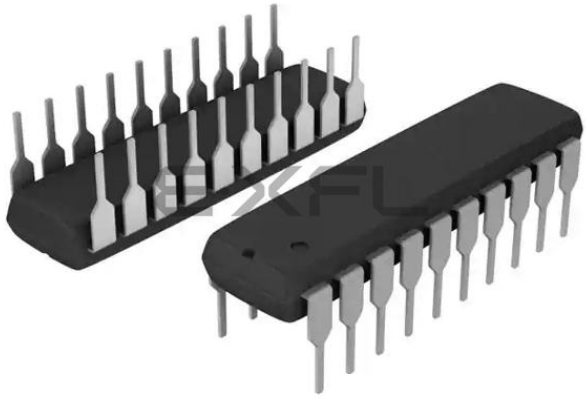


Welcome to E-XFL.COM

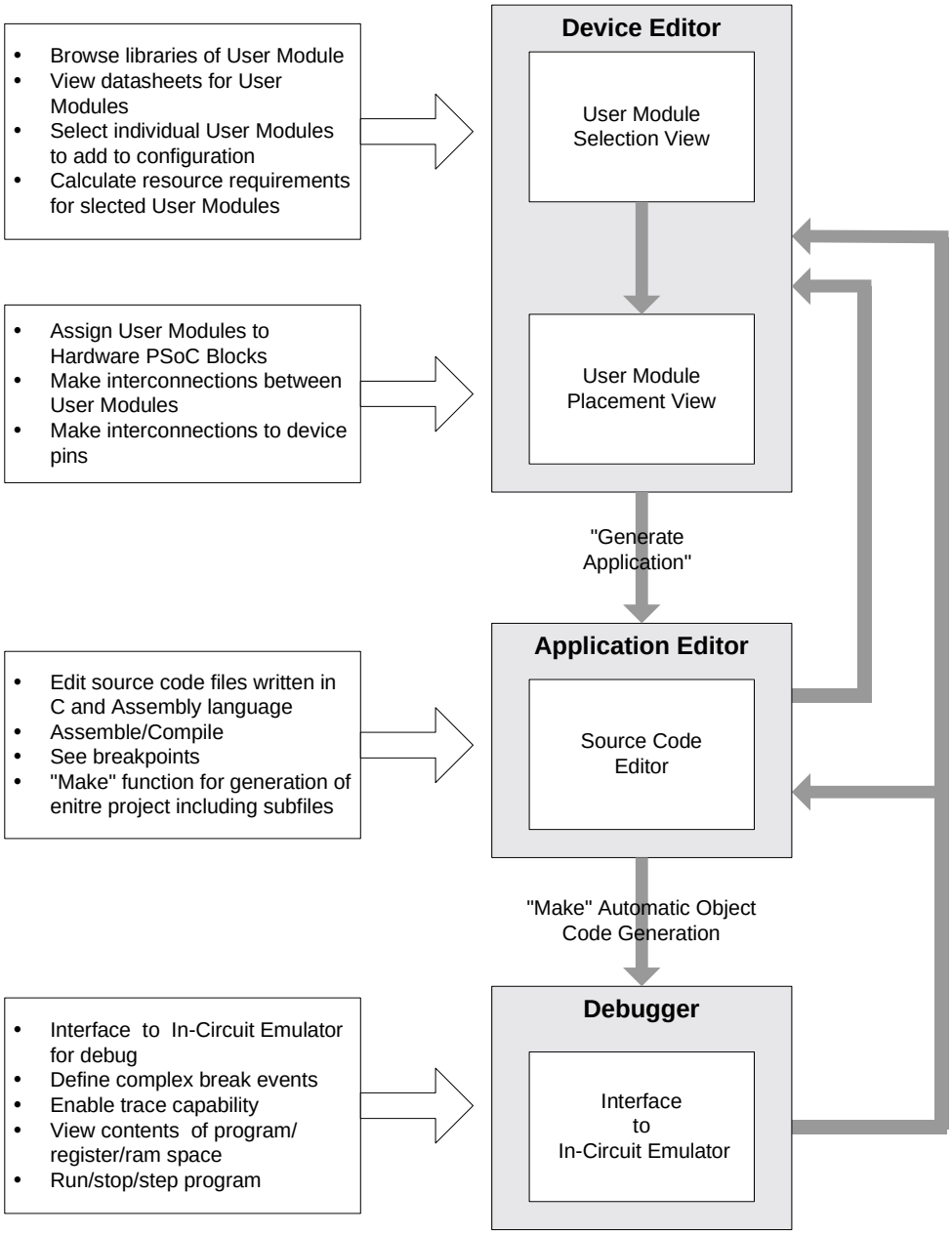
What is "Embedded - Microcontrollers"?

"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	16
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	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-DIP
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy8c24223-24pi

modules to hardware resources. You also make the interconnections between the user modules, and between the user modules and the I/O pins. This process step takes place in the Device Editor subsystem within PSoC Designer. There are two views inside this step: one for selecting user modules and one for assigning them to the hardware blocks and interconnecting them. The last action in this step is to "Generate Application," which causes the development software to automatically generate the required files for the selected configuration.



User Modules and Development Process Flow Chart

The next step in the process is to write your main program, and any other sub-routines required by your application. This step takes place in the Application Editor subsystem. You will have all the subroutines automatically generated for the user modules you have chosen and the source code for these routines can be viewed in this step as well. The different files created for the project are all contained in a tree structure for easy reference. The development software has a handy "Make" function, which assembles and compiles all source files, and links them into an object file ready for the debugging process.

The last step in development takes place in the Debugger subsystem. This is where the object code is downloaded into the In-Circuit Emulator and run. The Debugger is both the interface to the ICE and also contains an advanced set

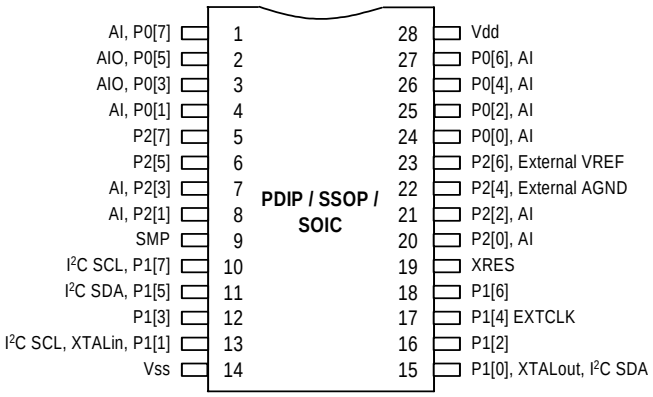
Table 1-4. 28-Pin Part Pinout (PDIP, SSOP, SOIC)

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^2C SCL

Pin No.	Description
11	P1[5], I^2C 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)

Pin No.	Description
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.



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.

The interrupt logic portion of the block is shown in Figure 6-2.

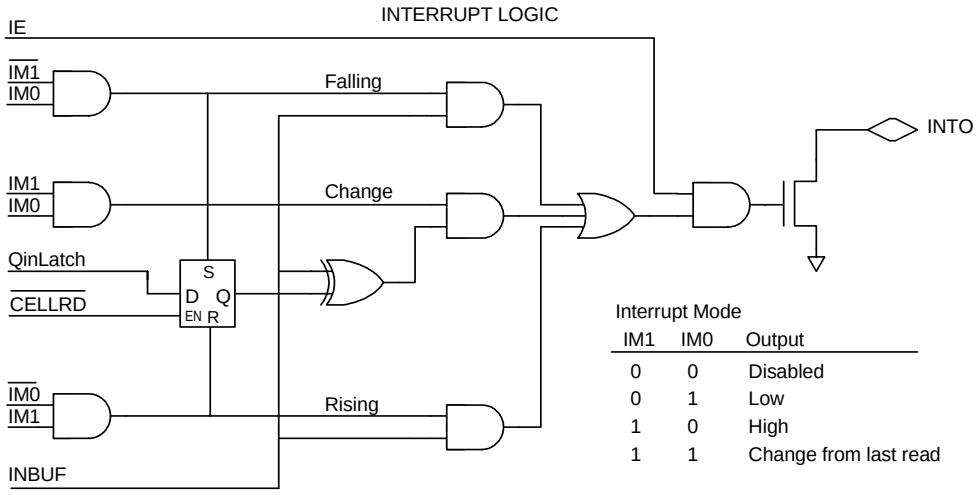


Figure 6-2. GPIO Interrupt Mode Block Diagram

6.2 Register Definitions

For a selected GPIO block, the individual registers are addressed as shown in Table 6-2. In the register names, the 'x' is the port number, configured at the chip level (x = 0 to 7 typically). DA[1:0] refers to the two LSB of the register address.

All register values are readable, except for the PRTxDR register; reads of this register return the pin state instead of the register bit state.

Table 6-2. Internal Register Bit Addressing

XOI	DA[1:0]	Register	Resets to:	(Name)	Function
0	00b	PRTxDR	0	DIN	Data
0	01b	PRTxIE	0	IE	Interrupt Enable
0	10b	PRTxGS	0	BYP	Global Select
0	11b	PRTxDM2	**1**	DM2	Drive Mode, Bit 2
1	00b	PRTxDM0	0	DM0	Drive Mode, Bit 0
1	01b	PRTxDM1	1	DM1	Drive Mode, BIt 1
1	10b	PRTxIC0	0	IM0	Intrpt. Mask, Bit 0
1	11b	PRTxIC1	0	IM1	Intrpt. Mask, Bit 1

6.2.1 PRTxDR Registers

Writing the PRTxDR register bit sets the output drive state for the pin to high (for DIN=1) or low (DIN=0), unless a bypass mode is selected (either I2C Enable=1 or the global select register written high).

Reading PRTxDR returns the actual pin state, as seen by the input buffer. This may not be the same as the expected output state, if the load pulls the pin more strongly than the pin's configured output drive.

For additional information, reference the PRTxDR register on page 88.

group. These are referred to as DM2, DM1, and DM0, or

6.2.2 PRTxIE Registers

The PRTxIE register is used to enable/disable the interrupt enable internal to the GPIO block. A '1' enables the INTO output at the block, a '0' disables INTO so it can only be Hi-Z.

For additional information, reference the PRTxIE register on page 89.

6.2.3 PRTxGS Registers

The PRTxGS register is used to select the block for connection to global inputs or outputs. Writing this register high enables the global bypass (BYP=1 in Figure 6-1). If the drive mode is set to digital Hi-Z (DM[2:0] = 010b), then the pin is selected for global input (PIN drives to the Global Input Bus). In non-Hi-Z modes, the block is selected for global output (the Global Output Bus drives to PIN), bypassing the data register value (assuming I2C Enable=0).

If the PRTxGS register is written to zero, the global in/out function is disabled for the pin.

For additional information, reference the PRTxGS register on page 90.

6.2.4 PRTxDMx Registers

There are eight possible drive modes for each port pin. Three mode bits are required to select one of these modes, and these three bits are spread into three different registers (PRTxDM0, PRTxDM1, and PRTxDM2). 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 three Drive Mode register bits that control the drive mode for that pin (Example: Bit[2] in PRT0DM0, bit[2] in PRT0DM1 and bit[2] in PRT0DM2). The three bits from the three registers are treated as a

Table 11-2. Sleep Interval Selections

Sleep Interval OSC_CR[4:3]	Sleep Timer Clocks	Sleep Period (nominal)	Watchdog Period (nominal)
00b (default)	64	1.95 ms	6 ms
01b	512	15.6 ms	47 ms
10b	4096	125 ms	375 ms
11b	32,768	1 sec	3 sec

Bits 2, 1, and 0: CPU Speed[2:0]. The PSoC M8C may operate over a range of CPU clock speeds (Table 11-3), allowing the M8C's performance and power requirements to be tailored to the application.

The reset value for the CPU Speed bits is zero. Therefore, the default CPU speed is one-eighth of the clock source. The internal main oscillator is the default clock source for the CPU speed circuit; therefore, the default CPU speed is 3 MHz.

The CPU frequency is changed with a write to the OSC_CR0 register. There are eight frequencies generated from a power-of-2 divide circuit, which are selected by a 3-bit code. At any given time, the CPU 8:1 clock multiplexer is selecting one of the available frequencies, which is re-synchronized to the 24 MHz master clock at the output.

Regardless of the CPU speed bit's setting, if the actual CPU speed is greater than 12 MHz, the 24 MHz operating requirements apply. An example of this scenario is a device that is configured to use an external clock, which is supplying a frequency of 20 MHz. If the CPU speed register's value is 0b011, the CPU clock will be 20 MHz. Therefore, the supply voltage requirements for the device are the same as if the part was operating at 24 MHz off of the internal main oscillator. The operating voltage requirements are not relaxed until the CPU speed is at 12.0 MHz or less.

Table 11-3. OSC_CR0[2:0] Bits: CPU Speed

Bits	Internal Main Oscillator	External Clock
000b	3 MHz	EXTCLK/ 8
001b	6 MHz	EXTCLK/ 4
010b	12 MHz	EXTCLK/ 2
011b	24 MHz	EXTCLK/ 1
100b	1.5 MHz	EXTCLK/ 16
101b	750 kHz	EXTCLK/ 32
110b	187.5 kHz	EXTCLK/ 128
111b	93.7 kHz	EXTCLK/ 256

For additional information, reference the OSC_CR0 register on page 176.

11.2.2 OSC_CR2 Register

Bit 7: PLLGAIN. This is the only bit in the OSC_CR2 register that directly influences the PLL. When set, this bit keeps the PLL in a low gain mode.

Bits 6 to 3: Reserved.

Bit 2: EXTCLKEN. 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. If an external clock is enabled, PLL mode should be off.

Bit 1: IMODIS. When set, the Internal Main Oscillator is disabled. If the doubler is enabled (SYSCLKX2DIS=0), the Internal Main oscillator will be forced on.

Bit 0: SYSCLKX2DIS. When set, the Internal Main Oscillator's doubler is disabled. This will result in a reduction of overall device power, on the order of 1 mA. It is advised that any application that does not require this doubled clock should have it turned off.

For additional information, reference the OSC_CR2 register on page 178.

13.1.11 DxBxxCR0

(CRCPRS Control)

Digital Basic/Communication Type B Block Control Register 0

Individual Register Names and Addresses

DBB00CR0: 0,23h DBB01CR0: 0,27h DCB02CR0: 0,2Bh DCB03CR0: 0,2Fh

	7	6	5	4	3	2	1	0
Access : POR							RW : 0	RW : 0
Bit Name							Pass Mode	Enable

For additional information, reference the “Register Definitions” on page 208 in the Digital Blocks chapter.

Bit	Name	Description
[7:2]	**Reserved**	
[1]	**Pass Mode**	The DATA input selection is driven directly to the primary output and the block interrupt output. The CLK input selection is driven directly to the auxiliary output. 0 Normal CRC/PRS outputs 1 Outputs are overridden.
[0]	**Enable**	0 CRC/PRS is not enabled. 1 CRC/PRS is enabled.

13.1.28 ASCxxCR3

Analog Switch Cap Type C Block Control Register 3

Individual Register Names and Addresses

ASC10CR3 : x,83h ASC21CR3 : x,97h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0		RW : 0	RW : 0	RW : 0		RW : 0	
Bit Name	ARefMux[1:0]		FSW1	FSW0	BMuxSC[1:0]		PWR[1:0]	

For additional information, reference the “Register Definitions” on page 251 in the Switched Capacitor Block chapter.

Bit	Name	Description
[7:6]	**ARefMux[1:0]**	Encoding for selecting reference input. 00b Analog ground is selected. 01b REFHI input selected. (This is usually the high reference.) 10b REFLO input selected. (This is usually the low reference.) 11b Reference selection is driven by the comparator. (When output comparator node is set high, the input is set to REFHI. When set low, the input is set to REFLO.)
[5]	**FSW1**	Bit for controlling gated switches. 0 Switch is disabled. 1 If the FSW1 bit is set to ‘1’, the state of the switch is determined by the AutoZero bit. If the AutoZero bit is ‘0’, the switch is enabled at all times. If the AutoZero bit is ‘1’, the switch is enabled only when the internal PHI2 is high.
[4]	**FSW0**	Bits for controlling gated switches. 0 Switch is disabled. 1 Switch is enabled when PHI1 is high.
[3:2]	**BMuxSC[1:0]**	Encoding for selecting B inputs. Note that the available mux inputs vary by individual PSoC block. **ASC10** **ASC21** 00b ACB00 ASD11 01b ASD11 ASD20 10b P2[3] P2[0] 11b ASD20 TrefGND
[1:0]	**PWR[1:0]**	Encoding for selecting one of four power levels. 00b Off 01b Low 10b Medium 11b High

13.1.50 INT_VC

Interrupt Vector Clear Register

Individual Register Names and Addresses

INT_VC: 0,E2h

	7	6	5	4	3	2	1	0
Access : POR	RC : 00							
Bit Name	Pending Interrupt[7:0]							

For additional information, reference the “Register Definitions” on page 55 in the Interrupt Controller chapter.

Bit	Name	Description	
[7:0]	**Pending Interrupt[7:0]**	Read	Returns vector for highest priority pending interrupt.
		Write	Clears all pending and posted interrupts.

13.2.6 DxBxxIN

Digital Basic/Communications Type B Block Input Register

Individual Register Names and Addresses

DBB00IN : 1,21h DBB01IN : 1,25h DCB02IN : 1,29h DCB03IN : 1,2Dh

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

Before changing any of the configuration registers (DxBxxFN, DxBxxIN, and DxBxxOU), disable the corresponding digital block by setting bit 0 in the CR0 or DxBxxCR0 register to ‘0’. The values in this register should not be changed while the block is enabled. After all configuration changes are made, enable the block by setting bit 0 in the CR0 register to ‘1’.

For additional information, reference the “Register Definitions” on page 208 in the Digital Blocks chapter.

Bit	Name	Description	
[7:4]	**Data Input[3:0]**	0h	Low (0)
		1h	High (1)
		2h	Row broadcast net
		3h	Chain function to previous block
		4h	Analog column comparator 0
		5h	Analog column comparator 1
		6h	Reserved
		7h	Reserved
		8h	Row output 0
		9h	Row output 1
		Ah	Row output 2
		Bh	Row output 3
		Ch	Row input 0
		Dh	Row input 1
		Eh	Row input 2
		Fh	Row input 3
[3:0]	**Clock Input[3:0]**	0h	Clock disabled (low)
		1h	VC3
		2h	Row broadcast net
		3h	Previous block primary output (low for DBB00)
		4h	SYSCLKX2
		5h	VC1
		6h	VC2
		7h	CLK32K
		8h	Row output 0
		9h	Row output 1
		Ah	Row output 2
		Bh	Row output 3
		Ch	Row input 0
		Dh	Row input 1
		Eh	Row input 2
		Fh	Row input 3

13.2.18 OSC_CR4

Oscillator Control Register 4

Individual Register Names and Addresses

OSC_CR4: 1,DEh

	7	6	5	4	3	2	1	0
Access : POR							RW : 0	
Bit Name							VC3 Input Select[1:0]	

For additional information, reference the “Register Definitions” on page 266 in the Digital Clocks chapter.

Bit	Name	Description
[7:2]	**Reserved**	
[1:0]	**VC3 Input Select[1:0]**	Selects the clocking source for the VC3 Clock Divider. 00b SYSCLK 01b VC1 10b VC2 11b SYSCLKX2

13.2.28 ECO_TR

External Crystal Oscillator Trim Register

Individual Register Names and Addresses

ECO_TR: 1,EBh

	7	6	5	4	3	2	1	0
Access : POR	W : 0							
Bit Name	PSSDC[1:0]							

The value of this register is used to trim the External Crystal Oscillator. Its value is set to the device specific, best value during boot. The value in this register should not be changed. For additional information, reference the “Register Definitions” on page 70 in the 32 kHz Crystal Oscillator chapter.

Bit	Name	Description
[7:6]	**PSSDC[1:0]**	Sleep duty cycle. Controls the ratios (in numbers of 32 kHz clock periods) of “on” time versus “off” time for PORLVD, Bandgap reference, and pspump. 00b 1 / 128 01b 1 / 512 10b 1 / 32 11b 1 / 8
[5:0]	**Reserved**	

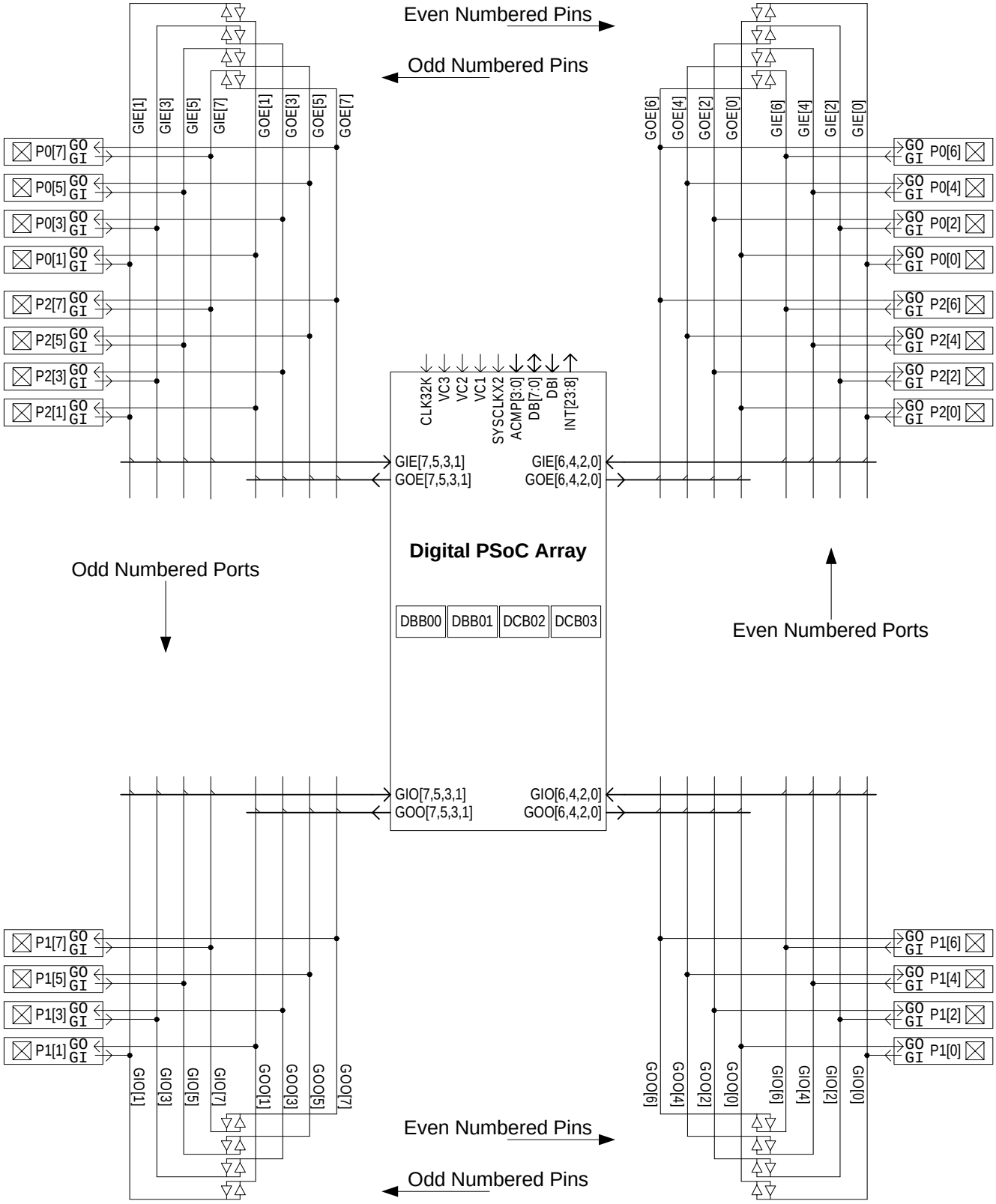


Figure 14-1. Global Interconnect Block Diagram

In Figure 15-1, the detailed view of a Digital PSoC block row has been replaced by a box labeled “Digital PSoC Block Row.” The rest of this figure illustrates how all rows are connected to the same globals, clocks, and so on. The figure also illustrates how the broadcast clock nets (BCxxxx) are connected between rows.

as the MS block polynomial (DR1) and 10h as the LS block polynomial value.

Determining the PRS Polynomial

Generally, PRS (pseudo random sequence) polynomials are selected from pre-computed reference tables. It is important to note that there are two common ways to specify a PRS polynomial: simple register configuration and modular configuration. In the simple method, a shift register is implemented with a reduction XOR of the MSB and feedback taps as input into the least significant bit. In the modular method, there is an XOR operation implemented between each register bit and each tap point enables the XOR with the MSB for that given bit. The CRCPRS function implements the modular approach.

Converting a Polynomial Spec to a Modular Spec

These are equivalent methods. However, there is a conversion that should be understood. If tables are specified in simple register format, then a conversion can be made to the modular format by subtracting each tap from the MS tap as shown in the following example.

To implement a 7-bit PRS of length 127, one possible code is [7,6,4,2]s, which is in simple format. The modular format would be [7,7-2,7-4,7-6]m or [7,5,3,2]m. Determining the polynomial to program is similar to the CRC example above. Set a binary bit for each tap (with bit 0 of the register corresponding to tap 1). Therefore, the code [7,5,3,2] would correspond to 01010110 or 56h.

In both the CRC and PRS cases, an appropriate Seed value should be selected that is greater than or equal in bit length.

17.1.8.1 Usability Exceptions

The following are usability exceptions for the CRCPRS function.

1. The polynomial register must only be written when the block is disabled.

17.1.8.2 Block Interrupt

The CRCPRS has one fixed interrupt source, which is the compare auxiliary output.

17.1.9 SPI Protocol Function

The Serial Peripheral Interface (SPI) is a Motorola specification for implementing full-duplex synchronous serial communication between devices. The 3-wire protocol uses both edges of the clock to enable synchronous communication, without the need for stringent setup and hold requirements.

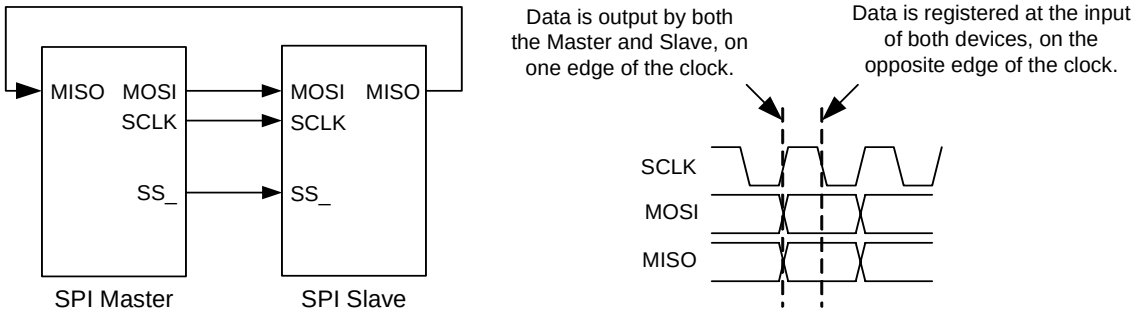


Figure 17-5. Basic SPI Configuration

A device can be a Master or Slave. A Master outputs clock and data to the Slave device and inputs Slave data. A Slave device inputs clock and data from the Master device and outputs data for input to the Master. The Master and Slave together are essentially a circular shift register, where the Master is generating the clocking and initiating data transfers.

A basic data transfer occurs when the Master sends 8 bits of data, along with eight clocks. In any transfer, both Master and Slave are transmitting and receiving simultaneously. If the Master is only sending data, the received data from the Slave is ignored. If the Master wishes to receive data from the Slave, the Master must send dummy bytes to generate the clocking for the Slave to send data back.

17.1.9.1 SPI Protocol Register Definitions

The SPI Protocol register definitions are located in Table 17-4. The use of the SS_ signal varies according to the capability of the Slave device.

Table 17-4. SPI Protocol Register Descriptions

Name	Function	Description
MOSI	Master Out Slave In	Master data output.
MISO	Master In Slave Out	Slave data output.
SCLK	Serial Clock	Clock generated by the Master.
SS_	Slave Select (active low)	This signal is provided to enable multi-slave connections to the MISO pin. The MOSI and SCLK pins can be connected to multiple slaves, and the SS_ input selects which slave will receive the input data and drive the MISO line.

22.1 Architectural Description

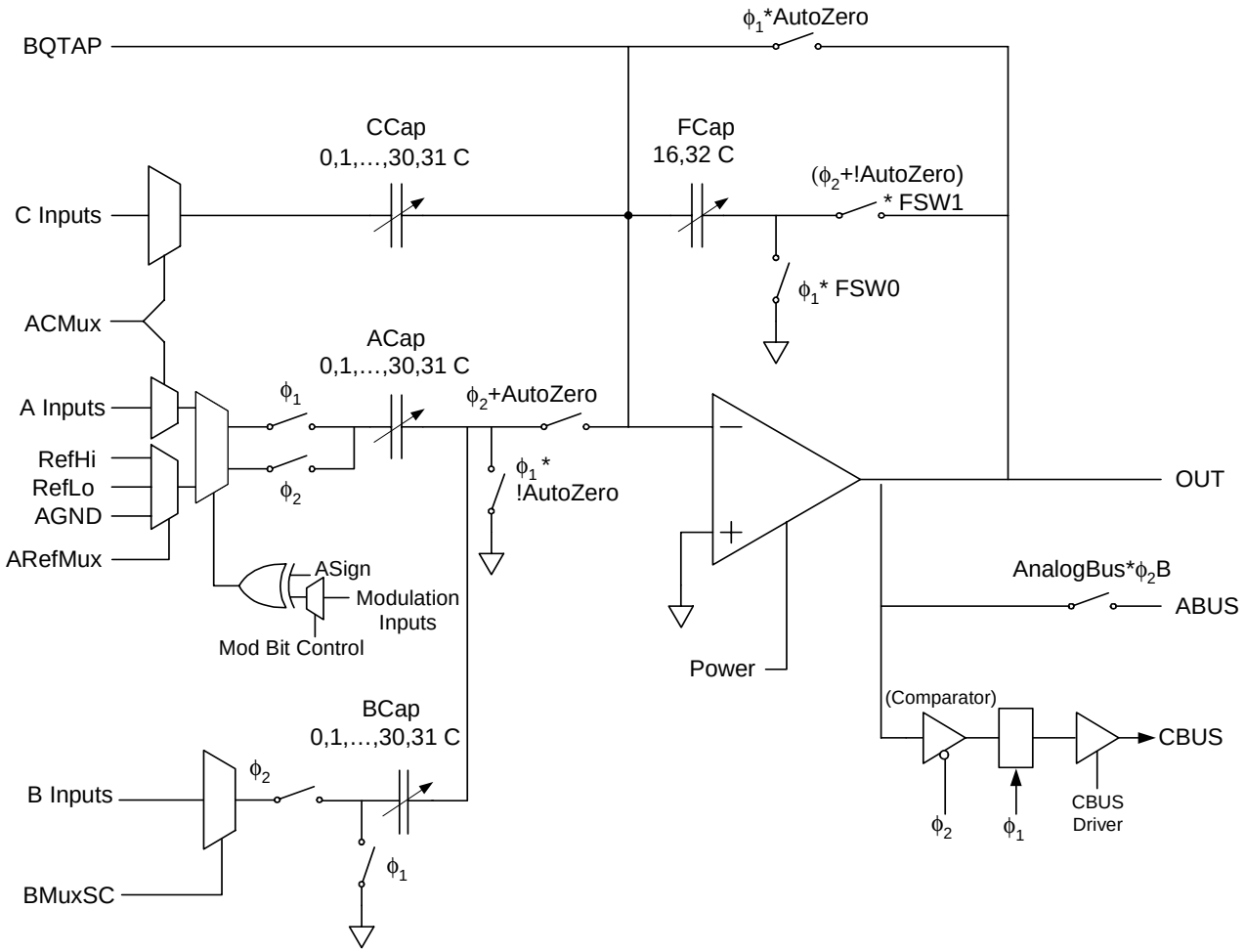


Figure 22-1. Analog Switch Cap Type C PSoC Blocks

24. Digital Clocks

This chapter discusses the Digital Clocks and its associated registers. It serves as an overview of the clocking options available in the PSoC devices. For detailed information on specific oscillators, see the individual oscillator chapters in the section called “CORE ARCHITECTURE” on page 33.

Table 24-1. Digital Clocking Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,DAh	INT_CLR0	VC3	Sleep	GPIO			Analog 1	Analog 0	V Monitor	RW : 00
0,E0h	INT_MSK0	VC3	Sleep	GPIO			Analog 1	Analog 0	V Monitor	RW : 00
1,DEh	OSC_CR4							VC3 Input Select[1:0]		RW : 00
1,DFh	OSC_CR3	VC3 Divider[7:0]								RW : 00
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00
1,E1h	OSC_CR1	VC1 Divider[3:0]				VC2 Divider[3:0]				RW : 00
1,E2h	OSC_CR2	PLLGAIN					EXTCLKEN	IMODIS	SYSCLKX2 DIS	RW : 00

24.1 Architectural Description

The PSoC M8C core has a large number of clock sources that increase the flexibility of the PSoC mixed signal arrays, as illustrated in Figure 24-1.

24.1.1 Internal Main Oscillator

The Internal Main Oscillator (IMO) is the foundation upon which almost all other clock sources in the PSoC mixed signal arrays are based. The default mode of the IMO creates a 24 MHz reference clock that is used by many other circuits in the chip. The IMO may also be configured to operate in a PLL mode where the oscillator is locked to a precision 32.768 kHz crystal reference. The PSoC device has an option to replace the IMO with an externally supplied clock that will become the base for all of the clocks the IMO normally serves.

Whether the external clock or the internal main oscillator is selected, all chip functions are clocked from a derivative of SYSCLK or are re-synchronized to SYSCLK. All external asynchronous signals (through row inputs), as well as the selected 32K oscillator, are resynchronized to SYSCLK for use in the digital blocks.

The IMO is discussed in detail in the chapter “Internal Main Oscillator (IMO)” on page 65.

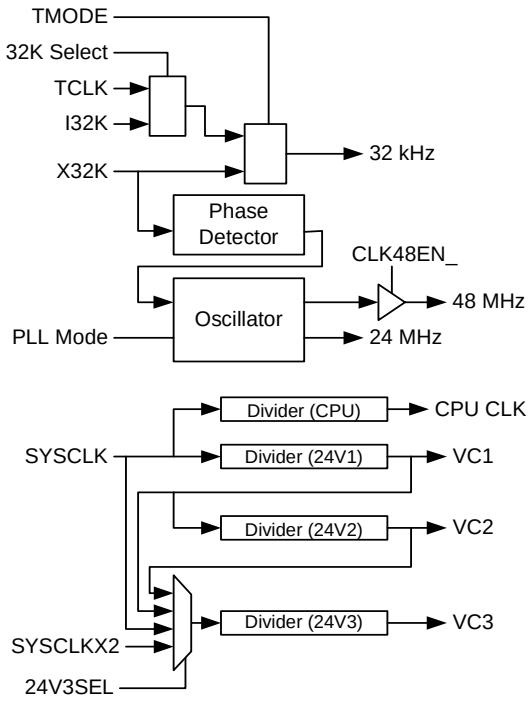


Figure 24-1. Overview of PSoC Clock Sources

27. I²C

This chapter explains the I2C block and its associated registers. The I2C communications block is a serial processor designed to implement a complete I2C Slave and/or Master.

Table 27-1. I2C Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D6h	I2C_CFG		PSelect	Bus Error IE	Stop IE	Clock Rate		Enable Master	Enable Slave	RW : 00
0,D7h	I2C_SCR	Bus Error	Lost Arb	Stop Status	ACK	Address	Transmit	LRB	Byte Complete	R : 00
0,D8h	I2C_DR	Data[7:0]								RW : 00
0,D9h	I2C_MSCR					Bus Busy	Master Mode	Restart Gen	Start Gen	R : 00

The I2C communications block is a serial to parallel processor designed to interface to the two-wire I2C serial communications bus. The block provides I2C specific support for status detection and generation of framing bits, to eliminate the need for excessive host processor intervention and overhead.

The I2C block will directly control the data (SDA) and clock (SCL) signals to the external I2C interface, through connections to two dedicated GPIO pins. The host firmware will interact with the block through IO register reads and writes, and firmware synchronization will be implemented through polling and/or interrupts.

Functionality requirements include:

- Master/Slave, Transmitter/Receiver Operation
- Byte processing for low CPU overhead
- Interrupt or Polling CPU interface
- Master Clock Rates: 50K, 100K, 400K
- Multi-Master Clock Synchronization
- Multi-Master Mode Arbitration support
- 7- or 10-bit addressing (through firmware support)
- SMBus operation (through firmware support)

Hardware functionality provides basic I2C control, data, and status primitives. A combination of hardware support and firmware command sequencing provides a high degree of flexibility for implementing the required I2C functionality.

Hardware limitations:

1. There is no hardware support for automatic address comparison. When Slave mode is enabled, every slave address will cause the block to interrupt the host and possibly stall the bus.
2. Since receive and transmitted data is not buffered, there is no support for automatic receive acknowledge. The host processor must intervene at the boundary of each byte and either send a byte or ACK received bytes.

27.1 Architectural Description

The I2C block is designed to support a set of primitive operations and detect a set of status conditions specific to the I2C protocol. These primitive operations and conditions are manipulated and combined at the firmware level to support the required data transfer modes. The host will set up control options and issue commands to the unit through IO Writes and obtain status through IO Reads and interrupts.

The block operates as either a Slave, a Master, or both. When enabled in Slave mode, the unit is always listening for a Start condition, or sending or receiving data. Master mode can work in conjunction with Slave mode. The Master supplies the ability to generate the START or STOP condition and determine if other masters are on the bus. For Multi Master mode, clock synchronization is supported. If Master mode is enabled and Slave mode is not enabled, the block does not generate interrupts on externally generated Start conditions.

27.1.1 Basic I2C Data Transfer

Figure 27-1 shows the basic form of data transfers on the I2C bus with a 7-bit address format. (For a more detailed description, see the I2C Specification, Version 2.1, by Phillips Semiconductor).

28. POR and LVD

This chapter briefly discusses the POR and LVD circuits and their associated registers.

Table 28-1. POR and LVD Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E3h	VLT_CR	SMP		PORLEV[1:0]		LVDTBEN	VM[2:0]			RW : 00
1,E4h	VLT_CMP						PUMP	LVD	PPOR	R : 00

28.1 Architectural Description

Power-on-Reset (POR) and Low Voltage Detect (LVD) circuits provide protection against low voltage conditions. The POR function senses Vdd and holds the system in reset, until the magnitude of Vdd will support operation to spec. The LVD function senses Vdd and provides an interrupt to the system when Vdd falls below a selected threshold. Other outputs and status bits are provided to indicate important voltage trip levels.

28.2 Register Definitions

This block contains two registers: VLT_CR and VLT_CMP (read only status bits).

28.2.1 VLT_CR Register

The VLT_CR register is cleared by all resets, which can cause reset-cycling during very slow supply ramps to 5V, when the POR Range is set for the 5V range. This is because the reset will clear the POR range setting back to 3V and a new boot/start-up occurs (possibly many times). The user can manage this with sleep mode and/or reading voltage status bits, if such cycling is an issue.

Bit 7: SMP. SMP low enables the switch mode pump circuit to run, when PUMP is high.

Bit 6: Reserved.

Bits 5 and 4: PORLEV[1:0]. PORLEV[1:0] sets the Vdd level at which PPOR switches.

Bit 3: LVDTBEN. LVDTBEN is AND'ed with LVD to produce a throttle-back signal that reduces CPU clock speed when low voltage conditions are detected.

Bits 2, 1, and 0: VM[2:0]. VM[2:0] sets the Vdd level of the LVD and the Pump Comparator switches.

For additional information, reference the VLT_CR register on page 179.

28.2.2 VLT_CMP Register

Bits 7 to 3: Reserved.

Bit 2: PUMP. PUMP reads the state of the Switch Mode Pump Vdd comparator. The trip points for both LVD and PUMP are set by VM[2:0] in the VLT_CR register.

Bit 1: LVD. LVD reads the state of the low voltage detect comparator. The trip points for both LVD and PUMP are set by VM[2:0] in the VLT_CR register.

Bit 0: PPOR. The PPOR bit reads back the state of the PPOR output. This can only be meaningfully read with PORLEV[1:0] set to disable PPOR. In that case, the PPOR status bit shows the comparator state directly.

For additional information, reference the VLT_CMP register on page 180.

31. System Resets

This chapter discusses the System Resets and its associated registers. The M8C supports several types of resets. The various resets are designed to provide error-free operation during power up for any voltage ramping profile, to allow for user-supplied external reset and to provide recovery from errant code operation.

Table 31-1. System Reset Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,FEh	CPU_SCR1								IRAMDIS	RW : 00
0,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	RW : XX

LEGEND
XX: The reset value is 10h after POR/XRES and 20h after a watchdog reset.

When reset is initiated, all registers are restored to their default states. Minor exceptions are explained below.

The following types of resets can occur:

- Power-on Reset (POR). This occurs at low supply voltage and is comprised of multiple sources.
- External Reset (XRES). This active high reset is driven into the chip, on parts that contain an Xres pin.
- Watchdog Reset (WDR). This optional reset occurs when a timer expires, before being cleared by user firmware.
- Internal Reset (IRES). This occurs during the boot sequence, if the SROM code determines that Flash reads are not valid.

The occurrence of a reset is recorded in the Status and Control Register (CPU_SCR, for POR/XRES/WDR), or in the System Status and Control Register 1 (CPU_SCR1, for IRESS). Firmware can interrogate these registers to determine the cause of a reset.

31.1 Register Definitions

31.1.1 CPU_SCR0 Register

The bits of the CPU_SCR0 register are used to convey status and control of events for various functions of a PSoC device.

Bit 7: GIES. The Global Interrupt Enable Status bit is a read only status bit and its use is discouraged. The GIES bit is a legacy bit which was used to provide the ability to read the GIE bit of the CPU_F register. However, the CPU_F register is now readable. When this bit is set, it indicates that the GIE bit in the CPU_F register is also set which, in turn, indicates that the microprocessor will service interrupts.

Bit 6: Reserved.

Bit 5: WDRS. The WatchDog Reset Status bit is normally zero, but set whenever a watchdog reset occurs. The bit is readable and clearable by writing a zero to its bit position in the CPU_SCR0 register. This bit may not be set.

Bit 4: PORS. The Power-On Reset Status (PORS) bit and watchdog enable bit will be set automatically by a POR or external reset. If the bit is cleared by user code, the watchdog timer will be enabled. Once cleared, the only way to reset the PORS bit is to go through a POR or external reset. Thus, there is no way to disable the watchdog timer, other than to go through a POR or external reset.

Bit 3: Sleep. The Sleep bit is used to enter low power Sleep mode when set, as described in this chapter.

Bits 2 and 1: Reserved.

Bit 0: STOP. The STOP bit is readable and writeable. When set, the PSoC M8C will stop executing code until a reset event occurs. This can be either a POR, watchdog reset, or external reset. If an application wants to stop code execution until a reset, the preferred method would be to use the HALT instruction rather than a register write to this bit.

For additional information, reference the CPU_SCR0 register on page 154.