Freescale Semiconductor
Application Note

Document Number: AN2919
Rev. 5, 12/2008

Determining the I^{2}C Frequency Divider Ratio for SCL

by *Networking and Multimedia Group*
Freescale Semiconductor, Inc.
Austin, TX

The I^{2}C interface is a two-wire, bidirectional serial bus developed by Philips that provides a simple, efficient way to exchange data between integrated circuit (IC) devices. The I^{2}C interface allows a device to exchange data with other I^{2}C devices such as microcontrollers, EEPROMs, real-time clock devices, A/D converters, and LCDs.

This document explains how the frequency divider to calculate the SCL speed of the I^{2}C interface is determined for the MPC824x, MPC83xx, MPC85xx, MPC86xx, and P2020 devices. For functional characteristics of the I^{2}C interface of a processor, refer to the device reference manual.

Contents

1 I^{2}C Signals and Registers

The I^{2}C interface uses serial data (SDA) and serial clock (SCL) signals for data transfer, as shown in Table 1. When the processor is idle or acts as a slave, SCL defaults as an input. The signal patterns driven on SDA represent address, data, or read/write information at different stages of the protocol.

Table 1. SCL and SDA Signals

Signal Name	Idle State	I/O	Description
SCL (serial clock)	HIGH	I	When the processor is idle or acts as a slave, SCL defaults as an input. The unit uses SCL to synchronize incoming data on SDA. The bus is assumed to be busy when SCL is detected low.
		O	As a master, the processor drives SCL along with SDA when transmitting. As a slave, the processor drives SCL low for data pacing.
SDA (serial data)	HIGH	I	When the processor is idle or in a receiving mode, SDA defaults as an input. The unit receives data from other I^{2}C devices on SDA. The bus is assumed to be busy when SDA is detected low.
		O	When transmitting as a master or slave, the processor drives (or transmits) data on SDA synchronous to SCL.

The I^{2}C registers are used for the address, data, configuration, control, and status of the I^{2}C controller. These registers are located in either the embedded utilities memory block or the configuration, control, and status block depending on the integrated processor. The I^{2}C registers considered in this document are responsible for the digital filter sampling rate and the frequency divider ratio.

2 I^{2}C Frequency Divider Register (I2CFDR/I2C*n*FDR)

The I^{2}C frequency divider register (I2CFDR) configures the I^{2}C sampling and clock rate for the MPC824x, MPC83xx, MPC85xx, and MPC86xx devices. This register is located at 0x0_3004 or 0x004 offset from the base address of one of the following registers, depending on the device:

- Embedded utilities memory block base address register (EUMBBAR for MPC824x devices)
- Configuration, control, and status register base address register (CCSRBAR for MPC85xx and MPC86xx devices)
- Internal memory map registers base address register (IMMRBAR for MPC83xx devices)

Some devices have two I^{2}C controllers and therefore have I2C*n*FDR instead of I2CFDR. In this document, unless otherwise stated, I2CFDR refers to both I2CFDR and I2C*n*FDR. For the integrated processors with two I^{2}C controllers, the offsets of the second set of I^{2}C registers is always 0x100 higher than the first set of registers. For example, if a register offset in the first controller is at 0x0_3000, the offset of that register in the second controller is at 0x0_3100. Because the offset of I2CFDR in the first controller is 0x0_3004, the offset of the I2CFDR register in the second controller is 0x0_3104. Otherwise, the definition for the second set of registers is exactly the same as that of the first.

The I^{2}C registers of MPC82xx devices are in the EUMBBAR, which is a PCI configuration register at PCI offset 0x78. The value programmed in this register is the base address for the embedded utilities memory block. The I^{2}C registers of MPC824x, MPC83xx, MPC85xx, and MPC86xx devices are in the CCSRBAR, which is at local offset 0x0_0000. The IMMRBAR is at local offset 0x0 for MPC83xx devices. The base address in both the CCSRBAR and the IMMRBAR is the 12 most-significant bits of the window for configuration accesses including I^2C registers. For some MPC85xx and MPC86xx devices, the base address of the CCSRBAR is the 16 most-significant address bits of the window used for configuration accesses, including I^2C registers.

The offset of the I^2C register must be added to that of the EUMBAR, CCSRBAR, or IMMRBAR to obtain the specific address location of that register. For details, refer to the device reference manual.

The bit assignments for the frequency divider ratio (FDR) in the I2CFDR differ among devices, as shown in Table 2, but the definition is the same. The FDR is used to create the SCL speed and to prescale the clock for bit rate selection. The SCL serial bit clock frequency is equal to the <source clock> divided by the divider. The frequency divider value can be changed at any point in a program.

Table 2. FDR Bit and Address Location Summary

Bit Name (and Assignment)	Register Name	Offset	Device Name/Family
FDR (5–0)	I^2C frequency divider register (I2CFDR)	0x0_3004	MPC824x
FDR (2–7)	I^2C frequency divider register (I2CFDR)	0x0_3004	MPC8540/41/60/55E
FDR (2–7)	I^2Cn frequency divider register (I2CFDR)	0x0_3004, 0x3104	MPC83xx, MPC85xx where xx is 33/36/43/44/45/47/48/67/68/72[E], MPC86xx, and P2020.

The bit values of the FDR field determine the divider ratio used to create the SCL speed from the source clock of the I^2C controller. The divider ratio is determined by the value programmed in the FDR as well as by the digital filter sampling rate programmed in the I^2C controller. The method used to determine this divider is discussed later in this document. The divider for an FDR value is based on the digital filter frequency sampling rate for the part, as follows:

- For MPC824x devices, it is determined by the value programmed in the DFFSR located in the I2CFDR.
- For MPC83xx, MPC85xx, and MPC86xx devices, it is determined by the value programmed in the DFSR located in the I^2C digital filter sampling rate register (I2CDFSRR).

The definition of the DFSSR and DFSR bits is the same. To assist in filtering out signal noise, the sample rate is programmable. This field is used to prescale the frequency at which the digital filter takes samples from the I^2C bus. The resulting sampling rate is the <source clock> divided by the non-zero value set in this field.

Table 3 provides the register information for programming the I^2C digital filter sampling rate for various integrated processors.

Table 3. Digital Filter Sampling Rate Bit and Address Location Summary

Bits Name (and Location)	Register Name	Offset	Device Name/Family
DFFSR (13–8)	I^2C frequency divider register (I2CFDR)	0x0_3004	MPC824x

Table 3. Digital Filter Sampling Rate Bit and Address Location Summary (continued)

Bits Name (and Location)	Register Name	Offset	Device Name/Family
DFSR (2–7)	I^{2}C digital filter sampling rate register (I2CDFSRR)	0x0_3014	MPC8540/41/60/55E
DFSR (2–7)	I^{2}C*n* digital filter sampling rate registers (I2CDFSRR)	0x0_3014, 0x3114	MPC83xx, MPC85xx where xx is 33/36/43/44/45/47/48/67/68/72[E], MPC86xx and P2020.

Notice that bits 13 through 8 in the DFFSR for the MPC824x devices correspond to bits 2 through 7 in the DFSR for the MPC83xx, MPC85xx, and MPC86xx devices. In this document, DFSR refers to both DFFSR and DFSR because their definitions are the same and they are used the same way to change the I^{2}C sampling rate of different integrated processors. For details on I2CFDR and I2CDFSRR, consult the device reference manual.

3 Determining the Frequency Divider

The bit values of the FDR and DFSR are used to determine the divider ratio between the source clock and the SCL speeds.

3.1 Source Clock

The frequency of SCL is based on a source clock divided by the FDR ratio. As Table 4 shows, this source clock is not the same for the all integrated processors. For details, consult the device reference manuals.

Table 4. Source Clock for SCL

Source Clock	Description	Device Name/Family
SDRAM clock	The value for the memory clock is based on the peripheral PLL ratio and the input clock for the part (OSC_IN/PCI_SYNC_IN)	MPC824x
MPX/CCB clock	The value for this clock is based on the platform PLL ratio and the input clock for the part (SYSCLK/PCI_CLK).	MPC8540/41/60/55E MPC8610
Half of MPX/CCB clock	The value for this clock is based on half of the platform PLL ratio and the input clock for the part (SYSCLK/PCI_CLK).	MPC85xx where xx is 36/43/45/47/48/67/68/72[E], MPC8641(D), and P2020.
Other CCB Clock ratio	The value for this clock is based on half of the platform PLL ratio and the input clock for the part, which is divided by the security PLL ratio.	MPC8533/44—the ratio can be 2:1 or 3:1(default) for I2C source clock: CCB clock depending on the CCB frequency (refer to the *MPC8544 PowerQUICC™ III Integrated Processor Hardware Specifications* for more details)
CSB clock	The value for this clock is based on the platform PLL ratio and the input clock for the part (SYSCLK/PCI_CLK). Note that the source clock for MPC83xx devices can be a divider of the CSB clock depending on the value programmed in the TSEC2CM bits of the system clock control register. For more details see the reference manual.	MPC83xx

3.2 DFSR

The 6-bit field of the DFSR represents the sampling rate of the digital filter, which factors into the frequency of the SCL because each time the SCL clock switches, the digital sampling logic has an extra period due to bit settings in the FDR. Therefore, the value of DFSR is part of the calculation of the SCL speed. The supported values for sampling rates in the DFSR are 0x01–0x3F or decimal 1 to 63. If 0x00 is written to the DFSR, for some devices, the register returns to a default value of 0x10. Please refer to the reference manual of the appropriate device to determine the reset value of the DFSR.

3.3 FDR Bits and Sub-Dividers

FDR bit assignments vary among devices. Bits 5–0 for MPC824x devices correspond to bits 2–7 of MPC83xx, MPC85xx, and MPC86xx devices. The bit ordering for MPC824x devices is little-endian where bit 0 is the least-significant, and the bit ordering on MPC83xx, MPC85xx, and MPC86xx devices is big-endian where bit 7 is the least-significant bit. For calculating the FDR divider, the 6-bit field of the FDR is partitioned into two groups (A and B). For MPC824x devices, group A consists of bits 5, 1, and 0 and group B consists of bits 4–2. For MPC83xx, MPC85xx and MPC86xx devices, group A consists of bits 2, 6, and 7 and group B consists of bits 3–5. Each group has its own set of dividers based on its bit pattern, as shown in Table 5.

Table 5. FDR Bit Patterns for MPC8xxx Devices

Bit Values	FDR Bit Locations	Divider
000	Group A Bits 5, 1, 0 for MPC824x devices Bits 2, 6, 7 for MPC83xx, MPC85xx, and MPC86xx devices	18
	Group B Bits 4, 3, 2 for MPC824x devices Bits 3, 4, 5 for MPC83xx, MPC85xx, and MPC86xx devices	16
001	Group A Bits 5, 1, 0 for MPC824x devices Bits 2, 6, 7 for MPC83xx, MPC85xx, and MPC86xx devices	20
	Group B Bits 4, 3, 2 for MPC824x devices Bits 3, 4, 5 for MPC83xx, MPC85xx, and MPC86xx devices	32
010	Group A Bits 5, 1, 0 for MPC824x devices Bits 2, 6, 7 for MPC83xx, MPC85xx, and MPC86xx devices	24
	Group B Bits 4, 3, 2 for MPC824x devices; its 3, 4,5 for MPC83xx, MPC85xx, and MPC86xx devices	64
011	Group A Bits 5, 1, 0 for MPC824x devices Bits 2, 6, 7 for MPC83xx, MPC85xx, and MPC86xx devices	30
	Group B Bits 4, 3, 2 for MPC824x devices Bits 3, 4, 5 for MPC83xx, MPC85xx, and MPC86xx devices	128

Table 5. FDR Bit Patterns for MPC8xxx Devices (continued)

Bit Values	FDR Bit Locations	Divider
100	Group A Bits 5, 1, 0 for MPC824x devices Bits 2, 6, 7 for MPC83xx, MPC85xx, and MPC86xx devices	10
	Group B Bits 4, 3, 2 for MPC824x devices Bits 3, 4, 5 for MPC83xx, MPC85xx, and MPC86xx devices	256
101	Group A Bits 5, 1, 0 for MPC824x devices Bits 2, 6, 7 for MPC83xx, MPC85xx, and MPC86xx devices	12
	Group B Bits 4, 3, 2 for MPC824x devices Bits 3, 4, 5 for MPC83xx, MPC85xx, and MPC86xx devices	512
110	Group A Bits 5, 1, 0 for MPC824x devices Bits 2, 6, 7 for MPC83xx, MPC85xx, and MPC86xx devices	14
	Group B Bits 4, 3, 2 for MPC824x devices Bits 3, 4, 5 for MPC83xx, MPC85xx, and MPC86xx devices	1024
111	Group A Bits 5, 1, 0 for MPC824x devices Bits 2, 6, 7 for MPC83xx, MPC85xx, and MPC86xx devices	16
	Group B Bits 4, 3, 2 for MPC824x devices Bits 3, 4, 5 for MPC83xx, MPC85xx, and MPC86xx devices	2048

For example, if the value written in the FDR is 0x0C (0b001100), the bit patterns assigned are 000 to group A and 011 to group B. As Table 5 indicates, these values correspond to a divider of 18 for group A and 128 for group B. The dividers of groups A and B are used in the calculations to determine the final FDR divider.

4 Obtaining the Divider Equation

To simplify the equations for calculating the final divider ratio of source clock to SCL speed, the following variables are used to refer to the FDR bits:

- Variable A represents the divider obtained through group A of Table 5.
- Variable B represents the divider obtained through group B of Table 5.
- Variable C represents the decimal equivalent of the value in the DFSR.

The steps to obtain the divider for calculating the SCL are as follows:

1. Take the decimal value of C; multiply it by 3; then divide the result by B.
2. Round the answer to step 1 down to the nearest whole number, that is, 1.x should be rounded down to 1, 2.x should be rounded down to 2, and so on.
3. Take the result of step 2 and multiply it by 2.

4. Add the value of A to the results of step 3.
5. Multiply the value of B by the results of step 4 to obtain the new divider.

The above steps can be summarized in the following equation:

$$\textit{Frequency divider} = B \times [A + (\mathrm{floor}(3 \times C \div B) \times 2)] \qquad \textit{Eqn. 1}$$

This equation, along with the source clock frequency, is needed to calculate the SCL frequency where:

$$\textit{SCL Frequency} = \textit{Source clock frequency} \div \textit{Frequency divider} \qquad \textit{Eqn. 2}$$

NOTE

In order for Equation 1 to work, the conditions described in Section 4.1, "Required Conditions for the Divider Equation," must be met.

4.1 Required Conditions for the Divider Equation

Two conditions must be met for the I^2C SCL frequency to match the frequency calculated by the divider equation obtained in Section 4, "Obtaining the Divider Equation." The following variables are used in stating the conditions:

- Variable T represents the period of the I^2C source clock in nanoseconds.
- Variable t_{I2CRM} represents the measured rise time in nanoseconds of the SCL signal from 10% to 70% of V_{CC}.
- Variable B represents the divider obtained through group B of Table 5 as explained in Section 4, "Obtaining the Divider Equation."
- Variable C represents the decimal equivalent of the value in the DFSR as explained in Section 4, "Obtaining the Divider Equation."

The two conditions can be stated as follows:

- ***Condition 1***: $C \leq 50 \div T$.

This means that the product of the decimal equivalent of the value in the DFSR and the source clock period must not exceed 50 ns. Thus, given a specific source clock period, the value in the DFSR must not be so high that it violates this condition.

- ***Condition 2***: $B \times T \geq t_{I2CRM} + 3 \times C \times T$.

Given a suitable value of DFSR, chosen to satisfy ***Condition 1***, this inequality must also be met to guarantee that the SCL frequency matches the SCL frequency calculated by the divider equation. It is important to note that t_{I2CRM} is the measured rise time of the SCL signal, which is defined as the time for the signal to rise from 10% to 70% of V_{CC}.

NOTE

Note that the rise time must not exceed 300 nanoseconds and that the above two conditions must both be satisfied to ensure that the actual SCL frequency values align with the calculated values. By meeting these conditions, the measured SCL frequency will match the calculated frequency to within 5 kHz. Ignoring either of these conditions may result in larger discrepancies between these frequency values.

4.2 Case Studies

To determine the correct values of FDR and DFSR to be used, the following steps should be considered:

1. Choose C (value for DFSR) so that ***Condition 1*** from Section 4.1, "Required Conditions for the Divider Equation" is met, where $C \leq 50 \div T$.
2. Choose B (Group B bits of FDR) so that ***Condition 2*** from Section 4.1, "Required Conditions for the Divider Equation" is met, where $B \times T \geq t_{I2CRM} + 3 \times C \times T$.
3. Evaluate A (Group A bits of FDR) so that ***Eqn.1*** and ***Eqn. 2*** from Section 4, "Obtaining the Divider Equation" are met so that:
   A = [Source clock frequency ÷ (B × Desired SCL Frequency)] – [2 × floor(3 × C ÷ B)].
4. Using the values of A, B, and C, determine the expected SCL frequency.

4.2.1 Example 1 (854x Device)

Consider the case of an 854x device where the desired SCL frequency is 400 kHz and the source clock frequency is 200 MHz. Hence, the desired frequency divider is 500.

From Step 1, in Section 4.2, "Case Studies," it can be seen that,

- For ***Condition 1***, $C \leq 50 \div T$, since the input source clock is 200 MHz, the clock cycle, T, is equal to 5 ns. Therefore, C must be ≤ 10. For this example, let C = 8.

From Step 2, in Section 4.2, "Case Studies," it can be seen that,

- In order for ***Condition 2*** to be met, the rise time of the clock seen at SCL needs to be measured from 10% to 70% of V_{CC}. Since the load and voltage of the system is not expected to change, the rise time measured at this point is expected to be the same rise time when the values of the DFSR and FDR bits are changed later on. Assuming the measured rise time is 120 ns, the value of B that should work can be calculated based on the equation and the value chosen for C, which is 8.
- Rewriting ***Condition 2*** with the known values, we have $B \times 5 \geq 120 + 3 \times 8 \times 5$. Therefore, B must be chosen so that it is ≥ 48. In Table 5, it can be seen that 64 is the smallest Group B number that is greater than or equal to 48. Therefore, we choose B as 64.

From Step 3, in Section 4.2, "Case Studies," it can be seen that,

- Replacing the known values into A where,

  A = [Source clock frequency ÷ (B × SCL Frequency)] – [2 × floor(3 × C ÷ B)]

  we have:

  $A = [500 \div 64] – [\text{floor}(3 \times 8 \div 64) \times 2] = [7.8125] – [0] = 7.8125$

- That means that A needs to be the closest value to 7.875 in Table 5. In the table, the closest Group A value to 7.875 is 10.

From Step 4, in Section 4.2, “Case Studies,” it can be seen that,

- Plugging the Table 5 values of A, B, and the chosen value of C back, we see that the actual frequency divider obtained is 64 × [10 + (floor(3×8÷64) × 2)], or 640.
- Therefore, we can program the FDR and DFSR. With C = 8, the DFSR is 0b001000. Since the Group B divider value used is 64, bits 3, 4 and 5 of the MPC854x device are 0, 1, and 0, respectively, according to Table 5. Finally, using Table 5 with the Group A divider of 10, bits 2, 6, and 7 of the MPC854x device are 1, 0, and 0, respectively. Hence, the value for the FDR is 0b101000.
- With FDR = 0b101000 and DFSR = 0b001000, the expected actual frequency of SCL is 200 MHz ÷ 640, or ~ 312.5 kHz.

4.2.2 Example 2 (8610 Device)

Consider the case of an 8610 device where the desired SCL frequency is 200 kHz and the source clock frequency is 533 MHz. Hence, the desired frequency divider is 2665.

From Step 1 in Section 4.2, “Case Studies,” it can be seen that,

- For ***Condition 1***, $C \leq 50 \div T$, since the input source clock is 200 MHz, the clock cycle, T, is equal to 1.88 ns. Therefore, C must be ≤ 26. For this example, let C = 25.

From Step 2, in Section 4.2, “Case Studies,” it can be seen that,

- In order for ***Condition 2*** to be met, the rise time of the clock seen at SCL needs to be measured from 10% to 70% of V_{CC}. Since the load and voltage of the system is not expected to change, the rise time measured at this point is expected to be the same rise time when the values of the DFSR and FDR bits are changed later on. Assuming the measured rise time is 50 ns, the value of B that should work can be calculated based on the equation and the value chosen for C, which is 25.
- Rewriting ***Condition 2*** with the known values, we have $B \times 1.88 \geq 50 + 3 \times 25 \times 1.88$. Therefore B must be chosen so that it is ≥ 102. In Ταβλε 5, 128 is the smallest Group B number that is greater than or equal to 102. Therefore, we choose B as 128.

From Step 3, in Section 4.2, “Case Studies,” it can be seen that,

- Replacing the known values into A where,

  A = [Source clock frequency ÷ (B × SCL Frequency)] – [2 × floor(3 × C ÷ B)],

  we have:

  A = [2665 ÷ 128] – [floor(3 × 25 ÷ 128) × 2] = [20.82] – [0] = 20.82
- That means that A needs to be the closest value to 20.82 in Table 5. The closest Group A value to 20.82 is 20.

From Step 4, in Section 4.2, “Case Studies” it can be seen that,

- Plugging the Table 5 values of A, B, and the chosen value of C back, we see that the actual frequency divider obtained is 128 × [20 + (floor(3 × 25 ÷ 128) × 2)], or 2560.

- Therefore, we can program the FDR and DFSR. With C = 25, the DFSR is 0b011001. Since the Group B divider value used is 128, then bits 3, 4 and 5 of the MPC8610 device are 0, 1, and 1, respectively, according to Table 5. Finally, using Table 5 with the Group A divider of 20, bits 2, 6, and 7 of the MPC8610 device are 0, 0, and 1, respectively. Hence, the value for the FDR is 0b001101.
- With FDR = 0b001101 and DFSR = 0b011001, the expected actual frequency of SCL is 533 MHz ÷ 2560, or ~ 195.3 kHz.

5 FDR Divider Tables

The divider value for any combination of FDR and DFSR values can be calculated by using the method described in Section 4, "Obtaining the Divider Equation." The supported FDR values are 0x00–0x3F. The FDR divider table in the reference manual is based on the value DFSR = 0b10000, as shown in Table 6.

Table 6. FDR Based on values for DFSR = 0b10000

FDR	Divider (Decimal)	FDR	Divider (Decimal)
0x00	384	0x20	256
0x01	416	0x21	288
0x02	480	0x22	320
0x03	576	0x23	352
0x04	640	0x24	384
0x05	704	0x25	448
0x06	832	0x26	512
0x07	1024	0x27	576
0x08	1152	0x28	640
0x09	1280	0x29	768
0x0A	1536	0x2A	896
0x0B	1920	0x2B	1024
0x0C	2304	0x2C	1280
0x0D	2560	0x2D	1536
0x0E	3072	0x2E	1792
0x0F	3840	0x2F	2048
0x10	4608	0x30	2560
0x11	5120	0x31	3072
0x12	6144	0x32	3584
0x13	7680	0x33	4096
0x14	9216	0x34	5120

Table 6. FDR Based on values for DFSR = 0b10000 (continued)

FDR	Divider (Decimal)	FDR	Divider (Decimal)
0x15	10240	0x35	6144
0x16	12288	0x36	7168
0x17	15360	0x37	8192
0x18	18432	0x38	10240
0x19	20480	0x39	12288
0x1A	24576	0x3A	14336
0x1B	30720	0x3B	16384
0x1C	36864	0x3C	20480
0x1D	40960	0x3D	24576
0x1E	49152	0x3E	28672
0x1F	61440	0x3F	32768

Table 7 through Table 9 show the FDR dividers for certain values of the DFSR using the method described in Section 4, “Obtaining the Divider Equation.” Table 7 shows the FDR dividers based on a DFSR value of 0x01.

Table 7. FDR Based on a DFSR Value of 0x01

FDR	Divider (Decimal)	FDR	Divider (Decimal)
0x00	288	0x20	160
0x01	320	0x21	192
0x02	384	0x22	224
0x03	480	0x23	256
0x04	576	0x24	320
0x05	640	0x25	384
0x06	768	0x26	448
0x07	960	0x27	512
0x08	1152	0x28	640
0x09	1280	0x29	768
0x0A	1536	0x2A	896
0x0B	1920	0x2B	1024
0x0C	2304	0x2C	1280
0x0D	2560	0x2D	1536
0x0E	3072	0x2E	1792
0x0F	3840	0x2F	2048

Table 7. FDR Based on a DFSR Value of 0x01 (continued)

FDR	Divider (Decimal)	FDR	Divider (Decimal)
0x10	4608	0x30	2560
0x11	5120	0x31	3072
0x12	6144	0x32	3584
0x13	7680	0x33	4096
0x14	9216	0x34	5120
0x15	10240	0x35	6144
0x16	12288	0x36	7168
0x17	15360	0x37	8192
0x18	18432	0x38	10240
0x19	20480	0x39	12288
0x1A	24576	0x3A	14336
0x1B	30720	0x3B	16384
0x1C	36864	0x3C	20480
0x1D	40960	0x3D	24576
0x1E	49152	0x3E	28672
0x1F	61440	0x3F	32768

Table 8 shows dividers based on a DFSR value of 0x23. Note the asterisk in the divider in Section 4.2.1, “Example 1 (854x Device),” for an FDR value of 0x03.

Table 8. FDR Based on a DFSR Value of 0x23

FDR	Divider (Decimal)	FDR	Divider (Decimal)
0x00	480	0x20	352
0x01	512	0x21	384
0x02	576	0x22	416
0x03	672*	0x23	448
0x04	768	0x24	512
0x05	832	0x25	576
0x06	960	0x26	640
0x07	1152	0x27	704
0x08	1280	0x28	768
0x09	1408	0x29	896
0x0A	1664	0x2A	1024
0x0B	2408	0x2B	1152

Table 8. FDR Based on a DFSR Value of 0x23 (continued)

FDR	Divider (Decimal)	FDR	Divider (Decimal)
0x0C	2304	0x2C	1280
0x0D	2560	0x2D	1536
0x0E	3072	0x2E	1792
0x0F	3840	0x2F	2048
0x10	4608	0x30	2560
0x11	5120	0x31	3072
0x12	6144	0x32	3584
0x13	7680	0x33	4096
0x14	9216	0x34	5120
0x15	10240	0x35	6144
0x16	12288	0x36	7168
0x17	15360	0x37	8192
0x18	18432	0x38	10240
0x19	20480	0x39	12288
0x1A	24576	0x3A	14336
0x1B	30720	0x3B	16384
0x1C	36864	0x3C	20480
0x1D	40960	0x3D	24576
0x1E	49152	0x3E	28672
0x1F	61440	0x3F	32768

Table 9 shows dividers based on a DFSR value of 0x34. Note the asterisk in the divider in Section 4.2.2, “Example 2 (8610 Device),” for an FDR value of 0x0E.

Table 9. FDR Based on a DFSR Value of 0x34

FDR	Divider (Decimal)	FDR	Divider (Decimal)
0x00	576	0x20	448
0x01	608	0x21	480
0x02	672	0x22	512
0x03	768	0x23	544
0x04	832	0x24	576
0x05	896	0x25	640
0x06	1024	0x26	704
0x07	1216	0x27	768

Table 9. FDR Based on a DFSR Value of 0x34 (continued)

FDR	Divider (Decimal)	FDR	Divider (Decimal)
0x08	1408	0x28	896
0x09	1536	0x29	1024
0x0A	1792	0x2A	1152
0x0B	2176	0x2B	1280
0x0C	2560	0x2C	1536
0x0D	2816	0x2D	1792
0x0E	3328*	0x2E	2048
0x0F	4096	0x2F	2304
0x10	4608	0x30	2560
0x11	5120	0x31	3072
0x12	6144	0x32	3584
0x13	7680	0x33	4096
0x14	9216	0x34	5120
0x15	10240	0x35	6144
0x16	12288	0x36	7168
0x17	15360	0x37	8192
0x18	18432	0x38	10240
0x19	20480	0x39	12288
0x1A	24576	0x3A	14336
0x1B	30720	0x3B	16384
0x1C	36864	0x3C	20480
0x1D	40960	0x3D	24576
0x1E	49152	0x3E	28672
0x1F	61440	0x3F	32768

In some cases, the same FDR divider table can be used for various DFSR values using the first three steps of the calculation method in Section 4, “Obtaining the Divider Equation.” The constants in the first three steps can be summed up as rounding down the decimal portion of the results of the DFSR (from 0.9 and below) divided by variable B and then multiplying the remaining integer by 6. Calculations show that the DFSR values with the same integer result when divided by 5.4 (that is, 0.9 × 6) have the same FDR divider table. Therefore, DFSR values ending in decimal 5 and below have the same integer result. Similarly, DFSR values ending in decimal 6–10 have the same FDR dividers, as well as those ending in decimal 11–15.

DFSR values of 0x*n*0–0x*n*5 have the same FDR divider table, DFSR of 0x*n*6–0x*n*A have the same divider table, and DFSR of 0x*n*B–0x*n*F have the same DFSR. The *n* must be the same for this shortcut to work.

An exception occurs when 0x00 is written to DFSR because for some devices it resets to a default value of 0x10. Therefore, if the FDR divider table for DFSR = 0x10 is calculated, the divider values are the same for DFSR = 0x10–0x15. Similarly, if the FDR divider table for DFSR = 0x3B, the divider values are the same for DFSR = 0x3B–0x35. Considering these results, it is clear that Table 7 works for DFSR values of 0x01–0x05, Table 6 works for DFSR values of 0x10–0x15, Table 8 works for DFSR values of 0x20–0x25, and Table 9 works for DFSR values of 0x30–0x35.

6 Conclusion

The speed of SCL can be calculated based on the source clock speed divided by a programmable divider. The divider is calculated on the basis of the entries in the DFSR and FDR bits of the I^2C controller, if the conditions in Section 4, “Obtaining the Divider Equation,” are satisfied. For bit-specific information on DFFSR/DFSR and FDR, consult the reference manuals of the MPC824x, MPC83xx, MPC85xx and MPC86xx devices. The method described Section 3, “Determining the Frequency Divider,” along with the conditions in Section 4, “Obtaining the Divider Equation,” can be used to obtain the FDR dividers for DFSR cases not already covered in this document.

NOTE

> The required conditions provided in Section 4, “Obtaining the Divider Equation,” must be met to accurately calculate the expected divider and SCL frequency (within 5 kHz). Although this method yields the expected divider used to determine the SCL value, allow for some margin of error depending on the hardware design and the impact of the slave on SCL frequency.

7 Revision History

Table 10 provides a revision history for this application note.

Table 10. History Table

Revision Number	Date	Substantive Change(s)
5	12/2008	• Added some equation format information to Section 4, “Obtaining the Divider Equation” • Added a note to Section 4.1, “Required Conditions for the Divider Equation.” • Added Section 4.2, “Case Studies” • Changed the FDR value and example details in Section 4.2.1, “Example 1 (854x Device)” and Section 4.2.2, “Example 2 (8610 Device).”
4	11/2008	• In Table 2, Table 3, and Table 4, updated device name/family information.” • Updated sentence below Table 5 to reflect the correct FDR value. • Updated the last sentence of the first paragraph and corresponding table in Section 4, “Obtaining the Divider Equation.” • Replaced Section 3.4.1 and 3.4.2 with Section 4.2.1 and Section 4.2.2 • Added Section 4, “Obtaining the Divider Equation.” • Updated Section 6, “Conclusion,” to reflect conditions required prior to calculations.
3	01/2008	• Added text relevant to the MPC8544[E], MPC8568[E], MPC8572[E], MPC8610 and MPC8641(D).

Table 10. History Table (continued)

Revision Number	Date	Substantive Change(s)
2	04/2006	• Updated Table 4, “Source Clock for SCL,” to include correct information for source clock of MPC85xx where xx is 43/45/47/48 [E]
1	01/2006	• Renamed document to “Determining the I^{2}C Frequency Divider Ratio for SCL.” Replaced all content specific to MPC824x with information applicable to MPC824x, MPC83xx, and MPC85xx devices. Rearranged and added sections and tables.
0	06/2005	• Initial public release.

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or
+1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064
Japan
0120 191014 or
+81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800 441-2447 or
+1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

Document Number: AN2919
Rev. 5
12/2008

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor 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 which may be provided in Freescale Semiconductor 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 Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.