Integrated Flash Controller (IFC) Configuration on QorIQ Custom Boards

The integrated flash controller (IFC) is used to interface with external asynchronous/synchronous NAND flash, asynchronous NOR flash, SRAM, generic ASIC memory and EPROM. This document introduces how to configure IFC controller on QorIQ LS, T and P series custom boards, uses LS1043 custom board integrating NAND Flash MT29F64G08CBCBBH1 as an example to demonstrate IFC flash timing parameters calculation and control registers configuration, CodeWarrior initialization file customization and u-boot source code porting.

1. IFC Memory Mapped Registers Introduction

CSPR_EXT: Extended Chip Select Property registers.

Field	Description
0–23 -	This field is reserved.
24–31 BA_EXT	Extended Base Address: This field contains the msbs of the base address. Complete base address can be represented by concatenating CSPRn_EXT[BA_EXT] and CSPR*n*[BA] fields. Each chip selects's base register value is compared to the address on the address bus to determine if the master is accessing a memory bank controlled by the IFC. Address decoding is performed by using the address mask bit in AMASK*n*[AM] field more details of decoding is given in AMASK register description.

CSPR: Chip-select property register contains the base address and memory attributes for each bank.

Field	Description
0–15 BA	Base Address: Each base register value is compared to the address on the address bus to determine if the master is accessing a memory bank controlled by the IFC. Used with the address mask bit.
16–22 -	This field is reserved.
23–24 PS	Port Size-Specifies the port size of this memory region. For CSPR0, PS is configured through reset configuration word as loaded during power on reset. For all other banks the value is reset to 00 (port size not defined). 00 Reserved 01 8 bit 10 16 bit 11 Reserved
25 WP	Write Protect: **NOTE:** 1. This bit is valid only for NAND and NOR; for GPCM this bit should not be set. **NOTE:** 2. If CS0 is used for booting from NAND or NOR, CSPR0[WP] will be set (write protected). Software must clear this bit after completing the boot operation to permit write operations on CS0 0 Read and write accesses are allowed. 1 Only read accesses are allowed. The memory controller does not assert chip-select on write cycles to this memory bank for NOR flash. (Refer Write protect for more details)
26 -	This field is reserved.
27 TE	External Transceiver Enable. It specifies the value that will be driven on TE pin when a particular CSn is selected. 0 Logic 0 will be driven on TE pin 1 Logic 1 will be driven on TE pin
28 -	This field is reserved.
29–30 MSEL	Machine Select 00 NOR flash 01 NAND flash 10 GPCM 11 Reserved
31 V	Valid: Indicates that the contents of CSPRn are valid. 1 Bank is valid 0 Bank is invalid

AMASK: Address mask register

Field	Description
0–15 AM	16-bit Address mask corresponding to memory bank
16–31 -	This field is reserved.

CSOR: Chip-select option register-NAND Flash mode.

Field	Description
0 ECC_ENC_EN	ECC Encoder Enable/Disable bit: **NOTE:** The encoder should be enabled only when performing full page operations. In case of partial page operations, the encoder should be disabled. 0 ECC encoding disabled 1 ECC Encoding enabled
1 -	This field is reserved.
2–3 ECC_MODE	ECC Mode of operation 00 4 bit correction per 512 byte data sector 01 8 bit correction per 512 byte data sector 10 24 bit correction per 1 KB sector 11 40 bit correction per 1 KB sector
4 -	This field is reserved.
5 ECC_DEC_EN	ECC Decoding Enable/Disable bit **NOTE:** The decoder should be enabled only when performing full page operations. In case of partial page operations, the decoder should be disabled 0 ECC decoding disabled 1 ECC decoding enabled
6 -	This field is reserved.
7–8 RAL	Row Address Length: Number of address bytes issued during page address operation **NOTE:** Bytes for column address are determined by page size 00 1 byte 01 2 bytes 10 3 bytes 11 4 bytes
9–10 -	This field is reserved.
11–12 PGS	Page Size 00 512 Bytes 01 2 KB 10 4 KB 11 8 KB
13–15 -	This field is reserved.
16–18 SPRZ	Spare size **NOTE:** Depending on the ECC mode and page size, a fixed value of the spare region is selected during boot. For more information, see Booting methods. Others Reserved

Field	Description
	000 16 Bytes 001 64 Bytes 010 128 Bytes 011 210 Bytes 100 218 Bytes 101 224 Bytes 110 Spare size information will be used from corresponding CSORn_EXT register.
19–20 -	This field is reserved.
21–23 PB	Pages per Block Others Reserved 000 32 Pages 001 64 Pages 010 128 Pages 011 256 Pages 100 512 Pages
24 -	This field is reserved.
25–26 NAND_MODE	NAND mode of operation Others Reserved 00 Asynchronous mode 01 NVDDR 10 Reserved 11 Reserved
27–29 TRHZ	Time for read enable high to output high impedance (Z). Number of clocks required for memory to go in high-Z after read enable deassertion. This field is used during last read data access. If the IFC is accessing the NAND flash for a read operation, then after the last byte is read the NAND FSM must wait for TRHZ clock cycles so that there is no contention on the external buffer. Other settings are Reserved. 000 Wait for 20 IP Clocks 001 Wait for 40 IP Clocks 010 Wait for 60 IP Clocks 011 Wait for 80 IP Clocks 100 Wait for 100 IP Clocks
30 -	This field is reserved.
31 BCTLD	Buffer control disable. This bit signifies presence or absence of external buffer. If buffer is absent then this bit should be set to 1. 0 BCTL is actively driven by IFC based on direction of access. 1 BCTL is set to its default value (high) irrespective of direction of access.

IFC_FTIM0_CSn_NAND: Flash Timing register 0 for Chip Select n, field description is as the following.

Field	Description
7 -	This field is reserved.
8–15 TWP	Write enable (WE) pulse width 00000000 Reserved 00000001 1 IFC module input clock 00000010 2 IFC module input clocks ... 11111111 255 IFC module input clocks
16–17 -	This field is reserved.
18–23 TWCHT	WE to command hold time 000000 Reserved 000001 1 IFC module input clock 000010 2 IFC module input clocks ... 111111 63 IFC module input clocks
24–25 -	This field is reserved.
26–31 TWH	WE high hold time 000000 Reserved 000001 1 IFC module input clock 000010 2 IFC module input clocks ... 111111 63 IFC module input clocks

IFC_FTIM1_CSn_NAND: Flash Timing register 1 for Chip Select n, field description is as the following.

Field	Description
0–7 TADLE	Effective address to data loading time 00000000 Reserved 00000001 1 IFC module input clock 00000010 2 IFC module input clocks ... 11111111 255 IFC module input clocks
8–15 TWB	Clock Rising Edge to SR[6] (R/B) low
16–17 -	This field is reserved.
18–23 TRR	Ready busy high to read enable (RE) low time 000000 Reserved 000001 1 IFC module input clock 000010 2 IFC module input clocks ... 111111 63 IFC module input clocks
24–31 TWRCK	W/R low to data output cycle

IFC_FTIM2_CSn_NAND: Flash Timing register 2 for Chip Select n, field description is as the following.

Field	Description
3–10 TRAD	Flash read access delay: It represents the data sampling time of read data. It should be programmed such that sampling is done at the center of the received data eye. For calculation of data eye width and appropriate data sampling time refer to NAND asynchronous mode calculating read data window width . 00000000 Reserved 00000001 1 IFC module input clock 00000010 2 IFC module input clocks ... 11111111 255 IFC module input clocks
11–14 -	This field is reserved.
15–20 TREH	RE_B High Time 000000 Reserved 000001 1 IFC module input clock 000010 2 IFC module input clocks ... 111111 63 IFC module input clocks
21–23 -	This field is reserved.
24–31 TWHRE	WE_B High to RE_B Low Effective 00000000 Reserved 00000001 1 IFC module input clock 00000010 2 IFC module input clocks ... 11111111 255 IFC module input clocks

2. Calculate IFC Flash Timing Values and Configure Control Registers

The user could use the attached table to calculate IFC timing FTIM0, FTIM1, FTIM2, CSPR and CSOR registers.

The following tables are from MT29F64G08CBCBBH1 data sheet, these parameters are related with IFC timing calculation.

Parameter	Symbol	Mode 0		Mode 1		Mode 2		Mode 3		Mode 4		Mode 5		Unit	Notes
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max		
WE# HIGH hold time	^{t}WH	30	–	15	–	15	–	10	–	10	–	7	–	ns	
WE# HIGH to RE# LOW	^{t}WHR	120	–	80	–	80	–	60	–	60	–	60	–	ns	
WE# pulse width	^{t}WP	50	–	25	–	17	–	15	–	12	–	10	–	ns	
WP# transition to WE# LOW	^{t}WW	100	–	100	–	100	–	100	–	100	–	100	–	ns	

Parameter	Symbol	Mode 0		Mode 1		Mode 2		Mode 3		Mode 4		Mode 5		Unit	Notes
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max		
Clock period		100		50		35		30		25		20		ns	
Frequency		≈10		≈20		≈28		≈33		≈40		≈50		MHz	
ALE to data start	^{t}ADL	200	–	100	–	100	–	100	–	70	–	70	–	ns	1
ALE hold time	^{t}ALH	20	–	10	–	10	–	5	–	5	–	5	–	ns	
ALE setup time	^{t}ALS	50	–	25	–	15	–	10	–	10	–	10	–	ns	
ALE to RE# delay	^{t}AR	25	–	10	–	10	–	10	–	10	–	10	–	ns	
CE# access time	^{t}CEA	–	100	–	45	–	30	–	25	–	25	–	25	ns	
CE# hold time	^{t}CH	20	–	10	–	10	–	5	–	5	–	5	–	ns	
CE# HIGH to output High-Z	^{t}CHZ	–	100	–	50	–	50	–	50	–	30	–	30	ns	2
CLE hold time	^{t}CLH	20	–	10	–	10	–	5	–	5	–	5	–	ns	
CLE to RE# delay	^{t}CLR	20	–	10	–	10	–	10	–	10	–	10	–	ns	
CLE setup time	^{t}CLS	50	–	25	–	15	–	10	–	10	–	10	–	ns	
CE# HIGH to output hold	^{t}COH	0	–	15	–	15	–	15	–	15	–	15	–	ns	
CE# setup time	^{t}CS	70	–	35	–	25	–	25	–	20	–	15	–	ns	
Data hold time	^{t}DH	20	–	10	–	5	–	5	–	5	–	5	–	ns	
Data setup time	^{t}DS	40	–	20	–	15	–	10	–	10	–	7	–	ns	
Output High-Z to RE# LOW	^{t}IR	10	–	0	–	0	–	0	–	0	–	0	–	ns	
RE# cycle time	^{t}RC	100	–	50	–	35	–	30	–	25	–	20	–	ns	
RE# access time	^{t}REA	–	40	–	30	–	25	–	20	–	20	–	16	ns	3
RE# HIGH hold time	^{t}REH	30	–	15	–	15	–	10	–	10	–	7	–	ns	3
RE# HIGH to output hold	^{t}RHOH	0	–	15	–	15	–	15	–	15	–	15	–	ns	3
RE# HIGH to WE# LOW	^{t}RHW	200	–	100	–	100	–	100	–	100	–	100	–	ns	
RE# HIGH to output High-Z	^{t}RHZ	–	200	–	100	–	100	–	100	–	100	–	100	ns	2, 3
RE# LOW to output hold	^{t}RLOH	0	–	0	–	0	–	0	–	5	–	5	–	ns	3
RE# pulse width	^{t}RP	50	–	25	–	17	–	15	–	12	–	10	–	ns	
Ready to RE# LOW	^{t}RR	40	–	20	–	20	–	20	–	20	–	20	–	ns	
WE# HIGH to R/B# LOW	^{t}WB	–	200	–	100	–	100	–	100	–	100	–	100	ns	4
WE# cycle time	^{t}WC	100	–	45	–	35	–	30	–	25	–	20	–	ns	

Fill the sheet “NandParams” in the attached table according to parameters from MT29F64G08CBCBBH1 data sheet.

Memory Model	tcls	tcs	twp	tdh	twh	trr	trp	trea	treh	twhr	twb	tadl	trc	twc
hynix_nandf_mlc_hy27uk08bgfm	15	25	15	5	10	20	15	25	10	60	100	100	30	30
hynix_nandf_slc_hy27sf161g2a	25	35	25	10	15	20	25	30	15	60	100	100	50	45
hynix_nandf_slc_hy27uf164g2b	12	20	12	5	10	20	12	20	10	80	100	70	25	25
hynix_nandf_slc_hy27us08281a	0	0	25	10	15	20	25	30	15	60	100	0	50	50
micron_nandf_onfi2_mt29h8g08aca_mode0	50	70	50	20	30	40	50	40	30	120	200	200	100	100
micron_nandf_slc_mt29f2g08aaa	0	0	25	10	15	20	25	35	15	60	100	0	50	45
micron_nandf_slc_mt29f2g16aac	25	35	25	10	15	20	25	30	15	60	100	100	50	45
MT29F64G08CBCBBH1	50	70	50	20	30	40	50	40	30	120	200	200	100	100

twc	trhz	pagePerBlock	pageDataSize	pageSpareSize	MLCdevic	datai	BlockCount	EDO	trhoh (to	trloh	trp(mod)	tww
30	50	64	2048	64	0	8	32768	15	15	5	20	
45	50	64	1024	32	0	16	1024	25	10	0	35	
25	100	64	1024	32	0	16	4096	13	15	5	15	
50	30	32	512	16	0	8	1024	25	10	0	35	
100	200	128	4096	224	1	8	2048	50	0	0	70	
45	30	64	2048	64	0	8	2048	25	15	0	35	
45	30	64	1024	32	0	16	2048	25	15	0	35	
100	200	256	8192	744	1	8	2048	50	0	0	50	100

Fill clock frequency in “NandIFCParams” sheet and get the calculated timing in the attached table, please refer to the following.

Clock Frequency
Pad Delay

S.No.	Register ->	FTIM0				FTIM1				FTIM2			FTIM3
	Memory Model	TCCST	TWP	TWCHT	TWH	TADLE	TWBE	TRR	TRP	TRAD	TREH	TWHRE	TWW
1	hynix_nandf_mlc_hy27uk08bgfm	4	7	2	4	27	30	7	7	10	4	17	
2	hynix_nandf_slc_hy27sf161g2a	4	10	4	5	25	29	7	12	12	5	15	
3	hynix_nandf_slc_hy27uf164g2b	3	5	2	4	18	30	7	5	8	4	24	
4	hynix_nandf_slc_hy27us08281a	1	12	4	5	1	29	7	12	12	5	15	
5	micron_nandf_onfi2_mt29h8g08aca_mode0	7	24	7	10	50	57	14	24	16	10	30	
6	micron_nandf_slc_mt29f2g08aaa	1	10	4	5	1	29	7	12	14	5	15	
7	micron_nandf_slc_mt29f2g16aac	4	10	4	5	25	29	7	12	12	5	15	
8	MT29F64G08CBCBBH1	7	24	7	10	50	57	14	24	15	10	30	
9													

330 MHz
2 ns

FTIM0				FTIM0	FTIM1				FTIM1	FTIM2			FTIM2	CSPR0	CS0R				NandMon	CSPR0	CSOR
NB0	NB1	NB2	NB3	[0:31]	NB0	NB1	NB2	NB3	[0:31]	[0:31]	[0:31]	[0:31]	[0:31]	[16:31]	RAL	PGS	SPRZ	PB	Trea	PS	THRZ
08	07	02	04	08070204	1B	1E	07	07	1B1E0707	014	020	11	01402011	0083	01	01	001	001	9	01	000
08	0A	04	05	080A0405	19	1D	07	0C	191D070C	018	028	0F	0180280F	0103	00	01	000	001	10	10	000
06	05	02	04	06050204	12	1E	07	05	121E0705	010	020	18	01002018	0103	01	01	000	001	7	10	001
02	0C	04	05	020C0405	01	1D	07	0C	011D070C	018	028	0F	0180280F	0083	00	00	000	000	10	01	000
0E	18	07	0A	0E18070A	32	39	0E	18	32390E18	020	050	1E	0200501E	0083	01	10	101	010	14	01	011
02	0A	04	05	020A0405	01	1D	07	0C	011D070C	01C	028	0F	01C0280F	0083	01	01	001	001	12	01	000
08	0A	04	05	080A0405	19	1D	07	0C	191D070C	018	028	0F	0180280F	0103	01	01	000	001	10	10	000
0E	18	07	0A	0E18070A	32	39	0E	18	32390E18	01E	050	1E	01E0501E	0083	01	11	111	011	14	01	011

3. Customize CodeWarrior Initialization File with the Calculated IFC Timing

```
##############################################################################
# IFC Initialization
##############################################################################
 # CSPR_EXT
   CCSR_M(0x0153000C + NAND_CS * 12, 0x00000000)
   # CSPR
   CCSR_M(0x01530010 + NAND_CS * 12, 0x7E800083)
   # AMASK
   CCSR_M(0x015300A0 + NAND_CS * 12, 0xFFFF0000)
   # CSOR
   CCSR_M(0x01530130 + NAND_CS * 12, 0x8498E30C)

   # IFC_FTIM0
   CCSR_M(0x015301C0 + NAND_CS * 48, 0x0E18070A)
   # IFC_FTIM1
   CCSR_M(0x015301C4 + NAND_CS * 48, 0x32390E18)
   # IFC_FTIM2
   CCSR_M(0x015301C8 + NAND_CS * 48, 0x01E0501E)
   # IFC_FTIM3
  CCSR_M(0x015301CC + NAND_CS * 48, 0x00000000)
```

4. Porting U-BOOT Source with the Calculated IFC Timing

Modify IFC configuration section in the file include/configs/ls1043ardb.h as the following.

```
#define CONFIG_NAND_FSL_IFC

#define CONFIG_SYS_NAND_BASE          0x7e800000

#define CONFIG_SYS_NAND_BASE_PHYS      CONFIG_SYS_NAND_BASE


#define CONFIG_SYS_NAND_CSPR_EXT       (0x0)
```

```
#define CONFIG_SYS_NAND_CSPR
(CSPR_PHYS_ADDR(CONFIG_SYS_NAND_BASE_PHYS) \
                | CSPR_PORT_SIZE_8      \
                | CSPR_MSEL_NAND        \
                | CSPR_V)
#define CONFIG_SYS_NAND_AMASK   IFC_AMASK(64*1024)
#define CONFIG_SYS_NAND_CSOR    (CSOR_NAND_ECC_ENC_EN   /* ECC on encode */ \
                | CSOR_NAND_ECC_DEC_EN  /* ECC on decode */ \
                | CSOR_NAND_ECC_MODE_4  /* 4-bit ECC */ \
                | CSOR_NAND_RAL_3       /* RAL = 3 Bytes */ \
                | CSOR_NAND_PGS_8K      /* Page Size = 8K */ \
                | CSOR_NAND_SPRZ_CSOR_EXT /* Spare size = 64 */ \
                | CSOR_NAND_PB(256))    /* 64 Pages Per Block */
#define CONFIG_SYS_NAND_ONFI_DETECTION
#define CONFIG_SYS_NAND_FTIM0           (FTIM0_NAND_TCCST(0x7) | \
                        FTIM0_NAND_TWP(0x18)   | \
                        FTIM0_NAND_TWCHT(0x7) | \
                        FTIM0_NAND_TWH(0xa))
#define CONFIG_SYS_NAND_FTIM1           (FTIM1_NAND_TADLE(0x32) | \
                        FTIM1_NAND_TWBE(0x39)  | \
                        FTIM1_NAND_TRR(0xe)   | \
                        FTIM1_NAND_TRP(0x18))
#define CONFIG_SYS_NAND_FTIM2    (FTIM2_NAND_TRAD(0xf) | \
                        FTIM2_NAND_TREH(0xa) | \
                        FTIM2_NAND_TWHRE(0x1e))
#define CONFIG_SYS_NAND_FTIM3           0x0
#define CONFIG_SYS_NAND_BASE_LIST       { CONFIG_SYS_NAND_BASE }
#define CONFIG_SYS_MAX_NAND_DEVICE      1
#define CONFIG_MTD_NAND_VERIFY_WRITE
#define CONFIG_CMD_NAND
#define CSOR_NAND_SPARE_BYTES 0x2E8
#define CONFIG_SYS_NAND_CSOR_EXT CSOR_NAND_SPARE_BYTES
```

```
#define CONFIG_SYS_NAND_BLOCK_SIZE      (2048 * 1024)
#ifdef CONFIG_NAND_BOOT
#define CONFIG_SPL_PAD_TO               0x20000         /* block aligned */
#define CONFIG_SYS_NAND_U_BOOT_OFFS     CONFIG_SPL_PAD_TO
#define CONFIG_SYS_NAND_U_BOOT_SIZE     (640 << 10)
#endif
```