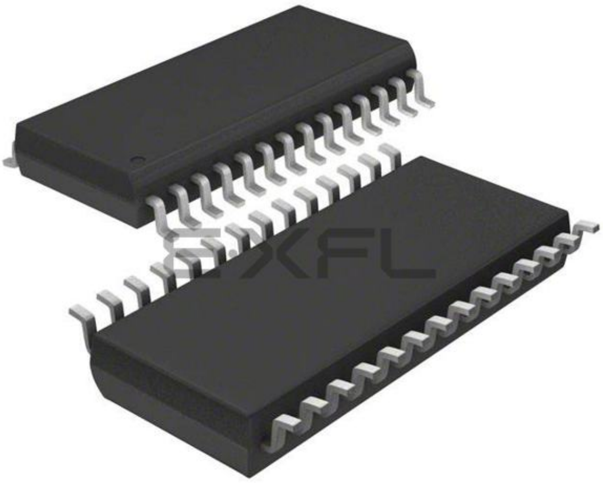
Infineon Technologies - CY8C24423-24PVIT Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

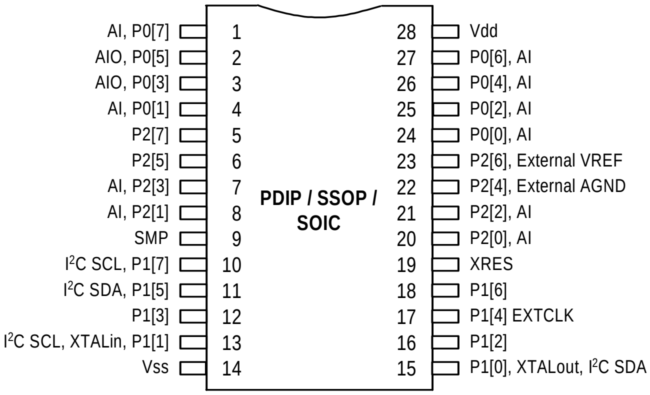
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	M8C
Core Size	8-Bit
Speed	24MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.25V
Data Converters	A/D 2x14b; D/A 2x9b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy8c24423-24pvit

Table 1-4. 28-Pin Part Pinout (PDIP, SSOP, SOIC)

Pin No.	Description
1	P0[7], A in
2	P0[5], A in, out
3	P0[3], A in, out
4	P0[1], A in
5	P2[7]
6	P2[5]
7	P2[3], A in (ASC10)
8	P2[1], A in (ASD20, ASC10)
9	SMP
10	P1[7], I^{2}C SCL
11	P1[5], I^{2}C SDA
12	P1[3]
13	P1[1], XTALin, I2C SCL
14	Vss
15	P1[0], XTALout, I2C SDA
16	P1[2],
17	P1[4], EXTCLK
18	P1[6]
19	XRES
20	P2[0], A in (ASC21)
21	P2[2], A in (ASD11, ASC21)
22	P2[4], External AGND
23	P2[6], External VREF
24	P0[0], A in
25	P0[2], A in
26	P0[4], A in
27	P0[6], A in
28	Vdd

LEGEND A: analog, D: digital, IO: input or output.



3.5.2 Source Direct

For these instructions the source address is stored in operand 1 of the instruction. During instruction execution the address will be used to retrieve the source value from RAM or register address space. The result of these instructions is placed in either the M8C A or X register as indicated by the instruction's opcode. All instructions using the Source Direct addressing mode are two bytes in length.

Table 3-7. Source Direct

Opcode	Operand 1
Instruction	Source Address

Source Direct Examples:

Source Code	**Machine Code**	**Comments**
`ADD      A, [7]`	`02 07`	The value in memory at address 7 is added to the Accumulator and the result is placed into the Accumulator.
`MOV      A, REG[8]`	`5D 08`	The value in the register space at address 8 is moved into the Accumulator.

3.5.3 Source Indexed

For these instructions the source offset from the X register is stored in operand 1 of the instruction. During instruction execution the current X register value is added to the signed offset to determine the address of the source value in RAM or register address space. The result of these instructions is placed in either the M8C A or X register as indicated by the instruction's opcode. All instructions using the Source Indexed addressing mode are two bytes in length.

Table 3-8. Source Indexed

Opcode	Operand 1
Instruction	Source Index

Source Indexed Examples:

Source Code	**Machine Code**	**Comments**
`ADD      A, [X+7]`	`03 07`	The value in memory at address X+7 is added to the Accumulator. The result is placed in the Accumulator.
`MOV      X, [X+8]`	`59 08`	The value in RAM at address X+8 is moved into the X register.

3.5.4 Destination Direct

For these instructions the destination address is stored in the machine code of the instruction. The source for the operation is either the M8C A or X register as indicated by the instruction's opcode. All instructions using the Destination Direct addressing mode are two bytes in length.

Table 3-9. Destination Direct

Opcode	Operand 1
Instruction	Destination Address

Destination Direct Examples:

Source Code	**Machine Code**	**Comments**
`ADD      [7], A`	`04 07`	The value in the Accumulator is added to memory, at address 7. The result is placed in memory at address 7. The Accumulator is unchanged.
`MOV      REG[8], A`	`60 08`	The Accumulator value is moved to register space at address 8. The Accumulator is unchanged.

Destination Indirect Post Increment Example:

Source Code	Machine Code	Comments
`MVI     [8], A`	`3F 08`	The value in memory at address 8 (the indirect address) points to a memory location in RAM. The Accumulator value is moved into the memory location pointed to by the indirect address. The indirect address in memory, at address 8, is then incremented.

3.6 Register Definitions

3.6.1 CPU_F (Flag) Register

The Flag register has four chip dependent bits (FL[7:4]) and four dedicated bits (FL[3:0]), as shown in Table 3-1.

3.6.1.1 Chip-Dependent Flag Bits

The chip-dependent Flag bits have no effect internally on the M8C. These bits are manipulated by the user with the Flag-Logic opcodes (for example, XOR F, 80h). Bit Definitions for the PSoC Mixed Signal Array family are as follows.

Bits 7, 6, and 5: Reserved.

Bit 4: XOI. IO Bank Select. This bit is used to select between register banks, in order to support more than 256 registers.

3.6.1.2 Dedicated Flag Bits

The dedicated Flag bits are described as follows.

Bit 3: Reserved.

Bit 2: Carry. Carry Flag. This bit is set or cleared in response to the result of several instructions. It may also be manipulated by the Flag-Logic opcodes (for example, OR F, 4). See the *PSoC Designer Assembly Guide User Manual* for more details.

Bit 1: Zero. Zero Flag. This bit is set or cleared in response to the result of several instructions. It may also be manipulated by the Flag-Logic opcodes (for example, OR F, 2). See the *PSoC Designer Assembly Guide User Manual* for more details.

Bit 0: GIE. Global Interrupt Enable. The state of this bit determines whether interrupts (by way of the IRQ) will be recognized by the M8C. This bit is set or cleared by the user, using the Flag-Logic opcodes (e.g., OR F, 1). GIE is also cleared automatically by the interrupt routine, after the flag byte has been stored on the stack.

For GIE=1, the M8C samples the IRQ input for each instruction. For GIE=0, the M8C ignores the IRQ.

For additional information, reference the CPU_F register on page 152.

4.1.2.5 *TableRead Function*

The TableRead function gives the user access to part-specific data stored in the Flash during manufacturing. It also returns a Revision ID for the die (not to be confused with the Silicon ID stored in Table 0).

Table 4-9. TableRead Parameters (06h)

Name	Address	Description
KEY1	0,F8h	3Ah
KEY2	0,F9h	Stack Pointer value, when SSC is executed.
BLOCKID	0,FAh	Table number to read.

Table 4-10. Table with Assigned Values in Flash Macro 0

	F8h	F9h	FAh	FBh	FCh	FDh	FEh	FFh
Table 0	Silicon ID		(May be used for serialization in the future.)					
Table 1	Voltage Reference trim for 3.3 V reg[1,EA]	Main Oscillator trim for 3.3 V reg[1,E8]	Room Temperature Calibration for 3.3V	Hot Temperature Calibration for 3.3V	Voltage Reference trim for 5 V reg[1,EA]	Main Oscillator trim for 5 V reg[1,E8]	Room Temperature Calibration for 5V	Hot Temperature Calibration for 5V
Table 2								
Table 3	M	B	Mult	M	B	Mult	00h	01h

4.1.2.6 *Checksum Function*

The Checksum function calculates a 16-bit checksum over a user specifiable number of blocks, within a single Flash macro (Bank) starting from block zero. The BLOCKID parameter is used to pass in the number of blocks to calculate the checksum over. A BLOCKID value of 1 will calculate the checksum of only block 0, while a BLOCKID value of 0 will calculate the checksum of all 256 user blocks.

The 16-bit checksum is returned in KEY1 and KEY2. The parameter KEY1 holds the lower 8 bits of the checksum and the parameter KEY2 holds the upper 8 bits of the checksum.

The checksum algorithm executes the following sequence of three instructions over the number of blocks times 64 to be checksumed.

```
romx
add [KEY1], A
adc [KEY2], 0
```

Table 4-11. Checksum Parameters (07h)

Name	Address	Description
KEY1	0,F8h	3Ah
KEY2	0,F9h	Stack Pointer value, when SSC is executed.
BLOCKID	0,FAh	Number of Flash blocks to calculate checksum on.

4.1.2.7 *Calibrate0 Function*

The Calibrate0 function transfers the calibration values stored in a special area of the Flash to their appropriate registers.

Table 4-12. Calibrate0 Parameters (08h)

Name	Address	Description
KEY1	0,F8h	3Ah
KEY2	0,F9h	Stack Pointer value, when SSC is executed.

4.1.2.8 *Calibrate1 Function*

While Calibrate1 is a completely separate function from Calibrate0, they perform the same function, which is to transfer the calibration values stored in a special area of the Flash to their appropriate registers. What is unique about Calibrate1 is that it calculates a checksum of the calibration data and, if that checksum is determined to be invalid, Calibrate1 will cause a hardware reset by setting the IRES bit of CPU_SCR1.

The Calibrate1 function uses SRAM to calculate a checksum of the calibration data. The POINTER value is used to indicate the address of a 30 byte buffer used by this function. When the function completes, the 30 bytes will be set to 00h.

Calibrate1 was created as a sub function of SWBootReset. However, the Calibrate1 function code was added to provide direct access. For more information on how Calibrate1 works, see the SWBootReset section.

Table 4-13. Calibrate1 Parameters (09h)

Name	Address	Description
KEY1	0,F8h	3Ah
KEY2	0,F9h	Stack Pointer value, when SSC is executed.
POINTER	0,FBh	First of 32 SRAM addresses used by this function.

together as DM[2:0]. Drive modes are shown in Table 6-3.

Table 6-3. Pin Drive Modes

Drive Mode DM[2:0]	Pin State	Description
000b	Resistive pull down	Strong high, resistive low
001b	Strong drive	Strong high, strong low
010b	High impedance	Hi-Z high and low, digital input enabled
011b	Resistive pull up	Resistive high, strong low
100b	Open drain high	Slow strong high, Hi-Z low
101b	Slow strong drive	Slow strong high, slow strong low
110b	High impedance, analog (**reset state**)	Hi-Z high and low, digital input disabled (for zero power) (**reset state**)
111b	Open drain low	Slow strong low, Hi-Z high

For analog IO, the drive mode should be set to one of the Hi-Z modes, either 010b or 110b. The 110b mode has the advantage that the block's digital input buffer is disabled, so no "crowbar" current flows even when the analog input is not close to either power rail. When digital inputs are needed on the same pin as analog inputs, the 010b Drive mode should be used. If the 110b Drive mode is used, the pin will always be read as a zero by the CPU and the pin will not be able to generate a useful interrupt. (It is not strictly required that a Hi-Z mode be selected for analog operation).

For global input modes, the drive mode must be set to 010b.

This GPIO provides a default drive mode of high impedance (Hi-Z). This is achieved by forcing the reset state of all PRTxDM1 and PRTxDM2 registers to FFh.

The resistive drive modes place a resistance in series with the output, for low outputs (mode 000b) or high outputs (mode 011b). Strong drive mode 001b gives the fastest edges at high DC drive strength. Mode 101b gives the same drive strength but with slower edges. The Open drain modes (100b and 111b) also use the slower edge rate drive. These modes enable open drain functions such as I^{2}C mode 111b (although the slow edge rate is not slow enough to meet the I^{2}C fast mode specification).

For additional information, reference the PRTxDM2 register on page 91, the PRTxDM0 register on page 155, and the PRTxDM1 register on page 156.

6.2.5 PRTxICx Registers

The interrupt mode for the pin is determined by bits in two registers: PRTxIC1 and PRTxIC0. These are referred to as IM1 and IM0, or together as IM[1:0].

There are four possible interrupt modes for each port pin. Two mode bits are required to select one of these modes and these two bits are spread into two different registers (PRTxIC0 and PRTxIC1). The bit position of the effected port pin (Example: Pin[2] in Port 0) is the same as the bit position of each of the Interrupt Control register bits that control the interrupt mode for that pin (Example: Bit[2] in PRT0IC0 and bit[2] in PRT0IC1). The two bits from the two registers are treated as a group.

The interrupt mode must be set to one of the non-zero modes listed in Table 6-4, in order to get an interrupt from the pin.

Table 6-4. GPIO Interrupt Modes

Interrupt Mode IM[1:0]	Description
00b	Bit interrupt disabled, INTO de-asserted
01b	Assert INTO when PIN = low
10b	Assert INTO when PIN = high
11b	Assert INTO when PIN = change from last read

The GPIO interrupt mode "disabled" (00b) disables interrupts from the pin, even if the GPIO's bit interrupt enable is on (from the PRTxIE register).

Interrupt mode 01b means that the block will assert the interrupt line (INTO) when the pin voltage is low, providing the block's bit interrupt enable line is set (high).

Interrupt mode 10b means that the block will assert the interrupt line (INTO), when the pin voltage is high, providing the block's bit interrupt enable line is set (high).

Interrupt mode 11b means that the block will assert the interrupt line (INTO) when the pin voltage is the opposite of the last state read from the pin (again providing the block's bit interrupt enable line is set high). This mode switches between low mode and high mode, depending on the last value that was read from the port during reads of the data register (PRTxDR). If the last value read from the GPIO was 0, the GPIO will subsequently be in interrupt high mode. If the last value read from the GPIO was 1, the GPIO will then be in interrupt low mode.

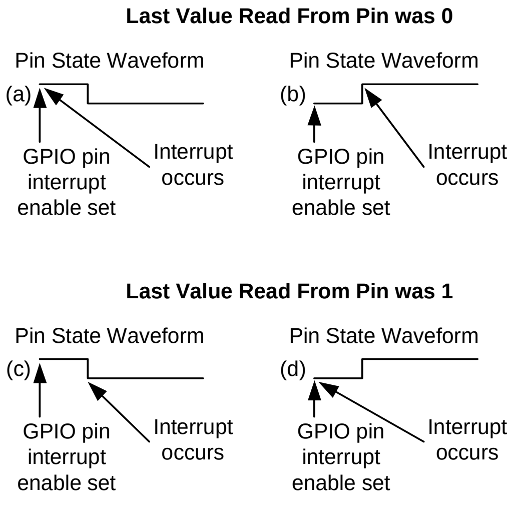

Last Value Read From Pin was 0
Pin State Waveform
(a)
GPIO pin interrupt enable set
Interrupt occurs
Pin State Waveform
(b)
GPIO pin interrupt enable set
Interrupt occurs
Last Value Read From Pin was 1
Pin State Waveform
(c)
GPIO pin interrupt enable set
Interrupt occurs
Pin State Waveform
(d)
GPIO pin interrupt enable set
Interrupt occurs

Figure 6-3. GPIO Interrupt Mode 11b

Figure 6-3 assumes that the GIE is set, GPIO interrupt mask is set, and that the GPIO interrupt mode has been set to 11b. The change interrupt mode is different from the other modes, in that it relies on the value of the GPIO's read latch to determine if the pin state has changed. Therefore, the port that contains the GPIO in question must be read during

7. Analog Output Drivers

This chapter presents the Analog Output Drivers and its associated register. The analog output drivers provide a means for driving analog signals off-chip.

Table 7-1. Analog Output Driver Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,62h	ABF_CR0	ACol1Mux		ABUF1EN0		ABUF0EN0		Bypass	PWR	RW : 00

7.1 Architectural Description

The PSoC device has two analog drivers used to output analog values on port pins. For a detailed drawing of the analog output drivers in relation to the analog system, reference the Analog Input Configuration chapter on page 245.

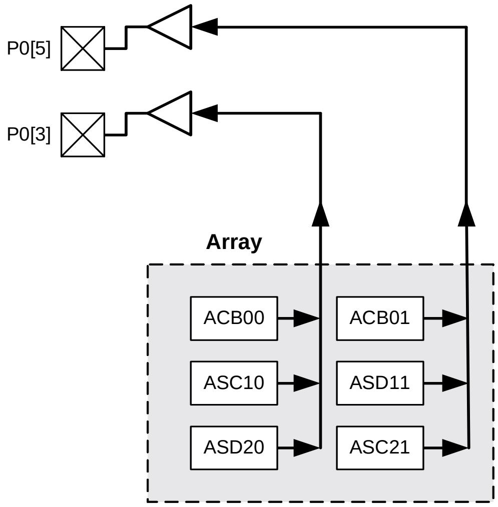


Figure 7-1. Analog Output Drivers

Each of these drivers is a resource available to all the analog blocks in a particular analog column. The user must select one analog block per column to drive a signal on its analog output bus (ABUS), to serve as the input to the analog driver for that column. The output from the analog output driver for each column can be enabled and disabled using the Analog Output Driver register ABF_CR0.

Figure 7-2.

7.2 Register Definitions

Table 7-1 presents an overview of all registers related to the analog output drivers. The following section presents a detail on the use of the register's bits.

7.2.1 ABF_CR0 Register

This register controls analog input muxes from Port 0, and the output buffer amplifiers that drive column outputs to device pins.

Bit 7: ACol1MUX. A mux selects the output of column 0 input mux or column 1 input mux. When set, this bit sets the column 1 input to column 0 input mux output.

Bit 6: Reserved.

Bits 5 and 3: ABUFxEN0. These bits enable or disable the column output amplifiers.

Bits 4 and 2: Reserved.

Bit 1: Bypass. Bypass mode connects the amplifier input directly to the output. When this bit is set, all amplifiers controlled by the register will be in bypass mode.

Bit 0: PWR. This bit is used to set the power level of the amplifiers. When this bit is set, all amplifiers controlled by the register will be in a high power.

For additional information, reference the ABF_CR0 register on page 166.

8. Internal Main Oscillator (IMO)

This chapter briefly presents the Internal Main Oscillator (IMO) and its associated register. The IMO produces clock signals of 24 MHz and 48 MHz.

Table 8-1. Internal Main Oscillator Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E8h	IMO_TR	Trim[7:0]								W : 00

8.1 Architectural Description

The Internal Main Oscillator outputs two clocks: a SYSCLK, which can be the internal 24 MHz clock or an external clock, and a SYSCLK2X that is always twice the SYSCLK frequency. In the absence of a high-precision input source from the 32 kHz crystal oscillator, the accuracy of the internal 24 MHz/48 MHz clocks will be +/-2.5% over temperature variation and two voltage ranges (3.3V +/-.3V and 5.0V +/-5%). No external components are required to achieve this level of accuracy.

There is an option to phase lock this oscillator to the External Crystal Oscillator. The choice of crystal and its inherent accuracy will determine the overall accuracy of the oscillator. The External Crystal Oscillator must be stable prior to locking the frequency of the Internal Main Oscillator to this reference source.

The IMO can be disabled when using an external clocking source. Also, the frequency doubler circuit, which produces SYSCLK2X, can be disabled to save power. Note that when using an external clock, if SYSCLK2X is needed, then the IMO can not be disabled. Registers for controlling these operations are found in the Digital Clocks chapter on page 263.

8.2 Register Definitions

8.2.1 IMO_TR Register

The device specific value for 5 volt operation is loaded into the Internal Main Oscillator Trim Register (IMO_TR) at boot time. The Internal Main oscillator will operate within specified tolerance over a voltage range of 4.75V to 5.25V, with no modification of this register. If the device is operated at a lower voltage, user code must modify the contents of this register. For operation in the voltage range of 3.3V +/-.3V, this can be accomplished with a Table Read command to the Supervisor ROM, which will supply a trim value for operation in this range. For operation between these Voltage ranges, user code can interpolate the best value using both available factory trim values.

Bits 7 to 0: Trim. These bits are used to trim the Internal Main Oscillator. A larger value in this register will increase the speed of the oscillator.

For additional information, reference the IMO_TR register on page 181.

13.1.18 CMP_CR0

Analog Comparator Bus 0 Register

Individual Register Names and Addresses

CMP_CR0: 0,64h

	7	6	5	4	3	2	1	0
Access : POR			R : 0	R : 0			RW : 0	RW : 0
Bit Name			COMP[1]	COMP[0]			AINT[1]	AINT[0]

For additional information, reference the “Register Definitions” on page 238 in the Analog Interface chapter.

Bit	Name	Description
[7:6]	**Reserved**	
[5]	**COMP[1]**	Comparator bus state for column 1. This bit is updated on the rising edge of PHI2, unless the comparator latch disable bits are set. If the comparator latch disable bits are set, then this bit is transparent to the comparator bus in the analog array.
[4]	**COMP[0]**	Comparator bus state for column 0. This bit is updated on the rising edge of PHI2, unless the comparator latch disable bits are set. If the comparator latch disable bits are set, then this bit is transparent to the comparator bus in the analog array.
[3:2]	**Reserved**	
[1]	**AINT[1]**	Controls the selection of the analog comparator interrupt for column 1. 0 The comparator data bit from the column is the input to the interrupt controller. 1 The falling edge of PHI2 for the column is the input to the interrupt controller.
[0]	**AINT[0]**	Controls the selection of the analog comparator interrupt for column 0. 0 The comparator data bit from the column is the input to the interrupt controller. 1 The falling edge of PHI2 for the column is the input to the interrupt controller.

13.1.37 RDIxLT1

Row Digital Interconnect Logic Table Register 1

Individual Register Names and Addresses

RDI0LT1 : x,B4h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0				RW : 0			
Bit Name	LUT3[3:0]				LUT2[3:0]			

For additional information, reference the “Register Definitions” on page 196 in the Row Digital Interconnect chapter.

Bit	Name	Description
[7:4]	**LUT3[3:0]**	Select logic function for LUT 3.

	Function
0h	FALSE
1h	A AND B
2h	A AND $\overline{B}$
3h	A
4h	$\overline{A}$ AND B
5h	B
6h	A XOR B
7h	A OR B
8h	A NOR B
9h	A XNOR B
Ah	$\overline{B}$
Bh	A OR $\overline{B}$
Ch	$\overline{A}$
Dh	$\overline{A}$ OR B
Eh	A NAND B
Fh	TRUE

Bit	Name	Description
[3:0]	**LUT2[3:0]**	Select logic function for LUT 2.

	Function
0h	FALSE
1h	A AND B
2h	A AND $\overline{B}$
3h	A
4h	$\overline{A}$ AND B
5h	B
6h	A XOR B
7h	A OR B
8h	A NOR B
9h	A XNOR B
Ah	$\overline{B}$
Bh	A OR $\overline{B}$
Ch	$\overline{A}$
Dh	$\overline{A}$ OR B
Eh	A NAND B
Fh	TRUE

13.1.60 MAC_X/ACC_DR1
Accumulator Data Register 1

Individual Register Names and Addresses

MAC_X/ACC_DR1: 0,ECh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

For additional information, reference the “Register Definitions” on page 270 in the Multiply Accumulate chapter.

Bit	Name	Description	
[7:0]	**Data[7:0]**	Read	Low byte of the MAC Accumulator sum. Returns the 2nd byte of the 32-bit accumulated value. The 2nd byte is next to the least significant byte for the accumulated value.
		Write	X multiplicand for the MAC 16-bit multiply and 32-bit accumulator.

Digital Register Summary

The table below lists all the PSoC registers in the digital system.

Summary Table of the Digital Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
GLOBAL DIGITAL INTERCONNECT (GDI) REGISTERS										
1,D0h	GDI_O_IN	GIONOUT7	GIONOUT6	GIONOUT5	GIONOUT4	GIONOUT3	GIONOUT2	GIONOUT1	GIONOUT0	RW : 00
1,D1h	GDI_E_IN	GIENOUT7	GIENOUT6	GIENOUT5	GIENOUT4	GIENOUT3	GIENOUT2	GIENOUT1	GIENOUT0	RW : 00
1,D2h	GDI_O_OU	GOOUTIN7	GOOUTIN6	GOOUTIN5	GOOUTIN4	GOOUTIN3	GOOUTIN2	GOOUTIN1	GOOUTIN0	RW : 00
1,D3h	GDI_E_OU	GOEUTIN7	GOEUTIN6	GOEUTIN5	GOEUTIN4	GOEUTIN3	GOEUTIN2	GOEUTIN1	GOEUTIN0	RW : 00
DIGITAL ROW REGISTERS										
x,B0h	RDI0RI	RI3[1:0]		RI2[1:0]		RI1[1:0]		RI0[1:0]		RW : 00
x,B1h	RDI0SYN					RI3SYN	RI2SYN	RI1SYN	RI0SYN	RW : 00
x,B2h	RDI0IS			BCSEL[1:0]		IS3	IS2	IS1	IS0	RW : 00
x,B3h	RDI0LT0	LUT1[3:0]				LUT0[3:0]				RW : 00
x,B4h	RDI0LT1	LUT3[3:0]				LUT2[3:0]				RW : 00
x,B5h	RDI0RO0	GOO5EN	GOO1EN	GOE5EN	GOE1EN	GOO4EN	GOO0EN	GOE4EN	GOE0EN	RW : 00
x,B6h	RDI0RO1	GOO7EN	GOO3EN	GOE7EN	GOE3EN	GOO6EN	GOO2EN	GOE6EN	GOE2EN	RW : 00
DIGITAL BLOCK REGISTERS										
Data and Control Registers										
0,20h	DBB00DR0	Data[7:0]								# : 00
0,21h	DBB00DR1	Data[7:0]								W : 00
0,22h	DBB00DR2	Data[7:0]								# : 00
0,23h	DBB00CR0	Function control/status bits for selected function[6:0]							Enable	# : 00
1,20h	DBB00FN	Data Invert	BCEN	End/Single	Mode[1:0]		Function[2:0]			RW : 00
1,21h	DBB00IN	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,22h	DBB00OU	AUXCLK		AUXEN	AUX IO Select[1:0]		OUTEN	Output Select[1:0]		RW : 00
0,24h	DBB01DR0	Data[7:0]								# : 00
0,25h	DBB01DR1	Data[7:0]								W : 00
0,26h	DBB01DR2	Data[7:0]								# : 00
0,27h	DBB01CR0	Function control/status bits for selected function[6:0]							Enable	# : 00
1,24h	DBB01FN	Data Invert	BCEN	End/Single	Mode[1:0]		Function[2:0]			RW : 00
1,25h	DBB01IN	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,26h	DBB01OU	AUXCLK		AUXEN	AUX IO Select[1:0]		OUTEN	Output Select[1:0]		RW : 00
0,28h	DCB02DR0	Data[7:0]								# : 00
0,29h	DCB02DR1	Data[7:0]								W : 00
0,2Ah	DCB02DR2	Data[7:0]								# : 00
0,2Bh	DCB02CR0	Function control/status bits for selected function[6:0]							Enable	# : 00
1,28h	DCB02FN	Data Invert	BCEN	End/Single	Mode[1:0]		Function[2:0]			RW : 00
1,29h	DCB02IN	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,2Ah	DCB02OU	AUXCLK		AUXEN	AUX IO Select[1:0]		OUTEN	Output Select[1:0]		RW : 00
0,2Ch	DCB03DR0	Data[7:0]								# : 00
0,2Dh	DCB03DR1	Data[7:0]								W : 00
0,2Eh	DCB03DR2	Data[7:0]								# : 00
0,2Fh	DCB03CR0	Function control/status bits for selected function[6:0]							Enable	# : 00
1,2Ch	DCB03FN	Data Invert	BCEN	End/Single	Mode[1:0]		Function[2:0]			RW : 00
1,2Dh	DCB03IN	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,2Eh	DCB03OU	AUXCLK		AUXEN	AUX IO Select[1:0]		OUTEN	Output Select[1:0]		RW : 00
Interrupt Mask Register										
0,E1h	INT_MSK1					DCB03	DCB02	DBB01	DBB00	RW : 00

LEGEND
#: Access is bit specific. Refer to register detail for additional information.
x: An “x” before the comma in the address field indicates that this register can be accessed or written to no matter what bank is used.

16. Row Digital Interconnect (RDI)

This chapter explains the Row Digital Interconnect (RDI) and its associated registers. This chapter discusses a single digital PSoC block row. It does not discuss the functions, inputs, or outputs for individual digital PSoC blocks. Information about individual digital PSoC blocks is covered in the Digital Blocks chapter on page 199.

Table 16-1. Digital PSoC Row Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,xxh	RDIxRI	RI3[1:0]		RI2[1:0]		RI1[1:0]		RI0[1:0]		RW : 00
x,xxh	RDIxSYN					RI3SYN	RI2SYN	RI1SYN	RI0SYN	RW : 00
x,xxh	RDIxIS			BCSEL[1:0]		IS3	IS2	IS1	IS0	RW : 00
x,xxh	RDIxLT0	LUT1[3:0]				LUT0[3:0]				RW : 00
x,xxh	RDIxLT1	LUT3[3:0]				LUT2[3:0]				RW : 00
x,xxh	RDIxRO0	GOO5EN	GOO1EN	GOE5EN	GOE1EN	GOO4EN	GOO0EN	GOE4EN	GOE0EN	RW : 00
x,xxh	RDIxRO1	GOO7EN	GOO3EN	GOE7EN	GOE3EN	GOO6EN	GOO2EN	GOE6EN	GOE2EN	RW : 00

LEGEND
x: An “x” before the comma in the address field indicates that the register exists in both register banks.
xx: An “x” after the comma in the address field indicates that there are multiple instances of the register. For a detailed address listing of these registers, refer to the “Digital Register Summary” on page 186.

Many signals pass through the Digital PSoC Block Row on their way to or from individual digital blocks. However, only a small number of signals pass though configurable circuits on their way to and from digital blocks. The configurable circuits allow for greater flexibility in the connections between digital blocks and global busses. What follows is a discussion of the signals that are configurable by way of the registers listed in Table 16-1.

16.1 Architectural Description

In Figure 16-1, within a Digital PSoC Block row, there are four digital PSoC Blocks. The first two blocks are of the type basic (DBB). The second two are of the type communication (DCB). This figure shows the connections between digital blocks within a row. Only the signals that pass through the bold line in Figure 16-1 are shown at the next level of hierarchy (Figure 16-2 on page 195).

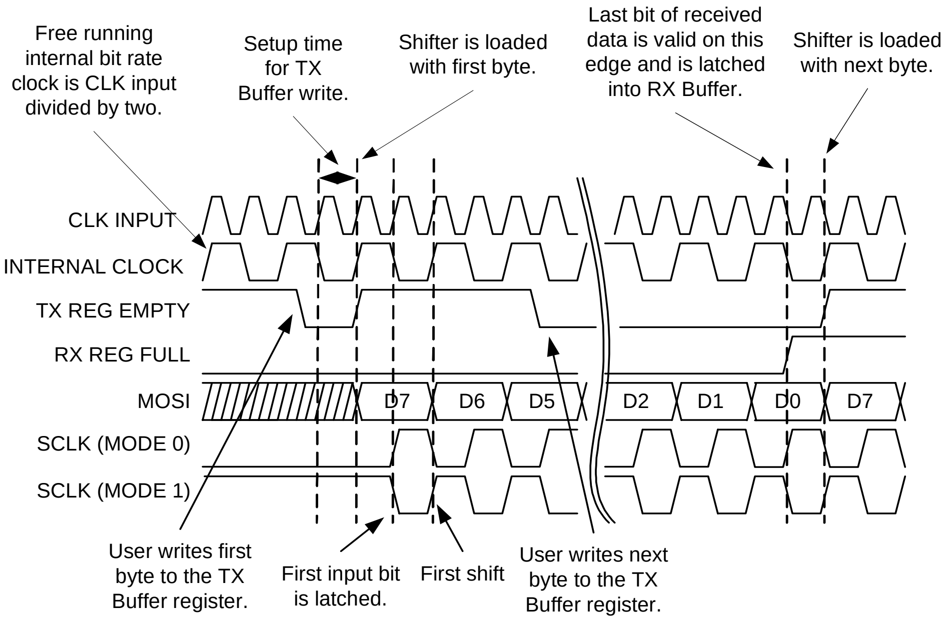
17.3.6 SPIM Timing

Enable/Disable Operation. As soon as the block is configured for SPIM, the primary output is the MSB or LSB of the shift register, depending on the LSBF configuration in bit 7 of the Control register. The auxiliary output is '1' or '0' depending on the idle clock state of the SPI mode. This is the idle state.

When the SPIM is enabled, the internal reset is released on the divide by 2 flip-flop, and on the next positive edge of the selected input clock. This 1 bit divider transitions to a '1' and remains free-running thereafter.

When the block is disabled, the SCLK and MOSI outputs revert to their idle state. All internal state is reset (including CR0 status) to its configuration specific reset state, except for DR0, DR1, and DR2, which are unaffected.

Normal Operation. Typical timing for a SPIM transfer is shown in Figure 17-15 and Figure 17-16. The user initially writes a byte to transmit when TX Reg Empty status is true.If no transmission is currently in progress, the data is loaded into the shifter and the transmission is initiated. The TX Reg Empty status is asserted again and the user is allowed to write the next byte to be transmitted to the TX Buffer register. After the last bit is output, if TX Buffer data is available with one-half clock setup time to the next clock, a new byte transmission will be initiated. A SPIM block receives a byte at the same time that it sends one. The SPI Complete or RX Reg Full can be used to determine when the input byte has been received.



Free running internal bit rate clock is CLK input divided by two.
Setup time for TX Buffer write.
Shifter is loaded with first byte.
Last bit of received data is valid on this edge and is latched into RX Buffer.
Shifter is loaded with next byte.
CLK INPUT
INTERNAL CLOCK
TX REG EMPTY
RX REG FULL
MOSI
SCLK (MODE 0)
SCLK (MODE 1)
D7
D6
D5
D2
D1
D0
D7
User writes first byte to the TX Buffer register.
First input bit is latched.
First shift
User writes next byte to the TX Buffer register.

Figure 17-15. Typical SPIM Timing in Mode 0 and 1

18. Analog Interface

This chapter explains the Analog System Interface and its associated registers. The analog system interface is a collection of system level interfaces to the analog array and analog reference block.

Table 18-1. Analog Interface Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,64h	CMP_CR0			COMP[1]	COMP[0]			AINT[1]	AINT[0]	RW : 00
0,66h	CMP_CR1			CLDIS[1]	CLDIS[0]					RW : 00
0,65h	ASY_CR		SARCNT[2:0]			SARSIGN	SARCOL[1:0]		SYNCEN	RW : 00
0,E6h	DEC_CR0	IGEN[3:0]				ICLKS0	DCOL[1:0]		DCLKS0	RW : 00
0,E7h	DEC_CR1	ECNT	IDEC			ICLKS1			DCLKS1	RW : 00
1,60h	CLK_CR0					Acolumn1[1:0]		Acolumn0[1:0]		RW : 00
1,61h	CLK_CR1		SHDIS	ACLK1[2:0]			ACLK0[2:0]			RW : 00
1,63h	AMD_CR0						AMOD0[2:0]			RW : 00
1,66h	AMD_CR1						AMOD1[2:0]			RW : 00
1,67h	ALT_CR0	LUT1[3:0]				LUT0[3:0]				RW : 00

18.1 Architectural Description

Figure 18-1 displays the top-level diagram of the PSoC device's analog system.

18.1.1 Analog Data Bus Interface

The Analog Bus Interface isolates the analog array and analog system interface registers from the CPU system data bus to reduce bus loading. Transceivers are implemented on the system data bus to isolate the analog data bus from the system data bus. This creates a local analog data bus.

18.1.2 Analog Comparator Bus Interface

Each analog column has a dedicated comparator bus associated with it. Every analog PSoC block has a comparator output that can drive this bus. However, only one analog block in a column can actively drive the comparator bus for a column at any one time. The output on the comparator bus can drive into the digital blocks as a data input. It also serves as an input to the decimator, as an interrupt input, and is available as read-only data in the Analog Comparator Control Register (CMP_CR0, Address = Bank 0,64H).

Figure 18-1 illustrates one column of the comparator bus. In the Continuous Time (CT) analog blocks, the CPhase and CLatch bits of CT Block Control Register 2 determine whether the output signal on the comparator bus is latched inside the block, and if it is, which clock phase it is latched on. In the Switched Capacitor (SC) analog blocks, the output on the comparator bus is always latched. The ClockPhase bit in SC Block Control Register 0 determines the phase on which this data is latched and available.

The comparator bus is latched before it is available, to either drive the digital blocks, interrupt, decimator, or be read in the CMP_CR0 register. The latch for each comparator bus is transparent (the output tracks the input) during the high period of PHI2. During the low period of PHI2, the latch retains the value on the comparator bus during the high-to-low transition of PHI2. The CMP_CR0 register is shown in Table 18-1. There is also an option to force the latch in each column into a transparent mode by setting bits in the CMP_CR1 register.

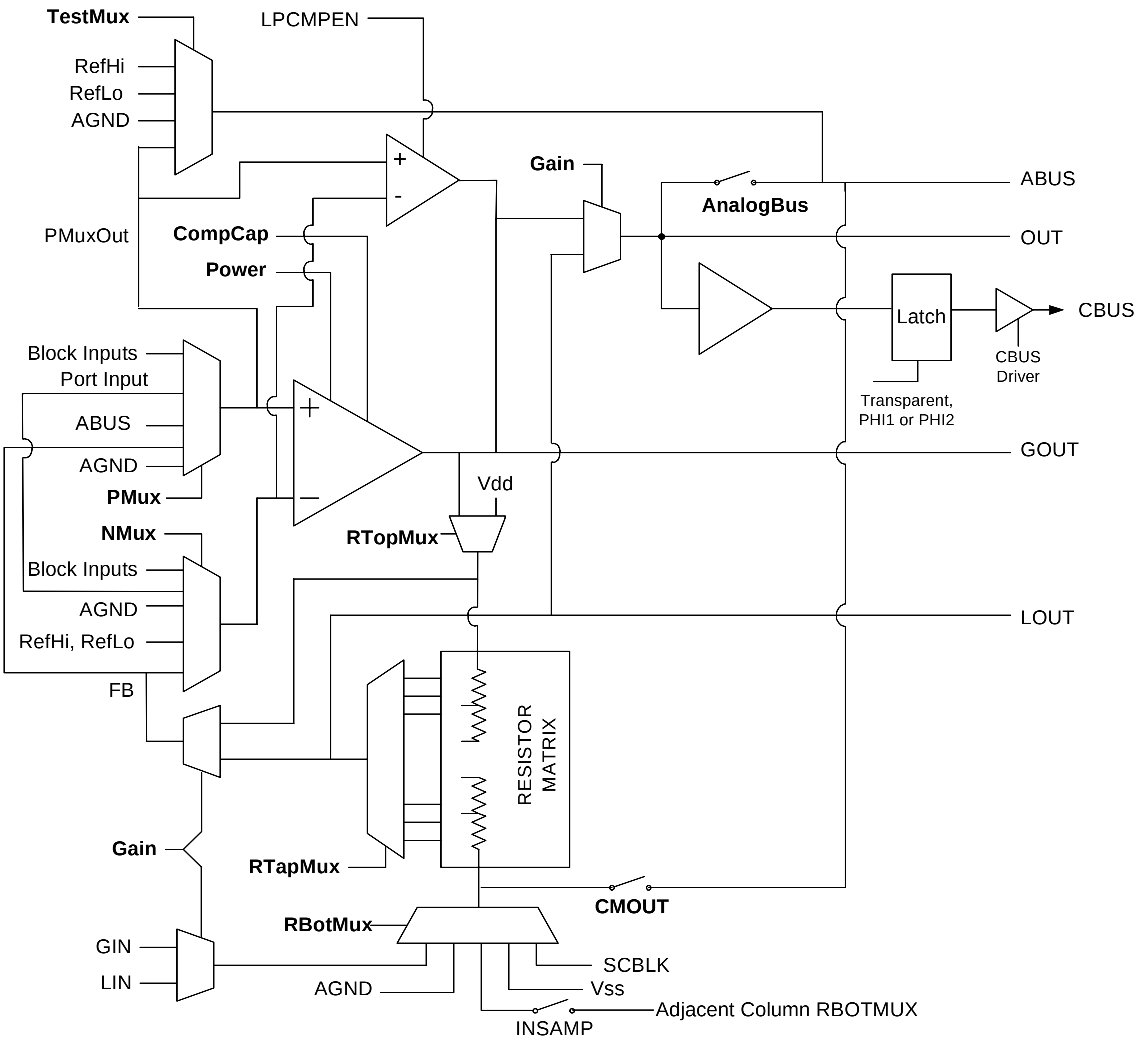


Figure 23-1. Analog Continuous Time Block Diagram

Bit 2: CMOUT. The analog array may be used to build two different forms of instrumentation amplifiers. Two continuous time blocks combine to make a 2-opamp instrumentation amplifier (see Figure 23-2).

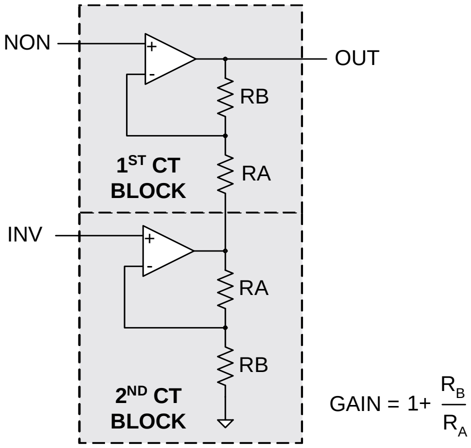


Figure 23-2. 2-Opamp Instrumentation Amplifier

Two continuous time blocks and one switched capacitor block combine to make a 3-opamp instrumentation amplifier (see Figure 23-3).

The 3-opamp instrumentation amplifier takes more resources, but handles a larger common mode input range. Bit2 (CMOUT) and bit1 (INSAMP) control switches are involved in the 3-opamp instrumentation amplifier. If bit2 (CMOUT) is high, then the node formed by the connection of the resistors between the continuous time blocks is connected to that continuous time block's ABus. This node is the common mode of the inputs to the instrumentation amplifier. The CMOUT bit is optional for the 3-opamp instrumentation amplifier.

Bit 1: INSAMP. This bit is used to connect the resistors of two continuous time blocks as part of a 3-opamp instrumentation amplifier. The INSAMP bit must be high for the 3-opamp instrumentation amplifier (see Figure 23-3).

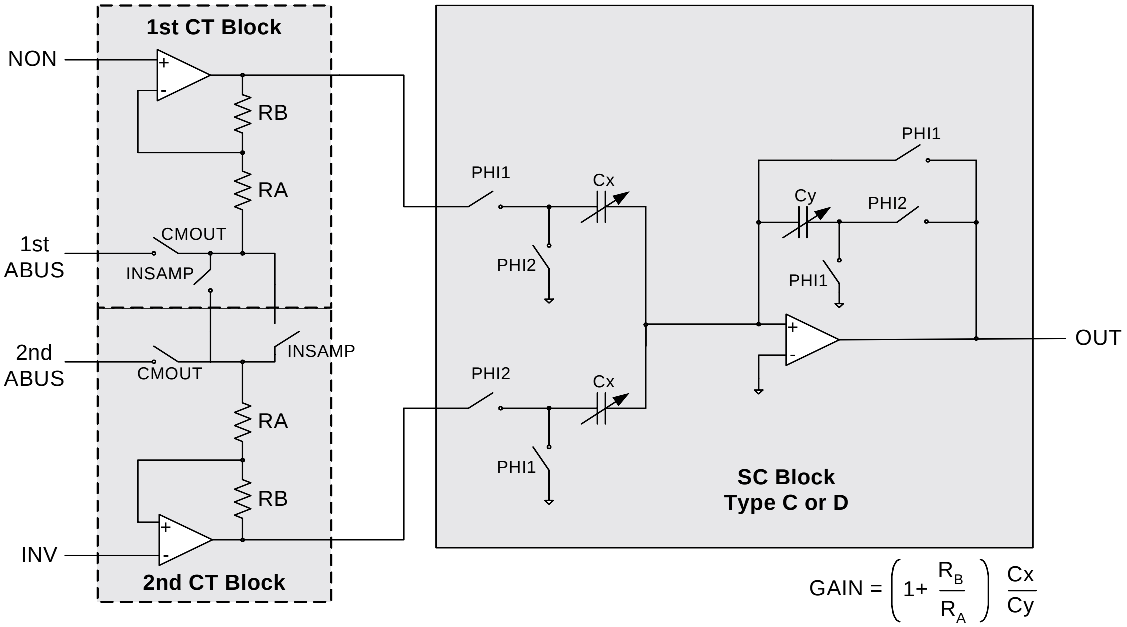


Figure 23-3. 3-Opamp Instrumentation Amplifier

24.1.2 Internal Low Speed Oscillator

The internal low speed oscillator (ILO), or sometimes referred to as the Sleep Oscillator, is always on unless the device is operating off a crystal. The ILO is available as a general clock, but is also the clock source for the sleep and watchdog timers.

The ILO is discussed in detail in the chapter "Internal Low Speed Oscillator (ILO)" on page 67.

24.1.3 32 kHz Crystal Oscillator

The PSoC may be configured to use an external crystal. When configured in this way the internal low speed oscillator is turned off and the crystal becomes the clock source for all 32 kHz clocks.

The Crystal Oscillator is discussed in detail in the chapter "32 kHz Crystal Oscillator (ECO)" on page 69.

24.1.4 External Clock

The ability to replace the 24 MHz internal main oscillator (IMO), as the device master system clock (SYSCLK) with an externally supplied clock, is a feature in the PSoC mixed signal arrays.

Pin P1[4] has been chosen as the input pin for the external clock. This pin was chosen because it is not associated with any special features such as analog IO, crystal, or In System Serial Programming (ISSP), and it is also not physically close to either the P1[0] and P1[1] crystal pins.

The user is able to supply an external clock with a frequency between 1 MHz and 24 MHz. The reset state of the EXTCLKEN bit is '0', and therefore, the device always boots up under control of the IMO. There is no way to start the system from a reset state with the external clock.

When the EXTCLKEN bit is set, the external clock becomes the source for the internal clock tree, SYSCLK, which drives most chip clocking functions. All external and internal signals, including the 32 kHz clock, whether derived from the internal low speed oscillator (ILO) or the crystal oscillator, are synchronized to this clock source.

24.1.4.1 Clock Doubler

One of the blocks driven by the system clock is the clock doubler circuit that drives the SYSCLKX2 output. This doubled clock, which is 48 MHz when the IMO is the selected clock, may be used as a clock source for the digital blocks. When the external clock is selected, the SYSCLKX2 signal is still available and serves as a doubler for whatever frequency is input on the external clock pin.

Following the spec for the external clock input ensures that the internal circuitry of the digital blocks, which is clocked by SYSCLKX2, will meet timing. However, since the doubled clock is generated from both edges of the input clock, clock jitter will be introduced if the duty cycle deviates greatly from fifty percent. Also, the high time of the clock out of the doubler is fixed at 21 ns, so the duty cycle of SYSCLKX2 will be proportional to the inverse of the frequency, as shown in Figure 24-2. Regardless of the input frequency, the high period of SYSCLKX2 is 21 ns nominal.

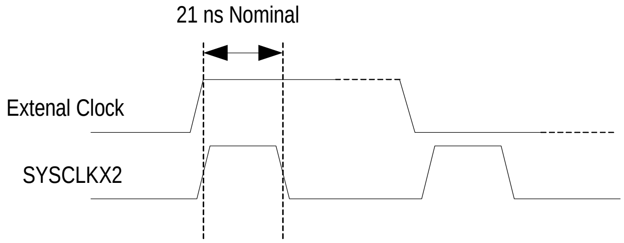


Figure 24-2. Operation of the Clock Doubler

24.1.4.2 Switch Operation

Switching between the IMO and the external clock may be done in firmware at any time and is transparent to the user. Since all chip resources run on clocks derived from or synchronized to SYSCLK, when the switch is made, analog and digital functions may be momentarily interrupted.

When a switch is made from the IMO to the external clock, the IMO may be turned off to save power. This can be done by setting the IMODIS bit and may be done immediately after the instruction that sets the EXTCLKEN bit. However, when switching back from an external clock to the IMO, the IMODIS bit must be cleared, and a firmware delay implemented. This gives the IMO sufficient start-up time before the EXTCLKEN bit may be cleared.

Switch timing depends on whether the CPU clock divider is set for divide by 1, or divide by 2 or greater. In the case where the CPU clock divider is set for divide by 2 or greater, as shown in Figure 24-3, the setting of the EXTCLKEN bit occurs shortly after the rising edge of SYSCLK. The SYSCLK output is then disabled after the next falling edge of SYSCLK, but before the next rising edge. This ensures a glitch free transition and provides a full cycle of setup time from SYSCLK to output disable. Once the current clock selection is disabled, the enable of the newly selected clock is double synchronized to that clock. After synchronization, on the subsequent negative edge, SYSCLK is enabled to output the newly selected clock.

In the 24 MHz case, as shown in Figure 24-4, the assertion of IOW_ and thus the setting of the EXTCLKEN bit occurs on the falling edge of SYSCLK. Since SYSCLK is already low, the output is immediately disabled and therefore, the setup time from SYSCLK to disable is one-half SYSCLK.

27.2.2 Master Operation

To prepare for a Master mode transaction, the host must determine if the bus is free. This can be done by polling the BusBusy status. If busy, interrupts can be enabled to detect a Stop condition. Once it is determined that the bus is available, firmware should write the address byte into the I2C_DR register and set the Start Gen bit in the I2C_MSCR register.

If the Slave sub-unit is not enabled, the block is in Master Only mode. In this mode, the unit does not generate interrupts or stall the bus on externally generated Start conditions.

In a Multi-Master environment there are two additional outcomes possible:

1. The host was too late to reserve the bus as a Master and another Master may have generated a Start and sent an Address/R/W byte. In this case, the unit as a Master will fail to generate a Start and will be forced into Slave mode. The Start will be pending and eventually occur at a later time when the bus becomes free. When the interrupt occurs in Slave mode, the host can determine that the Start command was unsuccessful by reading the I2C_MSCR register Start bit, which will be reset on successful Start from this unit as Master. If this bit is still a ‘1’ on the Start/Address interrupt, it means that the unit is operating in Slave mode. In this case, the data register will have the master’s address data.
2. If another Master starts a transmission at the same time as this unit, arbitration will occur. If this unit loses the arbitration, the LostArb status bit will be set. In this case, the block will release the bus and switch to Slave operation. When the Start/Address interrupt occurs, the data register will have the winning master’s address data.

Figure 27-3 is a graphical representation of a typical data transfer from the master perspective.

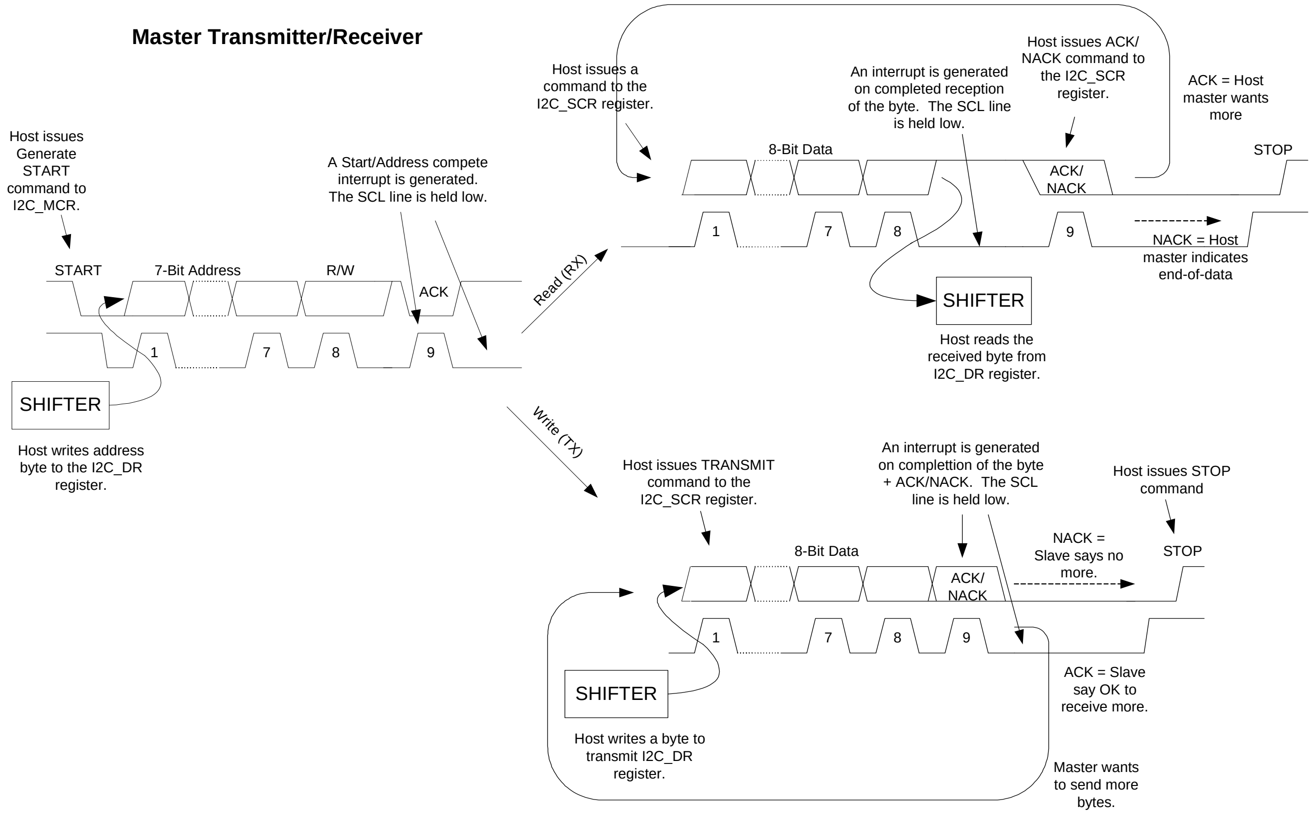

Master Transmitter/Receiver
Host issues Generate START command to I2C_MCR.
START
7-Bit Address
R/W
ACK
1
7
8
9
A Start/Address compete interrupt is generated. The SCL line is held low.
SHIFTER
Host writes address byte to the I2C_DR register.
Read (RX)
Write (TX)
Host issues a command to the I2C_SCR register.
8-Bit Data
An interrupt is generated on completed reception of the byte. The SCL line is held low.
Host issues ACK/ NACK command to the I2C_SCR register.
ACK/ NACK
ACK = Host master wants more
STOP
NACK = Host master indicates end-of-data
SHIFTER
Host reads the received byte from I2C_DR register.
Host issues TRANSMIT command to the I2C_SCR register.
8-Bit Data
An interrupt is generated on complettion of the byte + ACK/NACK. The SCL line is held low.
ACK/ NACK
NACK = Slave says no more.
Host issues STOP command
STOP
SHIFTER
Host writes a byte to transmit I2C_DR register.
ACK = Slave say OK to receive more.
Master wants to send more bytes.

Figure 27-3. Master Operation

Bit 0: Byte Complete. The I2C hardware operates on a byte basis. In Transmit mode, this bit is set and an interrupt is generated at the end of nine bits (the transmitted byte + the received ACK). In Receive mode, the bit is set after the eight bits of data have been received. When this bit is set, an interrupt is generated at these data sampling points, which are associated with the SCL input clock rising (see details in the Timing section). If the host responds with a write back to this register before the subsequent falling edge of SCL (which is approximately one-half bit time), the transfer will continue without interruption. However, if the host is unable to respond within that time, the hardware will hold the SCL line low, stalling the I2C bus. In both Master and Slave mode, a subsequent write to the I2C_SCR register will release the stall.

For additional information, reference the I2C_SCR register on page 128.

27.3.3 I2C_DR Register

This register is the I2C data register and provides read/write access to the shift register. It is not buffered and therefore, writes and valid data reads may only occur at specific points in the transfer. These cases are outlined as follows.

- **Master or Slave Receiver** Data in the I2C_DR register is only valid for reading, when the Byte Complete status bit is set. Data bytes must be read from the register before writing to the I2C_SCR register, which continues the transfer.
- **Master Start or Restart** Address bytes must be written in I2C_DR before the Start or Restart bit is set in the I2C_MSCR register, which causes the Start or Restart to be generated and the address shifted out.
- **Master or Slave Transmitter** Data bytes must be written to the I2C_DR register before the transmit bit is set in the I2C_SCR register, which causes the transfer to continue.

For additional information, reference the I2C_DR register on page 129.

27.3.4 I2C_MSCR Register

This register is the I2C master status and control register.

Table 27-6. I2C_MSCR Master Status and Control Register

Bit	Access	Description	Mode
0	R/W	Start Gen 1 = Generate a Start condition and send a byte (address) to the I2C bus. This bit is cleared by hardware when the Start generation is complete.	Master Only
1	R/W	Restart Gen 1 = Generate a Restart condition. This bit is cleared by hardware when the Start generation is complete.	Master Only

Table 27-6. I2C_MSCR Master Status and Control Register *(continued)*

Bit	Access	Description	Mode
2	RO	Master Mode This bit is set to ‘1’ when a start condition, generated by this block, is detected and reset to ‘0’ when a stop condition is detected.	Master Only
3	RO	Bus Busy This bit is set to ‘1’ when any Start condition is detected, and reset to ‘0’ when a Stop condition is detected.	Master Only

Bits 7 to 4: Reserved.

Bit 3: Bus Busy. This read only bit is set to ‘1’ by any Start condition and reset to ‘0’ by a Stop condition. It may be polled by firmware to determine when a bus transfer may be initiated.

Bit 2: Master Mode. This bit indicates that the device is operating as a Master. It is set in the detection of this block’s Start condition and reset in the detection of the subsequent Stop condition.

Bit 1: Restart Gen. This bit is only used at the end of a Master transfer (as noted in Other Cases 1 and 2 above of the Start Gen bit). If an address is loaded into the data register and this bit is set prior to NACKing (Master receiver) or resetting the transmit bit (Master transmitter), or after a Master transmitter is NACKed by the Slave, a Restart condition will be generated, followed by the transmission of the address byte.

Bit 0: Start Gen. Before setting this bit, firmware must write the address byte to send into the I2C_DR register. When this bit is set, the Start condition is generated, followed immediately by the transmission of the address byte. (No control in the I2C_SCR register is needed for the Master to initiate a transmission; the direction is inherently “transmit”.) The bit is automatically reset to ‘0’ after the Start has been generated.

There are three possible outcomes as a result of setting the Start Gen bit:

1. The bus is free and the Start condition is generated successfully. A Byte Complete interrupt will be generated after the Start and the address byte has been transmitted. If the address was ACKed by the receiver, the firmware may then proceed to send data bytes.
2. The Start command is too late. Another Master in a Multi-Master environment has generated a valid Start and the bus is busy. The resulting behavior depends upon whether Slave mode is enabled.

   Slave mode is enabled: A Start and address byte interrupt will be generated. When reading the I2C_MSCR, the Master will see the Start Gen bit still set and the I2C_SCR will have the Address bit set, indicating that the block has been addressed as a Slave.