_CH2_IRQ	MCS2_CH1_IRQ	MCS2_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	MCS1_CH7_IRQ	MCS1_CH6_IRQ	MCS1_CH5_IRQ	MCS1_CH4_IRQ	MCS1_CH3_IRQ	MCS1_CH2_IRQ	MCS1_CH1_IRQ	MCS1_CH0_IRQ	MCS0_CH7_IRQ	MCS0_CH6_IRQ	MCS0_CH5_IRQ	MCS0_CH4_IRQ	MCS0_CH3_IRQ	MCS0_CH2_IRQ	MCS0_CH1_IRQ	MCS0_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICM_IRQG_4 field descriptions

Field	Description
0 MCS3_CH7_IRQ	MCS3 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
1 MCS3_CH6_IRQ	MCS3 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
2 MCS3_CH5_IRQ	MCS3 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
3 MCS3_CH4_IRQ	MCS3 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
4 MCS3_CH3_IRQ	MCS3 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
5 MCS3_CH2_IRQ	MCS3 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
6 MCS3_CH1_IRQ	MCS3 channel 1 interrupt.

Table continues on the next page...

ICM_IRQG_4 field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
7 MCS3_CH0_IRQ	MCS3 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
8 MCS2_CH7_IRQ	MCS2 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
9 MCS2_CH6_IRQ	MCS2 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
10 MCS2_CH5_IRQ	MCS2 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
11 MCS2_CH4_IRQ	MCS2 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
12 MCS2_CH3_IRQ	MCS2 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
13 MCS2_CH2_IRQ	MCS2 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
14 MCS2_CH1_IRQ	MCS2 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
15 MCS2_CH0_IRQ	MCS2 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
16 MCS1_CH7_IRQ	MCS1 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
17 MCS1_CH6_IRQ	MCS1 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
18 MCS1_CH5_IRQ	MCS1 channel 5 interrupt.

Table continues on the next page...

ICM_IRQG_4 field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
19 MCS1_CH4_IRQ	MCS1 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
20 MCS1_CH3_IRQ	MCS1 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
21 MCS1_CH2_IRQ	MCS1 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
22 MCS1_CH1_IRQ	MCS1 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
23 MCS1_CH0_IRQ	MCS1 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
24 MCS0_CH7_IRQ	MCS0 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
25 MCS0_CH6_IRQ	MCS0 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
26 MCS0_CH5_IRQ	MCS0 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
27 MCS0_CH4_IRQ	MCS0 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
28 MCS0_CH3_IRQ	MCS0 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
29 MCS0_CH2_IRQ	MCS0 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
30 MCS0_CH1_IRQ	MCS0 channel 1 interrupt.

Table continues on the next page...

ICM_IRQG_4 field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
31 MCS0_CH0_IRQ	MCS0 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

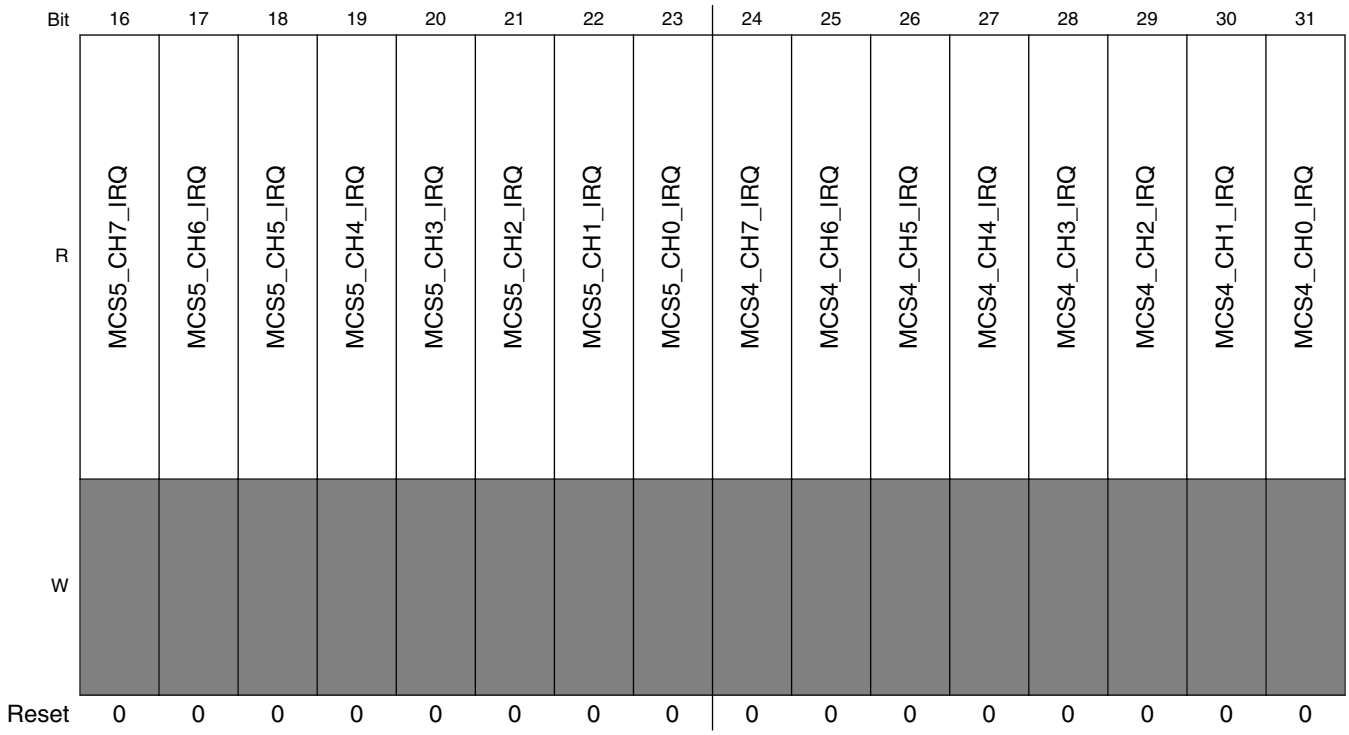
18.4.6 MCS Interrupt Group 1 Register (ICM_IRQG_5)

NOTE

Bit[8:31] is set only when the interrupt is enabled in the interrupt enable register of the corresponding submodule.

Address: 600h base + 14h offset = 614h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0								MCS6_CH7_IRQ	MCS6_CH6_IRQ	MCS6_CH5_IRQ	MCS6_CH4_IRQ	MCS6_CH3_IRQ	MCS6_CH2_IRQ	MCS6_CH1_IRQ	MCS6_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



ICM_IRQG_5 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 MCS6_CH7_IRQ	MCS6 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
9 MCS6_CH6_IRQ	MCS6 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
10 MCS6_CH5_IRQ	MCS6 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
11 MCS6_CH4_IRQ	MCS6 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
12 MCS6_CH3_IRQ	MCS6 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
13 MCS6_CH2_IRQ	MCS6 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_5 field descriptions (continued)

Field	Description
14 MCS6_CH1_IRQ	MCS6 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
15 MCS6_CH0_IRQ	MCS6 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
16 MCS5_CH7_IRQ	MCS5 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
17 MCS5_CH6_IRQ	MCS5 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
18 MCS5_CH5_IRQ	MCS5 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
19 MCS5_CH4_IRQ	MCS5 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
20 MCS5_CH3_IRQ	MCS5 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
21 MCS5_CH2_IRQ	MCS5 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
22 MCS5_CH1_IRQ	MCS5 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
23 MCS5_CH0_IRQ	MCS5 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
24 MCS4_CH7_IRQ	MCS4 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
25 MCS4_CH6_IRQ	MCS4 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_5 field descriptions (continued)

Field	Description
26 MCS4_CH5_IRQ	MCS4 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
27 MCS4_CH4_IRQ	MCS4 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
28 MCS4_CH3_IRQ	MCS4 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
29 MCS4_CH2_IRQ	MCS4 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
30 MCS4_CH1_IRQ	MCS4 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
31 MCS4_CH0_IRQ	MCS4 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

18.4.7 TOM Interrupt Group 0 Register (ICM_IRQG_6)

NOTE

Bit[0:31] is only set when the interrupt is enabled in the interrupt enable register of the corresponding submodule.

Address: 600h base + 18h offset = 618h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	TOM1_CH15_IRQ	TOM1_CH14_IRQ	TOM1_CH13_IRQ	TOM1_CH12_IRQ	TOM1_CH11_IRQ	TOM1_CH10_IRQ	TOM1_CH9_IRQ	TOM1_CH8_IRQ	TOM1_CH7_IRQ	TOM1_CH6_IRQ	TOM1_CH5_IRQ	TOM1_CH4_IRQ	TOM1_CH3_IRQ	TOM1_CH2_IRQ	TOM1_CH1_IRQ	TOM1_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TOM0_CH15_IRQ	TOM0_CH14_IRQ	TOM0_CH13_IRQ	TOM0_CH12_IRQ	TOM0_CH11_IRQ	TOM0_CH10_IRQ	TOM0_CH9_IRQ	TOM0_CH8_IRQ	TOM0_CH7_IRQ	TOM0_CH6_IRQ	TOM0_CH5_IRQ	TOM0_CH4_IRQ	TOM0_CH3_IRQ	TOM0_CH2_IRQ	TOM0_CH1_IRQ	TOM0_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICM_IRQG_6 field descriptions

Field	Description
0 TOM1_CH15_IRQ	TOM1 channel 15 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
1 TOM1_CH14_IRQ	TOM1 channel 14 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
2 TOM1_CH13_IRQ	TOM1 channel 13 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
3 TOM1_CH12_IRQ	TOM1 channel 12 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
4 TOM1_CH11_IRQ	TOM1 channel 11 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
5 TOM1_CH10_IRQ	TOM1 channel 10 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
6 TOM1_CH9_IRQ	TOM1 channel 9 shared interrupt.

Table continues on the next page...

ICM_IRQG_6 field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
7 TOM1_CH8_IRQ	TOM1 channel 8 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
8 TOM1_CH7_IRQ	TOM1 channel 7 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
9 TOM1_CH6_IRQ	TOM1 channel 6 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
10 TOM1_CH5_IRQ	TOM1 channel 5 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
11 TOM1_CH4_IRQ	TOM1 channel 4 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
12 TOM1_CH3_IRQ	TOM1 channel 3 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
13 TOM1_CH2_IRQ	TOM1 channel 2 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
14 TOM1_CH1_IRQ	TOM1 channel 1 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
15 TOM1_CH0_IRQ	TOM1 channel 0 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
16 TOM0_CH15_ IRQ	TOM0 channel 15 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
17 TOM0_CH14_ IRQ	TOM0 channel 14 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
18 TOM0_CH13_ IRQ	TOM0 channel 13 shared interrupt.

Table continues on the next page...

ICM_IRQG_6 field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
19 TOM0_CH12_IRQ	TOM0 channel 12 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
20 TOM0_CH11_IRQ	TOM0 channel 11 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
21 TOM0_CH10_IRQ	TOM0 channel 10 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
22 TOM0_CH9_IRQ	TOM0 channel 9 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
23 TOM0_CH8_IRQ	TOM0 channel 8 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
24 TOM0_CH7_IRQ	TOM0 channel 7 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
25 TOM0_CH6_IRQ	TOM0 channel 6 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
26 TOM0_CH5_IRQ	TOM0 channel 5 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
27 TOM0_CH4_IRQ	TOM0 channel 4 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
28 TOM0_CH3_IRQ	TOM0 channel 3 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
29 TOM0_CH2_IRQ	TOM0 channel 2 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
30 TOM0_CH1_IRQ	TOM0 channel 1 shared interrupt

Table continues on the next page...

ICM_IRQG_6 field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
31 TOM0_CH0_IRQ	TOM0 channel 0 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

18.4.8 TOM Interrupt Group 1 Register (ICM_IRQG_7)

NOTE

Bit[0:31] is only set when the interrupt is enabled in the interrupt enable register of the corresponding submodule.

Address: 600h base + 1Ch offset = 61Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	TOM3_CH15_IRQ	TOM3_CH14_IRQ	TOM3_CH13_IRQ	TOM3_CH12_IRQ	TOM3_CH11_IRQ	TOM3_CH10_IRQ	TOM3_CH9_IRQ	TOM3_CH8_IRQ	TOM3_CH7_IRQ	TOM3_CH6_IRQ	TOM3_CH5_IRQ	TOM3_CH4_IRQ	TOM3_CH3_IRQ	TOM3_CH2_IRQ	TOM3_CH1_IRQ	TOM3_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TOM2_CH15_IRQ	TOM2_CH14_IRQ	TOM2_CH13_IRQ	TOM2_CH12_IRQ	TOM2_CH11_IRQ	TOM2_CH10_IRQ	TOM2_CH9_IRQ	TOM2_CH8_IRQ	TOM2_CH7_IRQ	TOM2_CH6_IRQ	TOM2_CH5_IRQ	TOM2_CH4_IRQ	TOM2_CH3_IRQ	TOM2_CH2_IRQ	TOM2_CH1_IRQ	TOM2_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICM_IRQG_7 field descriptions

Field	Description
0 TOM3_CH15_IRQ	TOM3 channel 15 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
1 TOM3_CH14_IRQ	TOM3 channel 14 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
2 TOM3_CH13_IRQ	TOM3 channel 13 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
3 TOM3_CH12_IRQ	TOM3 channel 12 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
4 TOM3_CH11_IRQ	TOM3 channel 11 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
5 TOM3_CH10_IRQ	TOM3 channel 10 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
6 TOM3_CH9_IRQ	TOM3 channel 9 shared interrupt.

Table continues on the next page...

ICM_IRQG_7 field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
7 TOM3_CH8_IRQ	TOM3 channel 8 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
8 TOM3_CH7_IRQ	TOM3 channel 7 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
9 TOM3_CH6_IRQ	TOM3 channel 7 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
10 TOM3_CH5_IRQ	TOM3 channel 5 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
11 TOM3_CH4_IRQ	TOM3 channel 4 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
12 TOM3_CH3_IRQ	TOM3 channel 3 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
13 TOM3_CH2_IRQ	TOM3 channel 2 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
14 TOM3_CH1_IRQ	TOM3 channel 1 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
15 TOM3_CH0_IRQ	TOM3 channel 0 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
16 TOM2_CH15_ IRQ	TOM2 channel 15 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
17 TOM2_CH14_ IRQ	TOM2 channel 14 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
18 TOM2_CH13_ IRQ	TOM2 channel 13 shared interrupt.

Table continues on the next page...

ICM_IRQG_7 field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
19 TOM2_CH12_IRQ	TOM2 channel 12 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
20 TOM2_CH11_IRQ	TOM2 channel 11 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
21 TOM2_CH10_IRQ	TOM2 channel 10 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
22 TOM2_CH9_IRQ	TOM2 channel 9 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
23 TOM2_CH8_IRQ	TOM2 channel 8 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
24 TOM2_CH7_IRQ	TOM2 channel 7 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
25 TOM2_CH6_IRQ	TOM2 channel 6 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
26 TOM2_CH5_IRQ	TOM2 channel 5 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
27 TOM2_CH4_IRQ	TOM2 channel 4 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
28 TOM2_CH3_IRQ	TOM2 channel 3 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
29 TOM2_CH2_IRQ	TOM2 channel 2 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
30 TOM2_CH1_IRQ	TOM2 channel 1 shared interrupt

Table continues on the next page...

ICM_IRQG_7 field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
31 TOM2_CH0_IRQ	TOM2 channel 0 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

18.4.9 TOM Interrupt Group 2 Register (ICM_IRQG_8)

Bit[0:31] is only set when the interrupt is enabled in the interrupt enable register of the corresponding submodule.

Address: 600h base + 20h offset = 620h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	TOM5_CH15_IRQ	TOM5_CH14_IRQ	TOM5_CH13_IRQ	TOM5_CH12_IRQ	TOM5_CH11_IRQ	TOM5_CH10_IRQ	TOM5_CH9_IRQ	TOM5_CH8_IRQ	TOM5_CH7_IRQ	TOM5_CH6_IRQ	TOM5_CH5_IRQ	TOM5_CH4_IRQ	TOM5_CH3_IRQ	TOM5_CH2_IRQ	TOM5_CH1_IRQ	TOM5_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TOM4_CH15_IRQ	TOM4_CH14_IRQ	TOM4_CH13_IRQ	TOM4_CH12_IRQ	TOM4_CH11_IRQ	TOM4_CH10_IRQ	TOM4_CH9_IRQ	TOM4_CH8_IRQ	TOM4_CH7_IRQ	TOM4_CH6_IRQ	TOM4_CH5_IRQ	TOM4_CH4_IRQ	TOM4_CH3_IRQ	TOM4_CH2_IRQ	TOM4_CH1_IRQ	TOM4_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICM_IRQG_8 field descriptions

Field	Description
0 TOM5_CH15_IRQ	TOM5 channel 15 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
1 TOM5_CH14_IRQ	TOM5 channel 14 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
2 TOM5_CH13_IRQ	TOM5 channel 13 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
3 TOM5_CH12_IRQ	TOM5 channel 12 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
4 TOM5_CH11_IRQ	TOM5 channel 11 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
5 TOM5_CH10_IRQ	TOM5 channel 10 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
6 TOM5_CH9_IRQ	TOM5 channel 9 shared interrupt.

Table continues on the next page...

ICM_IRQG_8 field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
7 TOM5_CH8_IRQ	TOM5 channel 8 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
8 TOM5_CH7_IRQ	TOM5 channel 7 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
9 TOM5_CH6_IRQ	TOM5 channel 6 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
10 TOM5_CH5_IRQ	TOM5 channel 5 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
11 TOM5_CH4_IRQ	TOM5 channel 4 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
12 TOM5_CH3_IRQ	TOM5 channel 3 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
13 TOM5_CH2_IRQ	TOM5 channel 2 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
14 TOM5_CH1_IRQ	TOM5 channel 1 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
15 TOM5_CH0_IRQ	TOM5 channel 0 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
16 TOM4_CH15_ IRQ	TOM4 channel 15 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
17 TOM4_CH14_ IRQ	TOM4 channel 14 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
18 TOM4_CH13_ IRQ	TOM4 channel 13 shared interrupt.

Table continues on the next page...

ICM_IRQG_8 field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
19 TOM4_CH12_IRQ	TOM4 channel 12 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
20 TOM4_CH11_IRQ	TOM4 channel 11 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
21 TOM4_CH10_IRQ	TOM4 channel 10 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
22 TOM4_CH9_IRQ	TOM4 channel 9 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
23 TOM4_CH8_IRQ	TOM4 channel 8 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
24 TOM4_CH7_IRQ	TOM4 channel 7 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
25 TOM4_CH6_IRQ	TOM4 channel 6 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
26 TOM4_CH5_IRQ	TOM4 channel 5 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
27 TOM4_CH4_IRQ	TOM4 channel 4 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
28 TOM4_CH3_IRQ	TOM4 channel 3 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
29 TOM4_CH2_IRQ	TOM4 channel 2 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
30 TOM4_CH1_IRQ	TOM4 channel 1 shared interrupt

Table continues on the next page...

ICM_IRQG_8 field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
31 TOM4_CH0_IRQ	TOM4 channel 0 shared interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

18.4.10 ATOM Interrupt Group 0 Register (ICM_IRQG_9)

NOTE

Bit[0:31] is only set when the interrupt is enabled in the interrupt enable register of the corresponding submodule.

Address: 600h base + 24h offset = 624h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	ATOM3_CH7_IRQ	ATOM3_CH6_IRQ	ATOM3_CH5_IRQ	ATOM3_CH4_IRQ	ATOM3_CH3_IRQ	ATOM3_CH2_IRQ	ATOM3_CH1_IRQ	ATOM3_CH0_IRQ	ATOM2_CH7_IRQ	ATOM2_CH6_IRQ	ATOM2_CH5_IRQ	ATOM2_CH4_IRQ	ATOM2_CH3_IRQ	ATOM2_CH2_IRQ	ATOM2_CH1_IRQ	ATOM2_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	ATOM1_CH7_IRQ	ATOM1_CH6_IRQ	ATOM1_CH5_IRQ	ATOM1_CH4_IRQ	ATOM1_CH3_IRQ	ATOM1_CH2_IRQ	ATOM1_CH1_IRQ	ATOM1_CH0_IRQ	ATOM0_CH7_IRQ	ATOM0_CH6_IRQ	ATOM0_CH5_IRQ	ATOM0_CH4_IRQ	ATOM0_CH3_IRQ	ATOM0_CH2_IRQ	ATOM0_CH1_IRQ	ATOM0_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICM_IRQG_9 field descriptions

Field	Description
0 ATOM3_CH7_IRQ	ATOM3 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
1 ATOM3_CH6_IRQ	ATOM3 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
2 ATOM3_CH5_IRQ	ATOM3 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
3 ATOM3_CH4_IRQ	ATOM3 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
4 ATOM3_CH3_IRQ	ATOM3 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
5 ATOM3_CH2_IRQ	ATOM3 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_9 field descriptions (continued)

Field	Description
6 ATOM3_CH1_IRQ	ATOM3 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
7 ATOM3_CH0_IRQ	ATOM3 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
8 ATOM2_CH7_IRQ	ATOM2 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
9 ATOM2_CH6_IRQ	ATOM2 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
10 ATOM2_CH5_IRQ	ATOM2 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
11 ATOM2_CH4_IRQ	ATOM2 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
12 ATOM2_CH3_IRQ	ATOM2 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
13 ATOM2_CH2_IRQ	ATOM2 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
14 ATOM2_CH1_IRQ	ATOM2 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
15 ATOM2_CH0_IRQ	ATOM2 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
16 ATOM1_CH7_IRQ	ATOM1 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
17 ATOM1_CH6_IRQ	ATOM1 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_9 field descriptions (continued)

Field	Description
18 ATOM1_CH5_ IRQ	ATOM1 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
19 ATOM1_CH4_ IRQ	ATOM1 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
20 ATOM1_CH3_ IRQ	ATOM1 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
21 ATOM1_CH2_ IRQ	ATOM1 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
22 ATOM1_CH1_ IRQ	ATOM1 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
23 ATOM1_CH0_ IRQ	ATOM1 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
24 ATOM0_CH7_ IRQ	ATOM0 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
25 ATOM0_CH6_ IRQ	ATOM0 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
26 ATOM0_CH5_ IRQ	ATOM0 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
27 ATOM0_CH4_ IRQ	ATOM0 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
28 ATOM0_CH3_ IRQ	ATOM0 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
29 ATOM0_CH2_ IRQ	ATOM0 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_9 field descriptions (continued)

Field	Description
30 ATOM0_CH1_ IRQ	ATOM0 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
31 ATOM0_CH0_ IRQ	ATOM0 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

18.4.11 ATOM Interrupt Group 1 Register (ICM_IRQG_10)

NOTE

Bit[0:31] is only set when the interrupt is enabled in the interrupt enable register of the corresponding submodule.

Address: 600h base + 28h offset = 628h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	ATOM7_CH7_IRQ	ATOM7_CH6_IRQ	ATOM7_CH5_IRQ	ATOM7_CH4_IRQ	ATOM7_CH3_IRQ	ATOM7_CH2_IRQ	ATOM7_CH1_IRQ	ATOM7_CH0_IRQ	ATOM6_CH7_IRQ	ATOM6_CH6_IRQ	ATOM6_CH5_IRQ	ATOM6_CH4_IRQ	ATOM6_CH3_IRQ	ATOM6_CH2_IRQ	ATOM6_CH1_IRQ	ATOM6_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	ATOM5_CH7_IRQ	ATOM5_CH6_IRQ	ATOM5_CH5_IRQ	ATOM5_CH4_IRQ	ATOM5_CH3_IRQ	ATOM5_CH2_IRQ	ATOM5_CH1_IRQ	ATOM5_CH0_IRQ	ATOM4_CH7_IRQ	ATOM4_CH6_IRQ	ATOM4_CH5_IRQ	ATOM4_CH4_IRQ	ATOM4_CH3_IRQ	ATOM4_CH2_IRQ	ATOM4_CH1_IRQ	ATOM4_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICM_IRQG_10 field descriptions

Field	Description
0 ATOM7_CH7_IRQ	ATOM7 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
1 ATOM7_CH6_IRQ	ATOM7 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
2 ATOM7_CH5_IRQ	ATOM7 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
3 ATOM7_CH4_IRQ	ATOM7 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
4 ATOM7_CH3_IRQ	ATOM7 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
5 ATOM7_CH2_IRQ	ATOM7 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_10 field descriptions (continued)

Field	Description
6 ATOM7_CH1_IRQ	ATOM7 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
7 ATOM7_CH0_IRQ	ATOM7 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
8 ATOM6_CH7_IRQ	ATOM6 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
9 ATOM6_CH6_IRQ	ATOM6 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
10 ATOM6_CH5_IRQ	ATOM6 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
11 ATOM6_CH4_IRQ	ATOM channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
12 ATOM6_CH3_IRQ	ATOM6 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
13 ATOM6_CH2_IRQ	ATOM6 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
14 ATOM6_CH1_IRQ	ATOM6 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
15 ATOM6_CH0_IRQ	ATOM6 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
16 ATOM5_CH7_IRQ	ATOM5 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
17 ATOM5_CH6_IRQ	ATOM5 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_10 field descriptions (continued)

Field	Description
18 ATOM5_CH5_ IRQ	ATOM5 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
19 ATOM5_CH4_ IRQ	ATOM5 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
20 ATOM5_CH3_ IRQ	ATOM5 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
21 ATOM5_CH2_ IRQ	ATOM5 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
22 ATOM5_CH1_ IRQ	ATOM5 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
23 ATOM5_CH0_ IRQ	ATOM5 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
24 ATOM4_CH7_ IRQ	ATOM4 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
25 ATOM4_CH6_ IRQ	ATOM4 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
26 ATOM4_CH5_ IRQ	ATOM4 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
27 ATOM4_CH4_ IRQ	ATOM4 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
28 ATOM4_CH3_ IRQ	ATOM4 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
29 ATOM4_CH2_ IRQ	ATOM4 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_10 field descriptions (continued)

Field	Description
30 ATOM4_CH1_ IRQ	ATOM4 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
31 ATOM4_CH0_ IRQ	ATOM4 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

18.4.12 ATOM Interrupt Group 2 Register (ICM_IRQG_11)

NOTE

Bit[0:31] is only set when the interrupt is enabled in the interrupt enable register of the corresponding submodule.

Address: 600h base + 2Ch offset = 62Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	ATOM11_CH7_IRQ	ATOM11_CH6_IRQ	ATOM11_CH5_IRQ	ATOM11_CH4_IRQ	ATOM11_CH3_IRQ	ATOM11_CH2_IRQ	ATOM11_CH1_IRQ	ATOM11_CH0_IRQ	ATOM10_CH7_IRQ	ATOM10_CH6_IRQ	ATOM10_CH5_IRQ	ATOM10_CH4_IRQ	ATOM10_CH3_IRQ	ATOM10_CH2_IRQ	ATOM10_CH1_IRQ	ATOM10_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	ATOM9_CH7_IRQ	ATOM9_CH6_IRQ	ATOM9_CH5_IRQ	ATOM9_CH4_IRQ	ATOM9_CH3_IRQ	ATOM9_CH2_IRQ	ATOM9_CH1_IRQ	ATOM9_CH0_IRQ	ATOM8_CH7_IRQ	ATOM8_CH6_IRQ	ATOM8_CH5_IRQ	ATOM8_CH4_IRQ	ATOM4_CH3_IR8	ATOM8_CH2_IRQ	ATOM8_CH1_IRQ	ATOM8_CH0_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICM_IRQG_11 field descriptions

Field	Description
0 ATOM11_CH7_IRQ	ATOM11 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
1 ATOM11_CH6_IRQ	ATOM11 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
2 ATOM11_CH5_IRQ	ATOM11 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
3 ATOM11_CH4_IRQ	ATOM11 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
4 ATOM11_CH3_IRQ	ATOM11 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
5 ATOM11_CH2_IRQ	ATOM11 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_11 field descriptions (continued)

Field	Description
6 ATOM11_CH1_ IRQ	ATOM11 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
7 ATOM11_CH0_ IRQ	ATOM11 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
8 ATOM10_CH7_ IRQ	ATOM10 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
9 ATOM10_CH6_ IRQ	ATOM10 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
10 ATOM10_CH5_ IRQ	ATOM10 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
11 ATOM10_CH4_ IRQ	ATOM10 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
12 ATOM10_CH3_ IRQ	ATOM10 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
13 ATOM10_CH2_ IRQ	ATOM10 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
14 ATOM10_CH1_ IRQ	ATOM10 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
15 ATOM10_CH0_ IRQ	ATOM10 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
16 ATOM9_CH7_ IRQ	ATOM9 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
17 ATOM9_CH6_ IRQ	ATOM9 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_11 field descriptions (continued)

Field	Description
18 ATOM9_CH5_ IRQ	ATOM9 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
19 ATOM9_CH4_ IRQ	ATOM9 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
20 ATOM9_CH3_ IRQ	ATOM9 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
21 ATOM9_CH2_ IRQ	ATOM9 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
22 ATOM9_CH1_ IRQ	ATOM9 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
23 ATOM9_CH0_ IRQ	ATOM9 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
24 ATOM8_CH7_ IRQ	ATOM8 channel 7 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
25 ATOM8_CH6_ IRQ	ATOM8 channel 6 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
26 ATOM8_CH5_ IRQ	ATOM8 channel 5 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
27 ATOM8_CH4_ IRQ	ATOM8 channel 4 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
28 ATOM4_CH3_ IR8	ATOM8 channel 3 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
29 ATOM8_CH2_ IRQ	ATOM8 channel 2 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

Table continues on the next page...

ICM_IRQG_11 field descriptions (continued)

Field	Description
30 ATOM8_CH1_ IRQ	ATOM8 channel 1 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.
31 ATOM8_CH0_ IRQ	ATOM8 channel 0 interrupt. 0 No interrupt occurred. 1 Interrupt was raised by the corresponding submodule.

18.4.13 ICM IRQG Module Error Interrupt register (ICM_IRQG_MEI)

NOTE

Each bit can be set only if the error interrupt is enabled in the error interrupt enable register of the corresponding submodule

Address: 600h base + 30h offset = 630h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0						DPLL_EIRQ	CMP_EIRQ	SPE3_EIRQ	SPE2_EIRQ	SPE1_EIRQ	SPE0_EIRQ	0	MCS6_EIRQ	MCS5_EIRQ	MCS4_EIRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	MCS3_EIRQ	MCS2_EIRQ	MCS1_EIRQ	MCS0_EIRQ	0	TIM6_EIRQ	TIM5_EIRQ	TIM4_EIRQ	TIM3_EIRQ	TIM2_EIRQ	TIM1_EIRQ	TIM0_EIRQ	FIFO1_EIRQ	FIFO0_EIRQ	BRC_EIRQ	GTM_EIRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICM_IRQG_MEI field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 DPLL_EIRQ	DPLL error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
7 CMP_EIRQ	CMP error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
8 SPE3_EIRQ	SPE3 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
9 SPE2_EIRQ	SPE2 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
10 SPE1_EIRQ	SPE1 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
11 SPE0_EIRQ	SPE0 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 MCS6_EIRQ	MCS6 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
14 MCS5_EIRQ	MCS5 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
15 MCS4_EIRQ	MCS4 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
16 MCS3_EIRQ	MCS3 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
17 MCS2_EIRQ	MCS2 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule

Table continues on the next page...

ICM_IRQG_MEI field descriptions (continued)

Field	Description
18 MCS1_EIRQ	MCS1 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
19 MCS0_EIRQ	MCS0 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 TIM6_EIRQ	TIM6 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
22 TIM5_EIRQ	TIM5 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
23 TIM4_EIRQ	TIM4 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
24 TIM3_EIRQ	TIM3 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
25 TIM2_EIRQ	TIM2 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
26 TIM1_EIRQ	TIM1 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
27 TIM0_EIRQ	TIM0 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
28 FIFO1_EIRQ	FIFO1 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
29 FIFO0_EIRQ	FIFO0 error interrupt. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
30 BRC_EIRQ	BRC error interrupt.

Table continues on the next page...

ICM_IRQG_MEI field descriptions (continued)

Field	Description
	0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
31 GTM_EIRQ	GTM Error interrupt request. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule

18.4.14 ICM Channel Error Interrupt Request Group 0 register (ICM_IRQG_CEI0)

NOTE

Each bit can be set only if the error interrupt is enabled in the error interrupt enable register of the corresponding submodule

Address: 600h base + 34h offset = 634h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	FIFO1_CH7_EIRQ	FIFO1_CH6_EIRQ	FIFO1_CH5_EIRQ	FIFO1_CH4_EIRQ	FIFO1_CH3_EIRQ	FIFO1_CH2_EIRQ	FIFO1_CH1_EIRQ	FIFO1_CH0_EIRQ	FIFO0_CH7_EIRQ	FIFO0_CH6_EIRQ	FIFO0_CH5_EIRQ	FIFO0_CH4_EIRQ	FIFO0_CH3_EIRQ	FIFO0_CH2_EIRQ	FIFO0_CH1_EIRQ	FIFO0_CH0_EIRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICM_IRQG_CEI0 field descriptions

Field	Description
0–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 FIFO1_CH7_EIRQ	FIFO1 channel 7 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
17 FIFO1_CH6_EIRQ	FIFO1 channel 6 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
18 FIFO1_CH5_EIRQ	FIFO1 channel 5 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
19 FIFO1_CH4_EIRQ	FIFO1 channel 4 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
20 FIFO1_CH3_EIRQ	FIFO1 channel 3 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
21 FIFO1_CH2_EIRQ	FIFO1 channel 2 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
22 FIFO1_CH1_EIRQ	FIFO1 channel 1 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
23 FIFO1_CH0_EIRQ	FIFO1 channel 0 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
24 FIFO0_CH7_EIRQ	FIFO0 channel 7 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
25 FIFO0_CH6_EIRQ	FIFO0 channel 6 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
26 FIFO0_CH5_EIRQ	FIFO0 channel 5 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
27 FIFO0_CH4_EIRQ	FIFO0 channel 4 error interrupt

Table continues on the next page...

ICM_IRQG_CEI0 field descriptions (continued)

Field	Description
	0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
28 FIFO0_CH3_ EIRQ	FIFO0 channel 3 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
29 FIFO0_CH2_ EIRQ	FIFO0 channel 2 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
30 FIFO0_CH1_ EIRQ	FIFO0 channel 1 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
31 FIFO0_CH0_ EIRQ	FIFO0 channel 0 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule

18.4.15 Channel Error Interrupt Request Group 1 register (ICM_IRQG_CEI1)

NOTE

Each bit can be set only if the error interrupt is enabled in the error interrupt enable register of the corresponding submodule

Address: 600h base + 38h offset = 638h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	TIM3_CH7_EIRQ	TIM3_CH6_EIRQ	TIM3_CH5_EIRQ	TIM3_CH4_EIRQ	TIM3_CH3_EIRQ	TIM3_CH2_EIRQ	TIM3_CH1_EIRQ	TIM3_CH0_EIRQ	TIM2_CH7_EIRQ	TIM2_CH6_EIRQ	TIM2_CH5_EIRQ	TIM2_CH4_EIRQ	TIM2_CH3_EIRQ	TIM2_CH2_EIRQ	TIM2_CH1_EIRQ	TIM2_CH0_EIRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TIM1_CH7_EIRQ	TIM1_CH6_EIRQ	TIM1_CH5_EIRQ	TIM1_CH4_EIRQ	TIM1_CH3_EIRQ	TIM1_CH2_EIRQ	TIM1_CH1_EIRQ	TIM1_CH0_EIRQ	TIM0_CH7_EIRQ	TIM0_CH6_EIRQ	TIM0_CH5_EIRQ	TIM0_CH4_EIRQ	TIM0_CH3_EIRQ	TIM0_CH2_EIRQ	TIM0_CH1_EIRQ	TIM0_CH0_EIRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICM_IRQG_CEI1 field descriptions

Field	Description
0 TIM3_CH7_EIRQ	TIM3 Channel 7 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
1 TIM3_CH6_EIRQ	TIM3 Channel 6 Error interrupt No interrupt occurred Interrupt was asserted by the corresponding submodule
2 TIM3_CH5_EIRQ	TIM3 Channel 5 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
3 TIM3_CH4_EIRQ	TIM3 Channel 4 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
4 TIM3_CH3_EIRQ	TIM3 Channel 3 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
5 TIM3_CH2_EIRQ	TIM3 Channel 2 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
6 TIM3_CH1_EIRQ	TIM3 Channel 1 Error interrupt

Table continues on the next page...

ICM_IRQG_CEI1 field descriptions (continued)

Field	Description
	0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
7 TIM3_CH0_EIRQ	TIM3 Channel 0 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
8 TIM2_CH7_EIRQ	TIM2 Channel 7 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
9 TIM2_CH6_EIRQ	TIM2 Channel 6 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
10 TIM2_CH5_EIRQ	TIM2 Channel 5 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
11 TIM2_CH4_EIRQ	TIM2 Channel 4 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
12 TIM2_CH3_EIRQ	TIM2 Channel 3 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
13 TIM2_CH2_EIRQ	TIM2 Channel 2 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
14 TIM2_CH1_EIRQ	TIM2 Channel 1 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
15 TIM2_CH0_EIRQ	TIM2 Channel 0 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
16 TIM1_CH7_EIRQ	TIM1 Channel 7 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
17 TIM1_CH6_EIRQ	TIM1 Channel 6 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
18 TIM1_CH5_EIRQ	TIM1 Channel 5 Error interrupt

Table continues on the next page...

ICM_IRQG_CEI1 field descriptions (continued)

Field	Description
	0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
19 TIM1_CH4_EIRQ	TIM1 Channel 4 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
20 TIM1_CH3_EIRQ	TIM1 Channel 3 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
21 TIM1_CH2_EIRQ	TIM1 Channel 2 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
22 TIM1_CH1_EIRQ	TIM1 Channel 1 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
23 TIM1_CH0_EIRQ	TIM1 Channel 0 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
24 TIM0_CH7_EIRQ	TIM0 Channel 7 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
25 TIM0_CH6_EIRQ	TIM0 Channel 6 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
26 TIM0_CH5_EIRQ	TIM0 Channel 5 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
27 TIM0_CH4_EIRQ	TIM0 Channel 4 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
28 TIM0_CH3_EIRQ	TIM0 Channel 3 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
29 TIM0_CH2_EIRQ	TIM0 Channel 2 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
30 TIM0_CH1_EIRQ	TIM0 Channel 1 Error interrupt

Table continues on the next page...

ICM_IRQG_CEI1 field descriptions (continued)

Field	Description
	0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
31 TIM0_CH0_EIRQ	TIM0 Channel 0 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule

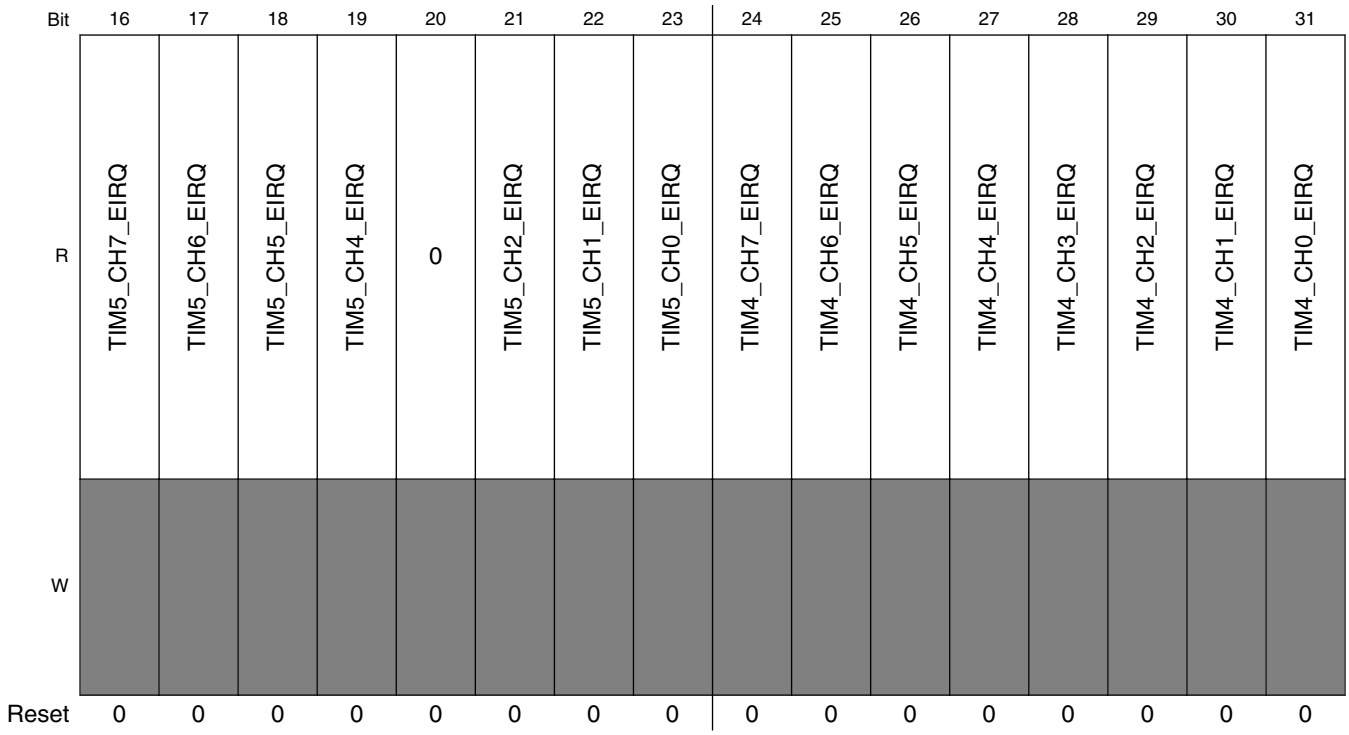
18.4.16 Channel Error Interrupt Request Group 2 register (ICM_IRQG_CEI2)

NOTE

Each bit can be set only if the error interrupt is enabled in the error interrupt enable register of the corresponding submodule

Address: 600h base + 3Ch offset = 63Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0								TIM6_CH7_EIRQ	TIM6_CH6_EIRQ	TIM6_CH5_EIRQ	TIM6_CH4_EIRQ	TIM6_CH3_EIRQ	TIM6_CH2_EIRQ	TIM6_CH1_EIRQ	TIM6_CH0_EIRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TIM5_CH7_EIRQ	TIM5_CH6_EIRQ	TIM5_CH5_EIRQ	TIM5_CH4_EIRQ	0	TIM5_CH2_EIRQ	TIM5_CH1_EIRQ	TIM5_CH0_EIRQ	TIM4_CH7_EIRQ	TIM4_CH6_EIRQ	TIM4_CH5_EIRQ	TIM4_CH4_EIRQ	TIM4_CH3_EIRQ	TIM4_CH2_EIRQ	TIM4_CH1_EIRQ	TIM4_CH0_EIRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICM_IRQG_CEI2 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 TIM6_CH7_EIRQ	TIM6 Channel 7 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
9 TIM6_CH6_EIRQ	TIM6 Channel 6 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
10 TIM6_CH5_EIRQ	TIM6 Channel 5 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
11 TIM6_CH4_EIRQ	TIM6 Channel 4 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
12 TIM6_CH3_EIRQ	TIM6 Channel 3 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
13 TIM6_CH2_EIRQ	TIM6 Channel 2 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule

Table continues on the next page...

ICM_IRQG_CEI2 field descriptions (continued)

Field	Description
14 TIM6_CH1_EIRQ	TIM6 Channel 1 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
15 TIM6_CH0_EIRQ	TIM6 Channel 0 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
16 TIM5_CH7_EIRQ	TIM5 Channel 7 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
17 TIM5_CH6_EIRQ	TIM5 Channel 6 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
18 TIM5_CH5_EIRQ	TIM5 Channel 5 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
19 TIM5_CH4_EIRQ	TIM5 Channel 4 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
20 Reserved	TIM5 Channel 3 Error interrupt This read-only field is reserved and always has the value 0. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
21 TIM5_CH2_EIRQ	TIM5 Channel 2 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
22 TIM5_CH1_EIRQ	TIM5 Channel 1 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
23 TIM5_CH0_EIRQ	TIM5 Channel 0 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
24 TIM4_CH7_EIRQ	TIM4 Channel 7 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
25 TIM4_CH6_EIRQ	TIM4 Channel 6 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule

Table continues on the next page...

ICM_IRQG_CEI2 field descriptions (continued)

Field	Description
26 TIM4_CH5_EIRQ	TIM4 Channel 5 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
27 TIM4_CH4_EIRQ	TIM4 Channel 4 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
28 TIM4_CH3_EIRQ	TIM4 Channel 3 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
29 TIM4_CH2_EIRQ	TIM4 Channel 2 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
30 TIM4_CH1_EIRQ	TIM4 Channel 1 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
31 TIM4_CH0_EIRQ	TIM4 Channel 0 Error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule

18.4.17 Channel Error Interrupt Request Group 3 register (ICM_IRQG_CEI3)

NOTE

Each bit can be set only if the error interrupt is enabled in the error interrupt enable register of the corresponding submodule

Address: 600h base + 40h offset = 640h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	MCS3_CH7_EIRQ	MCS3_CH6_EIRQ	MCS3_CH5_EIRQ	MCS3_CH4_EIRQ	MCS3_CH3_EIRQ	MCS3_CH2_EIRQ	MCS3_CH1_EIRQ	MCS3_CH0_EIRQ	MCS2_CH7_EIRQ	MCS2_CH6_EIRQ	MCS2_CH5_EIRQ	MCS2_CH4_EIRQ	MCS2_CH3_EIRQ	MCS2_CH2_EIRQ	MCS2_CH1_EIRQ	MCS2_CH0_EIRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	MCS1_CH7_EIRQ	MCS1_CH6_EIRQ	MCS1_CH5_EIRQ	MCS1_CH4_EIRQ	MCS1_CH3_EIRQ	MCS1_CH2_EIRQ	MCS1_CH1_EIRQ	MCS1_CH0_EIRQ	MCS0_CH7_EIRQ	MCS0_CH6_EIRQ	MCS0_CH5_EIRQ	MCS0_CH4_EIRQ	MCS0_CH3_EIRQ	MCS0_CH2_EIRQ	MCS0_CH1_EIRQ	MCS0_CH0_EIRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICM_IRQG_CEI3 field descriptions

Field	Description
0 MCS3_CH7_ EIRQ	MCS3 Channel 7 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
1 MCS3_CH6_ EIRQ	MCS3 Channel 6 error interrupt No interrupt occurred Interrupt was asserted by the corresponding submodule
2 MCS3_CH5_ EIRQ	MCS3 Channel 5 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
3 MCS3_CH4_ EIRQ	MCS3 Channel 4 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
4 MCS3_CH3_ EIRQ	MCS3 Channel 3 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
5 MCS3_CH2_ EIRQ	MCS3 Channel 2 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule

Table continues on the next page...

ICM_IRQG_CEI3 field descriptions (continued)

Field	Description
6 MCS3_CH1_ EIRQ	MCS3 Channel 1 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
7 MCS3_CH0_ EIRQ	MCS3 Channel 0 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
8 MCS2_CH7_ EIRQ	MCS2 Channel 7 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
9 MCS2_CH6_ EIRQ	MCS2 Channel 6 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
10 MCS2_CH5_ EIRQ	MCS2 Channel 5 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
11 MCS2_CH4_ EIRQ	MCS2 Channel 4 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
12 MCS2_CH3_ EIRQ	MCS2 Channel 3 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
13 MCS2_CH2_ EIRQ	MCS2 Channel 2 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
14 MCS2_CH1_ EIRQ	MCS2 Channel 1 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
15 MCS2_CH0_ EIRQ	MCS2 Channel 0 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
16 MCS1_CH7_ EIRQ	MCS1 Channel 7 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
17 MCS1_CH6_ EIRQ	MCS1 Channel 6 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule

Table continues on the next page...

ICM_IRQG_CEI3 field descriptions (continued)

Field	Description
18 MCS1_CH5_ EIRQ	MCS1 Channel 5 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
19 MCS1_CH4_ EIRQ	MCS1 Channel 4 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
20 MCS1_CH3_ EIRQ	MCS1 Channel 3 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
21 MCS1_CH2_ EIRQ	MCS1 Channel 2 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
22 MCS1_CH1_ EIRQ	MCS1 Channel 1 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
23 MCS1_CH0_ EIRQ	MCS1 Channel 0 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
24 MCS0_CH7_ EIRQ	MCS0 Channel 7 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
25 MCS0_CH6_ EIRQ	MCS0 Channel 6 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
26 MCS0_CH5_ EIRQ	MCS0 Channel 5 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
27 MCS0_CH4_ EIRQ	MCS0 Channel 4 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
28 MCS0_CH3_ EIRQ	MCS0 Channel 3 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
29 MCS0_CH2_ EIRQ	MCS0 Channel 2 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule

Table continues on the next page...

ICM_IRQG_CEI3 field descriptions (continued)

Field	Description
30 MCS0_CH1_ EIRQ	MCS0 Channel 1 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
31 MCS0_CH0_ EIRQ	MCS0 Channel 0 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule

18.4.18 Channel Error Interrupt Request Group 4 register (ICM_IRQG_CEI4)

NOTE

Each bit can be set only if the error interrupt is enabled in the error interrupt enable register of the corresponding submodule.

Address: 600h base + 44h offset = 644h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0								MCS6_CH7_EIRQ	MCS6_CH6_EIRQ	MCS6_CH5_EIRQ	MCS6_CH4_EIRQ	MCS6_CH3_EIRQ	MCS6_CH2_EIRQ	MCS6_CH1_EIRQ	MCS6_CH0_EIRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	MCS5_CH7_EIRQ	MCS5_CH6_EIRQ	MCS5_CH5_EIRQ	MCS5_CH4_EIRQ	0	MCS5_CH2_EIRQ	MCS5_CH1_EIRQ	MCS5_CH0_EIRQ	MCS4_CH7_EIRQ	MCS4_CH6_EIRQ	MCS4_CH5_EIRQ	MCS4_CH4_EIRQ	MCS4_CH3_EIRQ	MCS4_CH2_EIRQ	MCS4_CH1_EIRQ	MCS4_CH0_EIRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ICM_IRQG_CEI4 field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 MCS6_CH7_ EIRQ	MCS6 Channel 7 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
9 MCS6_CH6_ EIRQ	MCS6 Channel 6 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
10 MCS6_CH5_ EIRQ	MCS6 Channel 5 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
11 MCS6_CH4_ EIRQ	MCS6 Channel 4 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
12 MCS6_CH3_ EIRQ	MCS6 Channel 3 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
13 MCS6_CH2_ EIRQ	MCS6 Channel 2 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule

Table continues on the next page...

ICM_IRQG_CEI4 field descriptions (continued)

Field	Description
14 MCS6_CH1_ EIRQ	MCS6 Channel 1 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
15 MCS6_CH0_ EIRQ	MCS6 Channel 0 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
16 MCS5_CH7_ EIRQ	MCS5 Channel 7 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
17 MCS5_CH6_ EIRQ	MCS5 Channel 6 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
18 MCS5_CH5_ EIRQ	MCS5 Channel 5 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
19 MCS5_CH4_ EIRQ	MCS5 Channel 4 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
20 Reserved	MCS5 Channel 3 error interrupt This read-only field is reserved and always has the value 0. 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
21 MCS5_CH2_ EIRQ	MCS5 Channel 2 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
22 MCS5_CH1_ EIRQ	MCS5 Channel 1 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
23 MCS5_CH0_ EIRQ	MCS5 Channel 0 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
24 MCS4_CH7_ EIRQ	MCS4 Channel 7 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
25 MCS4_CH6_ EIRQ	MCS4 Channel 6 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule

Table continues on the next page...

ICM_IRQG_CEI4 field descriptions (continued)

Field	Description
26 MCS4_CH5_ EIRQ	MCS4 Channel 5 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
27 MCS4_CH4_ EIRQ	MCS4 Channel 4 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
28 MCS4_CH3_ EIRQ	MCS4 Channel 3 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
29 MCS4_CH2_ EIRQ	MCS4 Channel 2 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
30 MCS4_CH1_ EIRQ	MCS4 Channel 1 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule
31 MCS4_CH0_ EIRQ	MCS4 Channel 0 error interrupt 0 No interrupt occurred 1 Interrupt was asserted by the corresponding submodule

Chapter 19
Output Compare Unit (CMP)

19.1 CMP Overview

The Output Compare Unit (CMP) is designed for use in safety relevant applications, where multiple redundancy checks are required. That is, duplicate outputs from the TOM and ATOM submodules are compared in the CMP to assure data integrity. A simple EXOR function is used to facilitate compares. Therefore, it is necessary to ensure total, cycle accurate, output behavior from the TOM or ATOM submodules that source the signals that are to be compared. For example, cycle accurate output behavior must be assured when:

- two ATOM channels produce output signals that are triggered by the same time stamp value, or
- two TOM channels have the same configuration and start their output generation is triggered at the same time.

TOM channel triggering mechanisms are shown in TOM Overview.

The CMP performs a 2 x 24 comparison of neighboring channels from the TOM and ATOM submodules, respectively. Using three ATOM and two TOM modules (for example), channel 0 (from both the ATOM and TOM) is compared with channel 1, channel 2 with 3, and so on until channel 22 is compared with channel 23, as shown in the following figure.

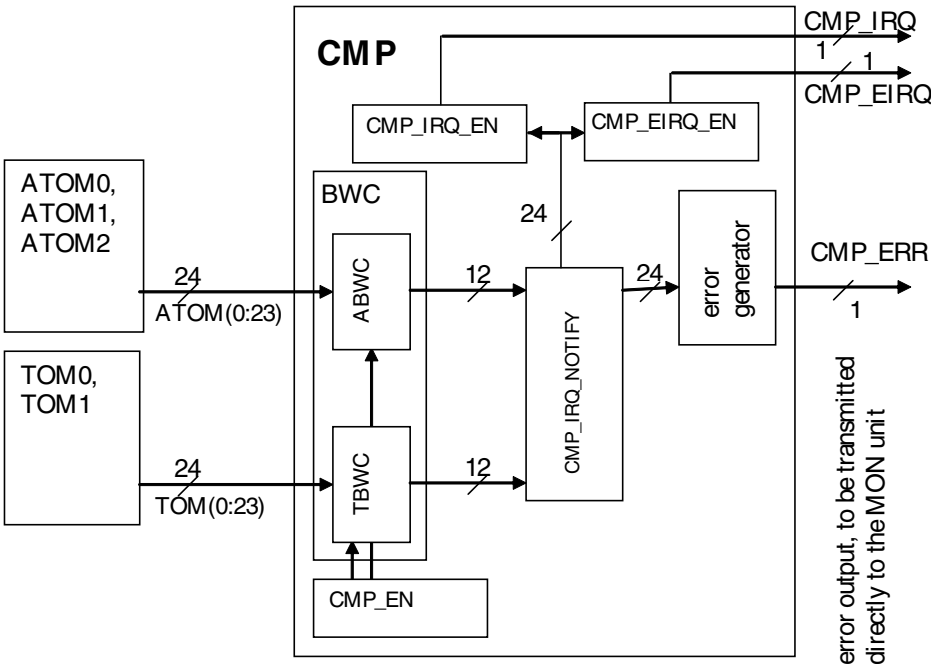


Figure 19-1. CMP architecture

19.2 Bitwise Compare Unit (BWC)

The BWC units (TBWC and ABWC) compare, in neighboring pairs, the channel combinations shown in Table 19-1.

Table 19-1. Bit-wise comparisons

TBWC/ABWC Comparator Number	TOM/ATOM Input Channel	TOM/ATOM Input Channel	Comparator Output Bit Number
0	0	1	0
1	2	3	1
2	4	5	2
3	6	7	3
4	8	9	4
5	10	11	5
6	12	13	6
7	14	15	7
8	16	17	8
9	18	19	9
10	20	21	10
11	22	23	11

19.3 Configuration of the Compare Unit

The Compare Unit consists of:

- 24 EXOR elements,
- a **CMP_EN** register that configures the corresponding comparisons, and
- a **CMP_IRQ_NOTIFY** status register that shows and stores each (configured) mismatch result.

When a mismatch error occurs, a *CMP_IRQ* interrupt request is asserted (if enabled in the CMP_IRQ_EN register) and a *CMP_EIRQ* interrupt request is asserted (if enabled in the CMP_EIRQ_EN register).

19.4 Error Generator

The error generator generates a CMP_ERR error signal that is routed directly to the Monitor (MON) unit, independently from the *CMP_IRQ* interrupt signal. The CMP_ERR error signal is asserted when at least one bit is set in the **CMP_IRQ_NOTIFY** register. For this purpose, the CMP_IRQ_NOTIFY bits are not masked by the CMP_IRQ_EN bits.

The MON stores the status of *CMP_ERR* signal its MON_STATUS register, which can be polled by the CPU.

19.5 CMP Interrupt Signal

The CMP submodule has two interrupt signals, one normal interrupt and one error interrupt. The source of both interrupts can be determined by reading the CMP_IRQ_NOTIFY, CMP_IRQ_EN and CMP_EIRQ_EN registers. Each source can be separately forced (for debug purposes) using the interrupt force CMP_IRQ_FORCINT register. CMP_IRQ_MODE configures interrupt output characteristics. All interrupt modes are described in GTM Interrupt Concept.

Table 19-2. CMP interrupts

Signal	Description
CMP_IRQ	Mismatching interrupt of outputs to be compared, when enabled
CMP_EIRQ	Mismatching interrupt of outputs to be compared, when enabled

19.6 Memory Map and Registers

The Compare (CMP) module registers are described as follows:

CMP memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Compare Enable Register (CMP_EN)	32	R/W	0000_0000h	19.6.1/816
4	Compare Interrupt Request Notification Register (CMP_IRQ_NOTIFY)	32	R/W	0000_0000h	19.6.2/818
8	Compare Interrupt Request Enable Register (CMP_IRQ_EN)	32	R/W	0000_0000h	19.6.3/820
C	Compare Force Interrupt Request Register (CMP_IRQ_FORCINT)	32	R/W	0000_0000h	19.6.4/822
10	Compare Interrupt Request Mode Register (CMP_IRQ_MODE)	32	R/W	See section	19.6.5/824
14	Compare Error Interrupt Request Enable register (CMP_EIRQ_EN)	32	R/W	0000_0000h	19.6.6/825

19.6.1 Compare Enable Register (CMP_EN)

Address: 200h base + 0h offset = 200h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0								TBWC11_EN	TBWC10_EN	TBWC9_EN	TBWC8_EN	TBWC7_EN	TBWC6_EN	TBWC5_EN	TBWC4_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TBWC3_EN	TBWC2_EN	TBWC1_EN	TBWC0_EN	ABWC11_EN	ABWC10_EN	ABWC9_EN	ABWC8_EN	ABWC7_EN	ABWC6_EN	ABWC5_EN	ABWC4_EN	ABWC3_EN	ABWC2_EN	ABWC1_EN	ABWC0_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMP_EN field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 TBWC11_EN	See TBWC0_EN.
9 TBWC10_EN	See TBWC0_EN.
10 TBWC9_EN	See TBWC0_EN.
11 TBWC8_EN	See TBWC0_EN.
12 TBWC7_EN	See TBWC0_EN.
13 TBWC6_EN	See TBWC0_EN.
14 TBWC5_EN	See TBWC0_EN.
15 TBWC4_EN	See TBWC0_EN.
16 TBWC3_EN	See TBWC0_EN.
17 TBWC2_EN	See TBWC0_EN.
18 TBWC1_EN	See TBWC0_EN.
19 TBWC0_EN	Enable corresponding comparator in TBWC. 0 Corresponding TBWC comparator is disabled. 1 Corresponding TBWC comparator is enabled.
20 ABWC11_EN	See ABWC0_EN.
21 ABWC10_EN	See ABWC0_EN.
22 ABWC9_EN	See ABWC0_EN.
23 ABWC8_EN	See ABWC0_EN.
24 ABWC7_EN	See ABWC0_EN.
25 ABWC6_EN	See ABWC0_EN.
26 ABWC5_EN	See ABWC0_EN.
27 ABWC4_EN	See ABWC0_EN.
28 ABWC3_EN	See ABWC0_EN.

Table continues on the next page...

CMP_EN field descriptions (continued)

Field	Description
29 ABWC2_EN	See ABWC0_EN.
30 ABWC1_EN	See ABWC0_EN.
31 ABWC0_EN	Enable corresponding comparator in ABWC. 0 Corresponding ABWC comparator is disabled. 1 Corresponding ABWC comparator is enabled.

19.6.2 Compare Interrupt Request Notification Register (CMP_IRQ_NOTIFY)

NOTE

Bit[8:31] will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Address: 200h base + 4h offset = 204h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0								TBWC11	TBWC10	TBWC9	TBWC8	TBCW7	TBWC6	TBWC5	TBWC4
W									w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TBWC3	TBWC2	TBWC1	TBWC0	ABWC11	ABWC10	ABWC9	ABWC8	ABWC7	ABWC6	ABWC5	ABWC4	ABWC3	ABWC2	ABWC1	ABWC0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMP_IRQ_NOTIFY field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 TBWC11	See TBWC0.
9 TBWC10	See TBWC0.
10 TBWC9	See TBWC0.
11 TBWC8	See TBWC0.
12 TBCW7	See TBWC0.
13 TBWC6	See TBWC0.
14 TBWC5	See TBWC0.
15 TBWC4	See TBWC0.
16 TBWC3	See TBWC0.
17 TBWC2	See TBWC0.
18 TBWC1	See TBWC0.
19 TBWC0	TOM sub modules outputs bitwise comparator 0 error indication. 0 No error recognized on TOM sub modules bits 0 and 1 (see chapter 19.2). 1 An error was recognized on corresponding TOM sub modules bits.
20 ABWC11	See ABWC0.
21 ABWC10	See ABWC0.
22 ABWC9	See ABWC0.
23 ABWC8	See ABWC0.
24 ABWC7	See ABWC0.
25 ABWC6	See ABWC0.
26 ABWC5	See ABWC0.
27 ABWC4	See ABWC0.
28 ABWC3	See ABWC0.

Table continues on the next page...

CMP_IRQ_NOTIFY field descriptions (continued)

Field	Description
29 ABWC2	See ABWC0.
30 ABWC1	See ABWC0.
31 ABWC0	ATOM sub modules outputs bitwise comparator error indication. 0 No error recognized on ATOM sub modules bits 0 and 1 (see chapter 19.2). 1 An error was recognized on corresponding ATOM sub modules bits.

19.6.3 Compare Interrupt Request Enable Register (CMP_IRQ_EN)

Address: 200h base + 8h offset = 208h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0								TBWC11_EN_IRQ	TBWC10_EN_IRQ	TBWC9_EN_IRQ	TBWC8_EN_IRQ	TBWC7_EN_IRQ	TBWC6_EN_IRQ	TBWC5_EN_IRQ	TBWC4_EN_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TBWC3_EN_IRQ	TBWC2_EN_IRQ	TBWC1_EN_IRQ	TBWC0_EN_IRQ	ABWC11_EN_IRQ	ABWC10_EN_IRQ	ABWC9_EN_IRQ	ABWC8_EN_IRQ	ABWC7_EN_IRQ	ABWC6_EN_IRQ	ABWC5_EN_IRQ	ABWC4_EN_IRQ	ABWC3_EN_IRQ	ABWC2_EN_IRQ	ABWC1_EN_IRQ	ABWC0_EN_IRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMP_IRQ_EN field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 TBWC11_EN_IRQ	See TBWC0_EN_IRQ.
9 TBWC10_EN_IRQ	See TBWC0_EN_IRQ.
10 TBWC9_EN_IRQ	See TBWC0_EN_IRQ.
11 TBWC8_EN_IRQ	See TBWC0_EN_IRQ.

Table continues on the next page...

CMP_IRQ_EN field descriptions (continued)

Field	Description
12 TBWC7_EN_IRQ	See TBWC0_EN_IRQ.
13 TBWC6_EN_IRQ	See TBWC0_EN_IRQ.
14 TBWC5_EN_IRQ	See TBWC0_EN_IRQ.
15 TBWC4_EN_IRQ	See TBWC0_EN_IRQ.
16 TBWC3_EN_IRQ	See TBWC0_EN_IRQ.
17 TBWC2_EN_IRQ	See TBWC0_EN_IRQ.
18 TBWC1_EN_IRQ	See TBWC0_EN_IRQ.
19 TBWC0_EN_IRQ	Enable corresponding TBWC interrupt source for CMP_IRQ line. 0 Corresponding interrupt source TBWC is disabled. 1 Corresponding interrupt source TBWC is enabled.
20 ABWC11_EN_IRQ	See ABWC0_EN_IRQ.
21 ABWC10_EN_IRQ	See ABWC0_EN_IRQ.
22 ABWC9_EN_IRQ	See ABWC0_EN_IRQ.
23 ABWC8_EN_IRQ	See ABWC0_EN_IRQ.
24 ABWC7_EN_IRQ	See ABWC0_EN_IRQ.
25 ABWC6_EN_IRQ	See ABWC0_EN_IRQ.
26 ABWC5_EN_IRQ	See ABWC0_EN_IRQ.
27 ABWC4_EN_IRQ	See ABWC0_EN_IRQ.
28 ABWC3_EN_IRQ	See ABWC0_EN_IRQ.
29 ABWC2_EN_IRQ	See ABWC0_EN_IRQ.
30 ABWC1_EN_IRQ	See ABWC0_EN_IRQ.
31 ABWC0_EN_IRQ	Enable corresponding ABWC interrupt source for CMP_IRQ line. 0 Corresponding interrupt source ABWC is disabled. 1 Corresponding interrupt source ABWC is enabled.

19.6.4 Compare Force Interrupt Request Register (CMP_IRQ_FORCINT)

NOTE

Bit[8:31] is write protected by bit RF_PROT of register GTM_CTRL.

Address: 200h base + Ch offset = 20Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0								TRG_TBWC11	TRG_TBWC10	TRG_TBWC9	TRG_TBWC8	TRG_TBWC7	TRG_TBWC6	TRG_TBWC5	TRG_TBWC4
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TRG_TBWC3	TRG_TBWC2	TRG_TBWC1	TRG_TBWC0	TRG_ABWC11	TRG_ABWC10	TRG_ABWC9	TRG_ABWC8	TRG_ABWC7	TRG_ABWC6	TRG_ABWC5	TRG_ABWC4	TRG_ABWC3	TRG_ABWC2	TRG_ABWC1	TRG_ABWC0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMP_IRQ_FORCINT field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 TRG_TBWC11	See TRG_TBWC0.
9 TRG_TBWC10	See TRG_TBWC0.
10 TRG_TBWC9	See TRG_TBWC0.
11 TRG_TBWC8	See TRG_TBWC0.
12 TRG_TBWC7	See TRG_TBWC0.
13 TRG_TBWC6	See TRG_TBWC0.
14 TRG_TBWC5	See TRG_TBWC0.
15 TRG_TBWC4	See TRG_TBWC0.

Table continues on the next page...

CMP_IRQ_FORCINT field descriptions (continued)

Field	Description
16 TRG_TBWC3	See TRG_TBWC0.
17 TRG_TBWC2	See TRG_TBWC0.
18 TRG_TBWC1	See TRG_TBWC0.
19 TRG_TBWC0	Trigger corresponding TBWC bit in CMP_IRQ_NOTIFY register by software. Bit[8:19] is cleared automatically after write. Bit[8:19] is write protected by bit GTM_CTRL[RF_PROT]. 0 No event triggering. 1 Assert corresponding field in CMP_IRQ_NOTIFY register.
20 TRG_ABWC11	See TRG_ABWC0.
21 TRG_ABWC10	See TRG_ABWC0.
22 TRG_ABWC9	See TRG_ABWC0.
23 TRG_ABWC8	See TRG_ABWC0.
24 TRG_ABWC7	See TRG_ABWC0.
25 TRG_ABWC6	See TRG_ABWC0.
26 TRG_ABWC5	See TRG_ABWC0.
27 TRG_ABWC4	See TRG_ABWC0.
28 TRG_ABWC3	See TRG_ABWC0.
29 TRG_ABWC2	See TRG_ABWC0.
30 TRG_ABWC1	See TRG_ABWC0.
31 TRG_ABWC0	Trigger corresponding ABWC bit in CMP_IRQ_NOTIFY register by software. Bit[20:31] is cleared automatically after write. 0 No event triggering. 1 Assert corresponding field in CMP_IRQ_NOTIFY register.

19.6.5 Compare Interrupt Request Mode Register (CMP_IRQ_MODE)

Address: 200h base + 10h offset = 210h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0															
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0														IRQ_MODE	
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- Bits 28-31 do not reset to a given value.

CMP_IRQ_MODE field descriptions

Field	Description
0–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–31 IRQ_MODE	IRQ mode selection. 00 Level mode. 01 Pulse mode. 10 Pulse-Notify mode. 11 Single-Pulse mode.

19.6.6 Compare Error Interrupt Request Enable register (CMP_EIRQ_EN)

Address: 200h base + 14h offset = 214h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0								TBWC11_EN_EIRQ	TBWC10_EN_EIRQ	TBWC9_EN_EIRQ	TBWC8_EN_EIRQ	TBWC7_EN_EIRQ	TBWC6_EN_EIRQ	TBWC5_EN_EIRQ	TBWC4_EN_EIRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TBWC3_EN_EIRQ	TBWC2_EN_EIRQ	TBWC1_EN_EIRQ	TBWC0_EN_EIRQ	ABWC11_EN_EIRQ	ABWC10_EN_EIRQ	ABWC9_EN_EIRQ	ABWC8_EN_EIRQ	ABWC7_EN_EIRQ	ABWC6_EN_EIRQ	ABWC5_EN_EIRQ	ABWC4_EN_EIRQ	ABWC3_EN_EIRQ	ABWC2_EN_EIRQ	ABWC1_EN_EIRQ	ABWC0_EN_EIRQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMP_EIRQ_EN field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 TBWC11_EN_EIRQ	See TBWC0_EN_EIRQ.
9 TBWC10_EN_EIRQ	See TBWC0_EN_EIRQ.
10 TBWC9_EN_EIRQ	See TBWC0_EN_EIRQ.
11 TBWC8_EN_EIRQ	See TBWC0_EN_EIRQ.
12 TBWC7_EN_EIRQ	See TBWC0_EN_EIRQ.
13 TBWC6_EN_EIRQ	See TBWC0_EN_EIRQ.
14 TBWC5_EN_EIRQ	See TBWC0_EN_EIRQ.

Table continues on the next page...

CMP_EIRQ_EN field descriptions (continued)

Field	Description
15 TBWC4_EN_ EIRQ	See TBWC0_EN_EIRQ.
16 TBWC3_EN_ EIRQ	See TBWC0_EN_EIRQ.
17 TBWC2_EN_ EIRQ	See TBWC0_EN_EIRQ.
18 TBWC1_EN_ EIRQ	See TBWC0_EN_EIRQ.
19 TBWC0_EN_ EIRQ	Enable comparator 0-11 in TBWC. 0 Interrupt source TBWC0-11 is disabled 1 Interrupt source TBWC0-11 is enabled
20 ABWC11_EN_ EIRQ	ABWC0_EN_EIRQ
21 ABWC10_EN_ EIRQ	ABWC0_EN_EIRQ
22 ABWC9_EN_ EIRQ	ABWC0_EN_EIRQ
23 ABWC8_EN_ EIRQ	ABWC0_EN_EIRQ
24 ABWC7_EN_ EIRQ	ABWC0_EN_EIRQ
25 ABWC6_EN_ EIRQ	ABWC0_EN_EIRQ
26 ABWC5_EN_ EIRQ	ABWC0_EN_EIRQ
27 ABWC4_EN_ EIRQ	ABWC0_EN_EIRQ
28 ABWC3_EN_ EIRQ	ABWC0_EN_EIRQ
29 ABWC2_EN_ EIRQ	ABWC0_EN_EIRQ
30 ABWC1_EN_ EIRQ	ABWC0_EN_EIRQ

Table continues on the next page...

CMP_EIRQ_EN field descriptions (continued)

Field	Description
31 ABWC0_EN_ EIRQ	Enable comparator 0-11 in ABWC. 0 Interrupt source ABWC0-11 is disabled 1 Interrupt source ABWC0-11 is enabled

Chapter 20
Monitor Unit (MON)

20.1 MON Overview

For safety relevant applications, the Monitor Unit (MON) is designed to provide supervision of commonly used circuitry and resources. The MON (shown in Figure 20-1) supervises the activity of clocks and the characteristics of output signals.

Output signal characteristics can be checked by routing them from the TOM or ATOM to the inputs of the TIM submodules, where they are routed (via the ARU) to an MCS channel. The corresponding MCS channel performs the check according to given properties. If the comparison fails, an error signal is generated in the MCS channel. Error signals from all of the channels within a single MCS are ORed and stored in the MCS[i]_STA[ERR] bit . The ERR bit (from each MCS) is routed to the MON, where it is stored in the MON_STATUS[MCS(3:0)_ERR] bit field.

The MCS generates an 8-bit MCS[3:0] _CH[7:0]_MCA activity signal for each comparison. The activity signal can also be used to implicitly check the activity of the TIM, ARU, and associated clocks. ARU cycle time can be compared, in an MCS channel, to a given value.

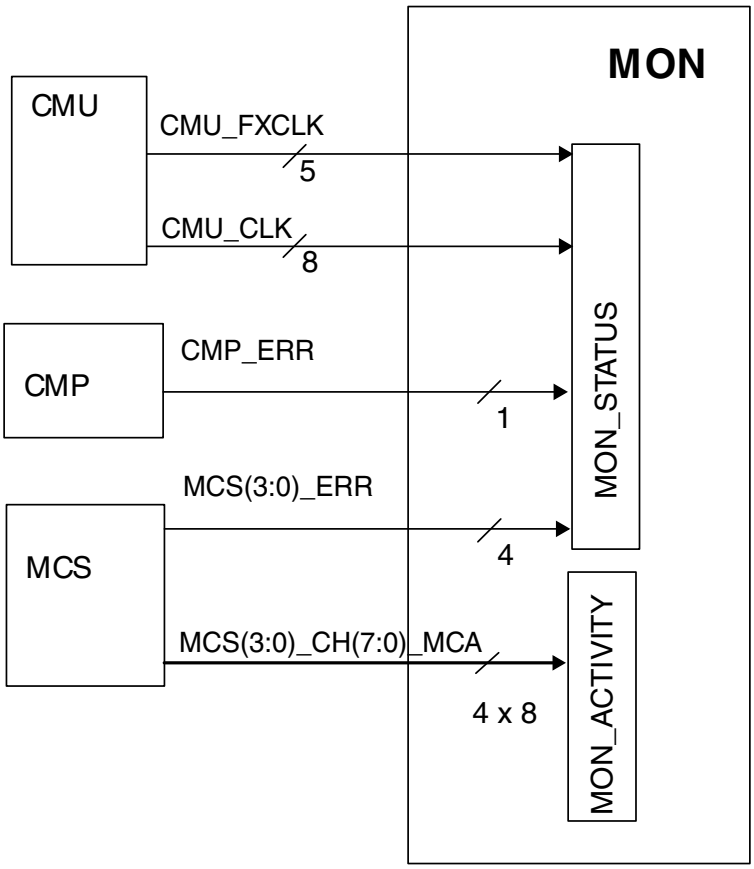


Figure 20-1. MON block diagram

20.1.1 Realization without Activity Checker for the clock signals

An activity checker for clock signals is not provided because the clock signals are used as enables for the system clock. Therefore, the clock signals (enables) only need to be checked to show when they are asserted (high).

20.2 Clock Monitoring

The *CMU_CLKx* (x=0..7) and *CMU_FXCLKy* (y=0..4) clock (enable) signals are routed from the CMU submodule to various destinations, including the MON.

Supervision of these clocks (enables) is accomplished by scanning for assertion (high) activity of each clock (enable). When a high value of the clock (enable) is detected, the corresponding bit in the **MON_STATUS** status register is set. to 1. A status register bit is reset by writing a 1 to it.

When the MON_STATUS register is polled by the CPU, and the time between two read accesses is longer than the period of the slowest clock, all bits for the corresponding clocks (enables) will have been set.

When the time between two read accesses is shorter than the period of the slowest clock, activity for all clocks (enables) cannot be shown, although the clocks (enables) might be working properly.

The MON does not have an enable register to control which clock (enable) status is stored in the MON_STATUS register. Therefore, monitoring of the MON_STATUS register must be confined only to those bits that have corresponding clocks enabled in the CMU_CLK_EN register.

20.3 CMP error Monitoring

The CMP_ERR signal is received directly from CMP submodule. It is set (high) if one or more errors are detected in the CMP.

20.4 MCS Checking for Characteristics of Signals

Using the MCS, some given properties of the TOM and ATOM submodule output signals can be checked. The output signals are routed to inputs of the TIM submodules, where they are routed (via the ARU) to MCS channels.

The corresponding MCS signal performs the check according to given properties. In this way signal high or low time as well as signal periods can be checked, also taking into account tolerances. When the check fails a MCS internal error signal is generated and ORed with the error signals of the other channels of the MCS module to an summarized error signal MCS[z]_ERR (z=0..3).

For each MCS, a summarized error signal is transmitted to MON and monitored in the MON_STATUS register. In order to check the execution of the comparison for each MCS channel an activity signal is generated. In the MCS[z]_CH[x]_MCA_i_x (x=0..7) vector always for each MCS 8 bits for each channels are combined. The activity signals are stored in the MON_ACTIVITY_0 register for MCS0 to MCS3 and in the MON_ACTIVITY_1 register for MCS4 to MCS6. The bits are set by a one signal and reset by writing a one to it (preferably after polling the status of the register).

Because an activity signal shows the execution of a comparison, the units that provide the signals for the comparison (e.g. TIM, ARU and MCS itself) as well as the involved clocks and time bases are implicitly checked for proper operation.

20.5 Checking ARU Cycle Time

An MCS channel can be used to check ARU cycle time. This check can be performed by an MCS channel. It should be noted that the MCS program for measuring the ARU round trip time must add a tolerance value. The resulting error is reported to the MON unit using the summarized MCS[z]_ERR error signal for each MCS module in addition to an interrupt, generated in the MCS. The same signals and status bits are used as in the case of checking the signal characteristics.

The corresponding MCS is programmed to get a fixed data value at address 0x1FF. The data value is always zero and is not blocked. When getting the access, the TBU_TS0 time stamp value is stored in a register. The next time getting the access, the new TBU_TS0 value is stored and the difference between both values is compared with a given value. When the comparison fails:

- the MCS[i]_STA[ERR] error flag is set to 1,
- an interrupt is generated by the MCS, and
- the MCS_ERR error signal is generated and routed to the MON.

When the check is performed, an MCS_CHx_MCA activity signal is provided for each MCS channel and it is routed to the MON, where it is stored in the MON_ACTIVITY register. The bits in the MON_ACTIVITY register are reset by writing a 1 to them.

When the check fails, an interrupt is generated and the MCS[z]_ERR error signal is provided for the MON unit.

20.6 MON Interrupt Signals

The MON submodule has no interrupt signals.

20.7 Memory Map and Registers

The Monitor (MON) module registers are described as follows:

MON memory map

Address offset (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Monitor Status Register (MON_STATUS)	32	R/W	0000_0000h	20.7.1/833
4	Monitor Acitivity 0 (MON_ACTIVITY_0)	32	R/W	0000_0000h	20.7.2/835
8	Monitor Acitivity 1 (MON_ACTIVITY_1)	32	R/W	0000_0000h	20.7.3/838

20.7.1 Monitor Status Register (MON_STATUS)

Address: 180h base + 0h offset = 180h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0					MCS6_ERR	MCS5_ERR	MCS4_ERR	MCS3_ERR	MCS2_ERR	MCS1_ERR	MCS0_ERR	0			CMP_ERR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0			ACT_CMUFX4	ACT_CMUFX3	ACT_CMUFX2	ACT_CMUFX1	ACT_CMUFX0	ACT_CMU7	ACT_CMU6	ACT_CMU5	ACT_CMU4	ACT_CMU3	ACT_CMU2	ACT_CMU1	ACT_CMU0
W				w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MON_STATUS field descriptions

Field	Description
0–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 MCS6_ERR	See MCS0_ERR.
6 MCS5_ERR	See MCS0_ERR.
7 MCS4_ERR	See MCS0_ERR.
8 MCS3_ERR	See MCS0_ERR.
9 MCS2_ERR	See MCS0_ERR.
10 MCS1_ERR	See MCS0_ERR.
11 MCS0_ERR	Error detected at corresponding MCS. **NOTE:** The MCS can be programmed to generate an error, when the comparison of signal values (duty time, cycle time) fails or also when the cycle time of the ARU (checking of the TBU_TS0 between two periodic accesses) is out of the expected range. **NOTE:** Bit[5:11] is set when the corresponding unit reports an error
12–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 CMP_ERR	Error detected at CMP. **NOTE:** Bit 15 is set when the corresponding unit reports an error.
16–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 ACT_CMUFX4	See ACT_CMUFX0.
20 ACT_CMUFX3	See ACT_CMUFX0.
21 ACT_CMUFX2	See ACT_CMUFX0.
22 ACT_CMUFX1	See ACT_CMUFX0.
23 ACT_CMUFX0	Corresponding CMU_CLKFX activity. Bit[19:23] is set when a high low slope is detected at the considered clock. **NOTE:** Bit[19:23] is cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
24 ACT_CMU7	See ACT_CMU0.
25 ACT_CMU6	See ACT_CMU0.
26 ACT_CMU5	See ACT_CMU0.

Table continues on the next page...

MON_STATUS field descriptions (continued)

Field	Description
27 ACT_CMU4	See ACT_CMU0.
28 ACT_CMU3	See ACT_CMU0.
29 ACT_CMU2	See ACT_CMU0.
30 ACT_CMU1	See ACT_CMU0.
31 ACT_CMU0	Corresponding CMU_CLK activity. Bit[24:31] is set when a high low slope is detected at the considered clock. **NOTE:** Bit[24:31] is cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

20.7.2 Monitor Acitivity 0 (MON_ACTIVITY_0)

NOTE

When not all MCS modules are implemented or the channels are not used for check purposes with supervising, the corresponding activity bits remain zero.

Address: 180h base + 4h offset = 184h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	MCA_3_7	MCA_3_6	MCA_3_5	MCA_3_4	MCA_3_3	MCA_3_2	MCA_3_1	MCA_3_0	MCA_2_7	MCA_2_6	MCA_2_5	MCA_2_4	MCA_2_3	MCA_2_2	MCA_2_1	MCA_2_0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	MCA_1_7	MCA_1_6	MCA_1_5	MCA_1_4	MCA_1_3	MCA_1_2	MCA_1_1	MCA_1_0	MCA_0_7	MCA_0_6	MCA_0_5	MCA_0_4	MCA_0_3	MCA_0_2	MCA_0_1	MCA_0_0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MON_ACTIVITY_0 field descriptions

Field	Description
0 MCA_3_7	See MCA_3_0.
1 MCA_3_6	See MCA_3_0.
2 MCA_3_5	See MCA_3_0.
3 MCA_3_4	See MCA_3_0.
4 MCA_3_3	See MCA_3_0.
5 MCA_3_2	See MCA_3_0.
6 MCA_3_1	See MCA_3_0.
7 MCA_3_0	Activity of check performed in module MCS[3] at corresponding channel. **NOTE:** Bit[0:7] will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
8 MCA_2_7	See MCA_2_0.
9 MCA_2_6	See MCA_2_0.
10 MCA_2_5	See MCA_2_0.
11 MCA_2_4	See MCA_2_0.
12 MCA_2_3	See MCA_2_0.
13 MCA_2_2	See MCA_2_0.
14 MCA_2_1	See MCA_2_0.
15 MCA_2_0	Activity of check performed in module MCS[2] at corresponding channel. **NOTE:** Bit[8:16] will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
16 MCA_1_7	See MCA_1_0.
17 MCA_1_6	See MCA_1_0.
18 MCA_1_5	See MCA_1_0.
19 MCA_1_4	See MCA_1_0.

Table continues on the next page...

MON_ACTIVITY_0 field descriptions (continued)

Field	Description
20 MCA_1_3	See MCA_1_0.
21 MCA_1_2	See MCA_1_0.
22 MCA_1_1	See MCA_1_0.
23 MCA_1_0	Activity of check performed in module MCS[1] at corresponding channel. **NOTE:** Bit[17:23] will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
24 MCA_0_7	See MCA_0_0.
25 MCA_0_6	See MCA_0_0.
26 MCA_0_5	See MCA_0_0.
27 MCA_0_4	See MCA_0_0.
28 MCA_0_3	See MCA_0_0.
29 MCA_0_2	See MCA_0_0.
30 MCA_0_1	See MCA_0_0.
31 MCA_0_0	Activity of check performed in module MCS[0] at corresponding channel. **NOTE:** Bit[24:31] will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

20.7.3 Monitor Acitivity 1 (MON_ACTIVITY_1)

NOTE

When not all MCS modules are implemented or the channels are not used for check purposes with supervising, the corresponding activity bits remain zero.

Address: 180h base + 8h offset = 188h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R									MCA_6_7	MCA_6_6	MCA_6_5	MCA_6_4	MCA_6_3	MCA_6_2	MCA_6_1	MCA_6_0
W	0								w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	MCA_5_7	MCA_5_6	MCA_5_5	MCA_5_4	MCA_5_3	MCA_5_2	MCA_5_1	MCA_5_0	MCA_4_7	MCA_4_6	MCA_4_5	MCA_4_4	MCA_4_3	MCA_4_2	MCA_4_1	MCA_4_0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MON_ACTIVITY_1 field descriptions

Field	Description
0–7 Reserved	This field is reserved.
8 MCA_6_7	See MCA_6_0.
9 MCA_6_6	See MCA_6_0.
10 MCA_6_5	See MCA_6_0.
11 MCA_6_4	See MCA_6_0.

Table continues on the next page...

MON_ACTIVITY_1 field descriptions (continued)

Field	Description
12 MCA_6_3	See MCA_6_0.
13 MCA_6_2	See MCA_6_0.
14 MCA_6_1	See MCA_6_0.
15 MCA_6_0	Activity of check performed in module MCS[6] at corresponding channel. **NOTE:** Bit[8:16] will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
16 MCA_5_7	See MCA_5_0.
17 MCA_5_6	See MCA_5_0.
18 MCA_5_5	See MCA_5_0.
19 MCA_5_4	See MCA_5_0.
20 MCA_5_3	See MCA_5_0.
21 MCA_5_2	See MCA_5_0.
22 MCA_5_1	See MCA_5_0.
23 MCA_5_0	Activity of check performed in module MCS[5] at corresponding channel. **NOTE:** Bit[17:23] will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
24 MCA_4_7	See MCA_4_0.
25 MCA_4_6	See MCA_4_0.
26 MCA_4_5	See MCA_4_0.
27 MCA_4_4	See MCA_4_0.
28 MCA_4_3	See MCA_4_0.
29 MCA_4_2	See MCA_4_0.
30 MCA_4_1	See MCA_4_0.
31 MCA_4_0	Activity of check performed in module MCS[4] at corresponding channel.

Table continues on the next page...

MON_ACTIVITY_1 field descriptions (continued)

Field	Description
	NOTE: Bit[24:31] will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Appendix A
Revision History

A.1 Revision history

Table A-1. Document revision history

Revision	Date	Description of changes
1	3 Feb 2015	Initial release.

Appendix B
GTM104 Appendix B

B.1 LEGAL NOTICE

EITHER EXPRESSLY OR IMPLICITLY, MAKES ANY REPRESENTATION OR WARRANTY REGARDING THE APPROPRIATENESS OF THE USE, OUTPUT, OR RESULTS OF THE USE OF THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, BEING CURRENT OR OTHERWISE. NOR DO THEY HAVE ANY OBLIGATION TO CORRECT ERRORS, MAKE CHANGES, SUPPORT THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO, DISTRIBUTE UPDATES, OR PROVIDE NOTIFICATION OF ANY ERROR OR DEFECT, KNOWN OR UNKNOWN. IF YOU RELY UPON THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO, YOU DO SO AT YOUR OWN RISK, AND YOU ASSUME THE RESPONSIBILITY FOR THE RESULTS. SHOULD THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL LOSSES, INCLUDING, BUT NOT LIMITED TO, ANY NECESSARY SERVICING, REPAIR OR CORRECTION OF ANY PROPERTY INVOLVED TO THE MAXIMUM EXTEND PERMITTED BY LAW.

DISCLAIMER: IN NO EVENT, UNLESS REQUIRED BY LAW OR AGREED TO IN WRITING, SHALL THE INTELLECTUAL PROPERTY OWNERS, COPYRIGHT HOLDERS OR ANY PERSON BE LIABLE FOR ANY LOSS, EXPENSE OR DAMAGE, OF ANY TYPE OR NATURE ARISING OUT OF THE USE OF, OR INABILITY TO USE THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO, INCLUDING, BUT NOT LIMITED TO, CLAIMS, SUITS OR CAUSES OF ACTION INVOLVING ALLEGED INFRINGEMENT OF COPYRIGHTS, PATENTS, TRADEMARKS, TRADE SECRETS, OR UNFAIR COMPETITION.

INDEMNIFICATION: TO THE MAXIMUM EXTEND PERMITTED BY LAW YOU AGREE TO INDEMNIFY AND HOLD HARMLESS THE INTELLECTUAL PROPERTY OWNERS, COPYRIGHT HOLDERS AND CONTRIBUTORS, AND EMPLOYEES, AND ANY PERSON FROM AND AGAINST ALL CLAIMS, LIABILITIES, LOSSES, CAUSES OF ACTION, DAMAGES, JUDGMENTS, AND EXPENSES, INCLUDING THE REASONABLE COST OF ATTORNEYS' FEES AND COURT COSTS, FOR INJURIES OR DAMAGES TO THE PERSON OR PROPERTY OF THIRD PARTIES, INCLUDING, WITHOUT LIMITATIONS, CONSEQUENTIAL, DIRECT AND INDIRECT DAMAGES AND ANY ECONOMIC LOSSES, THAT ARISE OUT OF OR IN CONNECTION WITH YOUR USE, MODIFICATION, OR DISTRIBUTION OF THIS SPECIFICATION, SOFTWARE RELATED THERETO, CODE AND/OR PROGRAM RELATED THERETO, ITS OUTPUT, OR ANY ACCOMPANYING DOCUMENTATION.

GOVERNING LAW: THE RELATIONSHIP BETWEEN YOU AND ROBERT BOSCH GMBH SHALL BE GOVERNED SOLELY BY THE LAWS OF THE FEDERAL REPUBLIC OF GERMANY. THE STIPULATIONS OF INTERNATIONAL CONVENTIONS REGARDING THE INTERNATIONAL SALE OF GOODS SHALL NOT BE APPLICABLE. THE EXCLUSIVE LEGAL VENUE SHALL BE DUESSELDORF, GERMANY.

MANDATORY LAW SHALL BE UNAFFECTED BY THE FOREGOING PARAGRAPHS.

INTELLECTUAL PROPERTY OWNERS/COPYRIGHT OWNERS/CONTRIBUTORS: ROBERT BOSCH GMBH, ROBERT BOSCH PLATZ 1, 70839 GERLINGEN, GERMANY AND ITS LICENSORS.

B.2 GTM configuration

Table B-1. GTM configuration

Module	Instances
Broadcast (BRC)	1
Parameter Storage Module (PSM)	2
Timer Input Module (TIM)	6
Timer Output Module (TOM)	5
ARI-connected Timer Output Module (ATOM)	9
Multi-Channel Sequencer (MCS)	6
Digital Phase-Locked Loop (DPLL)	1
Sensor Pattern Evaluation (SPE)	4
Output Compare Unit (CMP)	1
Monitor Unit (MON)	1
TIM0 Input Mapping Module (MAP)	1
Memory Configuration (MCFG)	1
Time Base Unit (TBU)	1
Advanced Routing Unit (ARU)	1
Clock Management Unit (CMU)	1
Interrupt Concentrator Module (ICM)	1
First-In-First-Out (FIFO)	2
FIFO-to-ARU (F2A)	2
AEI-to-FIFO Data Interface (AFD)	2
Architecture	1

B.3 Base addresses for GTM modules

Base addresses for GTM modules are shown in the following table.

Table B-2. Base addresses for GTM modules

Submodule	Base address
GTM Global	0x00000000
BRIDGE	0x00000030
MAP	0x00000F00
MCFG	0x00000F40
TBU	0x00000100
MON	0x00000180
CMP	0x00000200
ARU	0x00000280
CMU	0x00000300
ICM	0x00000600
SPE0	0x00000800
SPE1	0x00000880
SPE2	0x00000900
SPE3	0x00000980
TIM0	0x00001000
TIM1	0x00001800
TIM2	0x00002000
TIM3	0x00002800
TIM4	0x00003000
TIM5	0x00003800
TOM0	0x00008000
TOM1	0x00008800
TOM2	0x00009000
TOM3	0x00009800
TOM4	0x0000A000
ATOM0	0x0000D000
ATOM1	0x0000D800
ATOM2	0x0000E000
ATOM3	0x0000E800
ATOM4	0x0000F000
ATOM5	0x0000F800
ATOM6	0x00010000
ATOM7	0x00010800
ATOM8	0x00011000

Table continues on the next page...

Table B-2. Base addresses for GTM modules (continued)

F2A0	0x00018000
AFD0	0x00018080
FIFO0	0x00018400
FIFO0_MEMORY	0x00019000
F2A1	0x0001C000
AFD1	0x0001C080
FIFO1	0x0001C400
FIFO1_MEMORY	0x0001D000
DPLL	0x00028000
DPLL_RAM1A	0x00028200
DPLL_RAM1BC	0x00028400
DPLL_RAM2	0x0002C000
MCS0	0x00030000
MCS0_MEMORY	0x00038000
MCS1	0x00031000
MCS1_MEMORY	0x00040000
MCS2	0x00032000
MCS2_MEMORY	0x00048000
MCS3	0x00033000
MCS3_MEMORY	0x00050000
MCS4	0x00034000
MCS4_MEMORY	0x00058000
MCS5	0x00035000
MCS5_MEMORY	0x00060000

B.4 Full addresses for GTM registers

Table B-3. Full addresses of GTM registers

Address name	Address number
GTM_REV	0x00000000
GTM_RST	0x00000004
GTM_CTRL	0x00000008
GTM_AEI_ADDR_XPT	0x0000000C
GTM_IRQ_NOTIFY	0x00000010
GTM_IRQ_EN	0x00000014
GTM_IRQ_FORCINT	0x00000018
GTM_IRQ_MODE	0x0000001C
GTM_EIRQ_EN	0x00000020

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

GTM_BRIDGE_MODE	0x00000030
GTM_BRIDGE_PTR1	0x00000034
GTM_BRIDGE_PTR2	0x00000038
GTM_TIM0_AUX_IN_SRC	0x00000040
GTM_TIM1_AUX_IN_SRC	0x00000044
GTM_TIM2_AUX_IN_SRC	0x00000048
GTM_TIM3_AUX_IN_SRC	0x0000004C
GTM_TIM4_AUX_IN_SRC	0x00000050
GTM_TIM5_AUX_IN_SRC	0x00000054
MAP_CTRL_STAT	0x00000F00
MCFG_CTRL	0x00000F40
SVM_RESERVED0	0x000000C0
SVM_RESERVED1	0x000000C4
SVM_RESERVED2	0x000000C8
SVM_RESERVED3	0x000000CC
SVM_RESERVED4	0x000000D0
SVM_RESERVED5	0x000000D4
SVM_RESERVED6	0x000000D8
SVM_RESERVED7	0x000000DC
SVM_RESERVED8	0x000000E0
SVM_RESERVED9	0x000000E4
SVM_RESERVED10	0x000000E8
SVM_RESERVED11	0x000000EC
SVM_RESERVED12	0x000000F0
SVM_RESERVED13	0x000000F4
SVM_RESERVED14	0x000000F8
SVM_RESERVED15	0x000000FC
TBU_CHEN	0x00000100
TBU_CH0_CTRL	0x00000104
TBU_CH0_BASE	0x00000108
TBU_CH1_CTRL	0x0000010C
TBU_CH1_BASE	0x00000110
TBU_CH2_CTRL	0x00000114
TBU_CH2_BASE	0x00000118
MON_STATUS	0x00000180
MON_ACTIVITY_0	0x00000184
MON_ACTIVITY_1	0x00000188
CMP_EN	0x00000200
CMP_IRQ_NOTIFY	0x00000204
CMP_IRQ_EN	0x00000208

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

CMP_IRQ_FORCINT	0x0000020C
CMP_IRQ_MODE	0x00000210
CMP_EIRQ_EN	0x00000214
ARU_ACCESS	0x00000280
ARU_DATA_H	0x00000284
ARU_DATA_L	0x00000288
ARU_DBG_ACCESS0	0x0000028C
ARU_DBG_DATA0_H	0x00000290
ARU_DBG_DATA0_L	0x00000294
ARU_DBG_ACCESS1	0x00000298
ARU_DBG_DATA1_H	0x0000029C
ARU_DBG_DATA1_L	0x000002A0
ARU_IRQ_NOTIFY	0x000002A4
ARU_IRQ_EN	0x000002A8
ARU_IRQ_FORCINT	0x000002AC
ARU_IRQ_MODE	0x000002B0
CMU_CLK_EN	0x00000300
CMU_GCLK_NUM	0x00000304
CMU_GCLK_DEN	0x00000308
CMU_CLK_0_CTRL	0x0000030C
CMU_CLK_1_CTRL	0x00000310
CMU_CLK_2_CTRL	0x00000314
CMU_CLK_3_CTRL	0x00000318
CMU_CLK_4_CTRL	0x0000031C
CMU_CLK_5_CTRL	0x00000320
CMU_CLK_6_CTRL	0x00000324
CMU_CLK_7_CTRL	0x00000328
CMU_ECLK_0_NUM	0x0000032C
CMU_ECLK_0_DEN	0x00000330
CMU_ECLK_1_NUM	0x00000334
CMU_ECLK_1_DEN	0x00000338
CMU_ECLK_2_NUM	0x0000033C
CMU_ECLK_2_DEN	0x00000340
CMU_FXCLK_CTRL	0x00000344
BRC_SRC_0_ADDR	0x00000400
BRC_SRC_0_DEST	0x00000404
BRC_SRC_1_ADDR	0x00000408
BRC_SRC_1_DEST	0x0000040C
BRC_SRC_2_ADDR	0x00000410
BRC_SRC_2_DEST	0x00000414

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

BRC_SRC_3_ADDR	0x00000418
BRC_SRC_3_DEST	0x0000041C
BRC_SRC_4_ADDR	0x00000420
BRC_SRC_4_DEST	0x00000424
BRC_SRC_5_ADDR	0x00000428
BRC_SRC_5_DEST	0x0000042C
BRC_SRC_6_ADDR	0x00000430
BRC_SRC_6_DEST	0x00000434
BRC_SRC_7_ADDR	0x00000438
BRC_SRC_7_DEST	0x0000043C
BRC_SRC_8_ADDR	0x00000440
BRC_SRC_8_DEST	0x00000444
BRC_SRC_9_ADDR	0x00000448
BRC_SRC_9_DEST	0x0000044C
BRC_SRC_10_ADDR	0x00000450
BRC_SRC_10_DEST	0x00000454
BRC_SRC_11_ADDR	0x00000458
BRC_SRC_11_DEST	0x0000045C
BRC_IRQ_NOTIFY	0x00000460
BRC_IRQ_EN	0x00000464
BRC_IRQ_FORCINT	0x00000468
BRC_IRQ_MODE	0x0000046C
BRC_RST	0x00000470
BRC_EIRQ_EN	0x00000474
ICM_IRQG_0	0x00000600
ICM_IRQG_1	0x00000604
ICM_IRQG_2	0x00000608
ICM_IRQG_3	0x0000060C
ICM_IRQG_4	0x00000610
ICM_IRQG_5	0x00000614
ICM_IRQG_6	0x00000618
ICM_IRQG_7	0x0000061C
ICM_IRQG_8	0x00000620
ICM_IRQG_9	0x00000624
ICM_IRQG_10	0x00000628
ICM_IRQG_11	0x0000062C
ICM_IRQG_MEI	0x00000630
ICM_IRQG_CEI0	0x00000634
ICM_IRQG_CEI1	0x00000638
ICM_IRQG_CEI2	0x0000063C

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ICM_IRQG_CEI3	0x00000640
ICM_IRQG_CEI4	0x00000644
SPE0_CTRL_STAT	0x00000800
SPE0_PAT	0x00000804
SPE0_OUT_PAT0	0x00000808
SPE0_OUT_PAT1	0x0000080C
SPE0_OUT_PAT2	0x00000810
SPE0_OUT_PAT3	0x00000814
SPE0_OUT_PAT4	0x00000818
SPE0_OUT_PAT5	0x0000081C
SPE0_OUT_PAT6	0x00000820
SPE0_OUT_PAT7	0x00000824
SPE0_OUT_CTRL	0x00000828
SPE0_IRQ_NOTIFY	0x0000082C
SPE0_IRQ_EN	0x00000830
SPE0_IRQ_FORCINT	0x00000834
SPE0_IRQ_MODE	0x00000838
SPE0_EIRQ_EN	0x0000083C
SPE0_REV_CNT	0x00000840
SPE0_REV_CMP	0x00000844
SPE1_CTRL_STAT	0x00000880
SPE1_PAT	0x00000884
SPE1_OUT_PAT0	0x00000888
SPE1_OUT_PAT1	0x0000088C
SPE1_OUT_PAT2	0x00000890
SPE1_OUT_PAT3	0x00000894
SPE1_OUT_PAT4	0x00000898
SPE1_OUT_PAT5	0x0000089C
SPE1_OUT_PAT6	0x000008A0
SPE1_OUT_PAT7	0x000008A4
SPE1_OUT_CTRL	0x000008A8
SPE1_IRQ_NOTIFY	0x000008AC
SPE1_IRQ_EN	0x000008B0
SPE1_IRQ_FORCINT	0x000008B4
SPE1_IRQ_MODE	0x000008B8
SPE1_EIRQ_EN	0x000008BC
SPE1_REV_CNT	0x000008C0
SPE1_REV_CMP	0x000008C4
SPE2_CTRL_STAT	0x00000900
SPE2_PAT	0x00000904

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

SPE2_OUT_PAT0	0x00000908
SPE2_OUT_PAT1	0x0000090C
SPE2_OUT_PAT2	0x00000910
SPE2_OUT_PAT3	0x00000914
SPE2_OUT_PAT4	0x00000918
SPE2_OUT_PAT5	0x0000091C
SPE2_OUT_PAT6	0x00000920
SPE2_OUT_PAT7	0x00000924
SPE2_OUT_CTRL	0x00000928
SPE2_IRQ_NOTIFY	0x0000092C
SPE2_IRQ_EN	0x00000930
SPE2_IRQ_FORCINT	0x00000934
SPE2_IRQ_MODE	0x00000938
SPE2_EIRQ_EN	0x0000093C
SPE2_REV_CNT	0x00000940
SPE2_REV_CMP	0x00000944
SPE3_CTRL_STAT	0x00000980
SPE3_PAT	0x00000984
SPE3_OUT_PAT0	0x00000988
SPE3_OUT_PAT1	0x0000098C
SPE3_OUT_PAT2	0x00000990
SPE3_OUT_PAT3	0x00000994
SPE3_OUT_PAT4	0x00000998
SPE3_OUT_PAT5	0x0000099C
SPE3_OUT_PAT6	0x000009A0
SPE3_OUT_PAT7	0x000009A4
SPE3_OUT_CTRL	0x000009A8
SPE3_IRQ_NOTIFY	0x000009AC
SPE3_IRQ_EN	0x000009B0
SPE3_IRQ_FORCINT	0x000009B4
SPE3_IRQ_MODE	0x000009B8
SPE3_EIRQ_EN	0x000009BC
SPE3_REV_CNT	0x000009C0
SPE3_REV_CMP	0x000009C4
TIM0_CH0_GPR0	0x00001000
TIM0_CH0_GPR1	0x00001004
TIM0_CH0_CNT	0x00001008
TIM0_CH0_ECNT	0x0000100C
TIM0_CH0_CNTS	0x00001010
TIM0_CH0_TDUC	0x00001014

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM0_CH0_TDUV	0x00001018
TIM0_CH0_FLT_RE	0x0000101C
TIM0_CH0_FLT_FE	0x00001020
TIM0_CH0_CTRL	0x00001024
TIM0_CH0_ECTRL	0x00001028
TIM0_CH0_IRQ_NOTIFY	0x0000102C
TIM0_CH0_IRQ_EN	0x00001030
TIM0_CH0_IRQ_FORCINT	0x00001034
TIM0_CH0_IRQ_MODE	0x00001038
TIM0_CH0_EIRQ_EN	0x0000103C
TIM0_IN_SRC	0x00001078
TIM0_RST	0x0000107C
TIM0_CH1_GPR0	0x00001080
TIM0_CH1_GPR1	0x00001084
TIM0_CH1_CNT	0x00001088
TIM0_CH1_ECNT	0x0000108C
TIM0_CH1_CNTS	0x00001090
TIM0_CH1_TDUC	0x00001094
TIM0_CH1_TDUV	0x00001098
TIM0_CH1_FLT_RE	0x0000109C
TIM0_CH1_FLT_FE	0x000010A0
TIM0_CH1_CTRL	0x000010A4
TIM0_CH1_ECTRL	0x000010A8
TIM0_CH1_IRQ_NOTIFY	0x000010AC
TIM0_CH1_IRQ_EN	0x000010B0
TIM0_CH1_IRQ_FORCINT	0x000010B4
TIM0_CH1_IRQ_MODE	0x000010B8
TIM0_CH1_EIRQ_EN	0x000010BC
TIM0_CH2_GPR0	0x00001100
TIM0_CH2_GPR1	0x00001104
TIM0_CH2_CNT	0x00001108
TIM0_CH2_ECNT	0x0000110C
TIM0_CH2_CNTS	0x00001110
TIM0_CH2_TDUC	0x00001114
TIM0_CH2_TDUV	0x00001118
TIM0_CH2_FLT_RE	0x0000111C
TIM0_CH2_FLT_FE	0x00001120
TIM0_CH2_CTRL	0x00001124
TIM0_CH2_ECTRL	0x00001128
TIM0_CH2_IRQ_NOTIFY	0x0000112C

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM0_CH2_IRQ_EN	0x00001130
TIM0_CH2_IRQ_FORCINT	0x00001134
TIM0_CH2_IRQ_MODE	0x00001138
TIM0_CH2_EIRQ_EN	0x0000113C
TIM0_CH3_GPR0	0x00001180
TIM0_CH3_GPR1	0x00001184
TIM0_CH3_CNT	0x00001188
TIM0_CH3_ECNT	0x0000118C
TIM0_CH3_CNTS	0x00001190
TIM0_CH3_TDUC	0x00001194
TIM0_CH3_TDUV	0x00001198
TIM0_CH3_FLT_RE	0x0000119C
TIM0_CH3_FLT_FE	0x000011A0
TIM0_CH3_CTRL	0x000011A4
TIM0_CH3_ECTRL	0x000011A8
TIM0_CH3_IRQ_NOTIFY	0x000011AC
TIM0_CH3_IRQ_EN	0x000011B0
TIM0_CH3_IRQ_FORCINT	0x000011B4
TIM0_CH3_IRQ_MODE	0x000011B8
TIM0_CH3_EIRQ_EN	0x000011BC
TIM0_CH4_GPR0	0x00001200
TIM0_CH4_GPR1	0x00001204
TIM0_CH4_CNT	0x00001208
TIM0_CH4_ECNT	0x0000120C
TIM0_CH4_CNTS	0x00001210
TIM0_CH4_TDUC	0x00001214
TIM0_CH4_TDUV	0x00001218
TIM0_CH4_FLT_RE	0x0000121C
TIM0_CH4_FLT_FE	0x00001220
TIM0_CH4_CTRL	0x00001224
TIM0_CH4_ECTRL	0x00001228
TIM0_CH4_IRQ_NOTIFY	0x0000122C
TIM0_CH4_IRQ_EN	0x00001230
TIM0_CH4_IRQ_FORCINT	0x00001234
TIM0_CH4_IRQ_MODE	0x00001238
TIM0_CH4_EIRQ_EN	0x0000123C
TIM0_CH5_GPR0	0x00001280
TIM0_CH5_GPR1	0x00001284
TIM0_CH5_CNT	0x00001288
TIM0_CH5_ECNT	0x0000128C

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM0_CH5_CNTS	0x00001290
TIM0_CH5_TDUC	0x00001294
TIM0_CH5_TDUV	0x00001298
TIM0_CH5_FLT_RE	0x0000129C
TIM0_CH5_FLT_FE	0x000012A0
TIM0_CH5_CTRL	0x000012A4
TIM0_CH5_ECTRL	0x000012A8
TIM0_CH5_IRQ_NOTIFY	0x000012AC
TIM0_CH5_IRQ_EN	0x000012B0
TIM0_CH5_IRQ_FORCINT	0x000012B4
TIM0_CH5_IRQ_MODE	0x000012B8
TIM0_CH5_EIRQ_EN	0x000012BC
TIM0_CH6_GPR0	0x00001300
TIM0_CH6_GPR1	0x00001304
TIM0_CH6_CNT	0x00001308
TIM0_CH6_ECNT	0x0000130C
TIM0_CH6_CNTS	0x00001310
TIM0_CH6_TDUC	0x00001314
TIM0_CH6_TDUV	0x00001318
TIM0_CH6_FLT_RE	0x0000131C
TIM0_CH6_FLT_FE	0x00001320
TIM0_CH6_CTRL	0x00001324
TIM0_CH6_ECTRL	0x00001328
TIM0_CH6_IRQ_NOTIFY	0x0000132C
TIM0_CH6_IRQ_EN	0x00001330
TIM0_CH6_IRQ_FORCINT	0x00001334
TIM0_CH6_IRQ_MODE	0x00001338
TIM0_CH6_EIRQ_EN	0x0000133C
TIM0_CH7_GPR0	0x00001380
TIM0_CH7_GPR1	0x00001384
TIM0_CH7_CNT	0x00001388
TIM0_CH7_ECNT	0x0000138C
TIM0_CH7_CNTS	0x00001390
TIM0_CH7_TDUC	0x00001394
TIM0_CH7_TDUV	0x00001398
TIM0_CH7_FLT_RE	0x0000139C
TIM0_CH7_FLT_FE	0x000013A0
TIM0_CH7_CTRL	0x000013A4
TIM0_CH7_ECTRL	0x000013A8
TIM0_CH7_IRQ_NOTIFY	0x000013AC

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM0_CH7_IRQ_EN	0x000013B0
TIM0_CH7_IRQ_FORCINT	0x000013B4
TIM0_CH7_IRQ_MODE	0x000013B8
TIM0_CH7_EIRQ_EN	0x000013BC
TIM1_CH0_GPR0	0x00001800
TIM1_CH0_GPR1	0x00001804
TIM1_CH0_CNT	0x00001808
TIM1_CH0_ECNT	0x0000180C
TIM1_CH0_CNTS	0x00001810
TIM1_CH0_TDUC	0x00001814
TIM1_CH0_TDUV	0x00001818
TIM1_CH0_FLT_RE	0x0000181C
TIM1_CH0_FLT_FE	0x00001820
TIM1_CH0_CTRL	0x00001824
TIM1_CH0_ECTRL	0x00001828
TIM1_CH0_IRQ_NOTIFY	0x0000182C
TIM1_CH0_IRQ_EN	0x00001830
TIM1_CH0_IRQ_FORCINT	0x00001834
TIM1_CH0_IRQ_MODE	0x00001838
TIM1_CH0_EIRQ_EN	0x0000183C
TIM1_IN_SRC	0x00001878
TIM1_RST	0x0000187C
TIM1_CH1_GPR0	0x00001880
TIM1_CH1_GPR1	0x00001884
TIM1_CH1_CNT	0x00001888
TIM1_CH1_ECNT	0x0000188C
TIM1_CH1_CNTS	0x00001890
TIM1_CH1_TDUC	0x00001894
TIM1_CH1_TDUV	0x00001898
TIM1_CH1_FLT_RE	0x0000189C
TIM1_CH1_FLT_FE	0x000018A0
TIM1_CH1_CTRL	0x000018A4
TIM1_CH1_ECTRL	0x000018A8
TIM1_CH1_IRQ_NOTIFY	0x000018AC
TIM1_CH1_IRQ_EN	0x000018B0
TIM1_CH1_IRQ_FORCINT	0x000018B4
TIM1_CH1_IRQ_MODE	0x000018B8
TIM1_CH1_EIRQ_EN	0x000018BC
TIM1_CH2_GPR0	0x00001900
TIM1_CH2_GPR1	0x00001904

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM1_CH2_CNT	0x00001908
TIM1_CH2_ECNT	0x0000190C
TIM1_CH2_CNTS	0x00001910
TIM1_CH2_TDUC	0x00001914
TIM1_CH2_TDUV	0x00001918
TIM1_CH2_FLT_RE	0x0000191C
TIM1_CH2_FLT_FE	0x00001920
TIM1_CH2_CTRL	0x00001924
TIM1_CH2_ECTRL	0x00001928
TIM1_CH2_IRQ_NOTIFY	0x0000192C
TIM1_CH2_IRQ_EN	0x00001930
TIM1_CH2_IRQ_FORCINT	0x00001934
TIM1_CH2_IRQ_MODE	0x00001938
TIM1_CH2_EIRQ_EN	0x0000193C
TIM1_CH3_GPR0	0x00001980
TIM1_CH3_GPR1	0x00001984
TIM1_CH3_CNT	0x00001988
TIM1_CH3_ECNT	0x0000198C
TIM1_CH3_CNTS	0x00001990
TIM1_CH3_TDUC	0x00001994
TIM1_CH3_TDUV	0x00001998
TIM1_CH3_FLT_RE	0x0000199C
TIM1_CH3_FLT_FE	0x000019A0
TIM1_CH3_CTRL	0x000019A4
TIM1_CH3_ECTRL	0x000019A8
TIM1_CH3_IRQ_NOTIFY	0x000019AC
TIM1_CH3_IRQ_EN	0x000019B0
TIM1_CH3_IRQ_FORCINT	0x000019B4
TIM1_CH3_IRQ_MODE	0x000019B8
TIM1_CH3_EIRQ_EN	0x000019BC
TIM1_CH4_GPR0	0x00001A00
TIM1_CH4_GPR1	0x00001A04
TIM1_CH4_CNT	0x00001A08
TIM1_CH4_ECNT	0x00001A0C
TIM1_CH4_CNTS	0x00001A10
TIM1_CH4_TDUC	0x00001A14
TIM1_CH4_TDUV	0x00001A18
TIM1_CH4_FLT_RE	0x00001A1C
TIM1_CH4_FLT_FE	0x00001A20
TIM1_CH4_CTRL	0x00001A24

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM1_CH4_ECTRL	0x00001A28
TIM1_CH4_IRQ_NOTIFY	0x00001A2C
TIM1_CH4_IRQ_EN	0x00001A30
TIM1_CH4_IRQ_FORCINT	0x00001A34
TIM1_CH4_IRQ_MODE	0x00001A38
TIM1_CH4_EIRQ_EN	0x00001A3C
TIM1_CH5_GPR0	0x00001A80
TIM1_CH5_GPR1	0x00001A84
TIM1_CH5_CNT	0x00001A88
TIM1_CH5_ECNT	0x00001A8C
TIM1_CH5_CNTS	0x00001A90
TIM1_CH5_TDUC	0x00001A94
TIM1_CH5_TDUV	0x00001A98
TIM1_CH5_FLT_RE	0x00001A9C
TIM1_CH5_FLT_FE	0x00001AA0
TIM1_CH5_CTRL	0x00001AA4
TIM1_CH5_ECTRL	0x00001AA8
TIM1_CH5_IRQ_NOTIFY	0x00001AAC
TIM1_CH5_IRQ_EN	0x00001AB0
TIM1_CH5_IRQ_FORCINT	0x00001AB4
TIM1_CH5_IRQ_MODE	0x00001AB8
TIM1_CH5_EIRQ_EN	0x00001ABC
TIM1_CH6_GPR0	0x00001B00
TIM1_CH6_GPR1	0x00001B04
TIM1_CH6_CNT	0x00001B08
TIM1_CH6_ECNT	0x00001B0C
TIM1_CH6_CNTS	0x00001B10
TIM1_CH6_TDUC	0x00001B14
TIM1_CH6_TDUV	0x00001B18
TIM1_CH6_FLT_RE	0x00001B1C
TIM1_CH6_FLT_FE	0x00001B20
TIM1_CH6_CTRL	0x00001B24
TIM1_CH6_ECTRL	0x00001B28
TIM1_CH6_IRQ_NOTIFY	0x00001B2C
TIM1_CH6_IRQ_EN	0x00001B30
TIM1_CH6_IRQ_FORCINT	0x00001B34
TIM1_CH6_IRQ_MODE	0x00001B38
TIM1_CH6_EIRQ_EN	0x00001B3C
TIM1_CH7_GPR0	0x00001B80
TIM1_CH7_GPR1	0x00001B84

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM1_CH7_CNT	0x00001B88
TIM1_CH7_ECNT	0x00001B8C
TIM1_CH7_CNTS	0x00001B90
TIM1_CH7_TDUC	0x00001B94
TIM1_CH7_TDUV	0x00001B98
TIM1_CH7_FLT_RE	0x00001B9C
TIM1_CH7_FLT_FE	0x00001BA0
TIM1_CH7_CTRL	0x00001BA4
TIM1_CH7_ECTRL	0x00001BA8
TIM1_CH7_IRQ_NOTIFY	0x00001BAC
TIM1_CH7_IRQ_EN	0x00001BB0
TIM1_CH7_IRQ_FORCINT	0x00001BB4
TIM1_CH7_IRQ_MODE	0x00001BB8
TIM1_CH7_EIRQ_EN	0x00001BBC
TIM2_CH0_GPR0	0x00002000
TIM2_CH0_GPR1	0x00002004
TIM2_CH0_CNT	0x00002008
TIM2_CH0_ECNT	0x0000200C
TIM2_CH0_CNTS	0x00002010
TIM2_CH0_TDUC	0x00002014
TIM2_CH0_TDUV	0x00002018
TIM2_CH0_FLT_RE	0x0000201C
TIM2_CH0_FLT_FE	0x00002020
TIM2_CH0_CTRL	0x00002024
TIM2_CH0_ECTRL	0x00002028
TIM2_CH0_IRQ_NOTIFY	0x0000202C
TIM2_CH0_IRQ_EN	0x00002030
TIM2_CH0_IRQ_FORCINT	0x00002034
TIM2_CH0_IRQ_MODE	0x00002038
TIM2_CH0_EIRQ_EN	0x0000203C
TIM2_IN_SRC	0x00002078
TIM2_RST	0x0000207C
TIM2_CH1_GPR0	0x00002080
TIM2_CH1_GPR1	0x00002084
TIM2_CH1_CNT	0x00002088
TIM2_CH1_ECNT	0x0000208C
TIM2_CH1_CNTS	0x00002090
TIM2_CH1_TDUC	0x00002094
TIM2_CH1_TDUV	0x00002098
TIM2_CH1_FLT_RE	0x0000209C

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM2_CH1_FLT_FE	0x000020A0
TIM2_CH1_CTRL	0x000020A4
TIM2_CH1_ECTRL	0x000020A8
TIM2_CH1_IRQ_NOTIFY	0x000020AC
TIM2_CH1_IRQ_EN	0x000020B0
TIM2_CH1_IRQ_FORCINT	0x000020B4
TIM2_CH1_IRQ_MODE	0x000020B8
TIM2_CH1_EIRQ_EN	0x000020BC
TIM2_CH2_GPR0	0x00002100
TIM2_CH2_GPR1	0x00002104
TIM2_CH2_CNT	0x00002108
TIM2_CH2_ECNT	0x0000210C
TIM2_CH2_CNTS	0x00002110
TIM2_CH2_TDUC	0x00002114
TIM2_CH2_TDUV	0x00002118
TIM2_CH2_FLT_RE	0x0000211C
TIM2_CH2_FLT_FE	0x00002120
TIM2_CH2_CTRL	0x00002124
TIM2_CH2_ECTRL	0x00002128
TIM2_CH2_IRQ_NOTIFY	0x0000212C
TIM2_CH2_IRQ_EN	0x00002130
TIM2_CH2_IRQ_FORCINT	0x00002134
TIM2_CH2_IRQ_MODE	0x00002138
TIM2_CH2_EIRQ_EN	0x0000213C
TIM2_CH3_GPR0	0x00002180
TIM2_CH3_GPR1	0x00002184
TIM2_CH3_CNT	0x00002188
TIM2_CH3_ECNT	0x0000218C
TIM2_CH3_CNTS	0x00002190
TIM2_CH3_TDUC	0x00002194
TIM2_CH3_TDUV	0x00002198
TIM2_CH3_FLT_RE	0x0000219C
TIM2_CH3_FLT_FE	0x000021A0
TIM2_CH3_CTRL	0x000021A4
TIM2_CH3_ECTRL	0x000021A8
TIM2_CH3_IRQ_NOTIFY	0x000021AC
TIM2_CH3_IRQ_EN	0x000021B0
TIM2_CH3_IRQ_FORCINT	0x000021B4
TIM2_CH3_IRQ_MODE	0x000021B8
TIM2_CH3_EIRQ_EN	0x000021BC

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM2_CH4_GPR0	0x00002200
TIM2_CH4_GPR1	0x00002204
TIM2_CH4_CNT	0x00002208
TIM2_CH4_ECNT	0x0000220C
TIM2_CH4_CNTS	0x00002210
TIM2_CH4_TDUC	0x00002214
TIM2_CH4_TDUV	0x00002218
TIM2_CH4_FLT_RE	0x0000221C
TIM2_CH4_FLT_FE	0x00002220
TIM2_CH4_CTRL	0x00002224
TIM2_CH4_ECTRL	0x00002228
TIM2_CH4_IRQ_NOTIFY	0x0000222C
TIM2_CH4_IRQ_EN	0x00002230
TIM2_CH4_IRQ_FORCINT	0x00002234
TIM2_CH4_IRQ_MODE	0x00002238
TIM2_CH4_EIRQ_EN	0x0000223C
TIM2_CH5_GPR0	0x00002280
TIM2_CH5_GPR1	0x00002284
TIM2_CH5_CNT	0x00002288
TIM2_CH5_ECNT	0x0000228C
TIM2_CH5_CNTS	0x00002290
TIM2_CH5_TDUC	0x00002294
TIM2_CH5_TDUV	0x00002298
TIM2_CH5_FLT_RE	0x0000229C
TIM2_CH5_FLT_FE	0x000022A0
TIM2_CH5_CTRL	0x000022A4
TIM2_CH5_ECTRL	0x000022A8
TIM2_CH5_IRQ_NOTIFY	0x000022AC
TIM2_CH5_IRQ_EN	0x000022B0
TIM2_CH5_IRQ_FORCINT	0x000022B4
TIM2_CH5_IRQ_MODE	0x000022B8
TIM2_CH5_EIRQ_EN	0x000022BC
TIM2_CH6_GPR0	0x00002300
TIM2_CH6_GPR1	0x00002304
TIM2_CH6_CNT	0x00002308
TIM2_CH6_ECNT	0x0000230C
TIM2_CH6_CNTS	0x00002310
TIM2_CH6_TDUC	0x00002314
TIM2_CH6_TDUV	0x00002318
TIM2_CH6_FLT_RE	0x0000231C

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM2_CH6_FLT_FE	0x00002320
TIM2_CH6_CTRL	0x00002324
TIM2_CH6_ECTRL	0x00002328
TIM2_CH6_IRQ_NOTIFY	0x0000232C
TIM2_CH6_IRQ_EN	0x00002330
TIM2_CH6_IRQ_FORCINT	0x00002334
TIM2_CH6_IRQ_MODE	0x00002338
TIM2_CH6_EIRQ_EN	0x0000233C
TIM2_CH7_GPR0	0x00002380
TIM2_CH7_GPR1	0x00002384
TIM2_CH7_CNT	0x00002388
TIM2_CH7_ECNT	0x0000238C
TIM2_CH7_CNTS	0x00002390
TIM2_CH7_TDUC	0x00002394
TIM2_CH7_TDUV	0x00002398
TIM2_CH7_FLT_RE	0x0000239C
TIM2_CH7_FLT_FE	0x000023A0
TIM2_CH7_CTRL	0x000023A4
TIM2_CH7_ECTRL	0x000023A8
TIM2_CH7_IRQ_NOTIFY	0x000023AC
TIM2_CH7_IRQ_EN	0x000023B0
TIM2_CH7_IRQ_FORCINT	0x000023B4
TIM2_CH7_IRQ_MODE	0x000023B8
TIM2_CH7_EIRQ_EN	0x000023BC
TIM3_CH0_GPR0	0x00002800
TIM3_CH0_GPR1	0x00002804
TIM3_CH0_CNT	0x00002808
TIM3_CH0_ECNT	0x0000280C
TIM3_CH0_CNTS	0x00002810
TIM3_CH0_TDUC	0x00002814
TIM3_CH0_TDUV	0x00002818
TIM3_CH0_FLT_RE	0x0000281C
TIM3_CH0_FLT_FE	0x00002820
TIM3_CH0_CTRL	0x00002824
TIM3_CH0_ECTRL	0x00002828
TIM3_CH0_IRQ_NOTIFY	0x0000282C
TIM3_CH0_IRQ_EN	0x00002830
TIM3_CH0_IRQ_FORCINT	0x00002834
TIM3_CH0_IRQ_MODE	0x00002838
TIM3_CH0_EIRQ_EN	0x0000283C

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM3_IN_SRC	0x00002878
TIM3_RST	0x0000287C
TIM3_CH1_GPR0	0x00002880
TIM3_CH1_GPR1	0x00002884
TIM3_CH1_CNT	0x00002888
TIM3_CH1_ECNT	0x0000288C
TIM3_CH1_CNTS	0x00002890
TIM3_CH1_TDUC	0x00002894
TIM3_CH1_TDUV	0x00002898
TIM3_CH1_FLT_RE	0x0000289C
TIM3_CH1_FLT_FE	0x000028A0
TIM3_CH1_CTRL	0x000028A4
TIM3_CH1_ECTRL	0x000028A8
TIM3_CH1_IRQ_NOTIFY	0x000028AC
TIM3_CH1_IRQ_EN	0x000028B0
TIM3_CH1_IRQ_FORCINT	0x000028B4
TIM3_CH1_IRQ_MODE	0x000028B8
TIM3_CH1_EIRQ_EN	0x000028BC
TIM3_CH2_GPR0	0x00002900
TIM3_CH2_GPR1	0x00002904
TIM3_CH2_CNT	0x00002908
TIM3_CH2_ECNT	0x0000290C
TIM3_CH2_CNTS	0x00002910
TIM3_CH2_TDUV	0x00002918
TIM3_CH2_TDU	0x00002910
TIM3_CH2_FLT_RE	0x0000291C
TIM3_CH2_FLT_FE	0x00002920
TIM3_CH2_CTRL	0x00002924
TIM3_CH2_ECTRL	0x00002928
TIM3_CH2_IRQ_NOTIFY	0x0000292C
TIM3_CH2_IRQ_EN	0x00002930
TIM3_CH2_IRQ_FORCINT	0x00002934
TIM3_CH2_IRQ_MODE	0x00002938
TIM3_CH2_EIRQ_EN	0x0000293C
TIM3_CH3_GPR0	0x00002980
TIM3_CH3_GPR1	0x00002984
TIM3_CH3_CNT	0x00002988
TIM3_CH3_ECNT	0x0000298C
TIM3_CH3_CNTS	0x00002990
TIM3_CH3_TDUC	0x00002994

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM3_CH3_TDUV	0x00002998
TIM3_CH3_FLT_RE	0x0000299C
TIM3_CH3_FLT_FE	0x000029A0
TIM3_CH3_CTRL	0x000029A4
TIM3_CH3_ECTRL	0x000029A8
TIM3_CH3_IRQ_NOTIFY	0x000029AC
TIM3_CH3_IRQ_EN	0x000029B0
TIM3_CH3_IRQ_FORCINT	0x000029B4
TIM3_CH3_IRQ_MODE	0x000029B8
TIM3_CH3_EIRQ_EN	0x000029BC
TIM3_CH4_GPR0	0x00002A00
TIM3_CH4_GPR1	0x00002A04
TIM3_CH4_CNT	0x00002A08
TIM3_CH4_ECNT	0x00002A0C
TIM3_CH4_CNTS	0x00002A10
TIM3_CH4_TDUC	0x00002A14
TIM3_CH4_TDUV	0x00002A18
TIM3_CH4_FLT_RE	0x00002A1C
TIM3_CH4_FLT_FE	0x00002A20
TIM3_CH4_CTRL	0x00002A24
TIM3_CH4_ECTRL	0x00002A28
TIM3_CH4_IRQ_NOTIFY	0x00002A2C
TIM3_CH4_IRQ_EN	0x00002A30
TIM3_CH4_IRQ_FORCINT	0x00002A34
TIM3_CH4_IRQ_MODE	0x00002A38
TIM3_CH4_EIRQ_EN	0x00002A3C
TIM3_CH5_GPR0	0x00002A80
TIM3_CH5_GPR1	0x00002A84
TIM3_CH5_CNT	0x00002A88
TIM3_CH5_ECNT	0x00002A8C
TIM3_CH5_CNTS	0x00002A90
TIM3_CH5_TDUC	0x00002A94
TIM3_CH5_TDUV	0x00002A98
TIM3_CH5_FLT_RE	0x00002A9C
TIM3_CH5_FLT_FE	0x00002AA0
TIM3_CH5_CTRL	0x00002AA4
TIM3_CH5_ECTRL	0x00002AA8
TIM3_CH5_IRQ_NOTIFY	0x00002AAC
TIM3_CH5_IRQ_EN	0x00002AB0
TIM3_CH5_IRQ_FORCINT	0x00002AB4

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM3_CH5_IRQ_MODE	0x00002AB8
TIM3_CH5_EIRQ_EN	0x00002ABC
TIM3_CH6_GPR0	0x00002B00
TIM3_CH6_GPR1	0x00002B04
TIM3_CH6_CNT	0x00002B08
TIM3_CH6_ECNT	0x00002B0C
TIM3_CH6_CNTS	0x00002B10
TIM3_CH6_TDUC	0x00002B14
TIM3_CH6_TDUV	0x00002B18
TIM3_CH6_FLT_RE	0x00002B1C
TIM3_CH6_FLT_FE	0x00002B20
TIM3_CH6_CTRL	0x00002B24
TIM3_CH6_ECTRL	0x00002B28
TIM3_CH6_IRQ_NOTIFY	0x00002B2C
TIM3_CH6_IRQ_EN	0x00002B30
TIM3_CH6_IRQ_FORCINT	0x00002B34
TIM3_CH6_IRQ_MODE	0x00002B38
TIM3_CH6_EIRQ_EN	0x00002B3C
TIM3_CH7_GPR0	0x00002B80
TIM3_CH7_GPR1	0x00002B84
TIM3_CH7_CNT	0x00002B88
TIM3_CH7_ECNT	0x00002B8C
TIM3_CH7_CNTS	0x00002B90
TIM3_CH7_TDUC	0x00002B94
TIM3_CH7_TDUV	0x00002B98
TIM3_CH7_FLT_RE	0x00002B9C
TIM3_CH7_FLT_FE	0x00002BA0
TIM3_CH7_CTRL	0x00002BA4
TIM3_CH7_ECTRL	0x00002BA8
TIM3_CH7_IRQ_NOTIFY	0x00002BAC
TIM3_CH7_IRQ_EN	0x00002BB0
TIM3_CH7_IRQ_FORCINT	0x00002BB4
TIM3_CH7_IRQ_MODE	0x00002BB8
TIM3_CH7_EIRQ_EN	0x00002BBC
TIM4_CH0_GPR0	0x00003000
TIM4_CH0_GPR1	0x00003004
TIM4_CH0_CNT	0x00003008
TIM4_CH0_ECNT	0x0000300C
TIM4_CH0_CNTS	0x00003010
TIM4_CH0_TDUC	0x00003014

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM4_CH0_TDUV	0x00003018
TIM4_CH0_FLT_RE	0x0000301C
TIM4_CH0_FLT_FE	0x00003020
TIM4_CH0_CTRL	0x00003024
TIM4_CH0_ECTRL	0x00003028
TIM4_CH0_IRQ_NOTIFY	0x0000302C
TIM4_CH0_IRQ_EN	0x00003030
TIM4_CH0_IRQ_FORCINT	0x00003034
TIM4_CH0_IRQ_MODE	0x00003038
TIM4_CH0_EIRQ_EN	0x0000303C
TIM4_IN_SRC	0x00003078
TIM4_RST	0x0000307C
TIM4_CH1_GPR0	0x00003080
TIM4_CH1_GPR1	0x00003084
TIM4_CH1_CNT	0x00003088
TIM4_CH1_ECNT	0x0000308C
TIM4_CH1_CNTS	0x00003090
TIM4_CH1_TDUC	0x00003094
TIM4_CH1_TDUV	0x00003098
TIM4_CH1_FLT_RE	0x0000309C
TIM4_CH1_FLT_FE	0x000030A0
TIM4_CH1_CTRL	0x000030A4
TIM4_CH1_ECTRL	0x000030A8
TIM4_CH1_IRQ_NOTIFY	0x000030AC
TIM4_CH1_IRQ_EN	0x000030B0
TIM4_CH1_IRQ_FORCINT	0x000030B4
TIM4_CH1_IRQ_MODE	0x000030B8
TIM4_CH1_EIRQ_EN	0x000030BC
TIM4_CH2_GPR0	0x00003100
TIM4_CH2_GPR1	0x00003104
TIM4_CH2_CNT	0x00003108
TIM4_CH2_ECNT	0x0000310C
TIM4_CH2_CNTS	0x00003110
TIM4_CH2_TDUC	0x00003114
TIM4_CH2_TDUV	0x00003118
TIM4_CH2_FLT_RE	0x0000311C
TIM4_CH2_FLT_FE	0x00003120
TIM4_CH2_CTRL	0x00003124
TIM4_CH2_ECTRL	0x00003128
TIM4_CH2_IRQ_NOTIFY	0x0000312C

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM4_CH2_IRQ_EN	0x00003130
TIM4_CH2_IRQ_FORCINT	0x00003134
TIM4_CH2_IRQ_MODE	0x00003138
TIM4_CH2_EIRQ_EN	0x0000313C
TIM4_CH3_GPR0	0x00003180
TIM4_CH3_GPR1	0x00003184
TIM4_CH3_CNT	0x00003188
TIM4_CH3_ECNT	0x0000318C
TIM4_CH3_CNTS	0x00003190
TIM4_CH3_TDUC	0x00003194
TIM4_CH3_TDUV	0x00003198
TIM4_CH3_FLT_RE	0x0000319C
TIM4_CH3_FLT_FE	0x000031A0
TIM4_CH3_CTRL	0x000031A4
TIM4_CH3_ECTRL	0x000031A8
TIM4_CH3_IRQ_NOTIFY	0x000031AC
TIM4_CH3_IRQ_EN	0x000031B0
TIM4_CH3_IRQ_FORCINT	0x000031B4
TIM4_CH3_IRQ_MODE	0x000031B8
TIM4_CH3_EIRQ_EN	0x000031BC
TIM4_CH4_GPR0	0x00003200
TIM4_CH4_GPR1	0x00003204
TIM4_CH4_CNT	0x00003208
TIM4_CH4_ECNT	0x0000320C
TIM4_CH4_CNTS	0x00003210
TIM4_CH4_TDUC	0x00003214
TIM4_CH4_TDUV	0x00003218
TIM4_CH4_FLT_RE	0x0000321C
TIM4_CH4_FLT_FE	0x00003220
TIM4_CH4_CTRL	0x00003224
TIM4_CH4_ECTRL	0x00003228
TIM4_CH4_IRQ_NOTIFY	0x0000322C
TIM4_CH4_IRQ_EN	0x00003230
TIM4_CH4_IRQ_FORCINT	0x00003234
TIM4_CH4_IRQ_MODE	0x00003238
TIM4_CH4_EIRQ_EN	0x0000323C
TIM4_CH5_GPR0	0x00003280
TIM4_CH5_GPR1	0x00003284
TIM4_CH5_CNT	0x00003288
TIM4_CH5_ECNT	0x0000328C

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM4_CH5_CNTS	0x00003290
TIM4_CH5_TDUC	0x00003294
TIM4_CH5_TDUV	0x00003298
TIM4_CH5_FLT_RE	0x0000329C
TIM4_CH5_FLT_FE	0x000032A0
TIM4_CH5_CTRL	0x000032A4
TIM4_CH5_ECTRL	0x000032A8
TIM4_CH5_IRQ_NOTIFY	0x000032AC
TIM4_CH5_IRQ_EN	0x000032B0
TIM4_CH5_IRQ_FORCINT	0x000032B4
TIM4_CH5_IRQ_MODE	0x000032B8
TIM4_CH5_EIRQ_EN	0x000032BC
TIM4_CH6_GPR0	0x00003300
TIM4_CH6_GPR1	0x00003304
TIM4_CH6_CNT	0x00003308
TIM4_CH6_ECNT	0x0000330C
TIM4_CH6_CNTS	0x00003310
TIM4_CH6_TDUC	0x00003314
TIM4_CH6_TDUV	0x00003318
TIM4_CH6_FLT_RE	0x0000331C
TIM4_CH6_FLT_FE	0x00003320
TIM4_CH6_CTRL	0x00003324
TIM4_CH6_ECTRL	0x00003328
TIM4_CH6_IRQ_NOTIFY	0x0000332C
TIM4_CH6_IRQ_EN	0x00003330
TIM4_CH6_IRQ_FORCINT	0x00003334
TIM4_CH6_IRQ_MODE	0x00003338
TIM4_CH6_EIRQ_EN	0x0000333C
TIM4_CH7_GPR0	0x00003380
TIM4_CH7_GPR1	0x00003384
TIM4_CH7_CNT	0x00003388
TIM4_CH7_ECNT	0x0000338C
TIM4_CH7_CNTS	0x00003390
TIM4_CH7_TDUC	0x00003394
TIM4_CH7_TDUV	0x00003398
TIM4_CH7_FLT_RE	0x0000339C
TIM4_CH7_FLT_FE	0x000033A0
TIM4_CH7_CTRL	0x000033A4
TIM4_CH7_ECTRL	0x000033A8
TIM4_CH7_IRQ_NOTIFY	0x000033AC

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM4_CH7_IRQ_EN	0x000033B0
TIM4_CH7_IRQ_FORCINT	0x000033B4
TIM4_CH7_IRQ_MODE	0x000033B8
TIM4_CH7_EIRQ_EN	0x000033BC
TIM5_CH0_GPR0	0x00003800
TIM5_CH0_GPR1	0x00003804
TIM5_CH0_CNT	0x00003808
TIM5_CH0_ECNT	0x0000380C
TIM5_CH0_CNTS	0x00003810
TIM5_CH0_TDUC	0x00003814
TIM5_CH0_TDUV	0x00003818
TIM5_CH0_FLT_RE	0x0000381C
TIM5_CH0_FLT_FE	0x00003820
TIM5_CH0_CTRL	0x00003824
TIM5_CH0_ECTRL	0x00003828
TIM5_CH0_IRQ_NOTIFY	0x0000382C
TIM5_CH0_IRQ_EN	0x00003830
TIM5_CH0_IRQ_FORCINT	0x00003834
TIM5_CH0_IRQ_MODE	0x00003838
TIM5_CH0_EIRQ_EN	0x0000383C
TIM5_IN_SRC	0x00003878
TIM5_RST	0x0000387C
TIM5_CH1_GPR0	0x00003880
TIM5_CH1_GPR1	0x00003884
TIM5_CH1_CNT	0x00003888
TIM5_CH1_ECNT	0x0000388C
TIM5_CH1_CNTS	0x00003890
TIM5_CH1_TDUC	0x00003894
TIM5_CH1_TDUV	0x00003898
TIM5_CH1_FLT_RE	0x0000389C
TIM5_CH1_FLT_FE	0x000038A0
TIM5_CH1_CTRL	0x000038A4
TIM5_CH1_ECTRL	0x000038A8
TIM5_CH1_IRQ_NOTIFY	0x000038AC
TIM5_CH1_IRQ_EN	0x000038B0
TIM5_CH1_IRQ_FORCINT	0x000038B4
TIM5_CH1_IRQ_MODE	0x000038B8
TIM5_CH1_EIRQ_EN	0x000038BC
TIM5_CH2_GPR0	0x00003900
TIM5_CH2_GPR1	0x00003904

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM5_CH2_CNT	0x00003908
TIM5_CH2_ECNT	0x0000390C
TIM5_CH2_CNTS	0x00003910
TIM5_CH2_TDUC	0x00003914
TIM5_CH2_TDUV	0x00003918
TIM5_CH2_FLT_RE	0x0000391C
TIM5_CH2_FLT_FE	0x00003920
TIM5_CH2_CTRL	0x00003924
TIM5_CH2_ECTRL	0x00003928
TIM5_CH2_IRQ_NOTIFY	0x0000392C
TIM5_CH2_IRQ_EN	0x00003930
TIM5_CH2_IRQ_FORCINT	0x00003934
TIM5_CH2_IRQ_MODE	0x00003938
TIM5_CH2_EIRQ_EN	0x0000393C
TIM5_CH3_GPR0	0x00003980
TIM5_CH3_GPR1	0x00003984
TIM5_CH3_CNT	0x00003988
TIM5_CH3_ECNT	0x0000398C
TIM5_CH3_CNTS	0x00003990
TIM5_CH3_TDUC	0x00003994
TIM5_CH3_TDUV	0x00003998
TIM5_CH3_FLT_RE	0x0000399C
TIM5_CH3_FLT_FE	0x000039A0
TIM5_CH3_CTRL	0x000039A4
TIM5_CH3_ECTRL	0x000039A8
TIM5_CH3_IRQ_NOTIFY	0x000039AC
TIM5_CH3_IRQ_EN	0x000039B0
TIM5_CH3_IRQ_FORCINT	0x000039B4
TIM5_CH3_IRQ_MODE	0x000039B8
TIM5_CH3_EIRQ_EN	0x000039BC
TIM5_CH4_GPR0	0x00003A00
TIM5_CH4_GPR1	0x00003A04
TIM5_CH4_CNT	0x00003A08
TIM5_CH4_ECNT	0x00003A0C
TIM5_CH4_CNTS	0x00003A10
TIM5_CH4_TDUC	0x00003A14
TIM5_CH4_TDUV	0x00003A18
TIM5_CH4_FLT_RE	0x00003A1C
TIM5_CH4_FLT_FE	0x00003A20
TIM5_CH4_CTRL	0x00003A24

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM5_CH4_ECTRL	0x00003A28
TIM5_CH4_IRQ_NOTIFY	0x00003A2C
TIM5_CH4_IRQ_EN	0x00003A30
TIM5_CH4_IRQ_FORCINT	0x00003A34
TIM5_CH4_IRQ_MODE	0x00003A38
TIM5_CH4_EIRQ_EN	0x00003A3C
TIM5_CH5_GPR0	0x00003A80
TIM5_CH5_GPR1	0x00003A84
TIM5_CH5_CNT	0x00003A88
TIM5_CH5_ECNT	0x00003A8C
TIM5_CH5_CNTS	0x00003A90
TIM5_CH5_TDUC	0x00003A94
TIM5_CH5_TDUV	0x00003A98
TIM5_CH5_FLT_RE	0x00003A9C
TIM5_CH5_FLT_FE	0x00003AA0
TIM5_CH5_CTRL	0x00003AA4
TIM5_CH5_ECTRL	0x00003AA8
TIM5_CH5_IRQ_NOTIFY	0x00003AAC
TIM5_CH5_IRQ_EN	0x00003AB0
TIM5_CH5_IRQ_FORCINT	0x00003AB4
TIM5_CH5_IRQ_MODE	0x00003AB8
TIM5_CH5_EIRQ_EN	0x00003ABC
TIM5_CH6_GPR0	0x00003B00
TIM5_CH6_GPR1	0x00003B04
TIM5_CH6_CNT	0x00003B08
TIM5_CH6_ECNT	0x00003B0C
TIM5_CH6_CNTS	0x00003B10
TIM5_CH6_TDUC	0x00003B14
TIM5_CH6_TDUV	0x00003B18
TIM5_CH6_FLT_RE	0x00003B1C
TIM5_CH6_FLT_FE	0x00003B20
TIM5_CH6_CTRL	0x00003B24
TIM5_CH6_ECTRL	0x00003B28
TIM5_CH6_IRQ_NOTIFY	0x00003B2C
TIM5_CH6_IRQ_EN	0x00003B30
TIM5_CH6_IRQ_FORCINT	0x00003B34
TIM5_CH6_IRQ_MODE	0x00003B38
TIM5_CH6_EIRQ_EN	0x00003B3C
TIM5_CH7_GPR0	0x00003B80
TIM5_CH7_GPR1	0x00003B84

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TIM5_CH7_CNT	0x00003B88
TIM5_CH7_ECNT	0x00003B8C
TIM5_CH7_CNTS	0x00003B90
TIM5_CH7_TDUC	0x00003B94
TIM5_CH7_TDUV	0x00003B98
TIM5_CH7_FLT_RE	0x00003B9C
TIM5_CH7_FLT_FE	0x00003BA0
TIM5_CH7_CTRL	0x00003BA4
TIM5_CH7_ECTRL	0x00003BA8
TIM5_CH7_IRQ_NOTIFY	0x00003BAC
TIM5_CH7_IRQ_EN	0x00003BB0
TIM5_CH7_IRQ_FORCINT	0x00003BB4
TIM5_CH7_IRQ_MODE	0x00003BB8
TIM5_CH7_EIRQ_EN	0x00003BBC
TOM0_CH0_CTRL	0x00008000
TOM0_CH0_SR0	0x00008004
TOM0_CH0_SR1	0x00008008
TOM0_CH0_CM0	0x0000800C
TOM0_CH0_CM1	0x00008010
TOM0_CH0_CN0	0x00008014
TOM0_CH0_STAT	0x00008018
TOM0_CH0_IRQ_NOTIFY	0x0000801C
TOM0_CH0_IRQ_EN	0x00008020
TOM0_CH0_IRQ_FORCINT	0x00008024
TOM0_CH0_IRQ_MODE	0x00008028
TOM0_TGC0_GLB_CTRL	0x00008030
TOM0_TGC0_ACT_TB	0x00008034
TOM0_TGC0_FUPD_CTRL	0x00008038
TOM0_TGC0_INT_TRIG	0x0000803C
TOM0_CH1_CTRL	0x00008040
TOM0_CH1_SR0	0x00008044
TOM0_CH1_SR1	0x00008048
TOM0_CH1_CM0	0x0000804C
TOM0_CH1_CM1	0x00008050
TOM0_CH1_CN0	0x00008054
TOM0_CH1_STAT	0x00008058
TOM0_CH1_IRQ_NOTIFY	0x0000805C
TOM0_CH1_IRQ_EN	0x00008060
TOM0_CH1_IRQ_FORCINT	0x00008064
TOM0_CH1_IRQ_MODE	0x00008068

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM0_TGC0_ENDIS_CTRL	0x00008070
TOM0_TGC0_ENDIS_STAT	0x00008074
TOM0_TGC0_OUTEN_CTRL	0x00008078
TOM0_TGC0_OUTEN_STAT	0x0000807C
TOM0_CH2_CTRL	0x00008080
TOM0_CH2_SR0	0x00008084
TOM0_CH2_SR1	0x00008088
TOM0_CH2_CM0	0x0000808C
TOM0_CH2_CM1	0x00008090
TOM0_CH2_CN0	0x00008094
TOM0_CH2_STAT	0x00008098
TOM0_CH2_IRQ_NOTIFY	0x0000809C
TOM0_CH2_IRQ_EN	0x000080A0
TOM0_CH2_IRQ_FORCINT	0x000080A4
TOM0_CH2_IRQ_MODE	0x000080A8
TOM0_CH3_CTRL	0x000080C0
TOM0_CH3_SR0	0x000080C4
TOM0_CH3_SR1	0x000080C8
TOM0_CH3_CM0	0x000080CC
TOM0_CH3_CM1	0x000080D0
TOM0_CH3_CN0	0x000080D4
TOM0_CH3_STAT	0x000080D8
TOM0_CH3_IRQ_NOTIFY	0x000080DC
TOM0_CH3_IRQ_EN	0x000080E0
TOM0_CH3_IRQ_FORCINT	0x000080E4
TOM0_CH3_IRQ_MODE	0x000080E8
TOM0_CH4_CTRL	0x00008100
TOM0_CH4_SR0	0x00008104
TOM0_CH4_SR1	0x00008108
TOM0_CH4_CM0	0x0000810C
TOM0_CH4_CM1	0x00008110
TOM0_CH4_CN0	0x00008114
TOM0_CH4_STAT	0x00008118
TOM0_CH4_IRQ_NOTIFY	0x0000811C
TOM0_CH4_IRQ_EN	0x00008120
TOM0_CH4_IRQ_FORCINT	0x00008124
TOM0_CH4_IRQ_MODE	0x00008128
TOM0_CH5_CTRL	0x00008140
TOM0_CH5_SR0	0x00008144
TOM0_CH5_SR1	0x00008148

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM0_CH5_CM0	0x0000814C
TOM0_CH5_CM1	0x00008150
TOM0_CH5_CN0	0x00008154
TOM0_CH5_STAT	0x00008158
TOM0_CH5_IRQ_NOTIFY	0x0000815C
TOM0_CH5_IRQ_EN	0x00008160
TOM0_CH5_IRQ_FORCINT	0x00008164
TOM0_CH5_IRQ_MODE	0x00008168
TOM0_CH6_CTRL	0x00008180
TOM0_CH6_SR0	0x00008184
TOM0_CH6_SR1	0x00008188
TOM0_CH6_CM0	0x0000818C
TOM0_CH6_CM1	0x00008190
TOM0_CH6_CN0	0x00008194
TOM0_CH6_STAT	0x00008198
TOM0_CH6_IRQ_NOTIFY	0x0000819C
TOM0_CH6_IRQ_EN	0x000081A0
TOM0_CH6_IRQ_FORCINT	0x000081A4
TOM0_CH6_IRQ_MODE	0x000081A8
TOM0_CH7_CTRL	0x000081C0
TOM0_CH7_SR0	0x000081C4
TOM0_CH7_SR1	0x000081C8
TOM0_CH7_CM0	0x000081CC
TOM0_CH7_CM1	0x000081D0
TOM0_CH7_CN0	0x000081D4
TOM0_CH7_STAT	0x000081D8
TOM0_CH7_IRQ_NOTIFY	0x000081DC
TOM0_CH7_IRQ_EN	0x000081E0
TOM0_CH7_IRQ_FORCINT	0x000081E4
TOM0_CH7_IRQ_MODE	0x000081E8
TOM0_CH8_CTRL	0x00008200
TOM0_CH8_SR0	0x00008204
TOM0_CH8_SR1	0x00008208
TOM0_CH8_CM0	0x0000820C
TOM0_CH8_CM1	0x00008210
TOM0_CH8_CN0	0x00008214
TOM0_CH8_STAT	0x00008218
TOM0_CH8_IRQ_NOTIFY	0x0000821C
TOM0_CH8_IRQ_EN	0x00008220
TOM0_CH8_IRQ_FORCINT	0x00008224

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM0_CH8_IRQ_MODE	0x00008228
TOM0_TGC1_GLB_CTRL	0x00008230
TOM0_TGC1_ACT_TB	0x00008234
TOM0_TGC1_FUPD_CTRL	0x00008238
TOM0_TGC1_INT_TRIG	0x0000823C
TOM0_CH9_CTRL	0x00008240
TOM0_CH9_SR0	0x00008244
TOM0_CH9_SR1	0x00008248
TOM0_CH9_CM0	0x0000824C
TOM0_CH9_CM1	0x00008250
TOM0_CH9_CN0	0x00008254
TOM0_CH9_STAT	0x00008258
TOM0_CH9_IRQ_NOTIFY	0x0000825C
TOM0_CH9_IRQ_EN	0x00008260
TOM0_CH9_IRQ_FORCINT	0x00008264
TOM0_CH9_IRQ_MODE	0x00008268
TOM0_TGC1_ENDIS_CTRL	0x00008270
TOM0_TGC1_ENDIS_STAT	0x00008274
TOM0_TGC1_OUTEN_CTRL	0x00008278
TOM0_TGC1_OUTEN_STAT	0x0000827C
TOM0_CH10_CTRL	0x00008280
TOM0_CH10_SR0	0x00008284
TOM0_CH10_SR1	0x00008288
TOM0_CH10_CM0	0x0000828C
TOM0_CH10_CM1	0x00008290
TOM0_CH10_CN0	0x00008294
TOM0_CH10_STAT	0x00008298
TOM0_CH10_IRQ_NOTIFY	0x0000829C
TOM0_CH10_IRQ_EN	0x000082A0
TOM0_CH10_IRQ_FORCINT	0x000082A4
TOM0_CH10_IRQ_MODE	0x000082A8
TOM0_CH11_CTRL	0x000082C0
TOM0_CH11_SR0	0x000082C4
TOM0_CH11_SR1	0x000082C8
TOM0_CH11_CM0	0x000082CC
TOM0_CH11_CM1	0x000082D0
TOM0_CH11_CN0	0x000082D4
TOM0_CH11_STAT	0x000082D8
TOM0_CH11_IRQ_NOTIFY	0x000082DC
TOM0_CH11_IRQ_EN	0x000082E0

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM0_CH11_IRQ_FORCINT	0x000082E4
TOM0_CH11_IRQ_MODE	0x000082E8
TOM0_CH12_CTRL	0x00008300
TOM0_CH12_SR0	0x00008304
TOM0_CH12_SR1	0x00008308
TOM0_CH12_CM0	0x0000830C
TOM0_CH12_CM1	0x00008310
TOM0_CH12_CN0	0x00008314
TOM0_CH12_STAT	0x00008318
TOM0_CH12_IRQ_NOTIFY	0x0000831C
TOM0_CH12_IRQ_EN	0x00008320
TOM0_CH12_IRQ_FORCINT	0x00008324
TOM0_CH12_IRQ_MODE	0x00008328
TOM0_CH13_CTRL	0x00008340
TOM0_CH13_SR0	0x00008344
TOM0_CH13_SR1	0x00008348
TOM0_CH13_CM0	0x0000834C
TOM0_CH13_CM1	0x00008350
TOM0_CH13_CN0	0x00008354
TOM0_CH13_STAT	0x00008358
TOM0_CH13_IRQ_NOTIFY	0x0000835C
TOM0_CH13_IRQ_EN	0x00008360
TOM0_CH13_IRQ_FORCINT	0x00008364
TOM0_CH13_IRQ_MODE	0x00008368
TOM0_CH14_CTRL	0x00008380
TOM0_CH14_SR0	0x00008384
TOM0_CH14_SR1	0x00008388
TOM0_CH14_CM0	0x0000838C
TOM0_CH14_CM1	0x00008390
TOM0_CH14_CN0	0x00008394
TOM0_CH14_STAT	0x00008398
TOM0_CH14_IRQ_NOTIFY	0x0000839C
TOM0_CH14_IRQ_EN	0x000083A0
TOM0_CH14_IRQ_FORCINT	0x000083A4
TOM0_CH14_IRQ_MODE	0x000083A8
TOM0_CH15_CTRL	0x000083C0
TOM0_CH15_SR0	0x000083C4
TOM0_CH15_SR1	0x000083C8
TOM0_CH15_CM0	0x000083CC
TOM0_CH15_CM1	0x000083D0

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM0_CH15_CN0	0x000083D4
TOM0_CH15_STAT	0x000083D8
TOM0_CH15_IRQ_NOTIFY	0x000083DC
TOM0_CH15_IRQ_EN	0x000083E0
TOM0_CH15_IRQ_FORCINT	0x000083E4
TOM0_CH15_IRQ_MODE	0x000083E8
TOM1_CH0_CTRL	0x00008800
TOM1_CH0_SR0	0x00008804
TOM1_CH0_SR1	0x00008808
TOM1_CH0_CM0	0x0000880C
TOM1_CH0_CM1	0x00008810
TOM1_CH0_CN0	0x00008814
TOM1_CH0_STAT	0x00008818
TOM1_CH0_IRQ_NOTIFY	0x0000881C
TOM1_CH0_IRQ_EN	0x00008820
TOM1_CH0_IRQ_FORCINT	0x00008824
TOM1_CH0_IRQ_MODE	0x00008828
TOM1_TGC0_GLB_CTRL	0x00008830
TOM1_TGC0_ACT_TB	0x00008834
TOM1_TGC0_FUPD_CTRL	0x00008838
TOM1_TGC0_INT_TRIG	0x0000883C
TOM1_CH1_CTRL	0x00008840
TOM1_CH1_SR0	0x00008844
TOM1_CH1_SR1	0x00008848
TOM1_CH1_CM0	0x0000884C
TOM1_CH1_CM1	0x00008850
TOM1_CH1_CN0	0x00008854
TOM1_CH1_STAT	0x00008858
TOM1_CH1_IRQ_NOTIFY	0x0000885C
TOM1_CH1_IRQ_EN	0x00008860
TOM1_CH1_IRQ_FORCINT	0x00008864
TOM1_CH1_IRQ_MODE	0x00008868
TOM1_TGC0_ENDIS_CTRL	0x00008870
TOM1_TGC0_ENDIS_STAT	0x00008874
TOM1_TGC0_OUTEN_CTRL	0x00008878
TOM1_TGC0_OUTEN_STAT	0x0000887C
TOM1_CH2_CTRL	0x00008880
TOM1_CH2_SR0	0x00008884
TOM1_CH2_SR1	0x00008888
TOM1_CH2_CM0	0x0000888C

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM1_CH2_CM1	0x00008890
TOM1_CH2_CN0	0x00008894
TOM1_CH2_STAT	0x00008898
TOM1_CH2_IRQ_NOTIFY	0x0000889C
TOM1_CH2_IRQ_EN	0x000088A0
TOM1_CH2_IRQ_FORCINT	0x000088A4
TOM1_CH2_IRQ_MODE	0x000088A8
TOM1_CH3_CTRL	0x000088C0
TOM1_CH3_SR0	0x000088C4
TOM1_CH3_SR1	0x000088C8
TOM1_CH3_CM0	0x000088CC
TOM1_CH3_CM1	0x000088D0
TOM1_CH3_CN0	0x000088D4
TOM1_CH3_STAT	0x000088D8
TOM1_CH3_IRQ_NOTIFY	0x000088DC
TOM1_CH3_IRQ_EN	0x000088E0
TOM1_CH3_IRQ_FORCINT	0x000088E4
TOM1_CH3_IRQ_MODE	0x000088E8
TOM1_CH4_CTRL	0x00008900
TOM1_CH4_SR0	0x00008904
TOM1_CH4_SR1	0x00008908
TOM1_CH4_CM0	0x0000890C
TOM1_CH4_CM1	0x00008910
TOM1_CH4_CN0	0x00008914
TOM1_CH4_STAT	0x00008918
TOM1_CH4_IRQ_NOTIFY	0x0000891C
TOM1_CH4_IRQ_EN	0x00008920
TOM1_CH4_IRQ_FORCINT	0x00008924
TOM1_CH4_IRQ_MODE	0x00008928
TOM1_CH5_CTRL	0x00008940
TOM1_CH5_SR0	0x00008944
TOM1_CH5_SR1	0x00008948
TOM1_CH5_CM0	0x0000894C
TOM1_CH5_CM1	0x00008950
TOM1_CH5_CN0	0x00008954
TOM1_CH5_STAT	0x00008958
TOM1_CH5_IRQ_NOTIFY	0x0000895C
TOM1_CH5_IRQ_EN	0x00008960
TOM1_CH5_IRQ_FORCINT	0x00008964
TOM1_CH5_IRQ_MODE	0x00008968

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM1_CH6_CTRL	0x00008980
TOM1_CH6_SR0	0x00008984
TOM1_CH6_SR1	0x00008988
TOM1_CH6_CM0	0x0000898C
TOM1_CH6_CM1	0x00008990
TOM1_CH6_CN0	0x00008994
TOM1_CH6_STAT	0x00008998
TOM1_CH6_IRQ_NOTIFY	0x0000899C
TOM1_CH6_IRQ_EN	0x000089A0
TOM1_CH6_IRQ_FORCINT	0x000089A4
TOM1_CH6_IRQ_MODE	0x000089A8
TOM1_CH7_CTRL	0x000089C0
TOM1_CH7_SR0	0x000089C4
TOM1_CH7_SR1	0x000089C8
TOM1_CH7_CM0	0x000089CC
TOM1_CH7_CM1	0x000089D0
TOM1_CH7_CN0	0x000089D4
TOM1_CH7_STAT	0x000089D8
TOM1_CH7_IRQ_NOTIFY	0x000089DC
TOM1_CH7_IRQ_EN	0x000089E0
TOM1_CH7_IRQ_FORCINT	0x000089E4
TOM1_CH7_IRQ_MODE	0x000089E8
TOM1_CH8_CTRL	0x00008A00
TOM1_CH8_SR0	0x00008A04
TOM1_CH8_SR1	0x00008A08
TOM1_CH8_CM0	0x00008A0C
TOM1_CH8_CM1	0x00008A10
TOM1_CH8_CN0	0x00008A14
TOM1_CH8_STAT	0x00008A18
TOM1_CH8_IRQ_NOTIFY	0x00008A1C
TOM1_CH8_IRQ_EN	0x00008A20
TOM1_CH8_IRQ_FORCINT	0x00008A24
TOM1_CH8_IRQ_MODE	0x00008A28
TOM1_TGC1_GLB_CTRL	0x00008A30
TOM1_TGC1_ACT_TB	0x00008A34
TOM1_TGC1_FUPD_CTRL	0x00008A38
TOM1_TGC1_INT_TRIG	0x00008A3C
TOM1_CH9_CTRL	0x00008A40
TOM1_CH9_SR0	0x00008A44
TOM1_CH9_SR1	0x00008A48

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM1_CH9_CM0	0x00008A4C
TOM1_CH9_CM1	0x00008A50
TOM1_CH9_CN0	0x00008A54
TOM1_CH9_STAT	0x00008A58
TOM1_CH9_IRQ_NOTIFY	0x00008A5C
TOM1_CH9_IRQ_EN	0x00008A60
TOM1_CH9_IRQ_FORCINT	0x00008A64
TOM1_CH9_IRQ_MODE	0x00008A68
TOM1_TGC1_ENDIS_CTRL	0x00008A70
TOM1_TGC1_ENDIS_STAT	0x00008A74
TOM1_TGC1_OUTEN_CTRL	0x00008A78
TOM1_TGC1_OUTEN_STAT	0x00008A7C
TOM1_CH10_CTRL	0x00008A80
TOM1_CH10_SR0	0x00008A84
TOM1_CH10_SR1	0x00008A88
TOM1_CH10_CM0	0x00008A8C
TOM1_CH10_CM1	0x00008A90
TOM1_CH10_CN0	0x00008A94
TOM1_CH10_STAT	0x00008A98
TOM1_CH10_IRQ_NOTIFY	0x00008A9C
TOM1_CH10_IRQ_EN	0x00008AA0
TOM1_CH10_IRQ_FORCINT	0x00008AA4
TOM1_CH10_IRQ_MODE	0x00008AA8
TOM1_CH11_CTRL	0x00008AC0
TOM1_CH11_SR0	0x00008AC4
TOM1_CH11_SR1	0x00008AC8
TOM1_CH11_CM0	0x00008ACC
TOM1_CH11_CM1	0x00008AD0
TOM1_CH11_CN0	0x00008AD4
TOM1_CH11_STAT	0x00008AD8
TOM1_CH11_IRQ_NOTIFY	0x00008ADC
TOM1_CH11_IRQ_EN	0x00008AE0
TOM1_CH11_IRQ_FORCINT	0x00008AE4
TOM1_CH11_IRQ_MODE	0x00008AE8
TOM1_CH12_CTRL	0x00008B00
TOM1_CH12_SR0	0x00008B04
TOM1_CH12_SR1	0x00008B08
TOM1_CH12_CM0	0x00008B0C
TOM1_CH12_CM1	0x00008B10
TOM1_CH12_CN0	0x00008B14

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM1_CH12_STAT	0x00008B18
TOM1_CH12_IRQ_NOTIFY	0x00008B1C
TOM1_CH12_IRQ_EN	0x00008B20
TOM1_CH12_IRQ_FORCINT	0x00008B24
TOM1_CH12_IRQ_MODE	0x00008B28
TOM1_CH13_CTRL	0x00008B40
TOM1_CH13_SR0	0x00008B44
TOM1_CH13_SR1	0x00008B48
TOM1_CH13_CM0	0x00008B4C
TOM1_CH13_CM1	0x00008B50
TOM1_CH13_CN0	0x00008B54
TOM1_CH13_STAT	0x00008B58
TOM1_CH13_IRQ_NOTIFY	0x00008B5C
TOM1_CH13_IRQ_EN	0x00008B60
TOM1_CH13_IRQ_FORCINT	0x00008B64
TOM1_CH13_IRQ_MODE	0x00008B68
TOM1_CH14_CTRL	0x00008B80
TOM1_CH14_SR0	0x00008B84
TOM1_CH14_SR1	0x00008B88
TOM1_CH14_CM0	0x00008B8C
TOM1_CH14_CM1	0x00008B90
TOM1_CH14_CN0	0x00008B94
TOM1_CH14_STAT	0x00008B98
TOM1_CH14_IRQ_NOTIFY	0x00008B9C
TOM1_CH14_IRQ_EN	0x00008BA0
TOM1_CH14_IRQ_FORCINT	0x00008BA4
TOM1_CH14_IRQ_MODE	0x00008BA8
TOM1_CH15_CTRL	0x00008BC0
TOM1_CH15_SR0	0x00008BC4
TOM1_CH15_SR1	0x00008BC8
TOM1_CH15_CM0	0x00008BCC
TOM1_CH15_CM1	0x00008BD0
TOM1_CH15_CN0	0x00008BD4
TOM1_CH15_STAT	0x00008BD8
TOM1_CH15_IRQ_NOTIFY	0x00008BDC
TOM1_CH15_IRQ_EN	0x00008BE0
TOM1_CH15_IRQ_FORCINT	0x00008BE4
TOM1_CH15_IRQ_MODE	0x00008BE8
TOM2_CH0_CTRL	0x00009000
TOM2_CH0_SR0	0x00009004

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM2_CH0_SR1	0x00009008
TOM2_CH0_CM0	0x0000900C
TOM2_CH0_CM1	0x00009010
TOM2_CH0_CN0	0x00009014
TOM2_CH0_STAT	0x00009018
TOM2_CH0_IRQ_NOTIFY	0x0000901C
TOM2_CH0_IRQ_EN	0x00009020
TOM2_CH0_IRQ_FORCINT	0x00009024
TOM2_CH0_IRQ_MODE	0x00009028
TOM2_TGC0_GLB_CTRL	0x00009030
TOM2_TGC0_ACT_TB	0x00009034
TOM2_TGC0_FUPD_CTRL	0x00009038
TOM2_TGC0_INT_TRIG	0x0000903C
TOM2_CH1_CTRL	0x00009040
TOM2_CH1_SR0	0x00009044
TOM2_CH1_SR1	0x00009048
TOM2_CH1_CM0	0x0000904C
TOM2_CH1_CM1	0x00009050
TOM2_CH1_CN0	0x00009054
TOM2_CH1_STAT	0x00009058
TOM2_CH1_IRQ_NOTIFY	0x0000905C
TOM2_CH1_IRQ_EN	0x00009060
TOM2_CH1_IRQ_FORCINT	0x00009064
TOM2_CH1_IRQ_MODE	0x00009068
TOM2_TGC0_ENDIS_CTRL	0x00009070
TOM2_TGC0_ENDIS_STAT	0x00009074
TOM2_TGC0_OUTEN_CTRL	0x00009078
TOM2_TGC0_OUTEN_STAT	0x0000907C
TOM2_CH2_CTRL	0x00009080
TOM2_CH2_SR0	0x00009084
TOM2_CH2_SR1	0x00009088
TOM2_CH2_CM0	0x0000908C
TOM2_CH2_CM1	0x00009090
TOM2_CH2_CN0	0x00009094
TOM2_CH2_STAT	0x00009098
TOM2_CH2_IRQ_NOTIFY	0x0000909C
TOM2_CH2_IRQ_EN	0x000090A0
TOM2_CH2_IRQ_FORCINT	0x000090A4
TOM2_CH2_IRQ_MODE	0x000090A8
TOM2_CH3_CTRL	0x000090C0

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM2_CH3_SR0	0x000090C4
TOM2_CH3_SR1	0x000090C8
TOM2_CH3_CM0	0x000090CC
TOM2_CH3_CM1	0x000090D0
TOM2_CH3_CN0	0x000090D4
TOM2_CH3_STAT	0x000090D8
TOM2_CH3_IRQ_NOTIFY	0x000090DC
TOM2_CH3_IRQ_EN	0x000090E0
TOM2_CH3_IRQ_FORCINT	0x000090E4
TOM2_CH3_IRQ_MODE	0x000090E8
TOM2_CH4_CTRL	0x00009100
TOM2_CH4_SR0	0x00009104
TOM2_CH4_SR1	0x00009108
TOM2_CH4_CM0	0x0000910C
TOM2_CH4_CM1	0x00009110
TOM2_CH4_CN0	0x00009114
TOM2_CH4_STAT	0x00009118
TOM2_CH4_IRQ_NOTIFY	0x0000911C
TOM2_CH4_IRQ_EN	0x00009120
TOM2_CH4_IRQ_FORCINT	0x00009124
TOM2_CH4_IRQ_MODE	0x00009128
TOM2_CH5_CTRL	0x00009140
TOM2_CH5_SR0	0x00009144
TOM2_CH5_SR1	0x00009148
TOM2_CH5_CM0	0x0000914C
TOM2_CH5_CM1	0x00009150
TOM2_CH5_CN0	0x00009154
TOM2_CH5_STAT	0x00009158
TOM2_CH5_IRQ_NOTIFY	0x0000915C
TOM2_CH5_IRQ_EN	0x00009160
TOM2_CH5_IRQ_FORCINT	0x00009164
TOM2_CH5_IRQ_MODE	0x00009168
TOM2_CH6_CTRL	0x00009180
TOM2_CH6_SR0	0x00009184
TOM2_CH6_SR1	0x00009188
TOM2_CH6_CM0	0x0000918C
TOM2_CH6_CM1	0x00009190
TOM2_CH6_CN0	0x00009194
TOM2_CH6_STAT	0x00009198
TOM2_CH6_IRQ_NOTIFY	0x0000919C

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM2_CH6_IRQ_EN	0x000091A0
TOM2_CH6_IRQ_FORCINT	0x000091A4
TOM2_CH6_IRQ_MODE	0x000091A8
TOM2_CH7_CTRL	0x000091C0
TOM2_CH7_SR0	0x000091C4
TOM2_CH7_SR1	0x000091C8
TOM2_CH7_CM0	0x000091CC
TOM2_CH7_CM1	0x000091D0
TOM2_CH7_CN0	0x000091D4
TOM2_CH7_STAT	0x000091D8
TOM2_CH7_IRQ_NOTIFY	0x000091DC
TOM2_CH7_IRQ_EN	0x000091E0
TOM2_CH7_IRQ_FORCINT	0x000091E4
TOM2_CH7_IRQ_MODE	0x000091E8
TOM2_CH8_CTRL	0x00009200
TOM2_CH8_SR0	0x00009204
TOM2_CH8_SR1	0x00009208
TOM2_CH8_CM0	0x0000920C
TOM2_CH8_CM1	0x00009210
TOM2_CH8_CN0	0x00009214
TOM2_CH8_STAT	0x00009218
TOM2_CH8_IRQ_NOTIFY	0x0000921C
TOM2_CH8_IRQ_EN	0x00009220
TOM2_CH8_IRQ_FORCINT	0x00009224
TOM2_CH8_IRQ_MODE	0x00009228
TOM2_TGC1_GLB_CTRL	0x00009230
TOM2_TGC1_ACT_TB	0x00009234
TOM2_TGC1_FUPD_CTRL	0x00009238
TOM2_TGC1_INT_TRIG	0x0000923C
TOM2_CH9_CTRL	0x00009240
TOM2_CH9_SR0	0x00009244
TOM2_CH9_SR1	0x00009248
TOM2_CH9_CM0	0x0000924C
TOM2_CH9_CM1	0x00009250
TOM2_CH9_CN0	0x00009254
TOM2_CH9_STAT	0x00009258
TOM2_CH9_IRQ_NOTIFY	0x0000925C
TOM2_CH9_IRQ_EN	0x00009260
TOM2_CH9_IRQ_FORCINT	0x00009264
TOM2_CH9_IRQ_MODE	0x00009268

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM2_TGC1_ENDIS_CTRL	0x00009270
TOM2_TGC1_ENDIS_STAT	0x00009274
TOM2_TGC1_OUTEN_CTRL	0x00009278
TOM2_TGC1_OUTEN_STAT	0x0000927C
TOM2_CH10_CTRL	0x00009280
TOM2_CH10_SR0	0x00009284
TOM2_CH10_SR1	0x00009288
TOM2_CH10_CM0	0x0000928C
TOM2_CH10_CM1	0x00009290
TOM2_CH10_CN0	0x00009294
TOM2_CH10_STAT	0x00009298
TOM2_CH10_IRQ_NOTIFY	0x0000929C
TOM2_CH10_IRQ_EN	0x000092A0
TOM2_CH10_IRQ_FORCINT	0x000092A4
TOM2_CH10_IRQ_MODE	0x000092A8
TOM2_CH11_CTRL	0x000092C0
TOM2_CH11_SR0	0x000092C4
TOM2_CH11_SR1	0x000092C8
TOM2_CH11_CM0	0x000092CC
TOM2_CH11_CM1	0x000092D0
TOM2_CH11_CN0	0x000092D4
TOM2_CH11_STAT	0x000092D8
TOM2_CH11_IRQ_NOTIFY	0x000092DC
TOM2_CH11_IRQ_EN	0x000092E0
TOM2_CH11_IRQ_FORCINT	0x000092E4
TOM2_CH11_IRQ_MODE	0x000092E8
TOM2_CH12_CTRL	0x00009300
TOM2_CH12_SR0	0x00009304
TOM2_CH12_SR1	0x00009308
TOM2_CH12_CM0	0x0000930C
TOM2_CH12_CM1	0x00009310
TOM2_CH12_CN0	0x00009314
TOM2_CH12_STAT	0x00009318
TOM2_CH12_IRQ_NOTIFY	0x0000931C
TOM2_CH12_IRQ_EN	0x00009320
TOM2_CH12_IRQ_FORCINT	0x00009324
TOM2_CH12_IRQ_MODE	0x00009328
TOM2_CH13_CTRL	0x00009340
TOM2_CH13_SR0	0x00009344
TOM2_CH13_SR1	0x00009348

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM2_CH13_CM0	0x0000934C
TOM2_CH13_CM1	0x00009350
TOM2_CH13_CN0	0x00009354
TOM2_CH13_STAT	0x00009358
TOM2_CH13_IRQ_NOTIFY	0x0000935C
TOM2_CH13_IRQ_EN	0x00009360
TOM2_CH13_IRQ_FORCINT	0x00009364
TOM2_CH13_IRQ_MODE	0x00009368
TOM2_CH14_CTRL	0x00009380
TOM2_CH14_SR0	0x00009384
TOM2_CH14_SR1	0x00009388
TOM2_CH14_CM0	0x0000938C
TOM2_CH14_CM1	0x00009390
TOM2_CH14_CN0	0x00009394
TOM2_CH14_STAT	0x00009398
TOM2_CH14_IRQ_NOTIFY	0x0000939C
TOM2_CH14_IRQ_EN	0x000093A0
TOM2_CH14_IRQ_FORCINT	0x000093A4
TOM2_CH14_IRQ_MODE	0x000093A8
TOM2_CH15_CTRL	0x000093C0
TOM2_CH15_SR0	0x000093C4
TOM2_CH15_SR1	0x000093C8
TOM2_CH15_CM0	0x000093CC
TOM2_CH15_CM1	0x000093D0
TOM2_CH15_CN0	0x000093D4
TOM2_CH15_STAT	0x000093D8
TOM2_CH15_IRQ_NOTIFY	0x000093DC
TOM2_CH15_IRQ_EN	0x000093E0
TOM2_CH15_IRQ_FORCINT	0x000093E4
TOM2_CH15_IRQ_MODE	0x000093E8
TOM3_CH0_CTRL	0x00009800
TOM3_CH0_SR0	0x00009804
TOM3_CH0_SR1	0x00009808
TOM3_CH0_CM0	0x0000980C
TOM3_CH0_CM1	0x00009810
TOM3_CH0_CN0	0x00009814
TOM3_CH0_STAT	0x00009818
TOM3_CH0_IRQ_NOTIFY	0x0000981C
TOM3_CH0_IRQ_EN	0x00009820
TOM3_CH0_IRQ_FORCINT	0x00009824

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM3_CH0_IRQ_MODE	0x00009828
TOM3_TGC0_GLB_CTRL	0x00009830
TOM3_TGC0_ACT_TB	0x00009834
TOM3_TGC0_FUPD_CTRL	0x00009838
TOM3_TGC0_INT_TRIG	0x0000983C
TOM3_CH1_CTRL	0x00009840
TOM3_CH1_SR0	0x00009844
TOM3_CH1_SR1	0x00009848
TOM3_CH1_CM0	0x0000984C
TOM3_CH1_CM1	0x00009850
TOM3_CH1_CN0	0x00009854
TOM3_CH1_STAT	0x00009858
TOM3_CH1_IRQ_NOTIFY	0x0000985C
TOM3_CH1_IRQ_EN	0x00009860
TOM3_CH1_IRQ_FORCINT	0x00009864
TOM3_CH1_IRQ_MODE	0x00009868
TOM3_TGC0_ENDIS_CTRL	0x00009870
TOM3_TGC0_ENDIS_STAT	0x00009874
TOM3_TGC0_OUTEN_CTRL	0x00009878
TOM3_TGC0_OUTEN_STAT	0x0000987C
TOM3_CH2_CTRL	0x00009880
TOM3_CH2_SR0	0x00009884
TOM3_CH2_SR1	0x00009888
TOM3_CH2_CM0	0x0000988C
TOM3_CH2_CM1	0x00009890
TOM3_CH2_CN0	0x00009894
TOM3_CH2_STAT	0x00009898
TOM3_CH2_IRQ_NOTIFY	0x0000989C
TOM3_CH2_IRQ_EN	0x000098A0
TOM3_CH2_IRQ_FORCINT	0x000098A4
TOM3_CH2_IRQ_MODE	0x000098A8
TOM3_CH3_CTRL	0x000098C0
TOM3_CH3_SR0	0x000098C4
TOM3_CH3_SR1	0x000098C8
TOM3_CH3_CM0	0x000098CC
TOM3_CH3_CM1	0x000098D0
TOM3_CH3_CN0	0x000098D4
TOM3_CH3_STAT	0x000098D8
TOM3_CH3_IRQ_NOTIFY	0x000098DC
TOM3_CH3_IRQ_EN	0x000098E0

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM3_CH3_IRQ_FORCINT	0x000098E4
TOM3_CH3_IRQ_MODE	0x000098E8
TOM3_CH4_CTRL	0x00009900
TOM3_CH4_SR0	0x00009904
TOM3_CH4_SR1	0x00009908
TOM3_CH4_CM0	0x0000990C
TOM3_CH4_CM1	0x00009910
TOM3_CH4_CN0	0x00009914
TOM3_CH4_STAT	0x00009918
TOM3_CH4_IRQ_NOTIFY	0x0000991C
TOM3_CH4_IRQ_EN	0x00009920
TOM3_CH4_IRQ_FORCINT	0x00009924
TOM3_CH4_IRQ_MODE	0x00009928
TOM3_CH5_CTRL	0x00009940
TOM3_CH5_SR0	0x00009944
TOM3_CH5_SR1	0x00009948
TOM3_CH5_CM0	0x0000994C
TOM3_CH5_CM1	0x00009950
TOM3_CH5_CN0	0x00009954
TOM3_CH5_STAT	0x00009958
TOM3_CH5_IRQ_NOTIFY	0x0000995C
TOM3_CH5_IRQ_EN	0x00009960
TOM3_CH5_IRQ_FORCINT	0x00009964
TOM3_CH5_IRQ_MODE	0x00009968
TOM3_CH6_CTRL	0x00009980
TOM3_CH6_SR0	0x00009984
TOM3_CH6_SR1	0x00009988
TOM3_CH6_CM0	0x0000998C
TOM3_CH6_CM1	0x00009990
TOM3_CH6_CN0	0x00009994
TOM3_CH6_STAT	0x00009998
TOM3_CH6_IRQ_NOTIFY	0x0000999C
TOM3_CH6_IRQ_EN	0x000099A0
TOM3_CH6_IRQ_FORCINT	0x000099A4
TOM3_CH6_IRQ_MODE	0x000099A8
TOM3_CH7_CTRL	0x000099C0
TOM3_CH7_SR0	0x000099C4
TOM3_CH7_SR1	0x000099C8
TOM3_CH7_CM0	0x000099CC
TOM3_CH7_CM1	0x000099D0

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM3_CH7_CN0	0x000099D4
TOM3_CH7_STAT	0x000099D8
TOM3_CH7_IRQ_NOTIFY	0x000099DC
TOM3_CH7_IRQ_EN	0x000099E0
TOM3_CH7_IRQ_FORCINT	0x000099E4
TOM3_CH7_IRQ_MODE	0x000099E8
TOM3_CH8_CTRL	0x00009A00
TOM3_CH8_SR0	0x00009A04
TOM3_CH8_SR1	0x00009A08
TOM3_CH8_CM0	0x00009A0C
TOM3_CH8_CM1	0x00009A10
TOM3_CH8_CN0	0x00009A14
TOM3_CH8_STAT	0x00009A18
TOM3_CH8_IRQ_NOTIFY	0x00009A1C
TOM3_CH8_IRQ_EN	0x00009A20
TOM3_CH8_IRQ_FORCINT	0x00009A24
TOM3_CH8_IRQ_MODE	0x00009A28
TOM3_TGC1_GLB_CTRL	0x00009A30
TOM3_TGC1_ACT_TB	0x00009A34
TOM3_TGC1_FUPD_CTRL	0x00009A38
TOM3_TGC1_INT_TRIG	0x00009A3C
TOM3_CH9_CTRL	0x00009A40
TOM3_CH9_SR0	0x00009A44
TOM3_CH9_SR1	0x00009A48
TOM3_CH9_CM0	0x00009A4C
TOM3_CH9_CM1	0x00009A50
TOM3_CH9_CN0	0x00009A54
TOM3_CH9_STAT	0x00009A58
TOM3_CH9_IRQ_NOTIFY	0x00009A5C
TOM3_CH9_IRQ_EN	0x00009A60
TOM3_CH9_IRQ_FORCINT	0x00009A64
TOM3_CH9_IRQ_MODE	0x00009A68
TOM3_TGC1_ENDIS_CTRL	0x00009A70
TOM3_TGC1_ENDIS_STAT	0x00009A74
TOM3_TGC1_OUTEN_CTRL	0x00009A78
TOM3_TGC1_OUTEN_STAT	0x00009A7C
TOM3_CH10_CTRL	0x00009A80
TOM3_CH10_SR0	0x00009A84
TOM3_CH10_SR1	0x00009A88
TOM3_CH10_CM0	0x00009A8C

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM3_CH10_CM1	0x00009A90
TOM3_CH10_CN0	0x00009A94
TOM3_CH10_STAT	0x00009A98
TOM3_CH10_IRQ_NOTIFY	0x00009A9C
TOM3_CH10_IRQ_EN	0x00009AA0
TOM3_CH10_IRQ_FORCINT	0x00009AA4
TOM3_CH10_IRQ_MODE	0x00009AA8
TOM3_CH11_CTRL	0x00009AC0
TOM3_CH11_SR0	0x00009AC4
TOM3_CH11_SR1	0x00009AC8
TOM3_CH11_CM0	0x00009ACC
TOM3_CH11_CM1	0x00009AD0
TOM3_CH11_CN0	0x00009AD4
TOM3_CH11_STAT	0x00009AD8
TOM3_CH11_IRQ_NOTIFY	0x00009ADC
TOM3_CH11_IRQ_EN	0x00009AE0
TOM3_CH11_IRQ_FORCINT	0x00009AE4
TOM3_CH11_IRQ_MODE	0x00009AE8
TOM3_CH12_CTRL	0x00009B00
TOM3_CH12_SR0	0x00009B04
TOM3_CH12_SR1	0x00009B08
TOM3_CH12_CM0	0x00009B0C
TOM3_CH12_CM1	0x00009B10
TOM3_CH12_CN0	0x00009B14
TOM3_CH12_STAT	0x00009B18
TOM3_CH12_IRQ_NOTIFY	0x00009B1C
TOM3_CH12_IRQ_EN	0x00009B20
TOM3_CH12_IRQ_FORCINT	0x00009B24
TOM3_CH12_IRQ_MODE	0x00009B28
TOM3_CH13_CTRL	0x00009B40
TOM3_CH13_SR0	0x00009B44
TOM3_CH13_SR1	0x00009B48
TOM3_CH13_CM0	0x00009B4C
TOM3_CH13_CM1	0x00009B50
TOM3_CH13_CN0	0x00009B54
TOM3_CH13_STAT	0x00009B58
TOM3_CH13_IRQ_NOTIFY	0x00009B5C
TOM3_CH13_IRQ_EN	0x00009B60
TOM3_CH13_IRQ_FORCINT	0x00009B64
TOM3_CH13_IRQ_MODE	0x00009B68

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM3_CH14_CTRL	0x00009B80
TOM3_CH14_SR0	0x00009B84
TOM3_CH14_SR1	0x00009B88
TOM3_CH14_CM0	0x00009B8C
TOM3_CH14_CM1	0x00009B90
TOM3_CH14_CN0	0x00009B94
TOM3_CH14_STAT	0x00009B98
TOM3_CH14_IRQ_NOTIFY	0x00009B9C
TOM3_CH14_IRQ_EN	0x00009BA0
TOM3_CH14_IRQ_FORCINT	0x00009BA4
TOM3_CH14_IRQ_MODE	0x00009BA8
TOM3_CH15_CTRL	0x00009BC0
TOM3_CH15_SR0	0x00009BC4
TOM3_CH15_SR1	0x00009BC8
TOM3_CH15_CM0	0x00009BCC
TOM3_CH15_CM1	0x00009BD0
TOM3_CH15_CN0	0x00009BD4
TOM3_CH15_STAT	0x00009BD8
TOM3_CH15_IRQ_NOTIFY	0x00009BDC
TOM3_CH15_IRQ_EN	0x00009BE0
TOM3_CH15_IRQ_FORCINT	0x00009BE4
TOM3_CH15_IRQ_MODE	0x00009BE8
TOM4_CH0_CTRL	0x0000A000
TOM4_CH0_SR0	0x0000A004
TOM4_CH0_SR1	0x0000A008
TOM4_CH0_CM0	0x0000A00C
TOM4_CH0_CM1	0x0000A010
TOM4_CH0_CN0	0x0000A014
TOM4_CH0_STAT	0x0000A018
TOM4_CH0_IRQ_NOTIFY	0x0000A01C
TOM4_CH0_IRQ_EN	0x0000A020
TOM4_CH0_IRQ_FORCINT	0x0000A024
TOM4_CH0_IRQ_MODE	0x0000A028
TOM4_TGC0_GLB_CTRL	0x0000A030
TOM4_TGC0_ACT_TB	0x0000A034
TOM4_TGC0_FUPD_CTRL	0x0000A038
TOM4_TGC0_INT_TRIG	0x0000A03C
TOM4_CH1_CTRL	0x0000A040
TOM4_CH1_SR0	0x0000A044
TOM4_CH1_SR1	0x0000A048

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM4_CH1_CM0	0x0000A04C
TOM4_CH1_CM1	0x0000A050
TOM4_CH1_CN0	0x0000A054
TOM4_CH1_STAT	0x0000A058
TOM4_CH1_IRQ_NOTIFY	0x0000A05C
TOM4_CH1_IRQ_EN	0x0000A060
TOM4_CH1_IRQ_FORCINT	0x0000A064
TOM4_CH1_IRQ_MODE	0x0000A068
TOM4_TGC0_ENDIS_CTRL	0x0000A070
TOM4_TGC0_ENDIS_STAT	0x0000A074
TOM4_TGC0_OUTEN_CTRL	0x0000A078
TOM4_TGC0_OUTEN_STAT	0x0000A07C
TOM4_CH2_CTRL	0x0000A080
TOM4_CH2_SR0	0x0000A084
TOM4_CH2_SR1	0x0000A088
TOM4_CH2_CM0	0x0000A08C
TOM4_CH2_CM1	0x0000A090
TOM4_CH2_CN0	0x0000A094
TOM4_CH2_STAT	0x0000A098
TOM4_CH2_IRQ_NOTIFY	0x0000A09C
TOM4_CH2_IRQ_EN	0x0000A0A0
TOM4_CH2_IRQ_FORCINT	0x0000A0A4
TOM4_CH2_IRQ_MODE	0x0000A0A8
TOM4_CH3_CTRL	0x0000A0C0
TOM4_CH3_SR0	0x0000A0C4
TOM4_CH3_SR1	0x0000A0C8
TOM4_CH3_CM0	0x0000A0CC
TOM4_CH3_CM1	0x0000A0D0
TOM4_CH3_CN0	0x0000A0D4
TOM4_CH3_STAT	0x0000A0D8
TOM4_CH3_IRQ_NOTIFY	0x0000A0DC
TOM4_CH3_IRQ_EN	0x0000A0E0
TOM4_CH3_IRQ_FORCINT	0x0000A0E4
TOM4_CH3_IRQ_MODE	0x0000A0E8
TOM4_CH4_CTRL	0x0000A100
TOM4_CH4_SR0	0x0000A104
TOM4_CH4_SR1	0x0000A108
TOM4_CH4_CM0	0x0000A10C
TOM4_CH4_CM1	0x0000A110
TOM4_CH4_CN0	0x0000A114

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM4_CH4_STAT	0x0000A118
TOM4_CH4_IRQ_NOTIFY	0x0000A11C
TOM4_CH4_IRQ_EN	0x0000A120
TOM4_CH4_IRQ_FORCINT	0x0000A124
TOM4_CH4_IRQ_MODE	0x0000A128
TOM4_CH5_CTRL	0x0000A140
TOM4_CH5_SR0	0x0000A144
TOM4_CH5_SR1	0x0000A148
TOM4_CH5_CM0	0x0000A14C
TOM4_CH5_CM1	0x0000A150
TOM4_CH5_CN0	0x0000A154
TOM4_CH5_STAT	0x0000A158
TOM4_CH5_IRQ_NOTIFY	0x0000A15C
TOM4_CH5_IRQ_EN	0x0000A160
TOM4_CH5_IRQ_FORCINT	0x0000A164
TOM4_CH5_IRQ_MODE	0x0000A168
TOM4_CH6_CTRL	0x0000A180
TOM4_CH6_SR0	0x0000A184
TOM4_CH6_SR1	0x0000A188
TOM4_CH6_CM0	0x0000A18C
TOM4_CH6_CM1	0x0000A190
TOM4_CH6_CN0	0x0000A194
TOM4_CH6_STAT	0x0000A198
TOM4_CH6_IRQ_NOTIFY	0x0000A19C
TOM4_CH6_IRQ_EN	0x0000A1A0
TOM4_CH6_IRQ_FORCINT	0x0000A1A4
TOM4_CH6_IRQ_MODE	0x0000A1A8
TOM4_CH7_CTRL	0x0000A1C0
TOM4_CH7_SR0	0x0000A1C4
TOM4_CH7_SR1	0x0000A1C8
TOM4_CH7_CM0	0x0000A1CC
TOM4_CH7_CM1	0x0000A1D0
TOM4_CH7_CN0	0x0000A1D4
TOM4_CH7_STAT	0x0000A1D8
TOM4_CH7_IRQ_NOTIFY	0x0000A1DC
TOM4_CH7_IRQ_EN	0x0000A1E0
TOM4_CH7_IRQ_FORCINT	0x0000A1E4
TOM4_CH7_IRQ_MODE	0x0000A1E8
TOM4_CH8_CTRL	0x0000A200
TOM4_CH8_SR0	0x0000A204

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM4_CH8_SR1	0x0000A208
TOM4_CH8_CM0	0x0000A20C
TOM4_CH8_CM1	0x0000A210
TOM4_CH8_CN0	0x0000A214
TOM4_CH8_STAT	0x0000A218
TOM4_CH8_IRQ_NOTIFY	0x0000A21C
TOM4_CH8_IRQ_EN	0x0000A220
TOM4_CH8_IRQ_FORCINT	0x0000A224
TOM4_CH8_IRQ_MODE	0x0000A228
TOM4_TGC1_GLB_CTRL	0x0000A230
TOM4_TGC1_ACT_TB	0x0000A234
TOM4_TGC1_FUPD_CTRL	0x0000A238
TOM4_TGC1_INT_TRIG	0x0000A23C
TOM4_CH9_CTRL	0x0000A240
TOM4_CH9_SR0	0x0000A244
TOM4_CH9_SR1	0x0000A248
TOM4_CH9_CM0	0x0000A24C
TOM4_CH9_CM1	0x0000A250
TOM4_CH9_CN0	0x0000A254
TOM4_CH9_STAT	0x0000A258
TOM4_CH9_IRQ_NOTIFY	0x0000A25C
TOM4_CH9_IRQ_EN	0x0000A260
TOM4_CH9_IRQ_FORCINT	0x0000A264
TOM4_CH9_IRQ_MODE	0x0000A268
TOM4_TGC1_ENDIS_CTRL	0x0000A270
TOM4_TGC1_ENDIS_STAT	0x0000A274
TOM4_TGC1_OUTEN_CTRL	0x0000A278
TOM4_TGC1_OUTEN_STAT	0x0000A27C
TOM4_CH10_CTRL	0x0000A280
TOM4_CH10_SR0	0x0000A284
TOM4_CH10_SR1	0x0000A288
TOM4_CH10_CM0	0x0000A28C
TOM4_CH10_CM1	0x0000A290
TOM4_CH10_CN0	0x0000A294
TOM4_CH10_STAT	0x0000A298
TOM4_CH10_IRQ_NOTIFY	0x0000A29C
TOM4_CH10_IRQ_EN	0x0000A2A0
TOM4_CH10_IRQ_FORCINT	0x0000A2A4
TOM4_CH10_IRQ_MODE	0x0000A2A8
TOM4_CH11_CTRL	0x0000A2C0

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM4_CH11_SR0	0x0000A2C4
TOM4_CH11_SR1	0x0000A2C8
TOM4_CH11_CM0	0x0000A2CC
TOM4_CH11_CM1	0x0000A2D0
TOM4_CH11_CN0	0x0000A2D4
TOM4_CH11_STAT	0x0000A2D8
TOM4_CH11_IRQ_NOTIFY	0x0000A2DC
TOM4_CH11_IRQ_EN	0x0000A2E0
TOM4_CH11_IRQ_FORCINT	0x0000A2E4
TOM4_CH11_IRQ_MODE	0x0000A2E8
TOM4_CH12_CTRL	0x0000A300
TOM4_CH12_SR0	0x0000A304
TOM4_CH12_SR1	0x0000A308
TOM4_CH12_CM0	0x0000A30C
TOM4_CH12_CM1	0x0000A310
TOM4_CH12_CN0	0x0000A314
TOM4_CH12_STAT	0x0000A318
TOM4_CH12_IRQ_NOTIFY	0x0000A31C
TOM4_CH12_IRQ_EN	0x0000A320
TOM4_CH12_IRQ_FORCINT	0x0000A324
TOM4_CH12_IRQ_MODE	0x0000A328
TOM4_CH13_CTRL	0x0000A340
TOM4_CH13_SR0	0x0000A344
TOM4_CH13_SR1	0x0000A348
TOM4_CH13_CM0	0x0000A34C
TOM4_CH13_CM1	0x0000A350
TOM4_CH13_CN0	0x0000A354
TOM4_CH13_STAT	0x0000A358
TOM4_CH13_IRQ_NOTIFY	0x0000A35C
TOM4_CH13_IRQ_EN	0x0000A360
TOM4_CH13_IRQ_FORCINT	0x0000A364
TOM4_CH13_IRQ_MODE	0x0000A368
TOM4_CH14_CTRL	0x0000A380
TOM4_CH14_SR0	0x0000A384
TOM4_CH14_SR1	0x0000A388
TOM4_CH14_CM0	0x0000A38C
TOM4_CH14_CM1	0x0000A390
TOM4_CH14_CN0	0x0000A394
TOM4_CH14_STAT	0x0000A398
TOM4_CH14_IRQ_NOTIFY	0x0000A39C

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

TOM4_CH14_IRQ_EN	0x0000A3A0
TOM4_CH14_IRQ_FORCINT	0x0000A3A4
TOM4_CH14_IRQ_MODE	0x0000A3A8
TOM4_CH15_CTRL	0x0000A3C0
TOM4_CH15_SR0	0x0000A3C4
TOM4_CH15_SR1	0x0000A3C8
TOM4_CH15_CM0	0x0000A3CC
TOM4_CH15_CM1	0x0000A3D0
TOM4_CH15_CN0	0x0000A3D4
TOM4_CH15_STAT	0x0000A3D8
TOM4_CH15_IRQ_NOTIFY	0x0000A3DC
TOM4_CH15_IRQ_EN	0x0000A3E0
TOM4_CH15_IRQ_FORCINT	0x0000A3E4
TOM4_CH15_IRQ_MODE	0x0000A3E8
ATOM0_CH0_RDADDR	0x0000D000
ATOM0_CH0_CTRL	0x0000D004
ATOM0_CH0_SR0	0x0000D008
ATOM0_CH0_SR1	0x0000D00C
ATOM0_CH0_CM0	0x0000D010
ATOM0_CH0_CM1	0x0000D014
ATOM0_CH0_CN0	0x0000D018
ATOM0_CH0_STAT	0x0000D01C
ATOM0_CH0_IRQ_NOTIFY	0x0000D020
ATOM0_CH0_IRQ_EN	0x0000D024
ATOM0_CH0_IRQ_FORCINT	0x0000D028
ATOM0_CH0_IRQ_MODE	0x0000D02C
ATOM0_AGC_GLB_CTRL	0x0000D040
ATOM0_AGC_ENDIS_CTRL	0x0000D044
ATOM0_AGC_ENDIS_STAT	0x0000D048
ATOM0_AGC_ACT_TB	0x0000D04C
ATOM0_AGC_OUTEN_CTRL	0x0000D050
ATOM0_AGC_OUTEN_STAT	0x0000D054
ATOM0_AGC_FUPD_CTRL	0x0000D058
ATOM0_AGC_INT_TRIG	0x0000D05C
ATOM0_CH1_RDADDR	0x0000D080
ATOM0_CH1_CTRL	0x0000D084
ATOM0_CH1_SR0	0x0000D088
ATOM0_CH1_SR1	0x0000D08C
ATOM0_CH1_CM0	0x0000D090
ATOM0_CH1_CM1	0x0000D094

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM0_CH1_CN0	0x0000D098
ATOM0_CH1_STAT	0x0000D09C
ATOM0_CH1_IRQ_NOTIFY	0x0000D0A0
ATOM0_CH1_IRQ_EN	0x0000D0A4
ATOM0_CH1_IRQ_FORCINT	0x0000D0A8
ATOM0_CH1_IRQ_MODE	0x0000D0AC
ATOM0_CH2_RDADDR	0x0000D100
ATOM0_CH2_CTRL	0x0000D104
ATOM0_CH2_SR0	0x0000D108
ATOM0_CH2_SR1	0x0000D10C
ATOM0_CH2_CM0	0x0000D110
ATOM0_CH2_CM1	0x0000D114
ATOM0_CH2_CN0	0x0000D118
ATOM0_CH2_STAT	0x0000D11C
ATOM0_CH2_IRQ_NOTIFY	0x0000D120
ATOM0_CH2_IRQ_EN	0x0000D124
ATOM0_CH2_IRQ_FORCINT	0x0000D128
ATOM0_CH2_IRQ_MODE	0x0000D12C
ATOM0_CH3_RDADDR	0x0000D180
ATOM0_CH3_CTRL	0x0000D184
ATOM0_CH3_SR0	0x0000D188
ATOM0_CH3_SR1	0x0000D18C
ATOM0_CH3_CM0	0x0000D190
ATOM0_CH3_CM1	0x0000D194
ATOM0_CH3_CN0	0x0000D198
ATOM0_CH3_STAT	0x0000D19C
ATOM0_CH3_IRQ_NOTIFY	0x0000D1A0
ATOM0_CH3_IRQ_EN	0x0000D1A4
ATOM0_CH3_IRQ_FORCINT	0x0000D1A8
ATOM0_CH3_IRQ_MODE	0x0000D1AC
ATOM0_CH4_RDADDR	0x0000D200
ATOM0_CH4_CTRL	0x0000D204
ATOM0_CH4_SR0	0x0000D208
ATOM0_CH4_SR1	0x0000D20C
ATOM0_CH4_CM0	0x0000D210
ATOM0_CH4_CM1	0x0000D214
ATOM0_CH4_CN0	0x0000D218
ATOM0_CH4_STAT	0x0000D21C
ATOM0_CH4_IRQ_NOTIFY	0x0000D220
ATOM0_CH4_IRQ_EN	0x0000D224

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM0_CH4_IRQ_FORCINT	0x0000D228
ATOM0_CH4_IRQ_MODE	0x0000D22C
ATOM0_CH5_RDADDR	0x0000D280
ATOM0_CH5_CTRL	0x0000D284
ATOM0_CH5_SR0	0x0000D288
ATOM0_CH5_SR1	0x0000D28C
ATOM0_CH5_CM0	0x0000D290
ATOM0_CH5_CM1	0x0000D294
ATOM0_CH5_CN0	0x0000D298
ATOM0_CH5_STAT	0x0000D29C
ATOM0_CH5_IRQ_NOTIFY	0x0000D2A0
ATOM0_CH5_IRQ_EN	0x0000D2A4
ATOM0_CH5_IRQ_FORCINT	0x0000D2A8
ATOM0_CH5_IRQ_MODE	0x0000D2AC
ATOM0_CH6_RDADDR	0x0000D300
ATOM0_CH6_CTRL	0x0000D304
ATOM0_CH6_SR0	0x0000D308
ATOM0_CH6_SR1	0x0000D30C
ATOM0_CH6_CM0	0x0000D310
ATOM0_CH6_CM1	0x0000D314
ATOM0_CH6_CN0	0x0000D318
ATOM0_CH6_STAT	0x0000D31C
ATOM0_CH6_IRQ_NOTIFY	0x0000D320
ATOM0_CH6_IRQ_EN	0x0000D324
ATOM0_CH6_IRQ_FORCINT	0x0000D328
ATOM0_CH6_IRQ_MODE	0x0000D32C
ATOM0_CH7_RDADDR	0x0000D380
ATOM0_CH7_CTRL	0x0000D384
ATOM0_CH7_SR0	0x0000D388
ATOM0_CH7_SR1	0x0000D38C
ATOM0_CH7_CM0	0x0000D390
ATOM0_CH7_CM1	0x0000D394
ATOM0_CH7_CN0	0x0000D398
ATOM0_CH7_STAT	0x0000D39C
ATOM0_CH7_IRQ_NOTIFY	0x0000D3A0
ATOM0_CH7_IRQ_EN	0x0000D3A4
ATOM0_CH7_IRQ_FORCINT	0x0000D3A8
ATOM0_CH7_IRQ_MODE	0x0000D3AC
ATOM1_CH0_RDADDR	0x0000D800
ATOM1_CH0_CTRL	0x0000D804

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM1_CH0_SR0	0x0000D808
ATOM1_CH0_SR1	0x0000D80C
ATOM1_CH0_CM0	0x0000D810
ATOM1_CH0_CM1	0x0000D814
ATOM1_CH0_CN0	0x0000D818
ATOM1_CH0_STAT	0x0000D81C
ATOM1_CH0_IRQ_NOTIFY	0x0000D820
ATOM1_CH0_IRQ_EN	0x0000D824
ATOM1_CH0_IRQ_FORCINT	0x0000D828
ATOM1_CH0_IRQ_MODE	0x0000D82C
ATOM1_AGC_GLB_CTRL	0x0000D840
ATOM1_AGC_ENDIS_CTRL	0x0000D844
ATOM1_AGC_ENDIS_STAT	0x0000D848
ATOM1_AGC_ACT_TB	0x0000D84C
ATOM1_AGC_OUTEN_CTRL	0x0000D850
ATOM1_AGC_OUTEN_STAT	0x0000D854
ATOM1_AGC_FUPD_CTRL	0x0000D858
ATOM1_AGC_INT_TRIG	0x0000D85C
ATOM1_CH1_RDADDR	0x0000D880
ATOM1_CH1_CTRL	0x0000D884
ATOM1_CH1_SR0	0x0000D888
ATOM1_CH1_SR1	0x0000D88C
ATOM1_CH1_CM0	0x0000D890
ATOM1_CH1_CM1	0x0000D894
ATOM1_CH1_CN0	0x0000D898
ATOM1_CH1_STAT	0x0000D89C
ATOM1_CH1_IRQ_NOTIFY	0x0000D8A0
ATOM1_CH1_IRQ_EN	0x0000D8A4
ATOM1_CH1_IRQ_FORCINT	0x0000D8A8
ATOM1_CH1_IRQ_MODE	0x0000D8AC
ATOM1_CH2_RDADDR	0x0000D900
ATOM1_CH2_CTRL	0x0000D904
ATOM1_CH2_SR0	0x0000D908
ATOM1_CH2_SR1	0x0000D90C
ATOM1_CH2_CM0	0x0000D910
ATOM1_CH2_CM1	0x0000D914
ATOM1_CH2_CN0	0x0000D918
ATOM1_CH2_STAT	0x0000D91C
ATOM1_CH2_IRQ_NOTIFY	0x0000D920
ATOM1_CH2_IRQ_EN	0x0000D924

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM1_CH2_IRQ_FORCINT	0x0000D928
ATOM1_CH2_IRQ_MODE	0x0000D92C
ATOM1_CH3_RDADDR	0x0000D980
ATOM1_CH3_CTRL	0x0000D984
ATOM1_CH3_SR0	0x0000D988
ATOM1_CH3_SR1	0x0000D98C
ATOM1_CH3_CM0	0x0000D990
ATOM1_CH3_CM1	0x0000D994
ATOM1_CH3_CN0	0x0000D998
ATOM1_CH3_STAT	0x0000D99C
ATOM1_CH3_IRQ_NOTIFY	0x0000D9A0
ATOM1_CH3_IRQ_EN	0x0000D9A4
ATOM1_CH3_IRQ_FORCINT	0x0000D9A8
ATOM1_CH3_IRQ_MODE	0x0000D9AC
ATOM1_CH4_RDADDR	0x0000DA00
ATOM1_CH4_CTRL	0x0000DA04
ATOM1_CH4_SR0	0x0000DA08
ATOM1_CH4_SR1	0x0000DA0C
ATOM1_CH4_CM0	0x0000DA10
ATOM1_CH4_CM1	0x0000DA14
ATOM1_CH4_CN0	0x0000DA18
ATOM1_CH4_STAT	0x0000DA1C
ATOM1_CH4_IRQ_NOTIFY	0x0000DA20
ATOM1_CH4_IRQ_EN	0x0000DA24
ATOM1_CH4_IRQ_FORCINT	0x0000DA28
ATOM1_CH4_IRQ_MODE	0x0000DA2C
ATOM1_CH5_RDADDR	0x0000DA80
ATOM1_CH5_CTRL	0x0000DA84
ATOM1_CH5_SR0	0x0000DA88
ATOM1_CH5_SR1	0x0000DA8C
ATOM1_CH5_CM0	0x0000DA90
ATOM1_CH5_CM1	0x0000DA94
ATOM1_CH5_CN0	0x0000DA98
ATOM1_CH5_STAT	0x0000DA9C
ATOM1_CH5_IRQ_NOTIFY	0x0000DAA0
ATOM1_CH5_IRQ_EN	0x0000DAA4
ATOM1_CH5_IRQ_FORCINT	0x0000DAA8
ATOM1_CH5_IRQ_MODE	0x0000DAAC
ATOM1_CH6_RDADDR	0x0000DB00
ATOM1_CH6_CTRL	0x0000DB04

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM1_CH6_SR0	0x0000DB08
ATOM1_CH6_SR1	0x0000DB0C
ATOM1_CH6_CM0	0x0000DB10
ATOM1_CH6_CM1	0x0000DB14
ATOM1_CH6_CN0	0x0000DB18
ATOM1_CH6_STAT	0x0000DB1C
ATOM1_CH6_IRQ_NOTIFY	0x0000DB20
ATOM1_CH6_IRQ_EN	0x0000DB24
ATOM1_CH6_IRQ_FORCINT	0x0000DB28
ATOM1_CH6_IRQ_MODE	0x0000DB2C
ATOM1_CH7_RDADDR	0x0000DB80
ATOM1_CH7_CTRL	0x0000DB84
ATOM1_CH7_SR0	0x0000DB88
ATOM1_CH7_SR1	0x0000DB8C
ATOM1_CH7_CM0	0x0000DB90
ATOM1_CH7_CM1	0x0000DB94
ATOM1_CH7_CN0	0x0000DB98
ATOM1_CH7_STAT	0x0000DB9C
ATOM1_CH7_IRQ_NOTIFY	0x0000DBA0
ATOM1_CH7_IRQ_EN	0x0000DBA4
ATOM1_CH7_IRQ_FORCINT	0x0000DBA8
ATOM1_CH7_IRQ_MODE	0x0000DBAC
ATOM2_CH0_RDADDR	0x0000E000
ATOM2_CH0_CTRL	0x0000E004
ATOM2_CH0_SR0	0x0000E008
ATOM2_CH0_SR1	0x0000E00C
ATOM2_CH0_CM0	0x0000E010
ATOM2_CH0_CM1	0x0000E014
ATOM2_CH0_CN0	0x0000E018
ATOM2_CH0_STAT	0x0000E01C
ATOM2_CH0_IRQ_NOTIFY	0x0000E020
ATOM2_CH0_IRQ_EN	0x0000E024
ATOM2_CH0_IRQ_FORCINT	0x0000E028
ATOM2_CH0_IRQ_MODE	0x0000E02C
ATOM2_AGC_GLB_CTRL	0x0000E040
ATOM2_AGC_ENDIS_CTRL	0x0000E044
ATOM2_AGC_ENDIS_STAT	0x0000E048
ATOM2_AGC_ACT_TB	0x0000E04C
ATOM2_AGC_OUTEN_CTRL	0x0000E050
ATOM2_AGC_OUTEN_STAT	0x0000E054

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM2_AGC_FUPD_CTRL	0x0000E058
ATOM2_AGC_INT_TRIG	0x0000E05C
ATOM2_CH1_RDADDR	0x0000E080
ATOM2_CH1_CTRL	0x0000E084
ATOM2_CH1_SR0	0x0000E088
ATOM2_CH1_SR1	0x0000E08C
ATOM2_CH1_CM0	0x0000E090
ATOM2_CH1_CM1	0x0000E094
ATOM2_CH1_CN0	0x0000E098
ATOM2_CH1_STAT	0x0000E09C
ATOM2_CH1_IRQ_NOTIFY	0x0000E0A0
ATOM2_CH1_IRQ_EN	0x0000E0A4
ATOM2_CH1_IRQ_FORCINT	0x0000E0A8
ATOM2_CH1_IRQ_MODE	0x0000E0AC
ATOM2_CH2_RDADDR	0x0000E100
ATOM2_CH2_CTRL	0x0000E104
ATOM2_CH2_SR0	0x0000E108
ATOM2_CH2_SR1	0x0000E10C
ATOM2_CH2_CM0	0x0000E110
ATOM2_CH2_CM1	0x0000E114
ATOM2_CH2_CN0	0x0000E118
ATOM2_CH2_STAT	0x0000E11C
ATOM2_CH2_IRQ_NOTIFY	0x0000E120
ATOM2_CH2_IRQ_EN	0x0000E124
ATOM2_CH2_IRQ_FORCINT	0x0000E128
ATOM2_CH2_IRQ_MODE	0x0000E12C
ATOM2_CH3_RDADDR	0x0000E180
ATOM2_CH3_CTRL	0x0000E184
ATOM2_CH3_SR0	0x0000E188
ATOM2_CH3_SR1	0x0000E18C
ATOM2_CH3_CM0	0x0000E190
ATOM2_CH3_CM1	0x0000E194
ATOM2_CH3_CN0	0x0000E198
ATOM2_CH3_STAT	0x0000E19C
ATOM2_CH3_IRQ_NOTIFY	0x0000E1A0
ATOM2_CH3_IRQ_EN	0x0000E1A4
ATOM2_CH3_IRQ_FORCINT	0x0000E1A8
ATOM2_CH3_IRQ_MODE	0x0000E1AC
ATOM2_CH4_RDADDR	0x0000E200
ATOM2_CH4_CTRL	0x0000E204

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM2_CH4_SR0	0x0000E208
ATOM2_CH4_SR1	0x0000E20C
ATOM2_CH4_CM0	0x0000E210
ATOM2_CH4_CM1	0x0000E214
ATOM2_CH4_CN0	0x0000E218
ATOM2_CH4_STAT	0x0000E21C
ATOM2_CH4_IRQ_NOTIFY	0x0000E220
ATOM2_CH4_IRQ_EN	0x0000E224
ATOM2_CH4_IRQ_FORCINT	0x0000E228
ATOM2_CH4_IRQ_MODE	0x0000E22C
ATOM2_CH5_RDADDR	0x0000E280
ATOM2_CH5_CTRL	0x0000E284
ATOM2_CH5_SR0	0x0000E288
ATOM2_CH5_SR1	0x0000E28C
ATOM2_CH5_CM0	0x0000E290
ATOM2_CH5_CM1	0x0000E294
ATOM2_CH5_CN0	0x0000E298
ATOM2_CH5_STAT	0x0000E29C
ATOM2_CH5_IRQ_NOTIFY	0x0000E2A0
ATOM2_CH5_IRQ_EN	0x0000E2A4
ATOM2_CH5_IRQ_FORCINT	0x0000E2A8
ATOM2_CH5_IRQ_MODE	0x0000E2AC
ATOM2_CH6_RDADDR	0x0000E300
ATOM2_CH6_CTRL	0x0000E304
ATOM2_CH6_SR0	0x0000E308
ATOM2_CH6_SR1	0x0000E30C
ATOM2_CH6_CM0	0x0000E310
ATOM2_CH6_CM1	0x0000E314
ATOM2_CH6_CN0	0x0000E318
ATOM2_CH6_STAT	0x0000E31C
ATOM2_CH6_IRQ_NOTIFY	0x0000E320
ATOM2_CH6_IRQ_EN	0x0000E324
ATOM2_CH6_IRQ_FORCINT	0x0000E328
ATOM2_CH6_IRQ_MODE	0x0000E32C
ATOM2_CH7_RDADDR	0x0000E380
ATOM2_CH7_CTRL	0x0000E384
ATOM2_CH7_SR0	0x0000E388
ATOM2_CH7_SR1	0x0000E38C
ATOM2_CH7_CM0	0x0000E390
ATOM2_CH7_CM1	0x0000E394

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM2_CH7_CN0	0x0000E398
ATOM2_CH7_STAT	0x0000E39C
ATOM2_CH7_IRQ_NOTIFY	0x0000E3A0
ATOM2_CH7_IRQ_EN	0x0000E3A4
ATOM2_CH7_IRQ_FORCINT	0x0000E3A8
ATOM2_CH7_IRQ_MODE	0x0000E3AC
ATOM3_CH0_RDADDR	0x0000E800
ATOM3_CH0_CTRL	0x0000E804
ATOM3_CH0_SR0	0x0000E808
ATOM3_CH0_SR1	0x0000E80C
ATOM3_CH0_CM0	0x0000E810
ATOM3_CH0_CM1	0x0000E814
ATOM3_CH0_CN0	0x0000E818
ATOM3_CH0_STAT	0x0000E81C
ATOM3_CH0_IRQ_NOTIFY	0x0000E820
ATOM3_CH0_IRQ_EN	0x0000E824
ATOM3_CH0_IRQ_FORCINT	0x0000E828
ATOM3_CH0_IRQ_MODE	0x0000E82C
ATOM3_AGC_GLB_CTRL	0x0000E840
ATOM3_AGC_ENDIS_CTRL	0x0000E844
ATOM3_AGC_ENDIS_STAT	0x0000E848
ATOM3_AGC_ACT_TB	0x0000E84C
ATOM3_AGC_OUTEN_CTRL	0x0000E850
ATOM3_AGC_OUTEN_STAT	0x0000E854
ATOM3_AGC_FUPD_CTRL	0x0000E858
ATOM3_AGC_INT_TRIG	0x0000E85C
ATOM3_CH1_RDADDR	0x0000E880
ATOM3_CH1_CTRL	0x0000E884
ATOM3_CH1_SR0	0x0000E888
ATOM3_CH1_SR1	0x0000E88C
ATOM3_CH1_CM0	0x0000E890
ATOM3_CH1_CM1	0x0000E894
ATOM3_CH1_CN0	0x0000E898
ATOM3_CH1_STAT	0x0000E89C
ATOM3_CH1_IRQ_NOTIFY	0x0000E8A0
ATOM3_CH1_IRQ_EN	0x0000E8A4
ATOM3_CH1_IRQ_FORCINT	0x0000E8A8
ATOM3_CH1_IRQ_MODE	0x0000E8AC
ATOM3_CH2_RDADDR	0x0000E900
ATOM3_CH2_CTRL	0x0000E904

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM3_CH2_SR0	0x0000E908
ATOM3_CH2_SR1	0x0000E90C
ATOM3_CH2_CM0	0x0000E910
ATOM3_CH2_CM1	0x0000E914
ATOM3_CH2_CN0	0x0000E918
ATOM3_CH2_STAT	0x0000E91C
ATOM3_CH2_IRQ_NOTIFY	0x0000E920
ATOM3_CH2_IRQ_EN	0x0000E924
ATOM3_CH2_IRQ_FORCINT	0x0000E928
ATOM3_CH2_IRQ_MODE	0x0000E92C
ATOM3_CH3_RDADDR	0x0000E980
ATOM3_CH3_CTRL	0x0000E984
ATOM3_CH3_SR0	0x0000E988
ATOM3_CH3_SR1	0x0000E98C
ATOM3_CH3_CM0	0x0000E990
ATOM3_CH3_CM1	0x0000E994
ATOM3_CH3_CN0	0x0000E998
ATOM3_CH3_STAT	0x0000E99C
ATOM3_CH3_IRQ_NOTIFY	0x0000E9A0
ATOM3_CH3_IRQ_EN	0x0000E9A4
ATOM3_CH3_IRQ_FORCINT	0x0000E9A8
ATOM3_CH3_IRQ_MODE	0x0000E9AC
ATOM3_CH4_RDADDR	0x0000EA00
ATOM3_CH4_CTRL	0x0000EA04
ATOM3_CH4_SR0	0x0000EA08
ATOM3_CH4_SR1	0x0000EA0C
ATOM3_CH4_CM0	0x0000EA10
ATOM3_CH4_CM1	0x0000EA14
ATOM3_CH4_CN0	0x0000EA18
ATOM3_CH4_STAT	0x0000EA1C
ATOM3_CH4_IRQ_NOTIFY	0x0000EA20
ATOM3_CH4_IRQ_EN	0x0000EA24
ATOM3_CH4_IRQ_FORCINT	0x0000EA28
ATOM3_CH4_IRQ_MODE	0x0000EA2C
ATOM3_CH5_RDADDR	0x0000EA80
ATOM3_CH5_CTRL	0x0000EA84
ATOM3_CH5_SR0	0x0000EA88
ATOM3_CH5_SR1	0x0000EA8C
ATOM3_CH5_CM0	0x0000EA90
ATOM3_CH5_CM1	0x0000EA94

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM3_CH5_CN0	0x0000EA98
ATOM3_CH5_STAT	0x0000EA9C
ATOM3_CH5_IRQ_NOTIFY	0x0000EAA0
ATOM3_CH5_IRQ_EN	0x0000EAA4
ATOM3_CH5_IRQ_FORCINT	0x0000EAA8
ATOM3_CH5_IRQ_MODE	0x0000EAAC
ATOM3_CH6_RDADDR	0x0000EB00
ATOM3_CH6_CTRL	0x0000EB04
ATOM3_CH6_SR0	0x0000EB08
ATOM3_CH6_SR1	0x0000EB0C
ATOM3_CH6_CM0	0x0000EB10
ATOM3_CH6_CM1	0x0000EB14
ATOM3_CH6_CN0	0x0000EB18
ATOM3_CH6_STAT	0x0000EB1C
ATOM3_CH6_IRQ_NOTIFY	0x0000EB20
ATOM3_CH6_IRQ_EN	0x0000EB24
ATOM3_CH6_IRQ_FORCINT	0x0000EB28
ATOM3_CH6_IRQ_MODE	0x0000EB2C
ATOM3_CH7_RDADDR	0x0000EB80
ATOM3_CH7_CTRL	0x0000EB84
ATOM3_CH7_SR0	0x0000EB88
ATOM3_CH7_SR1	0x0000EB8C
ATOM3_CH7_CM0	0x0000EB90
ATOM3_CH7_CM1	0x0000EB94
ATOM3_CH7_CN0	0x0000EB98
ATOM3_CH7_STAT	0x0000EB9C
ATOM3_CH7_IRQ_NOTIFY	0x0000EBA0
ATOM3_CH7_IRQ_EN	0x0000EBA4
ATOM3_CH7_IRQ_FORCINT	0x0000EBA8
ATOM3_CH7_IRQ_MODE	0x0000EBAC
ATOM4_CH0_RDADDR	0x0000F000
ATOM4_CH0_CTRL	0x0000F004
ATOM4_CH0_SR0	0x0000F008
ATOM4_CH0_SR1	0x0000F00C
ATOM4_CH0_CM0	0x0000F010
ATOM4_CH0_CM1	0x0000F014
ATOM4_CH0_CN0	0x0000F018
ATOM4_CH0_STAT	0x0000F01C
ATOM4_CH0_IRQ_NOTIFY	0x0000F020
ATOM4_CH0_IRQ_EN	0x0000F024

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM4_CH0_IRQ_FORCINT	0x0000F028
ATOM4_CH0_IRQ_MODE	0x0000F02C
ATOM4_AGC_GLB_CTRL	0x0000F040
ATOM4_AGC_ENDIS_CTRL	0x0000F044
ATOM4_AGC_ENDIS_STAT	0x0000F048
ATOM4_AGC_ACT_TB	0x0000F04C
ATOM4_AGC_OUTEN_CTRL	0x0000F050
ATOM4_AGC_OUTEN_STAT	0x0000F054
ATOM4_AGC_FUPD_CTRL	0x0000F058
ATOM4_AGC_INT_TRIG	0x0000F05C
ATOM4_CH1_RDADDR	0x0000F080
ATOM4_CH1_CTRL	0x0000F084
ATOM4_CH1_SR0	0x0000F088
ATOM4_CH1_SR1	0x0000F08C
ATOM4_CH1_CM0	0x0000F090
ATOM4_CH1_CM1	0x0000F094
ATOM4_CH1_CN0	0x0000F098
ATOM4_CH1_STAT	0x0000F09C
ATOM4_CH1_IRQ_NOTIFY	0x0000F0A0
ATOM4_CH1_IRQ_EN	0x0000F0A4
ATOM4_CH1_IRQ_FORCINT	0x0000F0A8
ATOM4_CH1_IRQ_MODE	0x0000F0AC
ATOM4_CH2_RDADDR	0x0000F100
ATOM4_CH2_CTRL	0x0000F104
ATOM4_CH2_SR0	0x0000F108
ATOM4_CH2_SR1	0x0000F10C
ATOM4_CH2_CM0	0x0000F110
ATOM4_CH2_CM1	0x0000F114
ATOM4_CH2_CN0	0x0000F118
ATOM4_CH2_STAT	0x0000F11C
ATOM4_CH2_IRQ_NOTIFY	0x0000F120
ATOM4_CH2_IRQ_EN	0x0000F124
ATOM4_CH2_IRQ_FORCINT	0x0000F128
ATOM4_CH2_IRQ_MODE	0x0000F12C
ATOM4_CH3_RDADDR	0x0000F180
ATOM4_CH3_CTRL	0x0000F184
ATOM4_CH3_SR0	0x0000F188
ATOM4_CH3_SR1	0x0000F18C
ATOM4_CH3_CM0	0x0000F190
ATOM4_CH3_CM1	0x0000F194

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM4_CH3_CN0	0x0000F198
ATOM4_CH3_STAT	0x0000F19C
ATOM4_CH3_IRQ_NOTIFY	0x0000F1A0
ATOM4_CH3_IRQ_EN	0x0000F1A4
ATOM4_CH3_IRQ_FORCINT	0x0000F1A8
ATOM4_CH3_IRQ_MODE	0x0000F1AC
ATOM4_CH4_RDADDR	0x0000F200
ATOM4_CH4_CTRL	0x0000F204
ATOM4_CH4_SR0	0x0000F208
ATOM4_CH4_SR1	0x0000F20C
ATOM4_CH4_CM0	0x0000F210
ATOM4_CH4_CM1	0x0000F214
ATOM4_CH4_CN0	0x0000F218
ATOM4_CH4_STAT	0x0000F21C
ATOM4_CH4_IRQ_NOTIFY	0x0000F220
ATOM4_CH4_IRQ_EN	0x0000F224
ATOM4_CH4_IRQ_FORCINT	0x0000F228
ATOM4_CH4_IRQ_MODE	0x0000F22C
ATOM4_CH5_RDADDR	0x0000F280
ATOM4_CH5_CTRL	0x0000F284
ATOM4_CH5_SR0	0x0000F288
ATOM4_CH5_SR1	0x0000F28C
ATOM4_CH5_CM0	0x0000F290
ATOM4_CH5_CM1	0x0000F294
ATOM4_CH5_CN0	0x0000F298
ATOM4_CH5_STAT	0x0000F29C
ATOM4_CH5_IRQ_NOTIFY	0x0000F2A0
ATOM4_CH5_IRQ_EN	0x0000F2A4
ATOM4_CH5_IRQ_FORCINT	0x0000F2A8
ATOM4_CH5_IRQ_MODE	0x0000F2AC
ATOM4_CH6_RDADDR	0x0000F300
ATOM4_CH6_CTRL	0x0000F304
ATOM4_CH6_SR0	0x0000F308
ATOM4_CH6_SR1	0x0000F30C
ATOM4_CH6_CM0	0x0000F310
ATOM4_CH6_CM1	0x0000F314
ATOM4_CH6_CN0	0x0000F318
ATOM4_CH6_STAT	0x0000F31C
ATOM4_CH6_IRQ_NOTIFY	0x0000F320
ATOM4_CH6_IRQ_EN	0x0000F324

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM4_CH6_IRQ_FORCINT	0x0000F328
ATOM4_CH6_IRQ_MODE	0x0000F32C
ATOM4_CH7_RDADDR	0x0000F380
ATOM4_CH7_CTRL	0x0000F384
ATOM4_CH7_SR0	0x0000F388
ATOM4_CH7_SR1	0x0000F38C
ATOM4_CH7_CM0	0x0000F390
ATOM4_CH7_CM1	0x0000F394
ATOM4_CH7_CN0	0x0000F398
ATOM4_CH7_STAT	0x0000F39C
ATOM4_CH7_IRQ_NOTIFY	0x0000F3A0
ATOM4_CH7_IRQ_EN	0x0000F3A4
ATOM4_CH7_IRQ_FORCINT	0x0000F3A8
ATOM4_CH7_IRQ_MODE	0x0000F3AC
ATOM5_CH0_RDADDR	0x0000F800
ATOM5_CH0_CTRL	0x0000F804
ATOM5_CH0_SR0	0x0000F808
ATOM5_CH0_SR1	0x0000F80C
ATOM5_CH0_CM0	0x0000F810
ATOM5_CH0_CM1	0x0000F814
ATOM5_CH0_CN0	0x0000F818
ATOM5_CH0_STAT	0x0000F81C
ATOM5_CH0_IRQ_NOTIFY	0x0000F820
ATOM5_CH0_IRQ_EN	0x0000F824
ATOM5_CH0_IRQ_FORCINT	0x0000F828
ATOM5_CH0_IRQ_MODE	0x0000F82C
ATOM5_AGC_GLB_CTRL	0x0000F840
ATOM5_AGC_ENDIS_CTRL	0x0000F844
ATOM5_AGC_ENDIS_STAT	0x0000F848
ATOM5_AGC_ACT_TB	0x0000F84C
ATOM5_AGC_OUTEN_CTRL	0x0000F850
ATOM5_AGC_OUTEN_STAT	0x0000F854
ATOM5_AGC_FUPD_CTRL	0x0000F858
ATOM5_AGC_INT_TRIG	0x0000F85C
ATOM5_CH1_RDADDR	0x0000F880
ATOM5_CH1_CTRL	0x0000F884
ATOM5_CH1_SR0	0x0000F888
ATOM5_CH1_SR1	0x0000F88C
ATOM5_CH1_CM0	0x0000F890
ATOM5_CH1_CM1	0x0000F894

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM5_CH1_CN0	0x0000F898
ATOM5_CH1_STAT	0x0000F89C
ATOM5_CH1_IRQ_NOTIFY	0x0000F8A0
ATOM5_CH1_IRQ_EN	0x0000F8A4
ATOM5_CH1_IRQ_FORCINT	0x0000F8A8
ATOM5_CH1_IRQ_MODE	0x0000F8AC
ATOM5_CH2_RDADDR	0x0000F900
ATOM5_CH2_CTRL	0x0000F904
ATOM5_CH2_SR0	0x0000F908
ATOM5_CH2_SR1	0x0000F90C
ATOM5_CH2_CM0	0x0000F910
ATOM5_CH2_CM1	0x0000F914
ATOM5_CH2_CN0	0x0000F918
ATOM5_CH2_STAT	0x0000F91C
ATOM5_CH2_IRQ_NOTIFY	0x0000F920
ATOM5_CH2_IRQ_EN	0x0000F924
ATOM5_CH2_IRQ_FORCINT	0x0000F928
ATOM5_CH2_IRQ_MODE	0x0000F92C
ATOM5_CH3_RDADDR	0x0000F980
ATOM5_CH3_CTRL	0x0000F984
ATOM5_CH3_SR0	0x0000F988
ATOM5_CH3_SR1	0x0000F98C
ATOM5_CH3_CM0	0x0000F990
ATOM5_CH3_CM1	0x0000F994
ATOM5_CH3_CN0	0x0000F998
ATOM5_CH3_STAT	0x0000F99C
ATOM5_CH3_IRQ_NOTIFY	0x0000F9A0
ATOM5_CH3_IRQ_EN	0x0000F9A4
ATOM5_CH3_IRQ_FORCINT	0x0000F9A8
ATOM5_CH3_IRQ_MODE	0x0000F9AC
ATOM5_CH4_RDADDR	0x0000FA00
ATOM5_CH4_CTRL	0x0000FA04
ATOM5_CH4_SR0	0x0000FA08
ATOM5_CH4_SR1	0x0000FA0C
ATOM5_CH4_CM0	0x0000FA10
ATOM5_CH4_CM1	0x0000FA14
ATOM5_CH4_CN0	0x0000FA18
ATOM5_CH4_STAT	0x0000FA1C
ATOM5_CH4_IRQ_NOTIFY	0x0000FA20
ATOM5_CH4_IRQ_EN	0x0000FA24

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM5_CH4_IRQ_FORCINT	0x0000FA28
ATOM5_CH4_IRQ_MODE	0x0000FA2C
ATOM5_CH5_RDADDR	0x0000FA80
ATOM5_CH5_CTRL	0x0000FA84
ATOM5_CH5_SR0	0x0000FA88
ATOM5_CH5_SR1	0x0000FA8C
ATOM5_CH5_CM0	0x0000FA90
ATOM5_CH5_CM1	0x0000FA94
ATOM5_CH5_CN0	0x0000FA98
ATOM5_CH5_STAT	0x0000FA9C
ATOM5_CH5_IRQ_NOTIFY	0x0000FAA0
ATOM5_CH5_IRQ_EN	0x0000FAA4
ATOM5_CH5_IRQ_FORCINT	0x0000FAA8
ATOM5_CH5_IRQ_MODE	0x0000FAAC
ATOM5_CH6_RDADDR	0x0000FB00
ATOM5_CH6_CTRL	0x0000FB04
ATOM5_CH6_SR0	0x0000FB08
ATOM5_CH6_SR1	0x0000FB0C
ATOM5_CH6_CM0	0x0000FB10
ATOM5_CH6_CM1	0x0000FB14
ATOM5_CH6_CN0	0x0000FB18
ATOM5_CH6_STAT	0x0000FB1C
ATOM5_CH6_IRQ_NOTIFY	0x0000FB20
ATOM5_CH6_IRQ_EN	0x0000FB24
ATOM5_CH6_IRQ_FORCINT	0x0000FB28
ATOM5_CH6_IRQ_MODE	0x0000FB2C
ATOM5_CH7_RDADDR	0x0000FB80
ATOM5_CH7_CTRL	0x0000FB84
ATOM5_CH7_SR0	0x0000FB88
ATOM5_CH7_SR1	0x0000FB8C
ATOM5_CH7_CM0	0x0000FB90
ATOM5_CH7_CM1	0x0000FB94
ATOM5_CH7_CN0	0x0000FB98
ATOM5_CH7_STAT	0x0000FB9C
ATOM5_CH7_IRQ_NOTIFY	0x0000FBA0
ATOM5_CH7_IRQ_EN	0x0000FBA4
ATOM5_CH7_IRQ_FORCINT	0x0000FBA8
ATOM5_CH7_IRQ_MODE	0x0000FBAC
ATOM6_CH0_RDADDR	0x00010000
ATOM6_CH0_CTRL	0x00010004

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM6_CH0_SR0	0x00010008
ATOM6_CH0_SR1	0x0001000C
ATOM6_CH0_CM0	0x00010010
ATOM6_CH0_CM1	0x00010014
ATOM6_CH0_CN0	0x00010018
ATOM6_CH0_STAT	0x0001001C
ATOM6_CH0_IRQ_NOTIFY	0x00010020
ATOM6_CH0_IRQ_EN	0x00010024
ATOM6_CH0_IRQ_FORCINT	0x00010028
ATOM6_CH0_IRQ_MODE	0x0001002C
ATOM6_AGC_GLB_CTRL	0x00010040
ATOM6_AGC_ENDIS_CTRL	0x00010044
ATOM6_AGC_ENDIS_STAT	0x00010048
ATOM6_AGC_ACT_TB	0x0001004C
ATOM6_AGC_OUTEN_CTRL	0x00010050
ATOM6_AGC_OUTEN_STAT	0x00010054
ATOM6_AGC_FUPD_CTRL	0x00010058
ATOM6_AGC_INT_TRIG	0x0001005C
ATOM6_CH1_RDADDR	0x00010080
ATOM6_CH1_CTRL	0x00010084
ATOM6_CH1_SR0	0x00010088
ATOM6_CH1_SR1	0x0001008C
ATOM6_CH1_CM0	0x00010090
ATOM6_CH1_CM1	0x00010094
ATOM6_CH1_CN0	0x00010098
ATOM6_CH1_STAT	0x0001009C
ATOM6_CH1_IRQ_NOTIFY	0x000100A0
ATOM6_CH1_IRQ_EN	0x000100A4
ATOM6_CH1_IRQ_FORCINT	0x000100A8
ATOM6_CH1_IRQ_MODE	0x000100AC
ATOM6_CH2_RDADDR	0x00010100
ATOM6_CH2_CTRL	0x00010104
ATOM6_CH2_SR0	0x00010108
ATOM6_CH2_SR1	0x0001010C
ATOM6_CH2_CM0	0x00010110
ATOM6_CH2_CM1	0x00010114
ATOM6_CH2_CN0	0x00010118
ATOM6_CH2_STAT	0x0001011C
ATOM6_CH2_IRQ_NOTIFY	0x00010120
ATOM6_CH2_IRQ_EN	0x00010124

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM6_CH2_IRQ_FORCINT	0x00010128
ATOM6_CH2_IRQ_MODE	0x0001012C
ATOM6_CH3_RDADDR	0x00010180
ATOM6_CH3_CTRL	0x00010184
ATOM6_CH3_SR0	0x00010188
ATOM6_CH3_SR1	0x0001018C
ATOM6_CH3_CM0	0x00010190
ATOM6_CH3_CM1	0x00010194
ATOM6_CH3_CN0	0x00010198
ATOM6_CH3_STAT	0x0001019C
ATOM6_CH3_IRQ_NOTIFY	0x000101A0
ATOM6_CH3_IRQ_EN	0x000101A4
ATOM6_CH3_IRQ_FORCINT	0x000101A8
ATOM6_CH3_IRQ_MODE	0x000101AC
ATOM6_CH4_RDADDR	0x00010200
ATOM6_CH4_CTRL	0x00010204
ATOM6_CH4_SR0	0x00010208
ATOM6_CH4_SR1	0x0001020C
ATOM6_CH4_CM0	0x00010210
ATOM6_CH4_CM1	0x00010214
ATOM6_CH4_CN0	0x00010218
ATOM6_CH4_STAT	0x0001021C
ATOM6_CH4_IRQ_NOTIFY	0x00010220
ATOM6_CH4_IRQ_EN	0x00010224
ATOM6_CH4_IRQ_FORCINT	0x00010228
ATOM6_CH4_IRQ_MODE	0x0001022C
ATOM6_CH5_RDADDR	0x00010280
ATOM6_CH5_CTRL	0x00010284
ATOM6_CH5_SR0	0x00010288
ATOM6_CH5_SR1	0x0001028C
ATOM6_CH5_CM0	0x00010290
ATOM6_CH5_CM1	0x00010294
ATOM6_CH5_CN0	0x00010298
ATOM6_CH5_STAT	0x0001029C
ATOM6_CH5_IRQ_NOTIFY	0x000102A0
ATOM6_CH5_IRQ_EN	0x000102A4
ATOM6_CH5_IRQ_FORCINT	0x000102A8
ATOM6_CH5_IRQ_MODE	0x000102AC
ATOM6_CH6_RDADDR	0x00010300
ATOM6_CH6_CTRL	0x00010304

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM6_CH6_SR0	0x00010308
ATOM6_CH6_SR1	0x0001030C
ATOM6_CH6_CM0	0x00010310
ATOM6_CH6_CM1	0x00010314
ATOM6_CH6_CN0	0x00010318
ATOM6_CH6_STAT	0x0001031C
ATOM6_CH6_IRQ_NOTIFY	0x00010320
ATOM6_CH6_IRQ_EN	0x00010324
ATOM6_CH6_IRQ_FORCINT	0x00010328
ATOM6_CH6_IRQ_MODE	0x0001032C
ATOM6_CH7_RDADDR	0x00010380
ATOM6_CH7_CTRL	0x00010384
ATOM6_CH7_SR0	0x00010388
ATOM6_CH7_SR1	0x0001038C
ATOM6_CH7_CM0	0x00010390
ATOM6_CH7_CM1	0x00010394
ATOM6_CH7_CN0	0x00010398
ATOM6_CH7_STAT	0x0001039C
ATOM6_CH7_IRQ_NOTIFY	0x000103A0
ATOM6_CH7_IRQ_EN	0x000103A4
ATOM6_CH7_IRQ_FORCINT	0x000103A8
ATOM6_CH7_IRQ_MODE	0x000103AC
ATOM7_CH0_RDADDR	0x00010800
ATOM7_CH0_CTRL	0x00010804
ATOM7_CH0_SR0	0x00010808
ATOM7_CH0_SR1	0x0001080C
ATOM7_CH0_CM0	0x00010810
ATOM7_CH0_CM1	0x00010814
ATOM7_CH0_CN0	0x00010818
ATOM7_CH0_STAT	0x0001081C
ATOM7_CH0_IRQ_NOTIFY	0x00010820
ATOM7_CH0_IRQ_EN	0x00010824
ATOM7_CH0_IRQ_FORCINT	0x00010828
ATOM7_CH0_IRQ_MODE	0x0001082C
ATOM7_AGC_GLB_CTRL	0x00010840
ATOM7_AGC_ENDIS_CTRL	0x00010844
ATOM7_AGC_ENDIS_STAT	0x00010848
ATOM7_AGC_ACT_TB	0x0001084C
ATOM7_AGC_OUTEN_CTRL	0x00010850
ATOM7_AGC_OUTEN_STAT	0x00010854

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM7_AGC_FUPD_CTRL	0x00010858
ATOM7_AGC_INT_TRIG	0x0001085C
ATOM7_CH1_RDADDR	0x00010880
ATOM7_CH1_CTRL	0x00010884
ATOM7_CH1_SR0	0x00010888
ATOM7_CH1_SR1	0x0001088C
ATOM7_CH1_CM0	0x00010890
ATOM7_CH1_CM1	0x00010894
ATOM7_CH1_CN0	0x00010898
ATOM7_CH1_STAT	0x0001089C
ATOM7_CH1_IRQ_NOTIFY	0x000108A0
ATOM7_CH1_IRQ_EN	0x000108A4
ATOM7_CH1_IRQ_FORCINT	0x000108A8
ATOM7_CH1_IRQ_MODE	0x000108AC
ATOM7_CH2_RDADDR	0x00010900
ATOM7_CH2_CTRL	0x00010904
ATOM7_CH2_SR0	0x00010908
ATOM7_CH2_SR1	0x0001090C
ATOM7_CH2_CM0	0x00010910
ATOM7_CH2_CM1	0x00010914
ATOM7_CH2_CN0	0x00010918
ATOM7_CH2_STAT	0x0001091C
ATOM7_CH2_IRQ_NOTIFY	0x00010920
ATOM7_CH2_IRQ_EN	0x00010924
ATOM7_CH2_IRQ_FORCINT	0x00010928
ATOM7_CH2_IRQ_MODE	0x0001092C
ATOM7_CH3_RDADDR	0x00010980
ATOM7_CH3_CTRL	0x00010984
ATOM7_CH3_SR0	0x00010988
ATOM7_CH3_SR1	0x0001098C
ATOM7_CH3_CM0	0x00010990
ATOM7_CH3_CM1	0x00010994
ATOM7_CH3_CN0	0x00010998
ATOM7_CH3_STAT	0x0001099C
ATOM7_CH3_IRQ_NOTIFY	0x000109A0
ATOM7_CH3_IRQ_EN	0x000109A4
ATOM7_CH3_IRQ_FORCINT	0x000109A8
ATOM7_CH3_IRQ_MODE	0x000109AC
ATOM7_CH4_RDADDR	0x00010A00
ATOM7_CH4_CTRL	0x00010A04

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM7_CH4_SR0	0x00010A08
ATOM7_CH4_SR1	0x00010A0C
ATOM7_CH4_CM0	0x00010A10
ATOM7_CH4_CM1	0x00010A14
ATOM7_CH4_CN0	0x00010A18
ATOM7_CH4_STAT	0x00010A1C
ATOM7_CH4_IRQ_NOTIFY	0x00010A20
ATOM7_CH4_IRQ_EN	0x00010A24
ATOM7_CH4_IRQ_FORCINT	0x00010A28
ATOM7_CH4_IRQ_MODE	0x00010A2C
ATOM7_CH5_RDADDR	0x00010A80
ATOM7_CH5_CTRL	0x00010A84
ATOM7_CH5_SR0	0x00010A88
ATOM7_CH5_SR1	0x00010A8C
ATOM7_CH5_CM0	0x00010A90
ATOM7_CH5_CM1	0x00010A94
ATOM7_CH5_CN0	0x00010A98
ATOM7_CH5_STAT	0x00010A9C
ATOM7_CH5_IRQ_NOTIFY	0x00010AA0
ATOM7_CH5_IRQ_EN	0x00010AA4
ATOM7_CH5_IRQ_FORCINT	0x00010AA8
ATOM7_CH5_IRQ_MODE	0x00010AAC
ATOM7_CH6_RDADDR	0x00010B00
ATOM7_CH6_CTRL	0x00010B04
ATOM7_CH6_SR0	0x00010B08
ATOM7_CH6_SR1	0x00010B0C
ATOM7_CH6_CM0	0x00010B10
ATOM7_CH6_CM1	0x00010B14
ATOM7_CH6_CN0	0x00010B18
ATOM7_CH6_STAT	0x00010B1C
ATOM7_CH6_IRQ_NOTIFY	0x00010B20
ATOM7_CH6_IRQ_EN	0x00010B24
ATOM7_CH6_IRQ_FORCINT	0x00010B28
ATOM7_CH6_IRQ_MODE	0x00010B2C
ATOM7_CH7_RDADDR	0x00010B80
ATOM7_CH7_CTRL	0x00010B84
ATOM7_CH7_SR0	0x00010B88
ATOM7_CH7_SR1	0x00010B8C
ATOM7_CH7_CM0	0x00010B90
ATOM7_CH7_CM1	0x00010B94

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM7_CH7_CN0	0x00010B98
ATOM7_CH7_STAT	0x00010B9C
ATOM7_CH7_IRQ_NOTIFY	0x00010BA0
ATOM7_CH7_IRQ_EN	0x00010BA4
ATOM7_CH7_IRQ_FORCINT	0x00010BA8
ATOM7_CH7_IRQ_MODE	0x00010BAC
ATOM8_CH0_RDADDR	0x00011000
ATOM8_CH0_CTRL	0x00011004
ATOM8_CH0_SR0	0x00011008
ATOM8_CH0_SR1	0x0001100C
ATOM8_CH0_CM0	0x00011010
ATOM8_CH0_CM1	0x00011014
ATOM8_CH0_CN0	0x00011018
ATOM8_CH0_STAT	0x0001101C
ATOM8_CH0_IRQ_NOTIFY	0x00011020
ATOM8_CH0_IRQ_EN	0x00011024
ATOM8_CH0_IRQ_FORCINT	0x00011028
ATOM8_CH0_IRQ_MODE	0x0001102C
ATOM8_AGC_GLB_CTRL	0x00011040
ATOM8_AGC_ENDIS_CTRL	0x00011044
ATOM8_AGC_ENDIS_STAT	0x00011048
ATOM8_AGC_ACT_TB	0x0001104C
ATOM8_AGC_OUTEN_CTRL	0x00011050
ATOM8_AGC_OUTEN_STAT	0x00011054
ATOM8_AGC_FUPD_CTRL	0x00011058
ATOM8_AGC_INT_TRIG	0x0001105C
ATOM8_CH1_RDADDR	0x00011080
ATOM8_CH1_CTRL	0x00011084
ATOM8_CH1_SR0	0x00011088
ATOM8_CH1_SR1	0x0001108C
ATOM8_CH1_CM0	0x00011090
ATOM8_CH1_CM1	0x00011094
ATOM8_CH1_CN0	0x00011098
ATOM8_CH1_STAT	0x0001109C
ATOM8_CH1_IRQ_NOTIFY	0x000110A0
ATOM8_CH1_IRQ_EN	0x000110A4
ATOM8_CH1_IRQ_FORCINT	0x000110A8
ATOM8_CH1_IRQ_MODE	0x000110AC
ATOM8_CH2_RDADDR	0x00011100
ATOM8_CH2_CTRL	0x00011104

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM8_CH2_SR0	0x00011108
ATOM8_CH2_SR1	0x0001110C
ATOM8_CH2_CM0	0x00011110
ATOM8_CH2_CM1	0x00011114
ATOM8_CH2_CN0	0x00011118
ATOM8_CH2_STAT	0x0001111C
ATOM8_CH2_IRQ_NOTIFY	0x00011120
ATOM8_CH2_IRQ_EN	0x00011124
ATOM8_CH2_IRQ_FORCINT	0x00011128
ATOM8_CH2_IRQ_MODE	0x0001112C
ATOM8_CH3_RDADDR	0x00011180
ATOM8_CH3_CTRL	0x00011184
ATOM8_CH3_SR0	0x00011188
ATOM8_CH3_SR1	0x0001118C
ATOM8_CH3_CM0	0x00011190
ATOM8_CH3_CM1	0x00011194
ATOM8_CH3_CN0	0x00011198
ATOM8_CH3_STAT	0x0001119C
ATOM8_CH3_IRQ_NOTIFY	0x000111A0
ATOM8_CH3_IRQ_EN	0x000111A4
ATOM8_CH3_IRQ_FORCINT	0x000111A8
ATOM8_CH3_IRQ_MODE	0x000111AC
ATOM8_CH4_RDADDR	0x00011200
ATOM8_CH4_CTRL	0x00011204
ATOM8_CH4_SR0	0x00011208
ATOM8_CH4_SR1	0x0001120C
ATOM8_CH4_CM0	0x00011210
ATOM8_CH4_CM1	0x00011214
ATOM8_CH4_CN0	0x00011218
ATOM8_CH4_STAT	0x0001121C
ATOM8_CH4_IRQ_NOTIFY	0x00011220
ATOM8_CH4_IRQ_EN	0x00011224
ATOM8_CH4_IRQ_FORCINT	0x00011228
ATOM8_CH4_IRQ_MODE	0x0001122C
ATOM8_CH5_RDADDR	0x00011280
ATOM8_CH5_CTRL	0x00011284
ATOM8_CH5_SR0	0x00011288
ATOM8_CH5_SR1	0x0001128C
ATOM8_CH5_CM0	0x00011290
ATOM8_CH5_CM1	0x00011294

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

ATOM8_CH5_CN0	0x00011298
ATOM8_CH5_STAT	0x0001129C
ATOM8_CH5_IRQ_NOTIFY	0x000112A0
ATOM8_CH5_IRQ_EN	0x000112A4
ATOM8_CH5_IRQ_FORCINT	0x000112A8
ATOM8_CH5_IRQ_MODE	0x000112AC
ATOM8_CH6_RDADDR	0x00011300
ATOM8_CH6_CTRL	0x00011304
ATOM8_CH6_SR0	0x00011308
ATOM8_CH6_SR1	0x0001130C
ATOM8_CH6_CM0	0x00011310
ATOM8_CH6_CM1	0x00011314
ATOM8_CH6_CN0	0x00011318
ATOM8_CH6_STAT	0x0001131C
ATOM8_CH6_IRQ_NOTIFY	0x00011320
ATOM8_CH6_IRQ_EN	0x00011324
ATOM8_CH6_IRQ_FORCINT	0x00011328
ATOM8_CH6_IRQ_MODE	0x0001132C
ATOM8_CH7_RDADDR	0x00011380
ATOM8_CH7_CTRL	0x00011384
ATOM8_CH7_SR0	0x00011388
ATOM8_CH7_SR1	0x0001138C
ATOM8_CH7_CM0	0x00011390
ATOM8_CH7_CM1	0x00011394
ATOM8_CH7_CN0	0x00011398
ATOM8_CH7_STAT	0x0001139C
ATOM8_CH7_IRQ_NOTIFY	0x000113A0
ATOM8_CH7_IRQ_EN	0x000113A4
ATOM8_CH7_IRQ_FORCINT	0x000113A8
ATOM8_CH7_IRQ_MODE	0x000113AC
F2A0_CH0_ARU_RD_FIFO	0x00018000
F2A0_CH1_ARU_RD_FIFO	0x00018004
F2A0_CH2_ARU_RD_FIFO	0x00018008
F2A0_CH3_ARU_RD_FIFO	0x0001800C
F2A0_CH4_ARU_RD_FIFO	0x00018010
F2A0_CH5_ARU_RD_FIFO	0x00018014
F2A0_CH6_ARU_RD_FIFO	0x00018018
F2A0_CH7_ARU_RD_FIFO	0x0001801C
F2A0_CH0_STR_CFG	0x00018020
F2A0_CH1_STR_CFG	0x00018024

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

F2A0_CH2_STR_CFG	0x00018028
F2A0_CH3_STR_CFG	0x0001802C
F2A0_CH4_STR_CFG	0x00018030
F2A0_CH5_STR_CFG	0x00018034
F2A0_CH6_STR_CFG	0x00018038
F2A0_CH7_STR_CFG	0x0001803C
F2A0_ENABLE	0x00018040
AFD0_CH0_BUFFACC	0x00018080
AFD0_CH1_BUFFACC	0x00018090
AFD0_CH2_BUFFACC	0x000180A0
AFD0_CH3_BUFFACC	0x000180B0
AFD0_CH4_BUFFACC	0x000180C0
AFD0_CH5_BUFFACC	0x000180D0
AFD0_CH6_BUFFACC	0x000180E0
AFD0_CH7_BUFFACC	0x000180F0
FIFO0_CH0_CTRL	0x00018400
FIFO0_CH0_END_ADDR	0x00018404
FIFO0_CH0_START_ADDR	0x00018408
FIFO0_CH0_UPPER_WM	0x0001840C
FIFO0_CH0_LOWER_WM	0x00018410
FIFO0_CH0_STATUS	0x00018414
FIFO0_CH0_FILL_LEVEL	0x00018418
FIFO0_CH0_WR_PTR	0x0001841C
FIFO0_CH0_RD_PTR	0x00018420
FIFO0_CH0_IRQ_NOTIFY	0x00018424
FIFO0_CH0_IRQ_EN	0x00018428
FIFO0_CH0_IRQ_FORCINT	0x0001842C
FIFO0_CH0_IRQ_MODE	0x00018430
FIFO0_CH0_EIRQ_EN	0x00018434
FIFO0_CH1_CTRL	0x00018440
FIFO0_CH1_END_ADDR	0x00018444
FIFO0_CH1_START_ADDR	0x00018448
FIFO0_CH1_UPPER_WM	0x0001844C
FIFO0_CH1_LOWER_WM	0x00018450
FIFO0_CH1_STATUS	0x00018454
FIFO0_CH1_FILL_LEVEL	0x00018458
FIFO0_CH1_WR_PTR	0x0001845C
FIFO0_CH1_RD_PTR	0x00018460
FIFO0_CH1_IRQ_NOTIFY	0x00018464
FIFO0_CH1_IRQ_EN	0x00018468

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

FIFO0_CH1_IRQ_FORCINT	0x0001846C
FIFO0_CH1_IRQ_MODE	0x00018470
FIFO0_CH1_EIRQ_EN	0x00018474
FIFO0_CH2_CTRL	0x00018480
FIFO0_CH2_END_ADDR	0x00018484
FIFO0_CH2_START_ADDR	0x00018488
FIFO0_CH2_UPPER_WM	0x0001848C
FIFO0_CH2_LOWER_WM	0x00018490
FIFO0_CH2_STATUS	0x00018494
FIFO0_CH2_FILL_LEVEL	0x00018498
FIFO0_CH2_WR_PTR	0x0001849C
FIFO0_CH2_RD_PTR	0x000184A0
FIFO0_CH2_IRQ_NOTIFY	0x000184A4
FIFO0_CH2_IRQ_EN	0x000184A8
FIFO0_CH2_IRQ_FORCINT	0x000184AC
FIFO0_CH2_IRQ_MODE	0x000184B0
FIFO0_CH2_EIRQ_EN	0x000184B4
FIFO0_CH3_CTRL	0x000184C0
FIFO0_CH3_END_ADDR	0x000184C4
FIFO0_CH3_START_ADDR	0x000184C8
FIFO0_CH3_UPPER_WM	0x000184CC
FIFO0_CH3_LOWER_WM	0x000184D0
FIFO0_CH3_STATUS	0x000184D4
FIFO0_CH3_FILL_LEVEL	0x000184D8
FIFO0_CH3_WR_PTR	0x000184DC
FIFO0_CH3_RD_PTR	0x000184E0
FIFO0_CH3_IRQ_NOTIFY	0x000184E4
FIFO0_CH3_IRQ_EN	0x000184E8
FIFO0_CH3_IRQ_FORCINT	0x000184EC
FIFO0_CH3_IRQ_MODE	0x000184F0
FIFO0_CH3_EIRQ_EN	0x000184F4
FIFO0_CH4_CTRL	0x00018500
FIFO0_CH4_END_ADDR	0x00018504
FIFO0_CH4_START_ADDR	0x00018508
FIFO0_CH4_UPPER_WM	0x0001850C
FIFO0_CH4_LOWER_WM	0x00018510
FIFO0_CH4_STATUS	0x00018514
FIFO0_CH4_FILL_LEVEL	0x00018518
FIFO0_CH4_WR_PTR	0x0001851C
FIFO0_CH4_RD_PTR	0x00018520

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

FIFO0_CH4_IRQ_NOTIFY	0x00018524
FIFO0_CH4_IRQ_EN	0x00018528
FIFO0_CH4_IRQ_FORCINT	0x0001852C
FIFO0_CH4_IRQ_MODE	0x00018530
FIFO0_CH4_EIRQ_EN	0x00018534
FIFO0_CH5_CTRL	0x00018540
FIFO0_CH5_END_ADDR	0x00018544
FIFO0_CH5_START_ADDR	0x00018548
FIFO0_CH5_UPPER_WM	0x0001854C
FIFO0_CH5_LOWER_WM	0x00018550
FIFO0_CH5_STATUS	0x00018554
FIFO0_CH5_FILL_LEVEL	0x00018558
FIFO0_CH5_WR_PTR	0x0001855C
FIFO0_CH5_RD_PTR	0x00018560
FIFO0_CH5_IRQ_NOTIFY	0x00018564
FIFO0_CH5_IRQ_EN	0x00018568
FIFO0_CH5_IRQ_FORCINT	0x0001856C
FIFO0_CH5_IRQ_MODE	0x00018570
FIFO0_CH5_EIRQ_EN	0x00018574
FIFO0_CH6_CTRL	0x00018580
FIFO0_CH6_END_ADDR	0x00018584
FIFO0_CH6_START_ADDR	0x00018588
FIFO0_CH6_UPPER_WM	0x0001858C
FIFO0_CH6_LOWER_WM	0x00018590
FIFO0_CH6_STATUS	0x00018594
FIFO0_CH6_FILL_LEVEL	0x00018598
FIFO0_CH6_WR_PTR	0x0001859C
FIFO0_CH6_RD_PTR	0x000185A0
FIFO0_CH6_IRQ_NOTIFY	0x000185A4
FIFO0_CH6_IRQ_EN	0x000185A8
FIFO0_CH6_IRQ_FORCINT	0x000185AC
FIFO0_CH6_IRQ_MODE	0x000185B0
FIFO0_CH6_EIRQ_EN	0x000185B4
FIFO0_CH7_CTRL	0x000185C0
FIFO0_CH7_END_ADDR	0x000185C4
FIFO0_CH7_START_ADDR	0x000185C8
FIFO0_CH7_UPPER_WM	0x000185CC
FIFO0_CH7_LOWER_WM	0x000185D0
FIFO0_CH7_STATUS	0x000185D4
FIFO0_CH7_FILL_LEVEL	0x000185D8

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

FIFO0_CH7_WR_PTR	0x000185DC
FIFO0_CH7_RD_PTR	0x000185E0
FIFO0_CH7_IRQ_NOTIFY	0x000185E4
FIFO0_CH7_IRQ_EN	0x000185E8
FIFO0_CH7_IRQ_FORCINT	0x000185EC
FIFO0_CH7_IRQ_MODE	0x000185F0
FIFO0_CH7_EIRQ_EN	0x000185F4
FIFO0_MEMORY	0x00019000
FIFO0_MEMORY_END	0x00019FFC
F2A1_CH0_ARU_RD_FIFO	0x0001C000
F2A1_CH1_ARU_RD_FIFO	0x0001C004
F2A1_CH2_ARU_RD_FIFO	0x0001C008
F2A1_CH3_ARU_RD_FIFO	0x0001C00C
F2A1_CH4_ARU_RD_FIFO	0x0001C010
F2A1_CH5_ARU_RD_FIFO	0x0001C014
F2A1_CH6_ARU_RD_FIFO	0x0001C018
F2A1_CH7_ARU_RD_FIFO	0x0001C01C
F2A1_CH0_STR_CFG	0x0001C020
F2A1_CH1_STR_CFG	0x0001C024
F2A1_CH2_STR_CFG	0x0001C028
F2A1_CH3_STR_CFG	0x0001C02C
F2A1_CH4_STR_CFG	0x0001C030
F2A1_CH5_STR_CFG	0x0001C034
F2A1_CH6_STR_CFG	0x0001C038
F2A1_CH7_STR_CFG	0x0001C03C
F2A1_ENABLE	0x0001C040
AFD1_CH0_BUFFACC	0x0001C080
AFD1_CH1_BUFFACC	0x0001C090
AFD1_CH2_BUFFACC	0x0001C0A0
AFD1_CH3_BUFFACC	0x0001C0B0
AFD1_CH4_BUFFACC	0x0001C0C0
AFD1_CH5_BUFFACC	0x0001C0D0
AFD1_CH6_BUFFACC	0x0001C0E0
AFD1_CH7_BUFFACC	0x0001C0F0
FIFO1_CH0_CTRL	0x0001C400
FIFO1_CH0_END_ADDR	0x0001C404
FIFO1_CH0_START_ADDR	0x0001C408
FIFO1_CH0_UPPER_WM	0x0001C40C
FIFO1_CH0_LOWER_WM	0x0001C410
FIFO1_CH0_STATUS	0x0001C414

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

FIFO1_CH0_FILL_LEVEL	0x0001C418
FIFO1_CH0_WR_PTR	0x0001C41C
FIFO1_CH0_RD_PTR	0x0001C420
FIFO1_CH0_IRQ_NOTIFY	0x0001C424
FIFO1_CH0_IRQ_EN	0x0001C428
FIFO1_CH0_IRQ_FORCINT	0x0001C42C
FIFO1_CH0_IRQ_MODE	0x0001C430
FIFO1_CH0_EIRQ_EN	0x0001C434
FIFO1_CH1_CTRL	0x0001C440
FIFO1_CH1_END_ADDR	0x0001C444
FIFO1_CH1_START_ADDR	0x0001C448
FIFO1_CH1_UPPER_WM	0x0001C44C
FIFO1_CH1_LOWER_WM	0x0001C450
FIFO1_CH1_STATUS	0x0001C454
FIFO1_CH1_FILL_LEVEL	0x0001C458
FIFO1_CH1_WR_PTR	0x0001C45C
FIFO1_CH1_RD_PTR	0x0001C460
FIFO1_CH1_IRQ_NOTIFY	0x0001C464
FIFO1_CH1_IRQ_EN	0x0001C468
FIFO1_CH1_IRQ_FORCINT	0x0001C46C
FIFO1_CH1_IRQ_MODE	0x0001C470
FIFO1_CH1_EIRQ_EN	0x0001C474
FIFO1_CH2_CTRL	0x0001C480
FIFO1_CH2_END_ADDR	0x0001C484
FIFO1_CH2_START_ADDR	0x0001C488
FIFO1_CH2_UPPER_WM	0x0001C48C
FIFO1_CH2_LOWER_WM	0x0001C490
FIFO1_CH2_STATUS	0x0001C494
FIFO1_CH2_FILL_LEVEL	0x0001C498
FIFO1_CH2_WR_PTR	0x0001C49C
FIFO1_CH2_RD_PTR	0x0001C4A0
FIFO1_CH2_IRQ_NOTIFY	0x0001C4A4
FIFO1_CH2_IRQ_EN	0x0001C4A8
FIFO1_CH2_IRQ_FORCINT	0x0001C4AC
FIFO1_CH2_IRQ_MODE	0x0001C4B0
FIFO1_CH2_EIRQ_EN	0x0001C4B4
FIFO1_CH3_CTRL	0x0001C4C0
FIFO1_CH3_END_ADDR	0x0001C4C4
FIFO1_CH3_START_ADDR	0x0001C4C8
FIFO1_CH3_UPPER_WM	0x0001C4CC

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

FIFO1_CH3_LOWER_WM	0x0001C4D0
FIFO1_CH3_STATUS	0x0001C4D4
FIFO1_CH3_FILL_LEVEL	0x0001C4D8
FIFO1_CH3_WR_PTR	0x0001C4DC
FIFO1_CH3_RD_PTR	0x0001C4E0
FIFO1_CH3_IRQ_NOTIFY	0x0001C4E4
FIFO1_CH3_IRQ_EN	0x0001C4E8
FIFO1_CH3_IRQ_FORCINT	0x0001C4EC
FIFO1_CH3_IRQ_MODE	0x0001C4F0
FIFO1_CH3_EIRQ_EN	0x0001C4F4
FIFO1_CH4_CTRL	0x0001C500
FIFO1_CH4_END_ADDR	0x0001C504
FIFO1_CH4_START_ADDR	0x0001C508
FIFO1_CH4_UPPER_WM	0x0001C50C
FIFO1_CH4_LOWER_WM	0x0001C510
FIFO1_CH4_STATUS	0x0001C514
FIFO1_CH4_FILL_LEVEL	0x0001C518
FIFO1_CH4_WR_PTR	0x0001C51C
FIFO1_CH4_RD_PTR	0x0001C520
FIFO1_CH4_IRQ_NOTIFY	0x0001C524
FIFO1_CH4_IRQ_EN	0x0001C528
FIFO1_CH4_IRQ_FORCINT	0x0001C52C
FIFO1_CH4_IRQ_MODE	0x0001C530
FIFO1_CH4_EIRQ_EN	0x0001C534
FIFO1_CH5_CTRL	0x0001C540
FIFO1_CH5_END_ADDR	0x0001C544
FIFO1_CH5_START_ADDR	0x0001C548
FIFO1_CH5_UPPER_WM	0x0001C54C
FIFO1_CH5_LOWER_WM	0x0001C550
FIFO1_CH5_STATUS	0x0001C554
FIFO1_CH5_FILL_LEVEL	0x0001C558
FIFO1_CH5_WR_PTR	0x0001C55C
FIFO1_CH5_RD_PTR	0x0001C560
FIFO1_CH5_IRQ_NOTIFY	0x0001C564
FIFO1_CH5_IRQ_EN	0x0001C568
FIFO1_CH5_IRQ_FORCINT	0x0001C56C
FIFO1_CH5_IRQ_MODE	0x0001C570
FIFO1_CH5_EIRQ_EN	0x0001C574
FIFO1_CH6_CTRL	0x0001C580
FIFO1_CH6_END_ADDR	0x0001C584

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

FIFO1_CH6_START_ADDR	0x0001C588
FIFO1_CH6_UPPER_WM	0x0001C58C
FIFO1_CH6_LOWER_WM	0x0001C590
FIFO1_CH6_STATUS	0x0001C594
FIFO1_CH6_FILL_LEVEL	0x0001C598
FIFO1_CH6_WR_PTR	0x0001C59C
FIFO1_CH6_RD_PTR	0x0001C5A0
FIFO1_CH6_IRQ_NOTIFY	0x0001C5A4
FIFO1_CH6_IRQ_EN	0x0001C5A8
FIFO1_CH6_IRQ_FORCINT	0x0001C5AC
FIFO1_CH6_IRQ_MODE	0x0001C5B0
FIFO1_CH6_EIRQ_EN	0x0001C5B4
FIFO1_CH7_CTRL	0x0001C5C0
FIFO1_CH7_END_ADDR	0x0001C5C4
FIFO1_CH7_START_ADDR	0x0001C5C8
FIFO1_CH7_UPPER_WM	0x0001C5CC
FIFO1_CH7_LOWER_WM	0x0001C5D0
FIFO1_CH7_STATUS	0x0001C5D4
FIFO1_CH7_FILL_LEVEL	0x0001C5D8
FIFO1_CH7_WR_PTR	0x0001C5DC
FIFO1_CH7_RD_PTR	0x0001C5E0
FIFO1_CH7_IRQ_NOTIFY	0x0001C5E4
FIFO1_CH7_IRQ_EN	0x0001C5E8
FIFO1_CH7_IRQ_FORCINT	0x0001C5EC
FIFO1_CH7_IRQ_MODE	0x0001C5F0
FIFO1_CH7_EIRQ_EN	0x0001C5F4
FIFO1_MEMORY	0x0001D000
FIFO1_MEMORY_END	0x0001DFFC
DPLL_CTRL_0	0x00028000
DPLL_CTRL_1	0x00028004
DPLL_CTRL_2	0x00028008
DPLL_CTRL_3	0x0002800C
DPLL_CTRL_4	0x00028010
DPLL_CTRL_5	0x00028014
DPLL_ACT_STA	0x00028018
DPLL_OSW	0x0002801C
DPLL_AOSV_2	0x00028020
DPLL_APT	0x00028024
DPLL_APS	0x00028028
DPLL_APT_2C	0x0002802C

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

DPLL_APS_1C3	0x00028030
DPLL_NUTC	0x00028034
DPLL_NUSC	0x00028038
DPLL_NTI_CNT	0x0002803C
DPLL_IRQ_NOTIFY	0x00028040
DPLL_IRQ_EN	0x00028044
DPLL_IRQ_FORCINT	0x00028048
DPLL_IRQ_MODE	0x0002804C
DPLL_EIRQ_EN	0x00028050
DPLL_INC_CNT1	0x000280B0
DPLL_INC_CNT2	0x000280B4
DPLL_APT_SYNC	0x000280B8
DPLL_APS_SYNC	0x000280BC
DPLL_TBU_TS0_T	0x000280C0
DPLL_TBU_TS0_S	0x000280C4
DPLL_ADD_IN_LD1	0x000280C8
DPLL_ADD_IN_LD2	0x000280CC
DPLL_STATUS	0x000280FC
DPLL_ID_PMTR_0	0x00028100
DPLL_ID_PMTR_1	0x00028104
DPLL_ID_PMTR_2	0x00028108
DPLL_ID_PMTR_3	0x0002810C
DPLL_ID_PMTR_4	0x00028110
DPLL_ID_PMTR_5	0x00028114
DPLL_ID_PMTR_6	0x00028118
DPLL_ID_PMTR_7	0x0002811C
DPLL_ID_PMTR_8	0x00028120
DPLL_ID_PMTR_9	0x00028124
DPLL_ID_PMTR_10	0x00028128
DPLL_ID_PMTR_11	0x0002812C
DPLL_ID_PMTR_12	0x00028130
DPLL_ID_PMTR_13	0x00028134
DPLL_ID_PMTR_14	0x00028138
DPLL_ID_PMTR_15	0x0002813C
DPLL_ID_PMTR_16	0x00028140
DPLL_ID_PMTR_17	0x00028144
DPLL_ID_PMTR_18	0x00028148
DPLL_ID_PMTR_19	0x0002814C
DPLL_ID_PMTR_20	0x00028150
DPLL_ID_PMTR_21	0x00028154

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

DPLL_ID_PMTR_22	0x00028158
DPLL_ID_PMTR_23	0x0002815C
DPLL_ID_PMTR_24	0x00028160
DPLL_ID_PMTR_25	0x00028164
DPLL_ID_PMTR_26	0x00028168
DPLL_ID_PMTR_27	0x0002816C
DPLL_ID_PMTR_28	0x00028170
DPLL_ID_PMTR_29	0x00028174
DPLL_ID_PMTR_30	0x00028178
DPLL_ID_PMTR_31	0x0002817C
DPLL_CTRL_0_SHADOW_TRIGGER	0x000281E0
DPLL_CTRL_0_SHADOW_STATE	0x000281E4
DPLL_CTRL_1_SHADOW_TRIGGER	0x000281E8
DPLL_CTRL_1_SHADOW_STATE	0x000281EC
DPLL_RAM_INI	0x000281FC
DPLL_RR1A	0x00028200
DPLL_RR1A_END	0x000283FC
DPLL_RR1B	0x00028400
DPLL_RR1B_END	0x000285FC
DPLL_RR1C	0x00028600
DPLL_RR1C_END	0x000287FC
DPLL_RR1C_part2	0x00028800
DPLL_RR1C_part2_END	0x000289FC
DPLL_TSA_0	0x00028E00
DPLL_TSA_1	0x00028E04
DPLL_TSA_2	0x00028E08
DPLL_TSA_3	0x00028E0C
DPLL_TSA_4	0x00028E10
DPLL_TSA_5	0x00028E14
DPLL_TSA_6	0x00028E18
DPLL_TSA_7	0x00028E1C
DPLL_TSA_8	0x00028E20
DPLL_TSA_9	0x00028E24
DPLL_TSA_10	0x00028E28
DPLL_TSA_11	0x00028E2C
DPLL_TSA_12	0x00028E30
DPLL_TSA_13	0x00028E34
DPLL_TSA_14	0x00028E38
DPLL_TSA_15	0x00028E3C
DPLL_TSA_16	0x00028E40

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

DPLL_TSA_17	0x00028E44
DPLL_TSA_18	0x00028E48
DPLL_TSA_19	0x00028E4C
DPLL_TSA_20	0x00028E50
DPLL_TSA_21	0x00028E54
DPLL_TSA_22	0x00028E58
DPLL_TSA_23	0x00028E5C
DPLL_TSA_24	0x00028E60
DPLL_TSA_25	0x00028E64
DPLL_TSA_26	0x00028E68
DPLL_TSA_27	0x00028E6C
DPLL_TSA_28	0x00028E70
DPLL_TSA_29	0x00028E74
DPLL_TSA_30	0x00028E78
DPLL_TSA_31	0x00028E7C
DPLL_PSAC_0	0x00028E80
DPLL_PSAC_1	0x00028E84
DPLL_PSAC_2	0x00028E88
DPLL_PSAC_3	0x00028E8C
DPLL_PSAC_4	0x00028E90
DPLL_PSAC_5	0x00028E94
DPLL_PSAC_6	0x00028E98
DPLL_PSAC_7	0x00028E9C
DPLL_PSAC_8	0x00028EA0
DPLL_PSAC_9	0x00028EA4
DPLL_PSAC_10	0x00028EA8
DPLL_PSAC_11	0x00028EAC
DPLL_PSAC_12	0x00028EB0
DPLL_PSAC_13	0x00028EB4
DPLL_PSAC_14	0x00028EB8
DPLL_PSAC_15	0x00028EBC
DPLL_PSAC_16	0x00028EC0
DPLL_PSAC_17	0x00028EC4
DPLL_PSAC_18	0x00028EC8
DPLL_PSAC_19	0x00028ECC
DPLL_PSAC_20	0x00028ED0
DPLL_PSAC_21	0x00028ED4
DPLL_PSAC_22	0x00028ED8
DPLL_PSAC_23	0x00028EDC
DPLL_PSAC_24	0x00028EE0

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

DPLL_PSAC_25	0x00028EE4
DPLL_PSAC_26	0x00028EE8
DPLL_PSAC_27	0x00028EEC
DPLL_PSAC_28	0x00028EF0
DPLL_PSAC_29	0x00028EF4
DPLL_PSAC_30	0x00028EF8
DPLL_PSAC_31	0x00028EFC
DPLL_ACB_0	0x00028F00
DPLL_ACB_1	0x00028F04
DPLL_ACB_2	0x00028F08
DPLL_ACB_3	0x00028F0C
DPLL_ACB_4	0x00028F10
DPLL_ACB_5	0x00028F14
DPLL_ACB_6	0x00028F18
DPLL_ACB_7	0x00028F1C
DPLL_RR2	0x0002C000
DPLL_RR2_END	0x0002FFFC
MCS0_CH0_R0	0x00030000
MCS0_CH0_R1	0x00030004
MCS0_CH0_R2	0x00030008
MCS0_CH0_R3	0x0003000C
MCS0_CH0_R4	0x00030010
MCS0_CH0_R5	0x00030014
MCS0_CH0_R6	0x00030018
MCS0_CH0_R7	0x0003001C
MCS0_CH0_CTRL	0x00030020
MCS0_CH0_ACB	0x00030024
MCS0_CTRG	0x00030028
MCS0_STRG	0x0003002C
MCS0_CH0_PC	0x00030040
MCS0_CH0_IRQ_NOTIFY	0x00030044
MCS0_CH0_IRQ_EN	0x00030048
MCS0_CH0_IRQ_FORCINT	0x0003004C
MCS0_CH0_IRQ_MODE	0x00030050
MCS0_CH0_EIRQ_EN	0x00030054
MCS0_CTRL	0x00030074
MCS0_RST	0x00030078
MCS0_ERR	0x0003007C
MCS0_CH1_R0	0x00030080
MCS0_CH1_R1	0x00030084

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS0_CH1_R2	0x00030088
MCS0_CH1_R3	0x0003008C
MCS0_CH1_R4	0x00030090
MCS0_CH1_R5	0x00030094
MCS0_CH1_R6	0x00030098
MCS0_CH1_R7	0x0003009C
MCS0_CH1_CTRL	0x000300A0
MCS0_CH1_ACB	0x000300A4
MCS0_CH1_PC	0x000300C0
MCS0_CH1_IRQ_NOTIFY	0x000300C4
MCS0_CH1_IRQ_EN	0x000300C8
MCS0_CH1_IRQ_FORCINT	0x000300CC
MCS0_CH1_IRQ_MODE	0x000300D0
MCS0_CH1_EIRQ_EN	0x000300D4
MCS0_CH2_R0	0x00030100
MCS0_CH2_R1	0x00030104
MCS0_CH2_R2	0x00030108
MCS0_CH2_R3	0x0003010C
MCS0_CH2_R4	0x00030110
MCS0_CH2_R5	0x00030114
MCS0_CH2_R6	0x00030118
MCS0_CH2_R7	0x0003011C
MCS0_CH2_CTRL	0x00030120
MCS0_CH2_ACB	0x00030124
MCS0_CH2_PC	0x00030140
MCS0_CH2_IRQ_NOTIFY	0x00030144
MCS0_CH2_IRQ_EN	0x00030148
MCS0_CH2_IRQ_FORCINT	0x0003014C
MCS0_CH2_IRQ_MODE	0x00030150
MCS0_CH2_EIRQ_EN	0x00030154
MCS0_CH3_R0	0x00030180
MCS0_CH3_R1	0x00030184
MCS0_CH3_R2	0x00030188
MCS0_CH3_R3	0x0003018C
MCS0_CH3_R4	0x00030190
MCS0_CH3_R5	0x00030194
MCS0_CH3_R6	0x00030198
MCS0_CH3_R7	0x0003019C
MCS0_CH3_CTRL	0x000301A0
MCS0_CH3_ACB	0x000301A4

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS0_CH3_PC	0x000301C0
MCS0_CH3_IRQ_NOTIFY	0x000301C4
MCS0_CH3_IRQ_EN	0x000301C8
MCS0_CH3_IRQ_FORCINT	0x000301CC
MCS0_CH3_IRQ_MODE	0x000301D0
MCS0_CH3_EIRQ_EN	0x000301D4
MCS0_CH4_R0	0x00030200
MCS0_CH4_R1	0x00030204
MCS0_CH4_R2	0x00030208
MCS0_CH4_R3	0x0003020C
MCS0_CH4_R4	0x00030210
MCS0_CH4_R5	0x00030214
MCS0_CH4_R6	0x00030218
MCS0_CH4_R7	0x0003021C
MCS0_CH4_CTRL	0x00030220
MCS0_CH4_ACB	0x00030224
MCS0_CH4_PC	0x00030240
MCS0_CH4_IRQ_NOTIFY	0x00030244
MCS0_CH4_IRQ_EN	0x00030248
MCS0_CH4_IRQ_FORCINT	0x0003024C
MCS0_CH4_IRQ_MODE	0x00030250
MCS0_CH4_EIRQ_EN	0x00030254
MCS0_CH5_R0	0x00030280
MCS0_CH5_R1	0x00030284
MCS0_CH5_R2	0x00030288
MCS0_CH5_R3	0x0003028C
MCS0_CH5_R4	0x00030290
MCS0_CH5_R5	0x00030294
MCS0_CH5_R6	0x00030298
MCS0_CH5_R7	0x0003029C
MCS0_CH5_CTRL	0x000302A0
MCS0_CH5_ACB	0x000302A4
MCS0_CH5_PC	0x000302C0
MCS0_CH5_IRQ_NOTIFY	0x000302C4
MCS0_CH5_IRQ_EN	0x000302C8
MCS0_CH5_IRQ_FORCINT	0x000302CC
MCS0_CH5_IRQ_MODE	0x000302D0
MCS0_CH5_EIRQ_EN	0x000302D4
MCS0_CH6_R0	0x00030300
MCS0_CH6_R1	0x00030304

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS0_CH6_R2	0x00030308
MCS0_CH6_R3	0x0003030C
MCS0_CH6_R4	0x00030310
MCS0_CH6_R5	0x00030314
MCS0_CH6_R6	0x00030318
MCS0_CH6_R7	0x0003031C
MCS0_CH6_CTRL	0x00030320
MCS0_CH6_ACB	0x00030324
MCS0_CH6_PC	0x00030340
MCS0_CH6_IRQ_NOTIFY	0x00030344
MCS0_CH6_IRQ_EN	0x00030348
MCS0_CH6_IRQ_FORCINT	0x0003034C
MCS0_CH6_IRQ_MODE	0x00030350
MCS0_CH6_EIRQ_EN	0x00030354
MCS0_CH7_R0	0x00030380
MCS0_CH7_R1	0x00030384
MCS0_CH7_R2	0x00030388
MCS0_CH7_R3	0x0003038C
MCS0_CH7_R4	0x00030390
MCS0_CH7_R5	0x00030394
MCS0_CH7_R6	0x00030398
MCS0_CH7_R7	0x0003039C
MCS0_CH7_CTRL	0x000303A0
MCS0_CH7_ACB	0x000303A4
MCS0_CH7_PC	0x000303C0
MCS0_CH7_IRQ_NOTIFY	0x000303C4
MCS0_CH7_IRQ_EN	0x000303C8
MCS0_CH7_IRQ_FORCINT	0x000303CC
MCS0_CH7_IRQ_MODE	0x000303D0
MCS0_CH7_EIRQ_EN	0x000303D4
MCS1_CH0_R0	0x00031000
MCS1_CH0_R1	0x00031004
MCS1_CH0_R2	0x00031008
MCS1_CH0_R3	0x0003100C
MCS1_CH0_R4	0x00031010
MCS1_CH0_R5	0x00031014
MCS1_CH0_R6	0x00031018
MCS1_CH0_R7	0x0003101C
MCS1_CH0_CTRL	0x00031020
MCS1_CH0_ACB	0x00031024

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS1_CTRG	0x00031028
MCS1_STRG	0x0003102C
MCS1_CH0_PC	0x00031040
MCS1_CH0_IRQ_NOTIFY	0x00031044
MCS1_CH0_IRQ_EN	0x00031048
MCS1_CH0_IRQ_FORCINT	0x0003104C
MCS1_CH0_IRQ_MODE	0x00031050
MCS1_CH0_EIRQ_EN	0x00031054
MCS1_CTRL	0x00031074
MCS1_RST	0x00031078
MCS1_ERR	0x0003107C
MCS1_CH1_R0	0x00031080
MCS1_CH1_R1	0x00031084
MCS1_CH1_R2	0x00031088
MCS1_CH1_R3	0x0003108C
MCS1_CH1_R4	0x00031090
MCS1_CH1_R5	0x00031094
MCS1_CH1_R6	0x00031098
MCS1_CH1_R7	0x0003109C
MCS1_CH1_CTRL	0x000310A0
MCS1_CH1_ACB	0x000310A4
MCS1_CH1_PC	0x000310C0
MCS1_CH1_IRQ_NOTIFY	0x000310C4
MCS1_CH1_IRQ_EN	0x000310C8
MCS1_CH1_IRQ_FORCINT	0x000310CC
MCS1_CH1_IRQ_MODE	0x000310D0
MCS1_CH1_EIRQ_EN	0x000310D4
MCS1_CH2_R0	0x00031100
MCS1_CH2_R1	0x00031104
MCS1_CH2_R2	0x00031108
MCS1_CH2_R3	0x0003110C
MCS1_CH2_R4	0x00031110
MCS1_CH2_R5	0x00031114
MCS1_CH2_R6	0x00031118
MCS1_CH2_R7	0x0003111C
MCS1_CH2_CTRL	0x00031120
MCS1_CH2_ACB	0x00031124
MCS1_CH2_PC	0x00031140
MCS1_CH2_IRQ_NOTIFY	0x00031144
MCS1_CH2_IRQ_EN	0x00031148

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS1_CH2_IRQ_FORCINT	0x0003114C
MCS1_CH2_IRQ_MODE	0x00031150
MCS1_CH2_EIRQ_EN	0x00031154
MCS1_CH3_R0	0x00031180
MCS1_CH3_R1	0x00031184
MCS1_CH3_R2	0x00031188
MCS1_CH3_R3	0x0003118C
MCS1_CH3_R4	0x00031190
MCS1_CH3_R5	0x00031194
MCS1_CH3_R6	0x00031198
MCS1_CH3_R7	0x0003119C
MCS1_CH3_CTRL	0x000311A0
MCS1_CH3_ACB	0x000311A4
MCS1_CH3_PC	0x000311C0
MCS1_CH3_IRQ_NOTIFY	0x000311C4
MCS1_CH3_IRQ_EN	0x000311C8
MCS1_CH3_IRQ_FORCINT	0x000311CC
MCS1_CH3_IRQ_MODE	0x000311D0
MCS1_CH3_EIRQ_EN	0x000311D4
MCS1_CH4_R0	0x00031200
MCS1_CH4_R1	0x00031204
MCS1_CH4_R2	0x00031208
MCS1_CH4_R3	0x0003120C
MCS1_CH4_R4	0x00031210
MCS1_CH4_R5	0x00031214
MCS1_CH4_R6	0x00031218
MCS1_CH4_R7	0x0003121C
MCS1_CH4_CTRL	0x00031220
MCS1_CH4_ACB	0x00031224
MCS1_CH4_PC	0x00031240
MCS1_CH4_IRQ_NOTIFY	0x00031244
MCS1_CH4_IRQ_EN	0x00031248
MCS1_CH4_IRQ_FORCINT	0x0003124C
MCS1_CH4_IRQ_MODE	0x00031250
MCS1_CH4_EIRQ_EN	0x00031254
MCS1_CH5_R0	0x00031280
MCS1_CH5_R1	0x00031284
MCS1_CH5_R2	0x00031288
MCS1_CH5_R3	0x0003128C
MCS1_CH5_R4	0x00031290

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS1_CH5_R5	0x00031294
MCS1_CH5_R6	0x00031298
MCS1_CH5_R7	0x0003129C
MCS1_CH5_CTRL	0x000312A0
MCS1_CH5_ACB	0x000312A4
MCS1_CH5_PC	0x000312C0
MCS1_CH5_IRQ_NOTIFY	0x000312C4
MCS1_CH5_IRQ_EN	0x000312C8
MCS1_CH5_IRQ_FORCINT	0x000312CC
MCS1_CH5_IRQ_MODE	0x000312D0
MCS1_CH5_EIRQ_EN	0x000312D4
MCS1_CH6_R0	0x00031300
MCS1_CH6_R1	0x00031304
MCS1_CH6_R2	0x00031308
MCS1_CH6_R3	0x0003130C
MCS1_CH6_R4	0x00031310
MCS1_CH6_R5	0x00031314
MCS1_CH6_R6	0x00031318
MCS1_CH6_R7	0x0003131C
MCS1_CH6_CTRL	0x00031320
MCS1_CH6_ACB	0x00031324
MCS1_CH6_PC	0x00031340
MCS1_CH6_IRQ_NOTIFY	0x00031344
MCS1_CH6_IRQ_EN	0x00031348
MCS1_CH6_IRQ_FORCINT	0x0003134C
MCS1_CH6_IRQ_MODE	0x00031350
MCS1_CH6_EIRQ_EN	0x00031354
MCS1_CH7_R0	0x00031380
MCS1_CH7_R1	0x00031384
MCS1_CH7_R2	0x00031388
MCS1_CH7_R3	0x0003138C
MCS1_CH7_R4	0x00031390
MCS1_CH7_R5	0x00031394
MCS1_CH7_R6	0x00031398
MCS1_CH7_R7	0x0003139C
MCS1_CH7_CTRL	0x000313A0
MCS1_CH7_ACB	0x000313A4
MCS1_CH7_PC	0x000313C0
MCS1_CH7_IRQ_NOTIFY	0x000313C4
MCS1_CH7_IRQ_EN	0x000313C8

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS1_CH7_IRQ_FORCINT	0x000313CC
MCS1_CH7_IRQ_MODE	0x000313D0
MCS1_CH7_EIRQ_EN	0x000313D4
MCS2_CH0_R0	0x00032000
MCS2_CH0_R1	0x00032004
MCS2_CH0_R2	0x00032008
MCS2_CH0_R3	0x0003200C
MCS2_CH0_R4	0x00032010
MCS2_CH0_R5	0x00032014
MCS2_CH0_R6	0x00032018
MCS2_CH0_R7	0x0003201C
MCS2_CH0_CTRL	0x00032020
MCS2_CH0_ACB	0x00032024
MCS2_CTRG	0x00032028
MCS2_STRG	0x0003202C
MCS2_CH0_PC	0x00032040
MCS2_CH0_IRQ_NOTIFY	0x00032044
MCS2_CH0_IRQ_EN	0x00032048
MCS2_CH0_IRQ_FORCINT	0x0003204C
MCS2_CH0_IRQ_MODE	0x00032050
MCS2_CH0_EIRQ_EN	0x00032054
MCS2_CTRL	0x00032074
MCS2_RST	0x00032078
MCS2_ERR	0x0003207C
MCS2_CH1_R0	0x00032080
MCS2_CH1_R1	0x00032084
MCS2_CH1_R2	0x00032088
MCS2_CH1_R3	0x0003208C
MCS2_CH1_R4	0x00032090
MCS2_CH1_R5	0x00032094
MCS2_CH1_R6	0x00032098
MCS2_CH1_R7	0x0003209C
MCS2_CH1_CTRL	0x000320A0
MCS2_CH1_ACB	0x000320A4
MCS2_CH1_PC	0x000320C0
MCS2_CH1_IRQ_NOTIFY	0x000320C4
MCS2_CH1_IRQ_EN	0x000320C8
MCS2_CH1_IRQ_FORCINT	0x000320CC
MCS2_CH1_IRQ_MODE	0x000320D0
MCS2_CH1_EIRQ_EN	0x000320D4

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS2_CH2_R0	0x00032100
MCS2_CH2_R1	0x00032104
MCS2_CH2_R2	0x00032108
MCS2_CH2_R3	0x0003210C
MCS2_CH2_R4	0x00032110
MCS2_CH2_R5	0x00032114
MCS2_CH2_R6	0x00032118
MCS2_CH2_R7	0x0003211C
MCS2_CH2_CTRL	0x00032120
MCS2_CH2_ACB	0x00032124
MCS2_CH2_PC	0x00032140
MCS2_CH2_IRQ_NOTIFY	0x00032144
MCS2_CH2_IRQ_EN	0x00032148
MCS2_CH2_IRQ_FORCINT	0x0003214C
MCS2_CH2_IRQ_MODE	0x00032150
MCS2_CH2_EIRQ_EN	0x00032154
MCS2_CH3_R0	0x00032180
MCS2_CH3_R1	0x00032184
MCS2_CH3_R2	0x00032188
MCS2_CH3_R3	0x0003218C
MCS2_CH3_R4	0x00032190
MCS2_CH3_R5	0x00032194
MCS2_CH3_R6	0x00032198
MCS2_CH3_R7	0x0003219C
MCS2_CH3_CTRL	0x000321A0
MCS2_CH3_ACB	0x000321A4
MCS2_CH3_PC	0x000321C0
MCS2_CH3_IRQ_NOTIFY	0x000321C4
MCS2_CH3_IRQ_EN	0x000321C8
MCS2_CH3_IRQ_FORCINT	0x000321CC
MCS2_CH3_IRQ_MODE	0x000321D0
MCS2_CH3_EIRQ_EN	0x000321D4
MCS2_CH4_R0	0x00032200
MCS2_CH4_R1	0x00032204
MCS2_CH4_R2	0x00032208
MCS2_CH4_R3	0x0003220C
MCS2_CH4_R4	0x00032210
MCS2_CH4_R5	0x00032214
MCS2_CH4_R6	0x00032218
MCS2_CH4_R7	0x0003221C

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS2_CH4_CTRL	0x00032220
MCS2_CH4_ACB	0x00032224
MCS2_CH4_PC	0x00032240
MCS2_CH4_IRQ_NOTIFY	0x00032244
MCS2_CH4_IRQ_EN	0x00032248
MCS2_CH4_IRQ_FORCINT	0x0003224C
MCS2_CH4_IRQ_MODE	0x00032250
MCS2_CH4_EIRQ_EN	0x00032254
MCS2_CH5_R0	0x00032280
MCS2_CH5_R1	0x00032284
MCS2_CH5_R2	0x00032288
MCS2_CH5_R3	0x0003228C
MCS2_CH5_R4	0x00032290
MCS2_CH5_R5	0x00032294
MCS2_CH5_R6	0x00032298
MCS2_CH5_R7	0x0003229C
MCS2_CH5_CTRL	0x000322A0
MCS2_CH5_ACB	0x000322A4
MCS2_CH5_PC	0x000322C0
MCS2_CH5_IRQ_NOTIFY	0x000322C4
MCS2_CH5_IRQ_EN	0x000322C8
MCS2_CH5_IRQ_FORCINT	0x000322CC
MCS2_CH5_IRQ_MODE	0x000322D0
MCS2_CH5_EIRQ_EN	0x000322D4
MCS2_CH6_R0	0x00032300
MCS2_CH6_R1	0x00032304
MCS2_CH6_R2	0x00032308
MCS2_CH6_R3	0x0003230C
MCS2_CH6_R4	0x00032310
MCS2_CH6_R5	0x00032314
MCS2_CH6_R6	0x00032318
MCS2_CH6_R7	0x0003231C
MCS2_CH6_CTRL	0x00032320
MCS2_CH6_ACB	0x00032324
MCS2_CH6_PC	0x00032340
MCS2_CH6_IRQ_NOTIFY	0x00032344
MCS2_CH6_IRQ_EN	0x00032348
MCS2_CH6_IRQ_FORCINT	0x0003234C
MCS2_CH6_IRQ_MODE	0x00032350
MCS2_CH6_EIRQ_EN	0x00032354

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS2_CH7_R0	0x00032380
MCS2_CH7_R1	0x00032384
MCS2_CH7_R2	0x00032388
MCS2_CH7_R3	0x0003238C
MCS2_CH7_R4	0x00032390
MCS2_CH7_R5	0x00032394
MCS2_CH7_R6	0x00032398
MCS2_CH7_R7	0x0003239C
MCS2_CH7_CTRL	0x000323A0
MCS2_CH7_ACB	0x000323A4
MCS2_CH7_PC	0x000323C0
MCS2_CH7_IRQ_NOTIFY	0x000323C4
MCS2_CH7_IRQ_EN	0x000323C8
MCS2_CH7_IRQ_FORCINT	0x000323CC
MCS2_CH7_IRQ_MODE	0x000323D0
MCS2_CH7_EIRQ_EN	0x000323D4
MCS3_CH0_R0	0x00033000
MCS3_CH0_R1	0x00033004
MCS3_CH0_R2	0x00033008
MCS3_CH0_R3	0x0003300C
MCS3_CH0_R4	0x00033010
MCS3_CH0_R5	0x00033014
MCS3_CH0_R6	0x00033018
MCS3_CH0_R7	0x0003301C
MCS3_CH0_CTRL	0x00033020
MCS3_CH0_ACB	0x00033024
MCS3_CTRG	0x00033028
MCS3_STRG	0x0003302C
MCS3_CH0_PC	0x00033040
MCS3_CH0_IRQ_NOTIFY	0x00033044
MCS3_CH0_IRQ_EN	0x00033048
MCS3_CH0_IRQ_FORCINT	0x0003304C
MCS3_CH0_IRQ_MODE	0x00033050
MCS3_CH0_EIRQ_EN	0x00033054
MCS3_CTRL	0x00033074
MCS3_RST	0x00033078
MCS3_ERR	0x0003307C
MCS3_CH1_R0	0x00033080
MCS3_CH1_R1	0x00033084
MCS3_CH1_R2	0x00033088

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS3_CH1_R3	0x0003308C
MCS3_CH1_R4	0x00033090
MCS3_CH1_R5	0x00033094
MCS3_CH1_R6	0x00033098
MCS3_CH1_R7	0x0003309C
MCS3_CH1_CTRL	0x000330A0
MCS3_CH1_ACB	0x000330A4
MCS3_CH1_PC	0x000330C0
MCS3_CH1_IRQ_NOTIFY	0x000330C4
MCS3_CH1_IRQ_EN	0x000330C8
MCS3_CH1_IRQ_FORCINT	0x000330CC
MCS3_CH1_IRQ_MODE	0x000330D0
MCS3_CH1_EIRQ_EN	0x000330D4
MCS3_CH2_R0	0x00033100
MCS3_CH2_R1	0x00033104
MCS3_CH2_R2	0x00033108
MCS3_CH2_R3	0x0003310C
MCS3_CH2_R4	0x00033110
MCS3_CH2_R5	0x00033114
MCS3_CH2_R6	0x00033118
MCS3_CH2_R7	0x0003311C
MCS3_CH2_CTRL	0x00033120
MCS3_CH2_ACB	0x00033124
MCS3_CH2_PC	0x00033140
MCS3_CH2_IRQ_NOTIFY	0x00033144
MCS3_CH2_IRQ_EN	0x00033148
MCS3_CH2_IRQ_FORCINT	0x0003314C
MCS3_CH2_IRQ_MODE	0x00033150
MCS3_CH2_EIRQ_EN	0x00033154
MCS3_CH3_R0	0x00033180
MCS3_CH3_R1	0x00033184
MCS3_CH3_R2	0x00033188
MCS3_CH3_R3	0x0003318C
MCS3_CH3_R4	0x00033190
MCS3_CH3_R5	0x00033194
MCS3_CH3_R6	0x00033198
MCS3_CH3_R7	0x0003319C
MCS3_CH3_CTRL	0x000331A0
MCS3_CH3_ACB	0x000331A4
MCS3_CH3_PC	0x000331C0

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS3_CH3_IRQ_NOTIFY	0x000331C4
MCS3_CH3_IRQ_EN	0x000331C8
MCS3_CH3_IRQ_FORCINT	0x000331CC
MCS3_CH3_IRQ_MODE	0x000331D0
MCS3_CH3_EIRQ_EN	0x000331D4
MCS3_CH4_R0	0x00033200
MCS3_CH4_R1	0x00033204
MCS3_CH4_R2	0x00033208
MCS3_CH4_R3	0x0003320C
MCS3_CH4_R4	0x00033210
MCS3_CH4_R5	0x00033214
MCS3_CH4_R6	0x00033218
MCS3_CH4_R7	0x0003321C
MCS3_CH4_CTRL	0x00033220
MCS3_CH4_ACB	0x00033224
MCS3_CH4_PC	0x00033240
MCS3_CH4_IRQ_NOTIFY	0x00033244
MCS3_CH4_IRQ_EN	0x00033248
MCS3_CH4_IRQ_FORCINT	0x0003324C
MCS3_CH4_IRQ_MODE	0x00033250
MCS3_CH4_EIRQ_EN	0x00033254
MCS3_CH5_R0	0x00033280
MCS3_CH5_R1	0x00033284
MCS3_CH5_R2	0x00033288
MCS3_CH5_R3	0x0003328C
MCS3_CH5_R4	0x00033290
MCS3_CH5_R5	0x00033294
MCS3_CH5_R6	0x00033298
MCS3_CH5_R7	0x0003329C
MCS3_CH5_CTRL	0x000332A0
MCS3_CH5_ACB	0x000332A4
MCS3_CH5_PC	0x000332C0
MCS3_CH5_IRQ_NOTIFY	0x000332C4
MCS3_CH5_IRQ_EN	0x000332C8
MCS3_CH5_IRQ_FORCINT	0x000332CC
MCS3_CH5_IRQ_MODE	0x000332D0
MCS3_CH5_EIRQ_EN	0x000332D4
MCS3_CH6_R0	0x00033300
MCS3_CH6_R1	0x00033304
MCS3_CH6_R2	0x00033308

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS3_CH6_R3	0x0003330C
MCS3_CH6_R4	0x00033310
MCS3_CH6_R5	0x00033314
MCS3_CH6_R6	0x00033318
MCS3_CH6_R7	0x0003331C
MCS3_CH6_CTRL	0x00033320
MCS3_CH6_ACB	0x00033324
MCS3_CH6_PC	0x00033340
MCS3_CH6_IRQ_NOTIFY	0x00033344
MCS3_CH6_IRQ_EN	0x00033348
MCS3_CH6_IRQ_FORCINT	0x0003334C
MCS3_CH6_IRQ_MODE	0x00033350
MCS3_CH6_EIRQ_EN	0x00033354
MCS3_CH7_R0	0x00033380
MCS3_CH7_R1	0x00033384
MCS3_CH7_R2	0x00033388
MCS3_CH7_R3	0x0003338C
MCS3_CH7_R4	0x00033390
MCS3_CH7_R5	0x00033394
MCS3_CH7_R6	0x00033398
MCS3_CH7_R7	0x0003339C
MCS3_CH7_CTRL	0x000333A0
MCS3_CH7_ACB	0x000333A4
MCS3_CH7_PC	0x000333C0
MCS3_CH7_IRQ_NOTIFY	0x000333C4
MCS3_CH7_IRQ_EN	0x000333C8
MCS3_CH7_IRQ_FORCINT	0x000333CC
MCS3_CH7_IRQ_MODE	0x000333D0
MCS3_CH7_EIRQ_EN	0x000333D4
MCS4_CH0_R0	0x00034000
MCS4_CH0_R1	0x00034004
MCS4_CH0_R2	0x00034008
MCS4_CH0_R3	0x0003400C
MCS4_CH0_R4	0x00034010
MCS4_CH0_R5	0x00034014
MCS4_CH0_R6	0x00034018
MCS4_CH0_R7	0x0003401C
MCS4_CH0_CTRL	0x00034020
MCS4_CH0_ACB	0x00034024
MCS4_CTRG	0x00034028

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS4_STRG	0x0003402C
MCS4_CH0_PC	0x00034040
MCS4_CH0_IRQ_NOTIFY	0x00034044
MCS4_CH0_IRQ_EN	0x00034048
MCS4_CH0_IRQ_FORCINT	0x0003404C
MCS4_CH0_IRQ_MODE	0x00034050
MCS4_CH0_EIRQ_EN	0x00034054
MCS4_CTRL	0x00034074
MCS4_RST	0x00034078
MCS4_ERR	0x0003407C
MCS4_CH1_R0	0x00034080
MCS4_CH1_R1	0x00034084
MCS4_CH1_R2	0x00034088
MCS4_CH1_R3	0x0003408C
MCS4_CH1_R4	0x00034090
MCS4_CH1_R5	0x00034094
MCS4_CH1_R6	0x00034098
MCS4_CH1_R7	0x0003409C
MCS4_CH1_CTRL	0x000340A0
MCS4_CH1_ACB	0x000340A4
MCS4_CH1_PC	0x000340C0
MCS4_CH1_IRQ_NOTIFY	0x000340C4
MCS4_CH1_IRQ_EN	0x000340C8
MCS4_CH1_IRQ_FORCINT	0x000340CC
MCS4_CH1_IRQ_MODE	0x000340D0
MCS4_CH1_EIRQ_EN	0x000340D4
MCS4_CH2_R0	0x00034100
MCS4_CH2_R1	0x00034104
MCS4_CH2_R2	0x00034108
MCS4_CH2_R3	0x0003410C
MCS4_CH2_R4	0x00034110
MCS4_CH2_R5	0x00034114
MCS4_CH2_R6	0x00034118
MCS4_CH2_R7	0x0003411C
MCS4_CH2_CTRL	0x00034120
MCS4_CH2_ACB	0x00034124
MCS4_CH2_PC	0x00034140
MCS4_CH2_IRQ_NOTIFY	0x00034144
MCS4_CH2_IRQ_EN	0x00034148
MCS4_CH2_IRQ_FORCINT	0x0003414C

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS4_CH2_IRQ_MODE	0x00034150
MCS4_CH2_EIRQ_EN	0x00034154
MCS4_CH3_R0	0x00034180
MCS4_CH3_R1	0x00034184
MCS4_CH3_R2	0x00034188
MCS4_CH3_R3	0x0003418C
MCS4_CH3_R4	0x00034190
MCS4_CH3_R5	0x00034194
MCS4_CH3_R6	0x00034198
MCS4_CH3_R7	0x0003419C
MCS4_CH3_CTRL	0x000341A0
MCS4_CH3_ACB	0x000341A4
MCS4_CH3_PC	0x000341C0
MCS4_CH3_IRQ_NOTIFY	0x000341C4
MCS4_CH3_IRQ_EN	0x000341C8
MCS4_CH3_IRQ_FORCINT	0x000341CC
MCS4_CH3_IRQ_MODE	0x000341D0
MCS4_CH3_EIRQ_EN	0x000341D4
MCS4_CH4_R0	0x00034200
MCS4_CH4_R1	0x00034204
MCS4_CH4_R2	0x00034208
MCS4_CH4_R3	0x0003420C
MCS4_CH4_R4	0x00034210
MCS4_CH4_R5	0x00034214
MCS4_CH4_R6	0x00034218
MCS4_CH4_R7	0x0003421C
MCS4_CH4_CTRL	0x00034220
MCS4_CH4_ACB	0x00034224
MCS4_CH4_PC	0x00034240
MCS4_CH4_IRQ_NOTIFY	0x00034244
MCS4_CH4_IRQ_EN	0x00034248
MCS4_CH4_IRQ_FORCINT	0x0003424C
MCS4_CH4_IRQ_MODE	0x00034250
MCS4_CH4_EIRQ_EN	0x00034254
MCS4_CH5_R0	0x00034280
MCS4_CH5_R1	0x00034284
MCS4_CH5_R2	0x00034288
MCS4_CH5_R3	0x0003428C
MCS4_CH5_R4	0x00034290
MCS4_CH5_R5	0x00034294

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS4_CH5_R6	0x00034298
MCS4_CH5_R7	0x0003429C
MCS4_CH5_CTRL	0x000342A0
MCS4_CH5_ACB	0x000342A4
MCS4_CH5_PC	0x000342C0
MCS4_CH5_IRQ_NOTIFY	0x000342C4
MCS4_CH5_IRQ_EN	0x000342C8
MCS4_CH5_IRQ_FORCINT	0x000342CC
MCS4_CH5_IRQ_MODE	0x000342D0
MCS4_CH5_EIRQ_EN	0x000342D4
MCS4_CH6_R0	0x00034300
MCS4_CH6_R1	0x00034304
MCS4_CH6_R2	0x00034308
MCS4_CH6_R3	0x0003430C
MCS4_CH6_R4	0x00034310
MCS4_CH6_R5	0x00034314
MCS4_CH6_R6	0x00034318
MCS4_CH6_R7	0x0003431C
MCS4_CH6_CTRL	0x00034320
MCS4_CH6_ACB	0x00034324
MCS4_CH6_PC	0x00034340
MCS4_CH6_IRQ_NOTIFY	0x00034344
MCS4_CH6_IRQ_EN	0x00034348
MCS4_CH6_IRQ_FORCINT	0x0003434C
MCS4_CH6_IRQ_MODE	0x00034350
MCS4_CH6_EIRQ_EN	0x00034354
MCS4_CH7_R0	0x00034380
MCS4_CH7_R1	0x00034384
MCS4_CH7_R2	0x00034388
MCS4_CH7_R3	0x0003438C
MCS4_CH7_R4	0x00034390
MCS4_CH7_R5	0x00034394
MCS4_CH7_R6	0x00034398
MCS4_CH7_R7	0x0003439C
MCS4_CH7_CTRL	0x000343A0
MCS4_CH7_ACB	0x000343A4
MCS4_CH7_PC	0x000343C0
MCS4_CH7_IRQ_NOTIFY	0x000343C4
MCS4_CH7_IRQ_EN	0x000343C8
MCS4_CH7_IRQ_FORCINT	0x000343CC

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS4_CH7_IRQ_MODE	0x000343D0
MCS4_CH7_EIRQ_EN	0x000343D4
MCS5_CH0_R0	0x00035000
MCS5_CH0_R1	0x00035004
MCS5_CH0_R2	0x00035008
MCS5_CH0_R3	0x0003500C
MCS5_CH0_R4	0x00035010
MCS5_CH0_R5	0x00035014
MCS5_CH0_R6	0x00035018
MCS5_CH0_R7	0x0003501C
MCS5_CH0_CTRL	0x00035020
MCS5_CH0_ACB	0x00035024
MCS5_CTRG	0x00035028
MCS5_STRG	0x0003502C
MCS5_CH0_PC	0x00035040
MCS5_CH0_IRQ_NOTIFY	0x00035044
MCS5_CH0_IRQ_EN	0x00035048
MCS5_CH0_IRQ_FORCINT	0x0003504C
MCS5_CH0_IRQ_MODE	0x00035050
MCS5_CH0_EIRQ_EN	0x00035054
MCS5_CTRL	0x00035074
MCS5_RST	0x00035078
MCS5_ERR	0x0003507C
MCS5_CH1_R0	0x00035080
MCS5_CH1_R1	0x00035084
MCS5_CH1_R2	0x00035088
MCS5_CH1_R3	0x0003508C
MCS5_CH1_R4	0x00035090
MCS5_CH1_R5	0x00035094
MCS5_CH1_R6	0x00035098
MCS5_CH1_R7	0x0003509C
MCS5_CH1_CTRL	0x000350A0
MCS5_CH1_ACB	0x000350A4
MCS5_CH1_PC	0x000350C0
MCS5_CH1_IRQ_NOTIFY	0x000350C4
MCS5_CH1_IRQ_EN	0x000350C8
MCS5_CH1_IRQ_FORCINT	0x000350CC
MCS5_CH1_IRQ_MODE	0x000350D0
MCS5_CH1_EIRQ_EN	0x000350D4
MCS5_CH2_R0	0x00035100

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS5_CH2_R1	0x00035104
MCS5_CH2_R2	0x00035108
MCS5_CH2_R3	0x0003510C
MCS5_CH2_R4	0x00035110
MCS5_CH2_R5	0x00035114
MCS5_CH2_R6	0x00035118
MCS5_CH2_R7	0x0003511C
MCS5_CH2_CTRL	0x00035120
MCS5_CH2_ACB	0x00035124
MCS5_CH2_PC	0x00035140
MCS5_CH2_IRQ_NOTIFY	0x00035144
MCS5_CH2_IRQ_EN	0x00035148
MCS5_CH2_IRQ_FORCINT	0x0003514C
MCS5_CH2_IRQ_MODE	0x00035150
MCS5_CH2_EIRQ_EN	0x00035154
MCS5_CH3_R0	0x00035180
MCS5_CH3_R1	0x00035184
MCS5_CH3_R2	0x00035188
MCS5_CH3_R3	0x0003518C
MCS5_CH3_R4	0x00035190
MCS5_CH3_R5	0x00035194
MCS5_CH3_R6	0x00035198
MCS5_CH3_R7	0x0003519C
MCS5_CH3_CTRL	0x000351A0
MCS5_CH3_ACB	0x000351A4
MCS5_CH3_PC	0x000351C0
MCS5_CH3_IRQ_NOTIFY	0x000351C4
MCS5_CH3_IRQ_EN	0x000351C8
MCS5_CH3_IRQ_FORCINT	0x000351CC
MCS5_CH3_IRQ_MODE	0x000351D0
MCS5_CH3_EIRQ_EN	0x000351D4
MCS5_CH4_R0	0x00035200
MCS5_CH4_R1	0x00035204
MCS5_CH4_R2	0x00035208
MCS5_CH4_R3	0x0003520C
MCS5_CH4_R4	0x00035210
MCS5_CH4_R5	0x00035214
MCS5_CH4_R6	0x00035218
MCS5_CH4_R7	0x0003521C
MCS5_CH4_CTRL	0x00035220

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS5_CH4_ACB	0x00035224
MCS5_CH4_PC	0x00035240
MCS5_CH4_IRQ_NOTIFY	0x00035244
MCS5_CH4_IRQ_EN	0x00035248
MCS5_CH4_IRQ_FORCINT	0x0003524C
MCS5_CH4_IRQ_MODE	0x00035250
MCS5_CH4_EIRQ_EN	0x00035254
MCS5_CH5_R0	0x00035280
MCS5_CH5_R1	0x00035284
MCS5_CH5_R2	0x00035288
MCS5_CH5_R3	0x0003528C
MCS5_CH5_R4	0x00035290
MCS5_CH5_R5	0x00035294
MCS5_CH5_R6	0x00035298
MCS5_CH5_R7	0x0003529C
MCS5_CH5_CTRL	0x000352A0
MCS5_CH5_ACB	0x000352A4
MCS5_CH5_PC	0x000352C0
MCS5_CH5_IRQ_NOTIFY	0x000352C4
MCS5_CH5_IRQ_EN	0x000352C8
MCS5_CH5_IRQ_FORCINT	0x000352CC
MCS5_CH5_IRQ_MODE	0x000352D0
MCS5_CH5_EIRQ_EN	0x000352D4
MCS5_CH6_R0	0x00035300
MCS5_CH6_R1	0x00035304
MCS5_CH6_R2	0x00035308
MCS5_CH6_R3	0x0003530C
MCS5_CH6_R4	0x00035310
MCS5_CH6_R5	0x00035314
MCS5_CH6_R6	0x00035318
MCS5_CH6_R7	0x0003531C
MCS5_CH6_CTRL	0x00035320
MCS5_CH6_ACB	0x00035324
MCS5_CH6_PC	0x00035340
MCS5_CH6_IRQ_NOTIFY	0x00035344
MCS5_CH6_IRQ_EN	0x00035348
MCS5_CH6_IRQ_FORCINT	0x0003534C
MCS5_CH6_IRQ_MODE	0x00035350
MCS5_CH6_EIRQ_EN	0x00035354
MCS5_CH7_R0	0x00035380

Table continues on the next page...

Table B-3. Full addresses of GTM registers (continued)

MCS5_CH7_R1	0x00035384
MCS5_CH7_R2	0x00035388
MCS5_CH7_R3	0x0003538C
MCS5_CH7_R4	0x00035390
MCS5_CH7_R5	0x00035394
MCS5_CH7_R6	0x00035398
MCS5_CH7_R7	0x0003539C
MCS5_CH7_CTRL	0x000353A0
MCS5_CH7_ACB	0x000353A4
MCS5_CH7_PC	0x000353C0
MCS5_CH7_IRQ_NOTIFY	0x000353C4
MCS5_CH7_IRQ_EN	0x000353C8
MCS5_CH7_IRQ_FORCINT	0x000353CC
MCS5_CH7_IRQ_MODE	0x000353D0
MCS5_CH7_EIRQ_EN	0x000353D4
MCS0_MEM	0x00038000
MCS0_MEM_END	0x0003BFFC
MCS1_MEM	0x00040000
MCS1_MEM_END	0x00043FFC
MCS2_MEM	0x00048000
MCS2_MEM_END	0x0004BFFC
MCS3_MEM	0x00050000
MCS3_MEM_END	0x00053FFC
MCS4_MEM	0x00058000
MCS4_MEM_END	0x0005BFFC
MCS5_MEM	0x00060000
MCS5_MEM_END	0x00063FFC

B.5 Advanced Routing Unit (ARU)

ARU Round Trip Time:

- The ARU round trip time is 113 * SYS_CLK.

ARU-to-submodule write addresses are shown in the following table.

Table B-4. ARU-to-submodule write addresses

Name	Address
ARU_ACCESS	0x000
TIM [0–2]	

Table continues on the next page...

Table B-4. ARU-to-submodule write addresses (continued)

Name	Address
TIM0_WRADDR[0–7]	0x001–0x008
TIM1_WRADDR[0–7]	0x009–0x010
TIM2_WRADDR[0–7]	0x011–0x018
TIM3_WRADDR[0–7]	0x019–0x020
TIM4_WRADDR[0–7]	0x021–0x028
TIM5_WRADDR[0–7]	0x029–0x030
Unused	0x031–0x038
DPLL	
DPLL_WRADDR[0–23]	0x039–0x050
F2A	
F2A0_WRADDR[0–7]	0x051–0x058
F2A1_WRADDR[0–7]Unused	0x059–0x060
BRC	
BRC_WRADDR[0–21]	0x061–0x076
MCS [0–2]	
MCS0_WRADDR[0–23]	0x077–0x08E
MCS1_WRADDR[0–23]	0x08F–0x0A6
MCS2_WRADDR[0–23]	0x0A7–0x0BE
MCS3_WRADDR[0–23]	0x0BF–0x0D6
MCS4_WRADDR[0–23]	0x0D7–0x0EE
MCS5_WRADDR[0–23]	0x0EF–0x106
Unused	0x107–0x11E
ATOM [0–3]	
ATOM0_WRADDR[0–7]	0x11F–0x126
ATOM1_WRADDR[0–7]	0x127–0x12E
ATOM2_WRADDR[0–7]	0x12F–0x136
ATOM3_WRADDR[0–7]	0x137–0x13E
ATOM4_WRADDR[0–7]	0x13F–0x146
ATOM5_WRADDR[0–7]	0x147–0x14E
ATOM6_WRADDR[0–7]	0x14F–0x156
ATOM7_WRADDR[0–7]	0x157–0x15E
ATOM8_WRADDR[0–7]	0x15F–0x166
Unused	0x167–0x16E
Unused	0x16F–0x176
Unused	0x177–0x17E
Miscellaneous	
Unused	0x17F–0x1FD
ARU_EMPTY_ADDR	0x1FE
ARU_FULL_ADDR	0x1FF

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Document Number GTM104RM
Revision 1, 3 Feb 2015

freescale™