

15.3.10 I2C_SCR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D7h	I2C_SCR	Bus Error		Stop Status	ACK	Address	Transmit	LRB	Byte Complete	# : 00

LEGEND

Access is bit specific.

The I²C Status and Control Register (I2C_SCR) is used by the slave to control the flow of data bytes and to keep track of the bus state during a transfer.

This register contains status bits to determine the state of the current I²C transfer, and control bits to determine the actions for the next byte transfer. At the end of each byte transfer, the I²C hardware interrupts the M8C microcontroller and stalls the I²C bus on the subsequent low of the clock, until the PSoC device intervenes with the next command. This register may be read as many times as necessary; but on a subsequent write to this register, the bus stall is released and the current transfer continues.

There are five status bits: Byte Complete, LRB, Address, Stop Status, and Bus Error. These bits have Read/Clear (RC) access, which means that they are set by hardware but may be cleared by a write of '0' to the bit position. Under certain conditions, status is cleared automatically by the hardware.

There are two control bits: Transmit and ACK. These bits have RW access and may be cleared by hardware.

Bit 7: Bus Error. The Bus Error status detects misplaced Start or Stop conditions on the bus. These may be due to noise, rogue devices, or other devices that are not yet synchronized with the I²C bus traffic. According to the I²C specification, all compatible devices must reset their interface upon a received Start or Stop. This is a natural thing to do in slave mode because a Start initiates an address reception and a Stop idles the slave.

A bus error is defined as follows. A Start is only valid if the block is idle or a slave receiver is ready to receive the first bit of a new byte after an ACK. Any other timing for a Start condition sets the Bus Error bit. A Stop is only valid if the block is idle or a slave receiver is ready to receive the first bit of a new byte after an ACK. Any other timing for a Stop condition sets the Bus Error bit.

Bit 5: Stop Status. Stop status is set upon detection of an I²C Stop condition. This bit is sticky, which means that it remains set until a '0' is written back to it by the firmware. This bit may only be cleared if the Byte Complete status bit is set. If the Stop Interrupt Enable bit is set, an interrupt is also generated upon Stop detection. It is never automatically cleared. Using this bit, a slave can distinguish between a previous Stop or Restart upon a given address byte interrupt.

The selections are shown in the following table:

Bit	Access	Description
7	RC	Bus Error 1 = A misplaced Start or Stop condition was detected. This status bit must be cleared by firmware with a write of '0' to the bit position. It is never cleared by the hardware.
5	RC	Stop Status 1 = A Stop condition was detected. This status bit must be cleared by firmware with a write of '0' to the bit position. It is never cleared by the hardware.
4	RW	ACK: Acknowledge Out 0 = NACK the last received byte. 1 = ACK the last received byte. This bit is automatically cleared by hardware upon the following byte complete event.
3	RC	Address 1 = The transmitted or received byte is an address. This status bit must be cleared by firmware with a write of '0' to the bit position.
2	RW	Transmit 0 = Receive Mode. 1 = Transmit Mode. This bit is set by firmware to define the direction of the byte transfer. Any Start detect automatically clears this bit.
1	RC	LRB: Last Received Bit The value of the ninth bit in a transmit sequence, which is the acknowledge bit from the receiver. 0 = Last transmitted byte was ACK'ed by the receiver. 1 = Last transmitted byte was NACK'ed by the receiver. Any Start detect automatically clears this bit.
0	RC	Byte Complete Transmit Mode: 1 = 8 bits of data have been transmitted and an ACK or NACK has been received. Receive Mode: 1 = 8 bits of data have been received. Any Start detect automatically clears this bit.

Bit 4: ACK.

This control bit defines the acknowledge data bit that is transmitted out in response to a received byte. When receiving, a byte complete interrupt is generated after the eighth data bit is received. Upon the subsequent write to this register to continue (or terminate) the transfer, the state of this bit determines the next transmitted data bit. It is **active high**. A '1' sends an ACK and a '0' sends a NACK. A slave receiver sends a NACK to inform the master that it cannot receive any more bytes.

Bit 3: Address.

This bit is set when an address is received. This consists of a Start or Restart, and an address byte.

In slave mode when this status is set, firmware reads the received address from the data register and compares it with its own address. If the address does not match, the firmware writes a NACK indication to this register. No further interrupts occur until the next address is received. If the address does match, firmware must ACK the received byte, then byte complete interrupts are generated on subsequent bytes of the transfer.

When in I²C compatibility mode the Address bit should not be cleared between a start and the next byte complete. If it is cleared during this time the start status is cleared internally.

Bit 2: Transmit. This bit sets the direction of the shifter for a subsequent byte transfer. The shifter is always shifting in data from the I²C bus, but a write of '1' enables the output of the shifter to drive the SDA output line. Since a write to this register initiates the next transfer, data must be written to the data register before writing this bit. In receive mode, the previously received data must have been read from the data register before this write. Firmware derives this direction from the RW bit in the received slave address.

This direction control is only valid for data transfers. The direction of address bytes is determined by the hardware.

Bit 1: LRB. Last Received Bit. This is the last received bit in response to a previously transmitted byte. In transmit mode, the hardware sends a byte from the data register and

clock in an acknowledge bit from the receiver. Upon the subsequent byte complete interrupt, firmware checks the value of this bit. A '0' is the ACK value and a '1' is a NACK value.

The meaning of LRB depends upon the current operating mode.

'0': ACK. The master wants to read another byte. The slave loads the next byte into the [I2C_DR Register](#) and sets the Transmit bit in the I2C_SCR register to continue the transfer.

'1': NACK. The master is done reading bytes. The slave reverts to IDLE state on the subsequent I2C_SCR write (regardless of the value written).

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 are 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 [Timing Diagrams on page 130](#)). If the PSoC device responds with a write back to this register before the subsequent falling edge of SCL (which is approximately one-half bit time), the transfer continues without interruption. However, if the device cannot respond within that time, the hardware holds the SCL line low, stalling the I²C bus. A subsequent write to the I2C_SCR register releases the stall.

For additional information, refer to the [I2C_SCR register on page 240](#).

15.3.11 I2C_DR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D8h	I2C_DR	Data[7:0]								RW : 00

The I²C Data Register (I2C_DR) provides read/write access to the Shift register.

Bits 7 to 0: Data[7:0]. This register is not buffered; therefore, writes and valid data reads occur at specific points in the transfer. These cases are outlined as follows:

- **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 I2C_DR register

before writing to the [I2C_SCR Register](#), which continues the transfer.

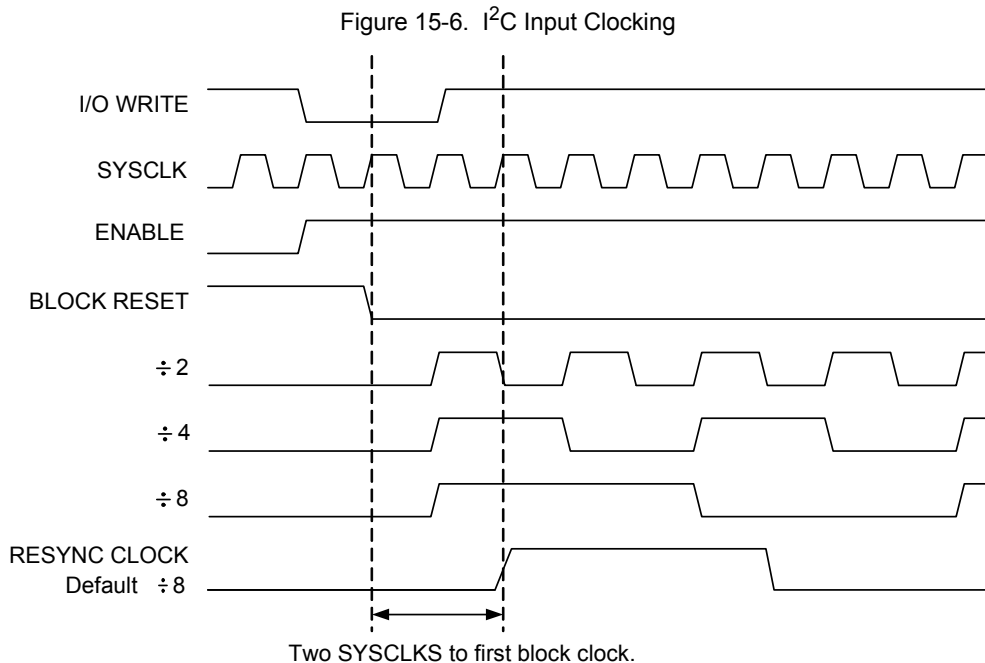
- **Slave Transmitter** – Data bytes must be written to the I2C_DR register before the Transmit bit is set in the [I2C_SCR Register](#), which continues the transfer.

For additional information, refer to the [I2C_DR register on page 241](#).

15.4 Timing Diagrams

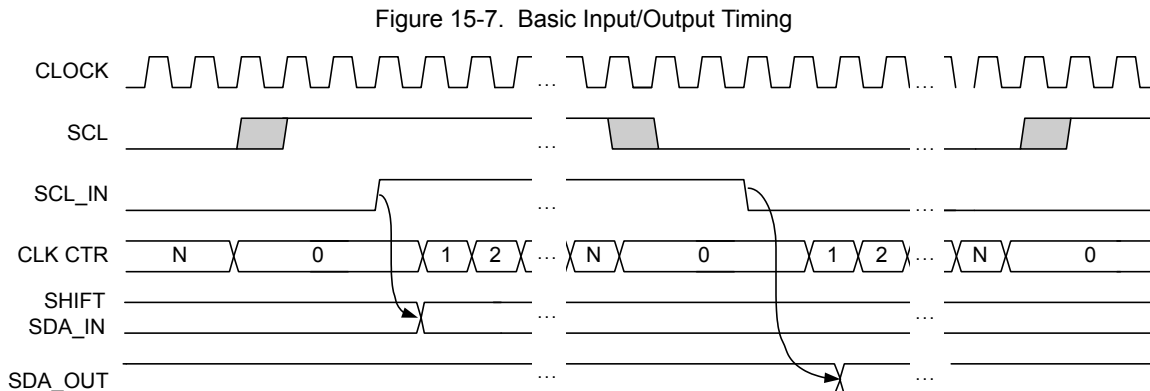
15.4.1 Clock Generation

Figure 15-6 illustrates the I²C input clocking scheme. The SYSCLK pin is an input into a three-stage ripple divider that provides the baud rate selections. When the block is disabled, all internal state is held in a reset state. When the Enable bit in the I2C_CFG Register is set, the reset is synchronously released and the clock generation is enabled. All three taps from the **ripple divider** are selectable ($/2$, $/4$, $/8$) from the clock rate bits in the I2C_CFG Register. If any of the three divider taps is selected, that clock is resynchronized to SYSCLK. The resulting clock is routed to all of the synchronous elements in the design.



15.4.2 Basic I/O Timing

Figure 15-7 illustrates basic input/output timing that is valid for both 16 times sampling and 32 times sampling. For 16 times sampling, $N=4$; for 32 times sampling, $N=12$. N is derived from the half-bit rate sampling of eight and 16 clocks, respectively, minus the input latency of three (count of 4 and 12 correspond to 5 and 13 clocks).



15.4.3 Status Timing

Figure 15-8 illustrates the interrupt timing for byte complete, which occurs on the positive edge of the ninth clock (byte + ACK/NACK) in transmit mode and on the positive edge of the eighth clock in receive mode. There is a maximum of three cycles of latency due to the input synchronizer/filter circuit. As shown, the interrupt occurs on the clock following a valid SCL positive edge input transition (after the synchronizers). The Address bit is set with the same timing but only after a slave address has been received. The LRB (Last Received Bit) status is also set with the same timing but only on the ninth bit after a transmitted byte.

Figure 15-8. Byte Complete, Address, LRB Timing

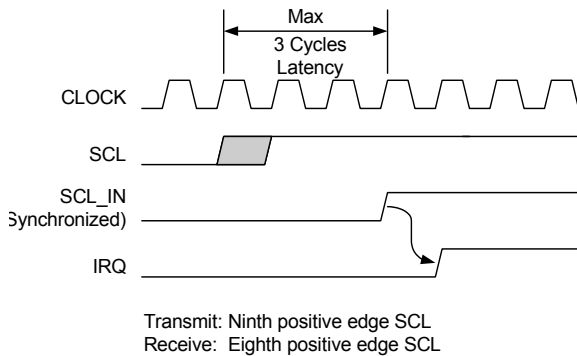


Figure 15-9 shows the timing for Stop status. This bit is set (and the interrupt occurs) two clocks after the synchronized and filtered SDA line transitions to a '1', when the SCL line is high.

Figure 15-9. Stop Status and Interrupt Timing

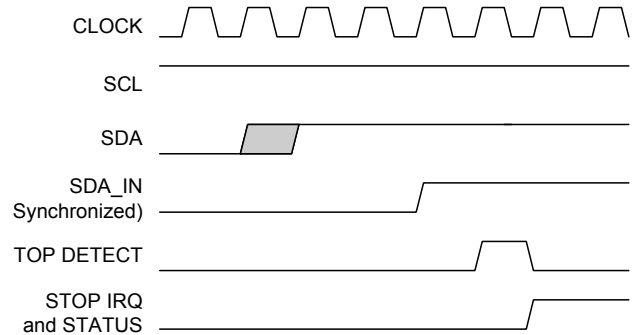
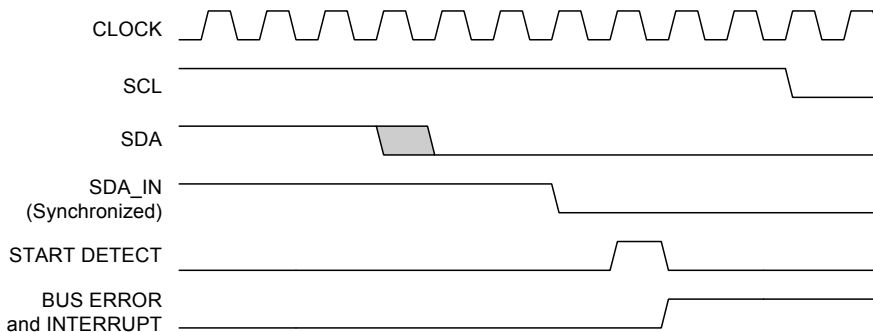


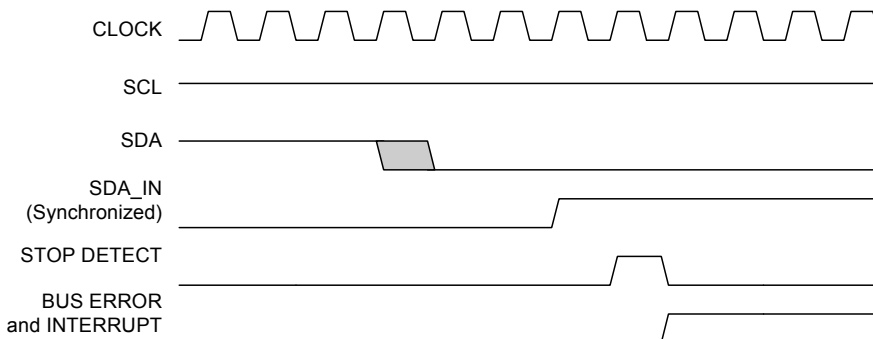
Figure 15-10 illustrates the timing for bus error interrupts. Bus Error status (and interrupt) occurs one cycle after the internal Start or Stop detect (two cycles after the filtered and synchronized SDA input transition).

Figure 15-10. Bus Error Interrupt Timing

Misplaced Start



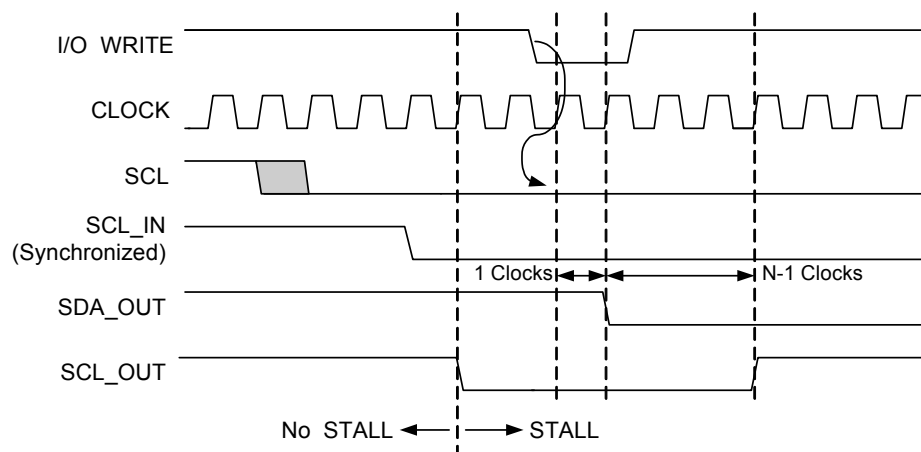
Misplaced Stop



15.4.4 Slave Stall Timing

When a byte complete interrupt occurs, the PSoC device firmware must respond with a write to the [I2C_SCR Register](#) to continue the transfer (or terminate the transfer). The interrupt occurs two clocks after the rising edge of SCL_IN (see [Status Timing on page 131](#)). As illustrated in [Figure 15-11](#), firmware has until one clock after the falling edge of SCL_IN to write to the [I2C_SCR Register](#); otherwise, a stall occurs. After stalled, the I/O write releases the stall. The setup time between data output and the next rising edge of SCL is always N-1 clocks.

Figure 15-11. Slave Stall Timing



15.4.5 Implementation

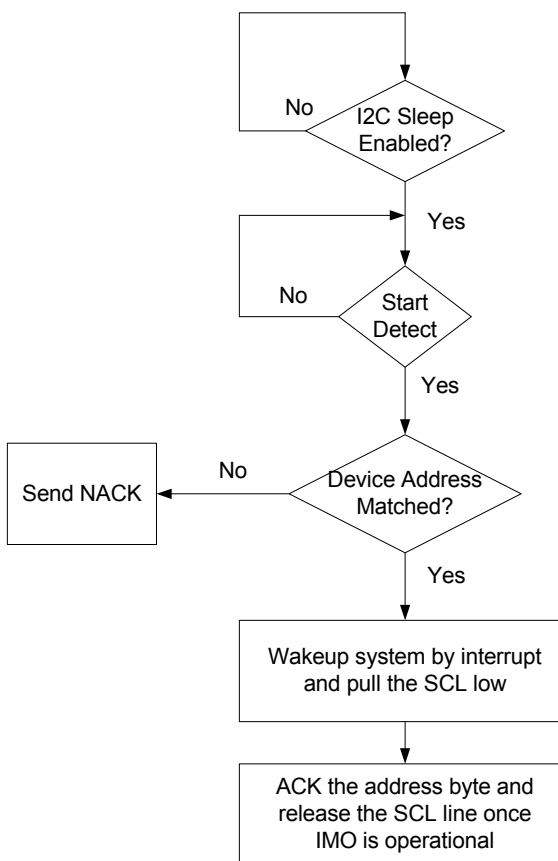
The I2C block responds to transactions during sleep if and only if:

- The I2C slave block is enabled, i.e., bit 0 of the [I2C_CFG Register](#) is set to 1'b1.
- If I2C_ON is set or the USB Enable bit of USB_CR0 is set.
- PD signal from the sleep controller is high.

To enable the wakeup through I2C, set the HW Addr EN bit so that the I2C slave block wakes the system if and only if the address matches. [Figure 15-12](#) depicts the wakeup sequence through I2C.

Note The last step in this flowchart where SCL is released in general, represents the configuration where buffer mode is enabled.

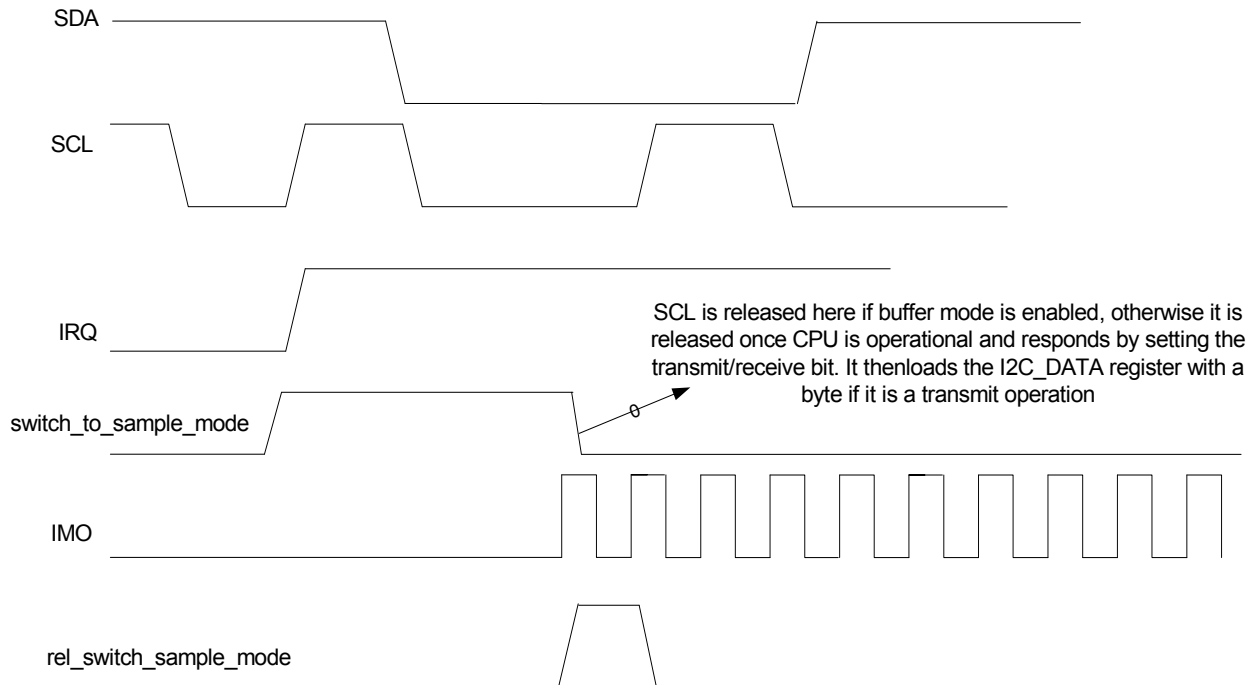
Figure 15-12. I2C Wakeup Sequence



15.4.6 Compatibility Mode Configuration

In compatibility mode, the SCL, as usual, is pulled low until the CPU responds by setting the Transmit/Receive bit and for loading a byte in the I2C_DATA register (in case of transmit operation) even though IMO is operational. Figure 15-13 illustrates the process of switching from direct clocking to sampled mode.

Figure 15-13. Direct Clocking to Sample Mode Timing



See the following notes regarding Figure 15-13:.

- After, the address matches IRQ and a signal called switch_to_sample_mode are asserted asynchronously.
- This IRQ goes as input to the sleep controller block. The sleep controller block wakes the system by clearing the sleep bit.
- After IMO is operational, the signal switch_to_sample_mode is registered and asserted high, indicating IMO is operational. When this registered signal becomes high, the switch_to_sample_mode signal deasserts asynchronously.
- The state machines and other logic are put at appropriate states when switching to sample mode.

I2C Slave

16. System Resets



This chapter discusses the System Resets and their associated registers. PSoC devices support 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. For a complete table of the System Reset registers, refer to the [Summary Table of the System Resource Registers on page 106](#). For a quick reference of all PSoC registers in address order, refer to the [Register Reference chapter on page 187](#).

16.1 Architectural Description

When reset is initiated, all registers are restored to their default states. In the [Register Reference chapter on page 187](#), this is indicated by the POR row in the register tables and elsewhere it is indicated in the Access column values on the right side of the colon, in the register tables. Minor exceptions are explained ahead.

The following types of resets occur in the PSoC device:

- **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 PSoC device on parts that contain an XRES pin.
- **Watchdog Reset (WDR).** This optional reset occurs when the watchdog timer expires before being cleared by user firmware. Watchdog resets default to off.
- **Internal Reset (IRES).** This occurs during the boot sequence if the SROM code determines that Flash reads are invalid.

The occurrence of a reset is recorded in the Status and Control registers (CPU_SCR0 for POR, XRES, and 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.

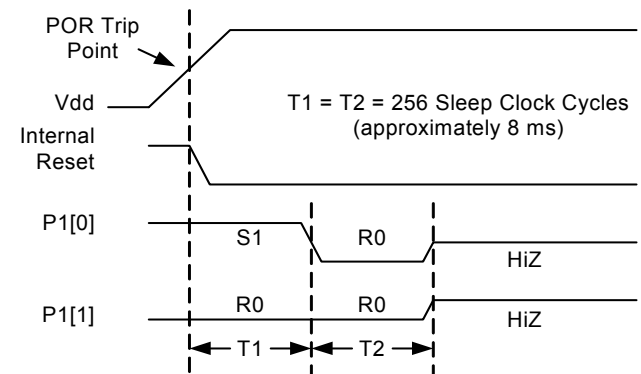
16.2 Pin Behavior During Reset

Power on Reset and External Reset cause toggling on two GPIO pins, P1[0] and P1[1], as described ahead and illustrated in [Figure 16-1](#) and [Figure 16-2](#). This allows programmers to synchronize with the PSoC device. All other GPIO pins are placed in a high impedance state during and immediately following reset.

16.2.1 GPIO Behavior on Power Up

At power up, the internal POR causes P1[0] to initially drive a strong high (1) while P1[1] drives a resistive low (0). After 256 sleep oscillator cycles (approximately 8 ms), the P1[0] signal transitions to a resistive low state. After an additional 256 sleep oscillator clocks, both pins transition to a high impedance state and normal CPU operation begins. This is illustrated in the following figure.

Figure 16-1. P1[1:0] Behavior on Power Up



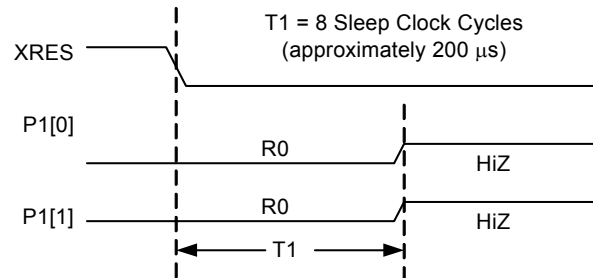
16.2.2 Powerup External Reset Behavior

The device's core runs on chip regulated supply, so there is a time delay in powering up the core. A short XRES pulse at power up causes an external reset startup behavior. However, the event is latched and applied only after the core has powered up (a delay of about 1 ms).

16.2.3 GPIO Behavior on External Reset

During External Reset (XRES=1), both P1[0] and P1[1] drive resistive low (0). After XRES deasserts, these pins continue to drive resistive low for another eight sleep clock cycles (approximately 200 μ s). After this time, both pins transition to a high impedance state and normal CPU operation begins. This is illustrated in [Figure 16-2](#).

Figure 16-2. P1[1:0] Behavior on External Reset (XRES)



16.3 Register Definitions

The following registers are associated with the PSoC System Resets and are listed in address order. Each register description has an associated register table showing the bit structure for that register. The bits in the tables that are grayed out are reserved bits and are not detailed in the register descriptions that follow. Always write reserved bits with a value of 0. For a complete table of system reset registers, refer to the [“Summary Table of the System Resource Registers” on page 106](#).

16.3.1 CPU_SCR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,FEh	CPU_SCR1	IRESS			SLIMO[1:0]				IRAMDIS	# : 0

LEGEND

x An “x” before the comma in the address field indicates that this register can be read or written to no matter what bank is used.
Access is bit specific. Refer to the [Register Reference chapter on page 187](#) for additional information.

The System Status and Control Register 1 (CPU_SCR1) conveys the status and control of events related to internal resets and watchdog reset.

Bit 7: IRESS. Internal Reset Status. This bit is a read only bit that determines if the booting process occurred more than once.

When this bit is set, it indicates that the SROM SWBootReset code executed more than once. If this bit is not set, the SWBootReset executed only once. In either case, the SWBootReset code does not allow execution from code stored in Flash until the M8C core is in a safe operating mode with respect to supply voltage and Flash operation. There is no need for concern when this bit is set. It is provided for systems that may be sensitive to boot time, so that they can determine if the normal one-pass boot time was exceeded.

For more information on the SWBootReset code see the [Supervisory ROM \(SROM\) chapter on page 89](#).

Bit 4:3 SLIMO[1:0]. These bits set the IMO frequency range. See the table ahead for more information. These

changes allow for optimization of speed and power. The IMO trim value must also be changed when SLIMO is changed (see [Engaging Slow IMO on page 63](#)).

When not in external clocking mode, the IMO is the source for SYSCLK; therefore, when the speed of the IMO changes so does SYSCLK.

SLIMO	CY8CTMG20x, CY8CTST200
00	12
01	6
10	24
11	Reserved

Bit 0: IRAMDIS. Initialize RAM Disable. This bit is a control bit that is readable and writeable. The **default value** for this bit is ‘0’, which indicates that the maximum amount of SRAM must be initialized on watchdog reset to a value of 00h. When the bit is ‘1’, the minimum amount of SRAM is initialized after a watchdog reset.

For additional information, refer to the [CPU_SCR1 register on page 257](#).

16.3.2 CPU_SCR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	# : XX

LEGEND

Access is bit specific. Refer to register detail for additional information.

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

The System Status and Control Register 0 (CPU_SCR0) is used to convey the status and control of events for various functions of a PSoC device.

Bit 7: GIES. Global Interrupt Enable Status. This bit is a read only status bit and its use is discouraged. The GIES bit is a legacy bit that 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 services interrupts.

Bit 5: WDRS. WatchDog Reset Status. This bit may not be set. It is normally '0' and automatically set whenever a watchdog reset occurs. The bit is readable and clearable by writing a '0' to its bit position in the CPU_SCR0 register.

Bit 4: PORS. Power On Reset Status. This bit, which is the watchdog enable bit, is set automatically by a POR or External Reset (XRES). If the bit is cleared by user code, the watchdog timer is enabled. After cleared, the only way to reset the PORS bit is to go through a POR or XRES. Thus, there is no way to disable the watchdog timer other than to go through a POR or XRES.

Bit 3: Sleep. This bit is used to enter Low Power Sleep mode when set. To wake up the system, this register bit is cleared asynchronously by any enabled interrupt. There are two special features of this bit that ensure proper sleep operation. First, the write to set the register bit is blocked if an interrupt is about to be taken on that instruction boundary (immediately after the write). Second, there is a hardware interlock to ensure that, once set, the SLEEP bit may not be cleared by an incoming interrupt until the sleep circuit has finished performing the sleep sequence and the system-wide power down signal has been asserted. This prevents the sleep circuit from being interrupted in the middle of the process of system power down, possibly leaving the system in an indeterminate state.

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

For additional information, refer to the [CPU_SCR0 register on page 258](#).

16.4 Timing Diagrams

16.4.1 Power On Reset

A Power on Reset (POR) is triggered whenever the supply voltage is below the POR trip point. POR ends once the supply voltage rises above this voltage. Refer to the [POR and LVD chapter on page 143](#) for more information on the operation of the POR block.

POR consists of two pieces: an Imprecise POR (IPOR) and a Precision POR (PPOR). “POR” refers to the OR of these two functions. IPOR has coarser accuracy and its trip point is typically lower than PPOR’s trip point. PPOR is derived from a circuit that is calibrated (during boot) for a very accurate location of the POR trip point.

During POR (POR=1), the IMO is powered off for low power during startup. After POR deasserts, the IMO is started (see [Figure 16-4](#)).

POR configures register reset status bits as shown in [16.4.4 Reset Details on page 141](#).

16.4.2 External Reset

An External Reset (XRES) is caused by pulling the XRES pin high. The XRES pin has an always-on, pull down resistor, so it does not require an external pull down for operation and can be tied directly to ground or left open. Behavior after XRES is similar to POR.

During XRES (XRES=1), the IMO is powered off for low power during startup. After XRES deasserts, the IMO is started (see [Figure 16-4](#)).

How the XRES configures register reset status bits is shown in [16.4.4 Reset Details on page 141](#).

16.4.3 Watchdog Timer Reset

The user has the option to enable the Watchdog Timer Reset (WDR), by clearing the PORS bit in the CPU_SCR0 register. After the PORS bit is cleared, the watchdog timer cannot be disabled. The only exception to this is if a POR/XRES event takes place, which disables the WDR. Note that a WDR does not clear the watchdog timer. See [Watchdog Timer on page 81](#) for details of the watchdog operation.

When the watchdog timer expires, a watchdog event occurs, resulting in the reset sequence. Some characteristics unique to the WDR are as follows.

- PSoC device reset asserts for one cycle of the CLK32K clock (at its reset state).
- The IMO is not halted during or after WDR (that is, the part does not go through a low power phase).
- CPU operation re-starts one CLK32K cycle after the internal reset deasserts (see [Figure 16-3](#)).

How the WDR configures register reset status bits is shown in [16.4.4 Reset Details on page 141](#).

Figure 16-3. Key Signals During WDR

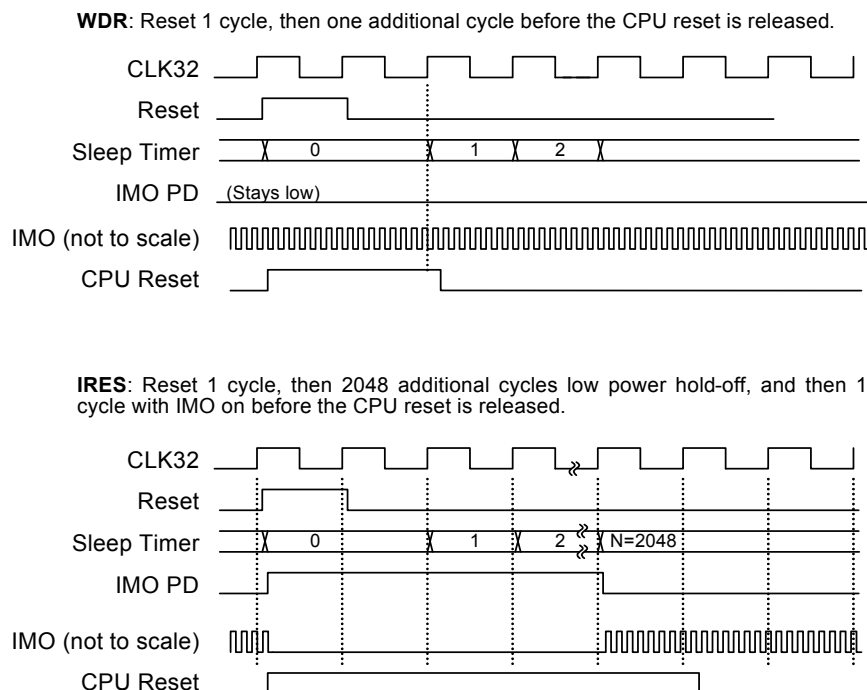
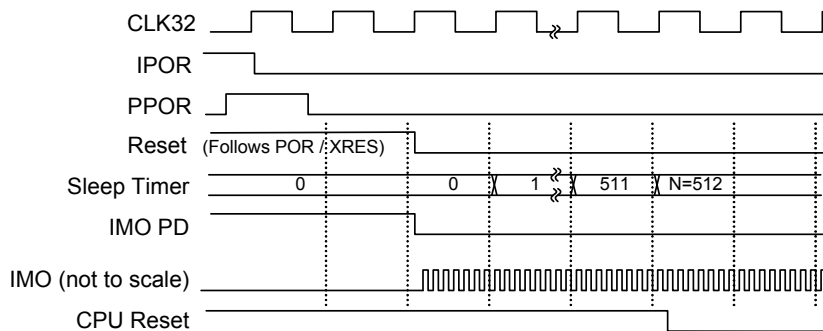
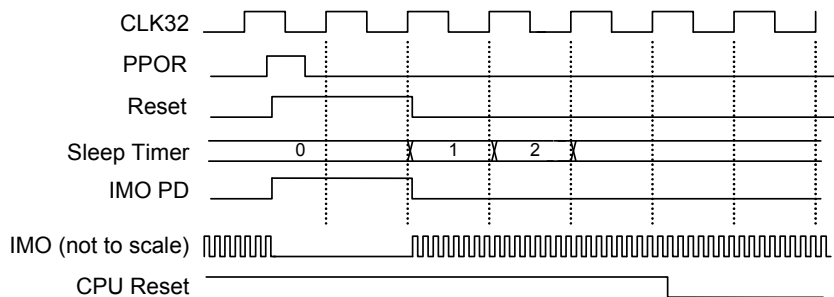


Figure 16-4. Key Signals During POR and XRES

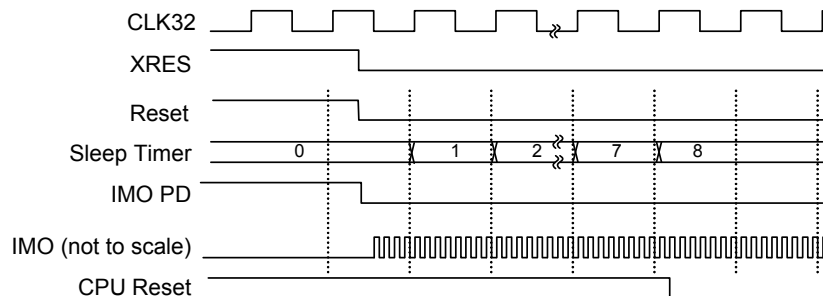
POR (IPOR followed by PPOR): Reset while POR is high (IMO off), then 511(+) cycles (IMO on), and then the CPU reset is released. **XRES** is the same, with N=8.



PPOR (with no IPOR): Reset while PPOR is high and to the end of the next 32K cycle (IMO off); 1 cycle IMO on before the CPU reset is released. Note that at the 3V level, PPOR tends to be brief because the reset clears the POR range register (VLT_CR) back to the default 2.4V setting.



XRES: Reset while XRES is high (IMO off), then 7(+) cycles (IMO on), and then the CPU reset is released.



16.4.4 Reset Details

Timing and functionality details are summarized in [Table 16-1](#). [Figure 16-4 on page 140](#) shows some of the relevant signals for IPOR, PPOR, XRES, and WDR.

Table 16-1. Reset Functionality

Item	IPOR (Part of POR)	PPOR (Part of POR)	XRES	WDR
Reset Length	While POR=1	While PPOR=1, plus 30-60 μ s (1-2 clocks)	While XRES=1	30 μ s (1 clock)
Low Power (IMO Off) During Reset?	Yes	Yes	Yes	No
Low Power Wait Following Reset?	No	No	No	No
CLK32K Cycles from End of Reset to CPU Reset Deasserts ^a	512	1	8	1
Register Reset (See Next Line for CPU_SCR0, CPU_SCR1)	All	All, except PPOR does not reset Bandgap Trim register	All	All
Reset Status Bits in CPU_SCR0, CPU_SCR1	Set PORS, Clear WDRS, Clear IRAMDIS	Set PORS, Clear WDRS, Clear IRAMDIS	Set PORS, Clear WDRS, Clear IRAMDIS	Clear PORS, Set WDRS, IRAMDIS unchanged
Bandgap Power	On	On	On	On
Boot Time ^b	2.2 ms	2.2 ms	2.2 ms	2.2 ms

a. CPU reset is released after synchronization with the CPU clock.

b. Measured from CPU reset release to execution of the code at Flash address 0x0000.

16.5 Power Modes

The ILO block drives the CLK32K clock used to time most events during the reset sequence. This clock is powered down by IPOR but not by any other reset. The sleep timer provides interval timing.

While POR or XRES assert, the IMO is powered off to reduce startup power consumption.

During and following IRES (for 64 ms nominally), the IMO is powered off for low average power during slow supply ramps.

During and after POR or XRES, the bandgap circuit is powered up.

The IMO is always on for at least one CLK32K cycle before CPU reset is deasserted.

17. POR and LVD



This chapter briefly discusses the Power on Reset (POR) and Low Voltage Detect (LVD) circuits and their associated registers. For a complete table of the POR registers, refer to the [Summary Table of the System Resource Registers on page 106](#). For a quick reference of all PSoC registers in address order, refer to the [Register Reference chapter on page 187](#).

17.1 Architectural Description

The Power on Reset (POR) and Low Voltage Detect (LVD) circuits provide protection against low voltage conditions. The POR function senses Vcc and Vcore (regulated voltage) holding the system in reset until the magnitude of Vcc and Vcore supports operation to specification. The LVD function senses Vcc and provides an interrupt to the system when Vcc falls below a selected threshold. Other outputs and status bits are provided to indicate important voltage trip levels. Refer to [Section 16.2 Pin Behavior During Reset](#) for a description of GPIO pin behavior during power up.

17.2 Register Definitions

The following registers are associated with the POR and LVD, and are listed in address order. The register descriptions below have an associated register table showing the bit structure. The bits that are grayed out in the register tables are reserved bits and are not detailed in the register descriptions that follow. Reserved bits must always be written with a value of '0'. For a complete table of the POR registers, refer to the [Summary Table of the System Resource Registers on page 106](#).

17.2.1 VLT_CR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E3h	VLT_CR			PORLEV[1:0]		LVDTBEN		VM[2:0]		RW : 00

The Voltage Monitor Control Register (VLT_CR) sets the trip points for POR, MON1 and LVD.

The VLT_CR register is cleared by all resets. This can cause reset cycling during very slow supply ramps to 5V when the MON1 range is set for the 5V range. This is because the reset clears the MON1 range setting back to 1.8V and a new boot or startup occurs (possibly many times). You can manage this with sleep mode or reading voltage status bits if such cycling is an issue.

Bits 5 and 4: PORLEV[1:0]. These bits set the Vcc level at which PPOR switches to one of three valid values. Do not use 11b because it is reserved.

See the “DC POR and LVD Specifications” table in the Electrical Specifications section of the PSoC device data sheet for voltage tolerances for each setting.

Bit 3: LVDTBEN. This bit is AND'ed with LVD to produce a throttle back signal that reduces CPU clock speed when low voltage conditions are detected. When the throttle back signal is asserted, the CPU speed bits in the OSC_CR0 register are reset, forcing the CPU speed to its reset state.

Bits 2 to 0: VM[2:0]. These bits set the Vdd level at which the LVD Comparator switches.

See the “DC POR and LVD Specifications” table in the Electrical Specifications section of the PSoC device data sheet for voltage tolerances for each setting.

For additional information, refer to the [VLT_CR register on page 279](#).

17.2.2 VLT_CMP Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E4h	VLT_CMP							LVD		RW : 0

The Voltage Monitor Comparators Register (VLT_CMP) reads the state of internal supply voltage monitors.

Bit 1: LVD. This bit reads the state of the LVD comparator. Zero Vdd is above the trip point. The trip points for LVD are set by VM[2:0] in the VLT_CR register.

For additional information, refer to the [VLT_CMP register on page 280](#).

18. SPI

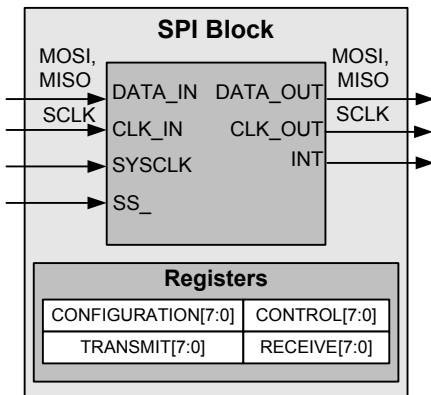


This chapter presents the Serial Peripheral Interconnect (SPI) and its associated registers. For a complete table of the SPI registers, refer to the [Summary Table of the System Resource Registers on page 106](#). For a quick reference of all PSoC registers in address order, refer to the [Register Reference chapter on page 187](#).

18.1 Architectural Description

The Serial Peripheral Interconnect (SPI) block is a dedicated master or slave SPI. The SPI slave function requires three inputs: Clock, Data, and SS_ (unless the SS_ is forced active with the SS_bit in the configuration register).

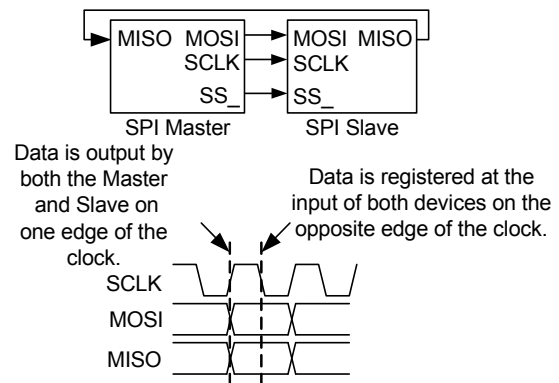
Figure 18-1. SPI Block Diagram



18.1.1 SPI Protocol Function

The 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 18-2](#) shows the basic signals in a simple connection.

Figure 18-2. 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. Together, the master and slave are essentially a circular Shift register, where the master generates the clocking and initiates data transfers.

A basic data transfer occurs when the master sends eight bits of data, along with eight clocks. In any transfer, both master and slave transmit and receive simultaneously. If the master only sends 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.

18.1.1.1 SPI Protocol Signal Definitions

The SPI protocol signal definitions are located in [Table 18-1](#). The use of the SS_ signal varies according to the capability of the slave device.

Table 18-1. SPI Protocol Signal Definitions

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 receives the input data and drives the MISO line.

18.1.2 SPI Master Function

The SPI Master (SPIM) offers SPI operating modes 0-3. By default, the most significant bit (MSb) of the data byte is shifted out first. An additional option can be set to reverse the direction and shift the data byte out the least significant bit (LSb) first. (Refer to the timing diagrams for this function on page [152](#).)

When configured for SPIM, DR0 functions as a Shift register with input from the DATA input (MISO) and output to the primary output F1 (MOSI). DR1 is the TX Buffer register and DR2 is the RX Buffer register.

The SPI protocol requires data to be registered at the device input, on the opposite edge of the clock that operates the output shifter. An additional register (RXD), at the input to the DR0 Shift register, has been implemented for this purpose. This register stores received data for one-half cycle before it is clocked into the Shift register.

The SPIM controls data transmission between master and slave because it generates the bit clock for internal clocking and for clocking the SPIS. The bit clock is derived from the CLK input selection.

There are four control bits and four status bits in the Control register (SPI_CR) that provide for device interfacing and synchronization.

The SPIM hardware has no support for driving the Slave Select (SS_) signal. The behavior and use of this signal is application and device dependent and, if required, must be implemented in firmware.

18.1.2.1 Usability Exceptions

The following are usability exceptions for the SPI Protocol function.

- The SPI_RXR (RX Buffer) register is not writeable.
- The SPI_TXR (TX Buffer) register is not readable.

18.1.2.2 Block Interrupt

The SPIM block has a selection of two interrupt sources: interrupt on TX Reg Empty (default) or interrupt on SPI Complete. Mode bit 1 in the Function register controls the selection. These modes are discussed in detail in [SPIM Timing on page 152](#).

If SPI Complete is selected as the block interrupt, the Control register must be read in the interrupt routine so that this status bit is cleared; otherwise, no subsequent interrupts are generated.

18.1.3 SPI Slave Function

The SPI Slave (SPIS) offers SPI operating modes 0-3. By default, the MSb of the data byte is shifted out first. An additional option can be set to reverse the direction and shift the data byte out LSb first. (Refer to the timing diagrams for this function on page [157](#).)

The SPI protocol requires data to be registered at the device input, on the opposite edge of the clock that operates the output shifter. An additional register (RXD), at the input to the DR0 Shift register, is implemented for this purpose. This register stores received data for one-half cycle before it is clocked into the Shift register.

The SPIS function derives all clocking from the SCLK input (typically an external SPI Master). This means that the master must initiate all transmissions. For example, to read a byte from the SPIS, the master must send a byte.

There are four control bits and four status bits in the Control register (SPI_CR) that provide for device interfacing and synchronization.

There is an additional data input in the SPIS, Slave Select (SS_), which is an active low signal. SS_ must be asserted to enable the SPIS to receive and transmit. SS_ has two high level functions: 1) To allow for the selection of a given slave in a multi-slave environment, and 2) To provide additional clocking for TX data queuing in SPI modes 0 and 1.

SS_ may be controlled from an external pin or can be controlled by way of user firmware.

When SS_ is negated, the SPIS ignores any MOSI/SCLK input from the master. In addition, the SPIS state machine is reset and the MISO output is forced to idle at logic 1. This allows for a wired-AND connection in a multi-slave environment. Note that if High Z output is required when the slave is not selected, this behavior must be implemented in firmware with I/O writes to the Port Drive register.

18.1.3.1 Usability Exceptions

The following are usability exceptions for the SPI Slave function.

- The SPI_RXR (RX Buffer) register is not writeable.
- The SPI_TXR (TX Buffer) register is not readable.

18.1.3.2 Block Interrupt

The SPIS block has a selection of two interrupt sources: Interrupt on TX Reg Empty (default) or interrupt on SPI Complete (same selection as the SPIM). Mode bit 1 in the Function register controls the selection.

If SPI Complete is selected as the block interrupt, the Control register must still be read in the interrupt routine so that this status bit is cleared; otherwise, no subsequent interrupts are generated.

18.1.4 Input Synchronization

All pin inputs are double synchronized to SYSCLK by default. Synchronization can be bypassed by setting the BYPS bit in the SPI_CFG register.

18.2 Register Definitions

The following registers are associated with the SPI and are listed in address order. The register descriptions have an associated register table showing the bit structure for that register. For a complete table of SPI registers, refer to the [Summary Table of the System Resource Registers on page 106](#).

Data Registers

18.2.1 SPI_TXR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,29h	SPI_TXR	Data[7:0]								W : 00

The SPI Transmit Data Register (SPI_TXR) is the SPI's transmit data register.

Bits 7 to 0: Data[7:0]. These bits encompass the SPI Transmit register. They are discussed by function type in [Table 18-2](#) and [Table 18-3](#).

For additional information, refer to the [SPI_TXR register on page 190](#).

18.2.2 SPI_RXR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,2Ah	SPI_RXR	Data[7:0]								R : 00

The SPI Receive Data Register (SPI_RXR) is the SPI's receive data register. A write to this register clears the RX Reg Full status bit in the Control register (SPI_CR).

Bits 7 to 0: Data[7:0]. These bits encompass the SPI Receive register. They are discussed by function type in [Table 18-2](#) and [Table 18-3](#).

For additional information, refer to the [SPI_RXR register on page 191](#).

18.2.2.1 SPI Master Data Register Definitions

There are two 8-bit Data registers and one 8-bit Control/Status register. [Table 18-2](#) explains the meaning of the Transmit and Receive registers in the context of SPIM operation.

Table 18-2. SPIM Data Register Descriptions

Name	Function	Description
SPI_TXR	TX Buffer	Write only register. If no transmission is in progress and this register is written to, the data from this register is loaded into the Shift register on the following clock edge, and a transmission is initiated. If a transmission is currently in progress, this register serves as a buffer for TX data. This register must only be written to when TX Reg Empty status is set and the write clears the TX Reg Empty status bit in the Control register. When the data is transferred from this register to the Shift register, then TX Reg Empty status is set.
SPI_RXR	RX Buffer	Read only register. When a byte transmission/reception is complete, the data in the shifter is transferred into the RX Buffer register and RX Reg Full status is set in the Control register. A read from this register clears the RX Reg Full status bit in the Control register.

18.2.2.2 SPI Slave Data Register Definitions

There are two 8-bit Data registers and one 8-bit Control/Status register. [Table 18-3](#) explains the meaning of the Transmit and Receive registers in the context of SPIS operation.

Table 18-3. SPIS Data Register Descriptions

Name	Function	Description
SPI_TXR	TX Buffer	Write only register. This register must only be written to when TX Reg Empty status is set and the write clears the TX Reg Empty status bit in the Control register. When the data is transferred from this register to the Shift register, then TX Reg Empty status is set.
SPI_RXR	RX Buffer	Read only register. When a byte transmission/reception is complete, the data in the shifter is transferred into the RX Buffer register and RX Reg Full status is set in the Control register. A read from this register clears the RX Reg Full status bit in the Control register.

Control Register

18.2.3 SPI_CR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,2Bh	SPI_CR	LSb First	Overrun	SPI Complete	TX Reg Empty	RX Reg Full	Clock Phase	Clock Polarity	Enable	# : 00

LEGEND

Access is bit specific. Refer to the register detail for additional information.

The SPI Control Register (SPI_CR) is the SPI's control register.

Bit 7: LSb First. This bit determines how the serial data is shifted out, either LSb or MSb first.

Bit 6: Overrun. This status bit indicates whether or not there was a receive buffer overrun. A read from the receive buffer after each received byte must be performed before the reception of the next byte in order to avoid an overrun condition.

Bit 5: SPI Complete. This status bit indicates the completion of a transaction. A read from this register clears this bit.

Bit 4: TX Reg Empty. This status bit indicates whether or not the Transmit register is empty.

Bit 3: RX Reg Full. This status bit indicates a Receive register full condition.

Bit 2: Clock Phase. This bit determines the edge (rising or falling) on which the data changes.

Bit 1: Clock Polarity. This bit determines the logic level the clock codes to in its idle state.

Bit 0: Enable. This bit enables the SPI block.

For additional information, refer to the [SPI_CR register on page 192](#).

18.2.3.1 SPI Control Register Definitions

Table 18-4. SPI Control Register Descriptions

Bit #	Name	Access	Description
7	LSb First	Read/Write	0 = Data shifted out MSb First. 1 = Data shifted out LSb First.
6	Overrun	Read Only	0 = No overrun. 1 = Indicates new byte received before previous one is read.
5	SPI Complete	Read Only	0 = Transaction in progress. 1 = Transaction is complete. Reading SPI_CR clears this bit.
4	TX Reg Empty	Read Only	0 = TX register is full. 1 = TX register is empty. Writing SPI_TXR register clears this bit.
3	RX Reg Full	Read Only	0 = RX register is not full. 1 = RX register is full. Reading SPI_RXR register clears this bit.
2	Clock Phase	Read/Write	0 = Data changes on trailing edge. 1 = Data changes on leading clock edge.
1	Clock Polarity	Read/Write	0 = Non-inverted, clock idles low (modes 0, 2). 1 = Inverted, clock idles high (modes 1, 3).
0	Enable	Read/Write	0 = Disable SPI function. 1 = Enable SPI function.

Configuration Register

The configuration block contains 1 register. This register must not be changed while the block is enabled. Note that the SPI Configuration register is located in bank 1 of the PSoC device's memory map.

18.2.4 SPI_CFG Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,29h	SPI_CFG	Clock Sel[2:0]			Bypass	SS_	SS_EN_	Int Sel	Slave	RW : 00

The SPI Configuration Register (SPI_CFG) is used to configure the SPI.

Bits 7 to 5: Clock Sel [2:0]. Clock Selection. These bits determine the operating frequency of the SPI Master.

Bit 4: Bypass. This bit determines whether or not the inputs are synchronized to SYSCLK.

Bit 3: SS_. Slave Select. This bit determines the logic value of the SS_ signal when the SS_EN_ signal is asserted (SS_EN_ = 0).

Bit 2: SS_EN_. Slave Select Enable. This active low bit determines if the slave select (SS_) signal is driven internally. If it is driven internally, its logic level is determined by the SS_ bit. If it is driven externally, its logic level is determined by the external pin.

Bit 1: Int Sel. Interrupt Select. This bit selects which condition produces an interrupt.

Bit 0: Slave. This bit determines whether the block functions as a master or slave.

For additional information, refer to the [SPI_CFG register on page 261](#).

18.2.4.1 SPI Configuration Register Definitions

Table 18-5. SPI Configuration Register Descriptions

Bit #	Name	Access	Mode	Description
7:5	Clock Sel	Read/Write	Master	SYSCLK 000b / 2 001b / 4 010b / 8 011b / 16 100b / 32 101b / 64 110b / 128 111b / 256
4	Bypass	Read/Write	Master/Slave	0 = All pin inputs are doubled, synchronized. 1 = Input synchronization is bypassed.
3	SS_	Read/Write	Slave	0 = Slave selected. 1 = Slave selection is determined from external SS_ pin.
2	SS_EN_	Read/Write	Slave	0 = Slave selection determined from SS_ bit. 1 = Slave selection determined from external SS_ pin.
1	Int Sel	Read/Write	Master/Slave	0 = Interrupt on TX Reg Empty. 1 = Interrupt on SPI Complete.
0	Slave	Read/Write	Master/Slave	0 = Operates as a master. 1 = Operates as a slave.

18.2.5 Related Registers

- [IO_CFG1 Register on page 272](#).

18.3 Timing Diagrams

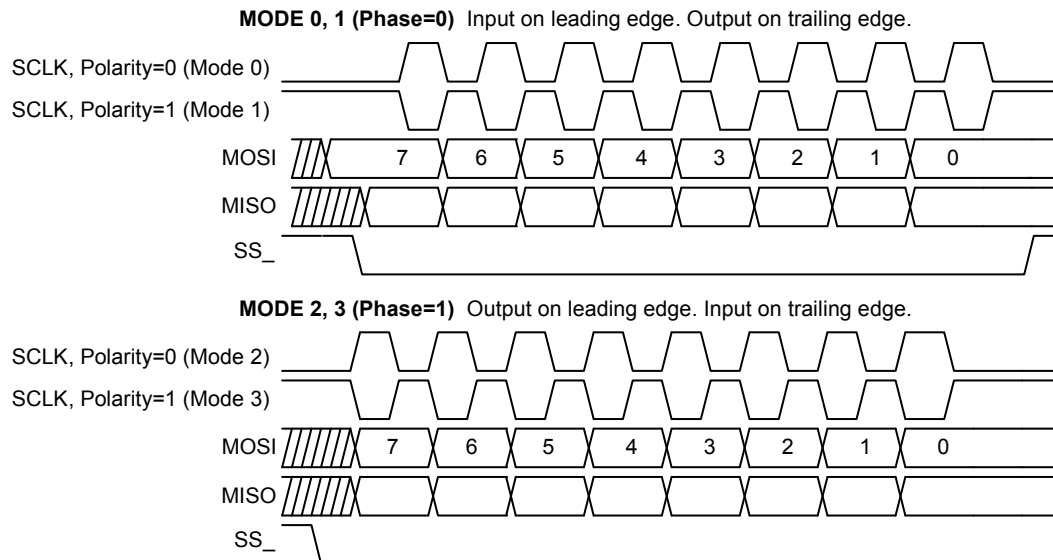
18.3.1 SPI Mode Timing

Figure 18-3 shows the SPI modes that are typically defined as 0, 1, 2, or 3. These mode numbers are an encoding of two control bits: Clock Phase and Clock Polarity.

Clock Phase indicates the relationship of the clock to the data. When the clock phase is '0', it means that the data is registered as an input on the leading edge of the clock and the next data is output on the trailing edge of the clock. When the clock phase is '1', it means that the next data is output on the leading edge of the clock and that data is registered as an input on the trailing edge of the clock.

Clock Polarity controls clock inversion. When clock polarity is set to '1', the clock idle state is high.

Figure 18-3. SPI Mode Timing



18.3.2 SPIM Timing

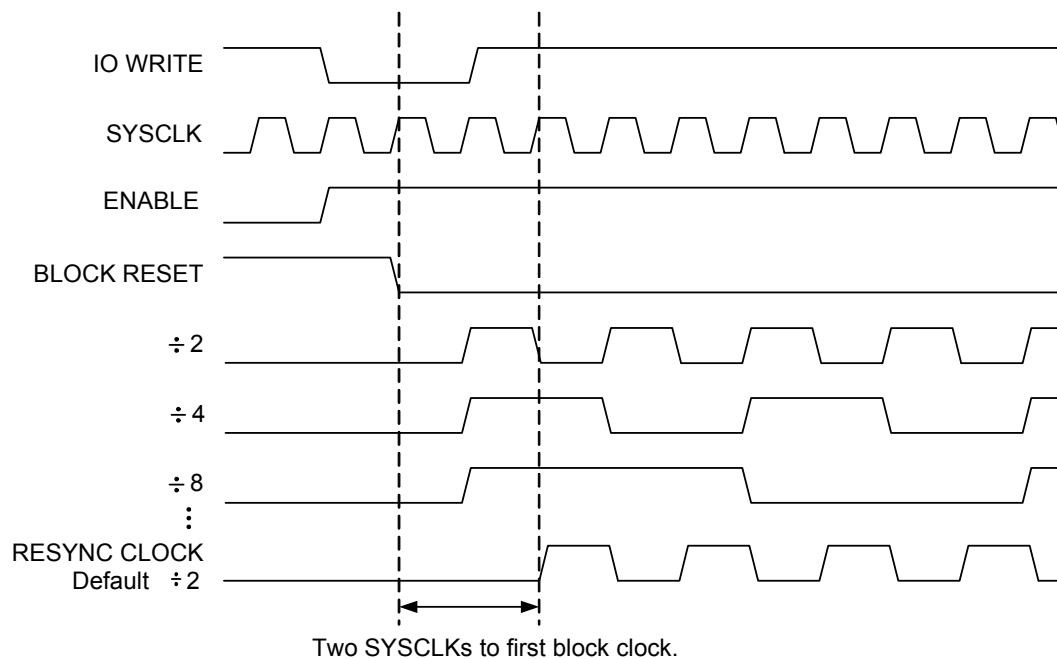
Enable/Disable Operation. As soon as the block is configured for SPIM, the primary output is the MSb or LSb of the Shift register, depending on the LSb First configuration in bit 7 of the Control register. The auxiliary output is '1' or '0', depending on the idle clock state of the SPI mode. This is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

Clock Generation. Figure 18-4 illustrates the SPIM input clocking scheme. The SYSCLK pin is an input into an eight-stage ripple divider that provides the baud rate selections. When the block is disabled, all internal state is held in a reset state.

When the Enable bit in the SPI_CR register is set, the reset is synchronously released and the clock generation is enabled. All eight taps from the ripple divider are selectable (/2, /4, /8, /16, /32, /64, /128, /256) from the Clock Sel bits in the SPI_CFG register. The selected divider tap is resynchronized to SYSCLK. The resulting clock is routed to all of the synchronous elements in the design.

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.

Figure 18-4. SPI Input Clocking



Normal Operation. Typical timing for an SPIM transfer is shown in Figure 18-5 and Figure 18-6. 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 is initiated. An 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.

Figure 18-5. Typical SPIM Timing in Mode 0 and 1

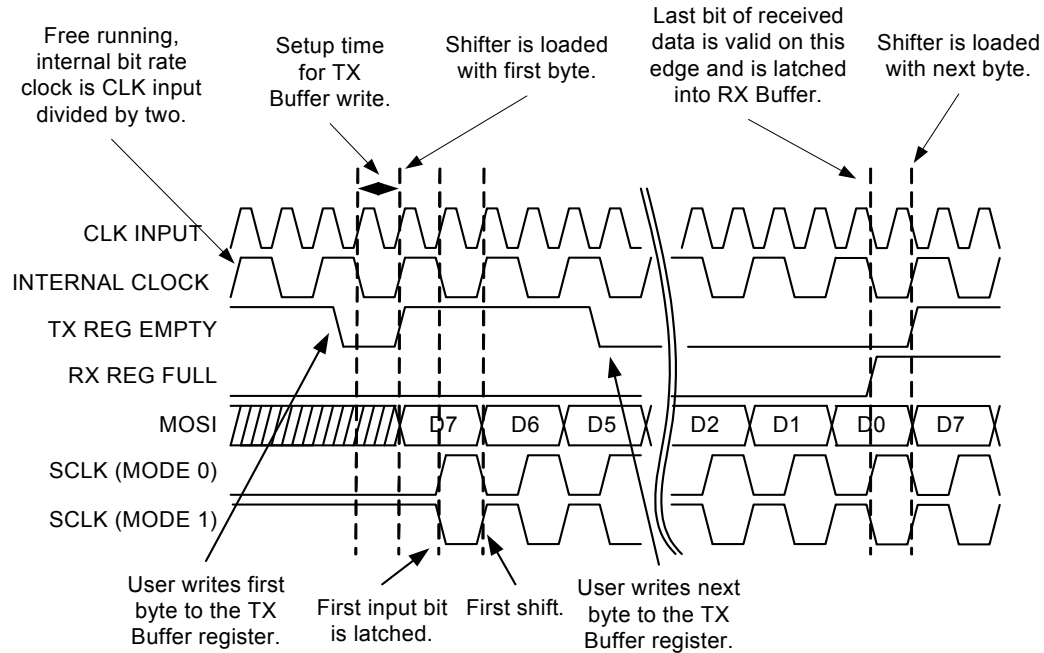
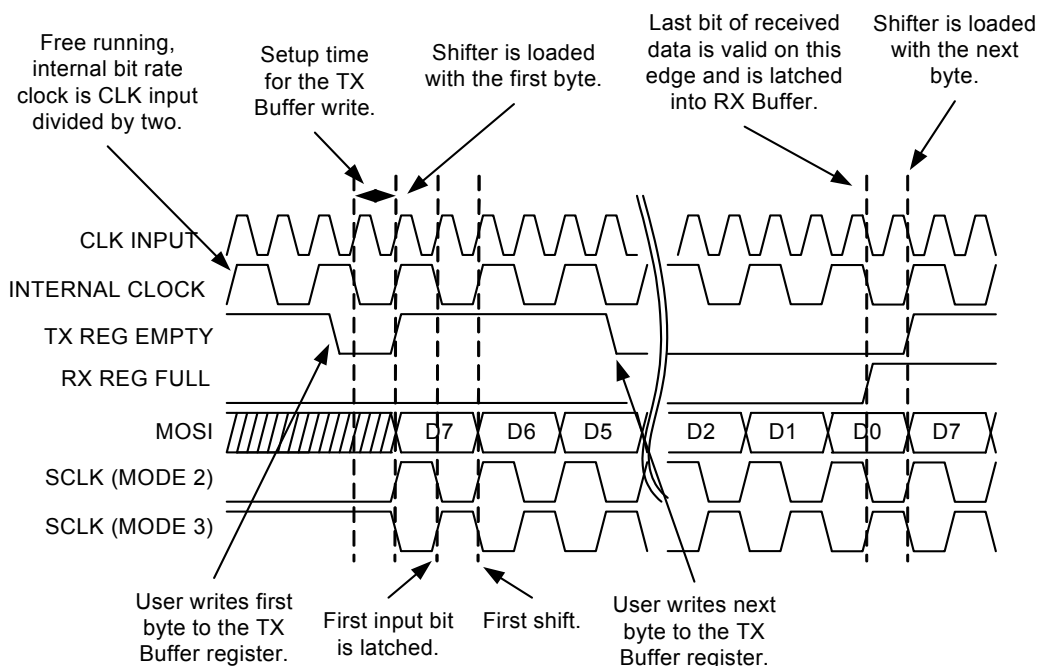


Figure 18-6. Typical SPIM Timing in Mode 2 and 3



Status Generation and Interrupts. There are four status bits in an SPI block: TX Reg Empty, RX Reg Full, SPI Complete, and Overrun.

TX Reg Empty indicates that a new byte can be written to the TX Buffer register. When the block is enabled, this status bit is immediately asserted. This status bit is cleared when the user writes a byte of data to the TX Buffer register. TX Reg Empty is a control input to the state machine and, if a transmission is not already in progress, the assertion of this control signal initiates one. This is the default SPIM block interrupt. However, an initial interrupt is not generated when the block is enabled. The user must write a byte to the TX Buffer register and that byte must be loaded into the shifter before interrupts generated from the TX Reg Empty status bit are enabled.

RX Reg Full is asserted on the edge that captures the eighth bit of receive data. This status bit is cleared when the user reads the RX Buffer register (DR2).

SPI Complete is an optional interrupt and is generated when eight bits of data and clock have been sent. In modes 0 and 1, this occurs one-half cycle after RX Reg Full is set; because in these modes, data is latched on the leading edge of the clock and there is an additional one-half cycle remaining to complete that clock. In modes 2 and 3, this occurs at the same edge that the receive data is latched. This signal may be used to read the received byte or it may be used by the SPIM to disable the block after data transmission is complete.

Overrun status is set if RX Reg Full is still asserted from a previous byte when a new byte is about to be loaded into the RX Buffer register. Because the RX Buffer register is imple-

mented as a latch, Overrun status is set one-half bit clock before RX Reg Full status.

See [Figure 18-7](#) and [Figure 18-8](#) for status timing relationships.

Figure 18-7. SPI Status Timing for Modes 0 and 1

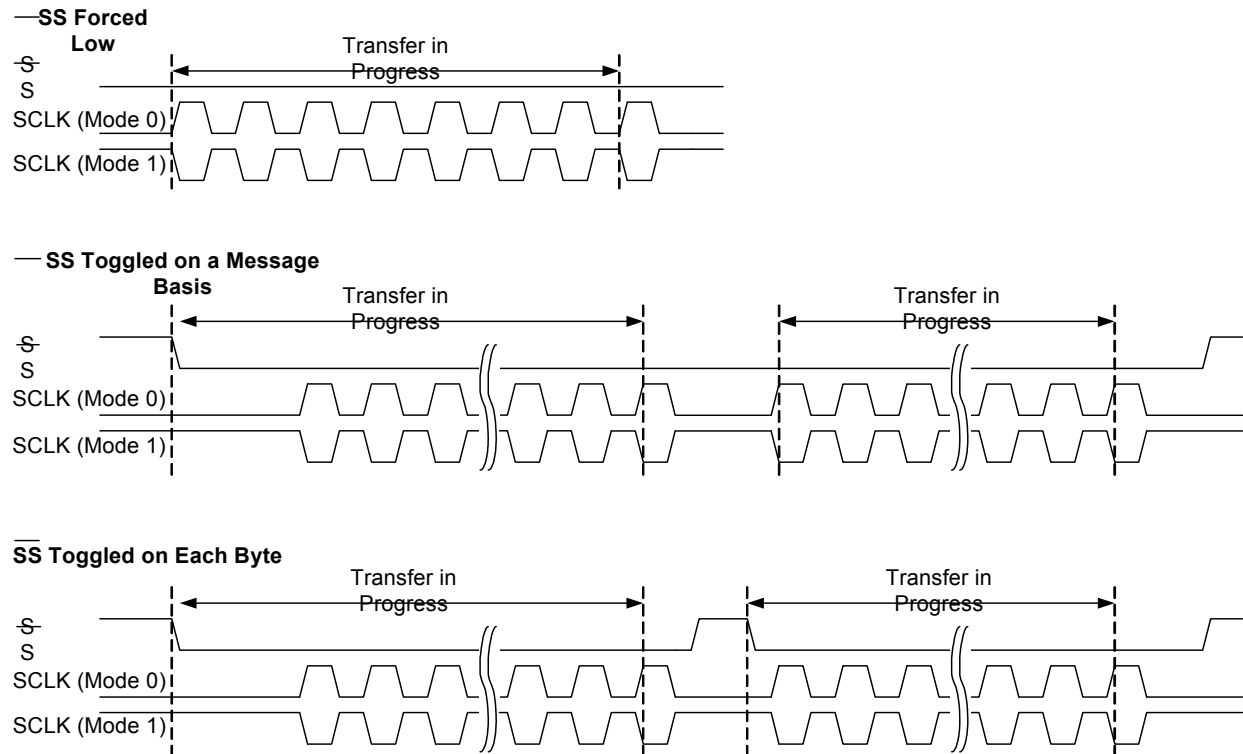
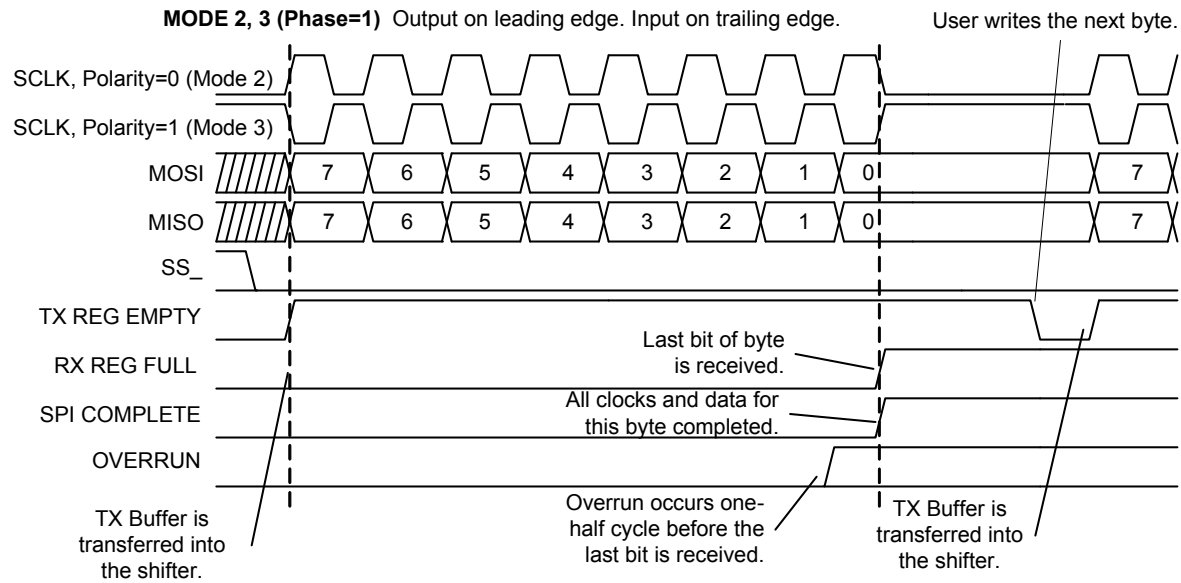


Figure 18-8. SPI Status Timing for Modes 2 and 3



18.3.3 SPIS Timing

Enable/Disable Operation. As soon as the block is configured for SPI Slave and before enabling, the MISO output is set to idle at logic 1. The Enable bit must be set and the SS_ asserted (either driven externally or forced by firmware programming) for the block to output data. When enabled, the primary output is the MSb or LSb of the Shift register, depending on the LSb First configuration in bit 7 of the Control register. The auxiliary output of the SPIS is always forced into tristate.

Since the SPIS has no internal clock, it must be enabled with setup time to any external master supplying the clock. Setup time is also required for a TX Buffer register write before the first edge of the clock or the first falling edge of SS_, depending on the mode. This setup time must be assured through the protocol and an understanding of the timing between the master and slave in a system.

When the block is disabled, the MISO output reverts to its idle 1 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 an SPIS transfer is shown in Figure 18-9 and Figure 18-10. If the SPIS is primarily being used as a receiver, the RX Reg Full (polling only) or SPI Complete (polling or interrupt) status may be used to determine when a byte has been received. In this way, the SPIS operates identically with the SPIM. However, there are two main areas in which the SPIS operates differently: 1) SPIS behavior related to the SS_ signal, and 2) TX data queuing (loading the TX Buffer register).

Figure 18-9. Typical SPIS Timing in Modes 0 and 1

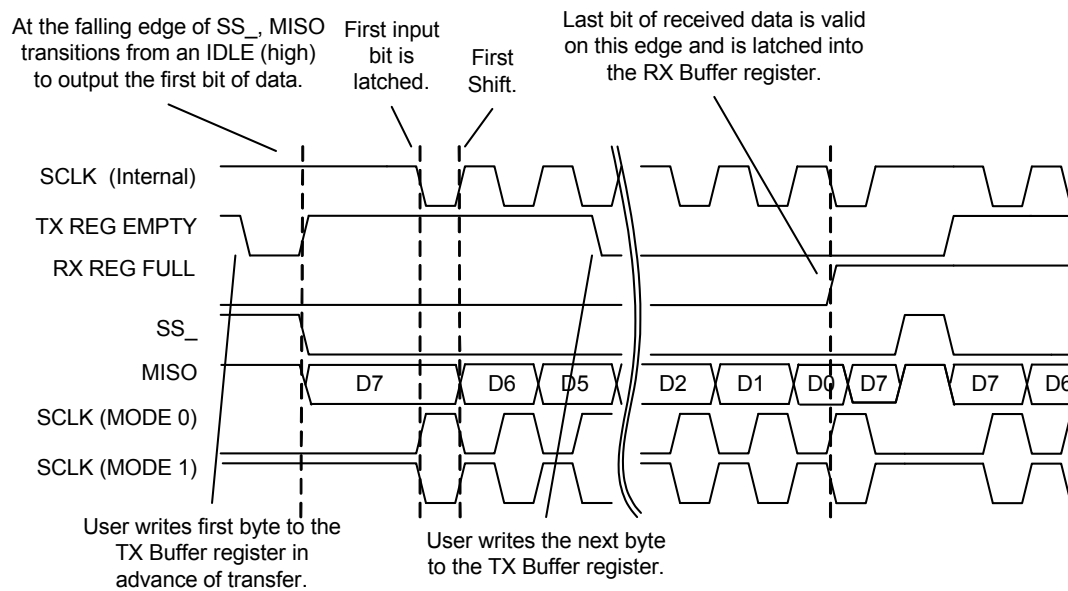
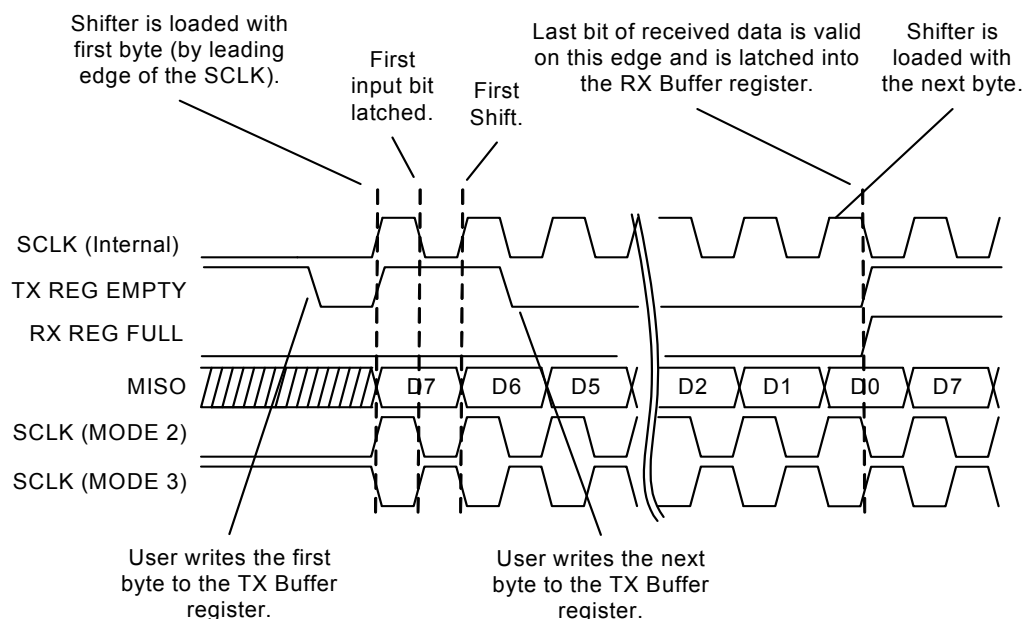


Figure 18-10. Typical SPIS Timing in Modes 2 and 3



Slave Select (SS_, Active Low). Slave Select must be asserted to enable the SPIS for receive and transmit. There are two ways to do this:

- Drive the auxiliary input from a pin (selected by the Aux I/O Select bits in the output register). This gives the SPI master control of the slave selection in a multi-slave environment.
- SS_ may be controlled in firmware with register writes to the output register. When Aux I/O Enable = 1, Aux I/O Select bit 0 becomes the SS_ input. This allows the user to save an input pin in single-slave environments.

When SS_ is negated (whether from an external or internal source), the SPIS state machine is reset and the MISO output is forced to idle at logic 1. In addition, the SPIS ignores any incoming MOSI/SCLK input from the master.

Status Generation and Interrupts. There are four status bits in the SPIS block: TX Reg Empty, RX Reg Full, SPI Complete, and Overrun. The timing of these status bits are identical to the SPIM, with the exception of TX Reg Empty, which is covered in the section on TX data queuing.

Status Clear On Read. Refer to the same subsection in [SPIM Timing on page 152](#).

TX Data Queuing. Most SPI applications call for data to be sent back from the slave to the master. Writing firmware to accomplish this requires an understanding of how the Shift register is loaded from the TX Buffer register.

All modes use the following mechanism: 1) If there is no transfer in progress, 2) if the shifter is empty, and 3) if data is available in the TX Buffer register, the byte is loaded into the shifter.

The only difference between the modes is that the definition of “transfer in progress” is slightly different between modes 0 and 1, and modes 2 and 3.

Figure 18-11 illustrates TX data loading in modes 0 and 1. A transfer in progress is defined to be from the falling edge of $\overline{SS}_-$ to the point at which the RX Buffer register is loaded with the received byte. This means that to send a byte in the next transfer, it must be loaded into the TX Buffer register before the falling edge of $\overline{SS}_-$. This ensures a minimum setup time for the first bit, since the leading edge of the first SCLK must latch in the received data. If $\overline{SS}_-$ is not toggled between each byte or is forced low through the configuration register, the leading edge of SCLK is used to define the start of transfer. However, in this case, the user must provide the required setup time (one-half clock minimum before the leading edge) with a knowledge of system latencies and response times.

Figure 18-11. Mode 0 and 1 Transfer in Progress

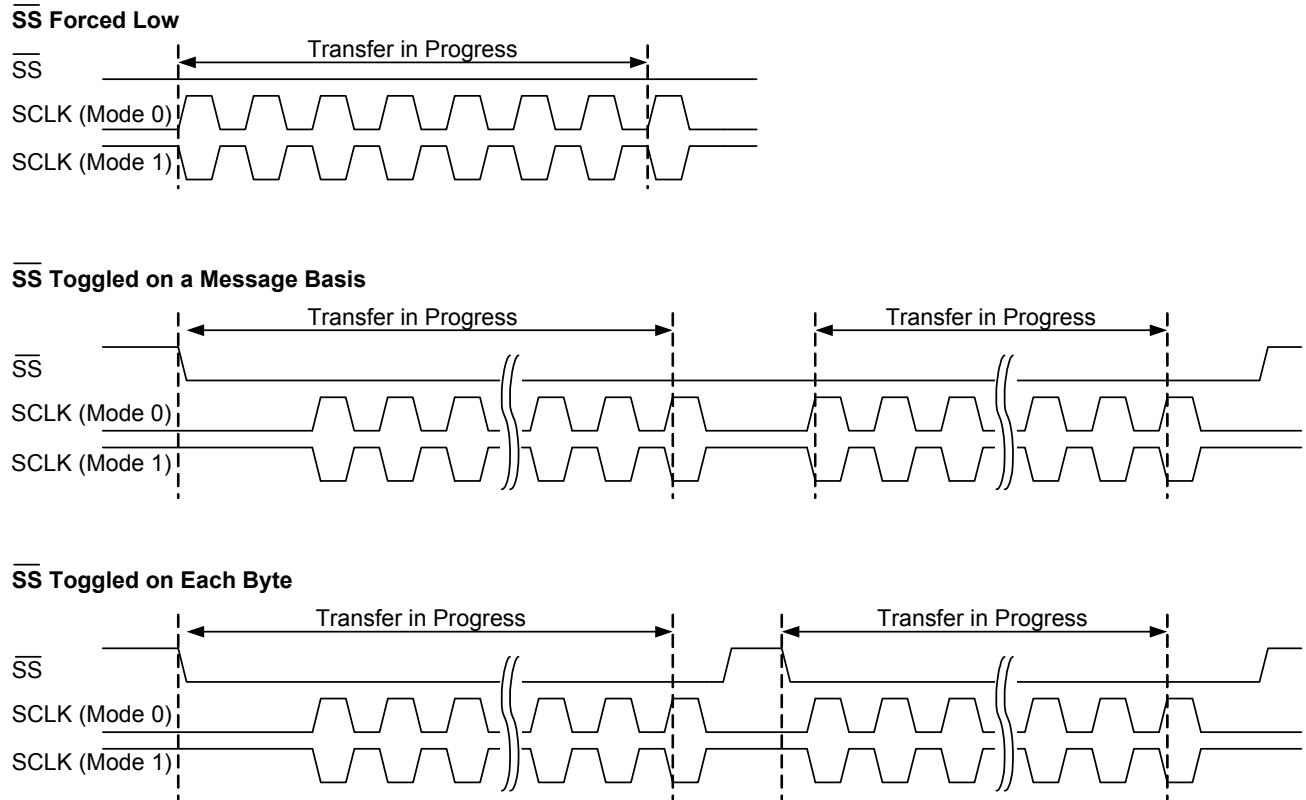
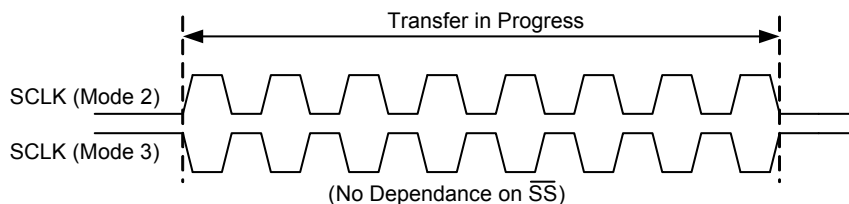


Figure 18-12 illustrates TX data loading in modes 2 and 3. In this case, a transfer in progress is defined to be from the leading edge of the first SCLK to the point at which the RX Buffer register is loaded with the received byte. Loading the shifter by the leading edge of the clock has the effect of providing the required one-half clock setup time, as the data is latched into the receiver on the trailing edge of the SCLK in these modes.

Figure 18-12. Mode 2 and 3 Transfer in Progress



SPI

19. Programmable Timer

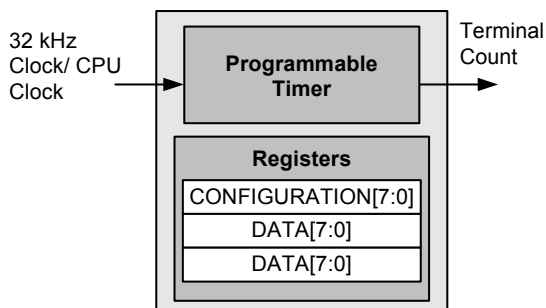


This chapter presents the Programmable Timer and its associated registers. For a complete table of the programmable timer registers, refer to the [Summary Table of the System Resource Registers on page 106](#). For a quick reference of all PSoC registers in address order, refer to the [Register Reference chapter on page 187](#).

19.1 Architectural Description

The device has three programmable timers (TIMER0, TIMER1, TIMER2). All three timers are individually controlled. The programmable timers are 16-bit down counters for the device. TIMER0 has a terminal count output. The timers have one configuration and two data registers associated with them. You start these timers by setting the START bit in their configuration registers (PT0_CFG, PT1_CFG, PT2_CFG). When started, the timers always start counting down from the value loaded into their data registers (PT*_DATA1, PT*_DATA0). The timers have a one-shot mode, in which the timers complete one full count cycle and stop. In one-shot mode the START bit in the configuration register is cleared after completion of one full count cycle. Setting the START bit restarts the timer.

Figure 19-1. Programmable Timer Block Diagram



19.1.1 Operation

When started, the programmable timer loads the value contained in its data registers and counts down to its terminal count of zero. The timers output an active high terminal count pulse for one clock cycle upon reaching the terminal count. The low time of the terminal count pulse is equal to the loaded decimal count value, multiplied by the clock period ($TC_{pw} = COUNT\ VALUE_{decimal} * CLK_{period}$). The period of the terminal count output is the pulse width of the terminal count, plus one clock period ($TC_{period} = TC_{pw} + CLK_{period}$). Refer to [Figure 19-2](#) and [Figure 19-3](#).

Only TIMER0 outputs this terminal count output. TIMER1 and TIMER2 do not have a terminal count output.

The timers work on either the 32 kHz clock or CPU clock. This clock selection is done using the CLKSEL bits in the respective configuration registers (PT0_CFG, PT1_CFG, PT2_CFG). Make clock selections before setting the START bit so that the timing is not affected and clock frequency does not change while the timer is running.

TIMER1 works on prescaled IMO clock (IMO-P) when the CSD_MODE bit in the CS_CR0 (0,A0) register is set to '1'. TIMER1 outputs the START signal, which is used in the TrueTouch module during CSD mode. Refer to [11.1.1.5 Sigma Delta on page 89](#) for more details on TrueTouch CSD mode. When CSD_MODE is set to '0', it works on either the 32 kHz clock or CPU clock, depending on the CLKSEL bit setting.

Figure 19-2. Continuous Operation Example

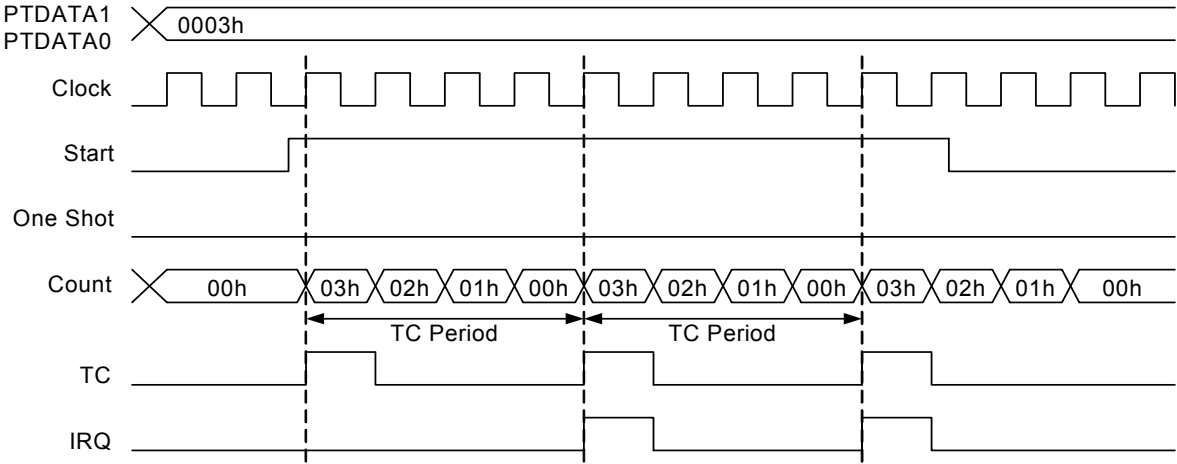
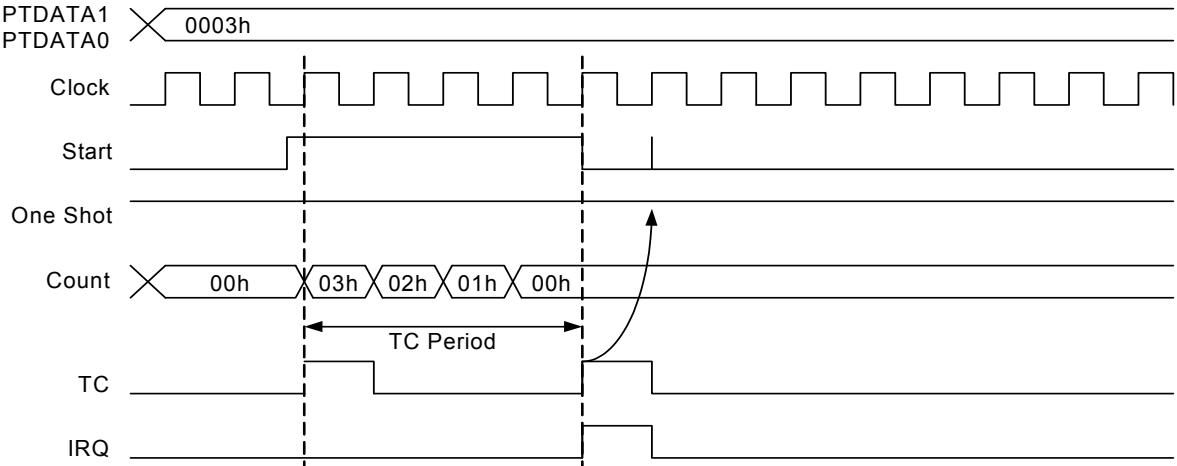


Figure 19-3. One-Shot Operation Example



19.2 Register Definitions

The following registers are associated with the Programmable Timer and are listed in address order. The register descriptions have an associated register table showing the bit structure for that register. The bits in the tables that are grayed out are reserved bits and are not detailed in the register descriptions that follow. Reserved bits must always be written with a value of '0'. For a complete table of programmable timer registers, refer to the [Summary Table of the System Resource Registers on page 106](#).

19.2.1 PT0_CFG Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,B0h	PT0_CFG						CLKSEL	One Shot	START	RW : 0

The Programmable Timer Configuration Register (PT0_CFG) configures the PSoC's programmable timer.

Bit 2: CLKSEL. This bit determines if the timer runs on the 32 kHz clock or CPU clock. If the bit is set to 1'b1, the timer runs on the CPU clock, otherwise, the timer runs on the 32 kHz clock.

Bit 1: One Shot. This bit determines if the timer runs in one-shot mode or continuous mode. In one-shot mode the timer completes one full count cycle and terminates. Upon termination, the START bit in this register is cleared. In continuous mode, the timer reloads the count value each time upon completion of its count cycle and repeats.

Bit 0: START. This bit starts the timer counting from a full count. The full count is determined by the value loaded into the data registers. This bit is cleared when the timer is running in one-shot mode upon completion of a full count cycle.

For additional information, refer to the [PT0_CFG register on page 221](#).

19.2.2 PT1_CFG Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,B3h	PT1_CFG						CLKSEL	One Shot	START	RW : 0

The Programmable Timer Configuration Register (PT1_CFG) configures the PSoC's programmable timer.

Bit 2: CLKSEL. This bit determines if the timer runs on the 32 kHz clock or CPU clock. If the bit is set to 1'b1, the timer runs on the CPU clock, otherwise, the timer runs on the 32 kHz clock.

Bit 1: One Shot. This bit determines if the timer runs in one-shot mode or continuous mode. In one-shot mode the timer completes one full count cycle and terminates. Upon termination, the START bit in this register is cleared. In continuous mode, the timer reloads the count value each time upon completion of its count cycle and repeats.

Bit 0: START. This bit starts the timer counting from a full count. The full count is determined by the value loaded into the data registers. This bit is cleared when the timer is running in one-shot mode upon completion of a full count cycle.

For additional information, refer to the [PT1_CFG register on page 224](#).

19.2.3 PT2_CFG Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,B6h	PT2_CFG						CLKSEL	One Shot	START	RW : 0

The Programmable Timer Configuration Register (PT2_CFG) configures the PSoC's programmable timer.

Bit 2: CLKSEL. This bit determines if the timer runs on the 32 kHz clock or CPU clock. If the bit is set to 1'b1, the timer runs on the CPU clock, otherwise, the timer runs on the 32 kHz clock.

Bit 1: One Shot. This bit determines if the timer runs in one-shot mode or continuous mode. In one-shot mode the timer completes one full count cycle and terminates. Upon termination, the START bit in this register is cleared. In con-

tinuous mode, the timer reloads the count value each time upon completion of its count cycle and repeats.

Bit 0: START. This bit starts the timer counting from a full count. The full count is determined by the value loaded into the data registers. This bit is cleared when the timer is running in one-shot mode upon completion of a full count cycle.

For additional information, refer to the [PT2_CFG register on page 225](#).

19.2.4 PTx_DATA0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,B2h	PT0_DATA0									RW : 00
0,B5h	PT1_DATA0									RW : 00
0,B8h	PT2_DATA0									RW : 00

The Programmable Timer Data Register 0 (PT0_DATA0, PT1_DATA0, PT2_DATA0) holds the lower 8 bits of the programmable timer's count value.

For additional information, refer to the [PT0_DATA0 register on page 223](#), [PT1_DATA0 register on page 223](#) and [PT2_DATA0 register on page 223](#).

Bits 7 to 0: DATA[7:0]. This is the lower byte of a 16-bit timer. The upper byte is in the corresponding PTxDATA1 register.

19.2.5 PTx_DATA1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,B1h	PT0_DATA1									RW : 00
0,B4h	PT1_DATA1									RW : 00
0,B7h	PT2_DATA1									RW : 00

The Programmable Timer Data Register 1 (PT0_DATA1, PT1_DATA1, PT2_DATA1) holds the 8 bits of the programmable timer's count value for the device

For additional information, refer to the [PT0_DATA1 register on page 222](#), [PT1_DATA1 register on page 222](#), and [PT2_DATA1 register on page 222](#).

Bits 7 to 0: DATA[7:0]. This is the upper byte of a 16-bit timer. The lower byte is in the corresponding PTx_DATA0 register.

20. Full-Speed USB



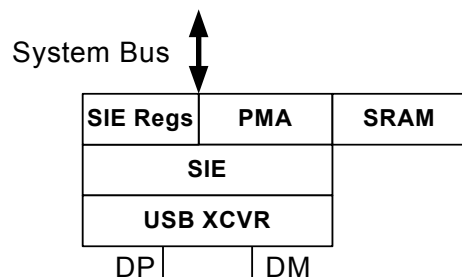
This chapter explains the Full-Speed USB (Universal Serial Bus) resource and its associated registers. For a quick reference of all PSoC registers in address order, refer to the [Register Reference chapter on page 187](#).

20.1 Architectural Description

The PSoC USB system resource adheres to the USB 2.0 Specification for full-speed devices operating at 12 Mbps with one upstream port and one USB address. PSoC USB consists of these components:

- Serial Interface Engine (SIE) block
- PSoC Memory Arbiter (PMA) block
- 512 bytes of dedicated SRAM
- A Full-Speed USB Transceiver with internal regulator and two dedicated USB pins

Figure 20-1. USB Block Diagram



At the PSoC system level, the full-speed USB system resource interfaces to the rest of the PSoC by way of the M8C's register access instructions and to the outside world by way of the two USB pins. The SIE supports nine endpoints including a bidirectional control endpoint (endpoint 0) and eight uni-directional data endpoints (endpoints 1 to 8). The uni-directional data endpoints are individually configurable as either IN or OUT.

20.2 Application Description

The individual components and issues of the USB system are described in detail in the following sections.

20.2.1 USB SIE

The USB Serial Interface Engine (SIE) allows the PSoC device to communicate with the USB host at full-speed data rates (12 Mbps). The SIE simplifies the interface to USB traffic by automatically handling the following USB processing tasks without firmware intervention:

- Translates the encoded received data and formats the data to be transmitted on the bus.
- Generates and checks CRCs. Incoming packets failing checksum verification are ignored.
- Checks addresses. Ignores all transactions not addressed to the device.
- Sends appropriate ACK/NAK/Stall handshakes.
- Identifies token type (SETUP, IN, OUT) and sets the appropriate token bit once a valid token is received.
- Identifies Start-of-Frame (SOF) and saves the frame count.
- Sends data to or retrieves data from the USB SRAM, by way of the PSoC Memory Arbiter (PMA).

Firmware is required to handle various parts of the USB interface. The SIE issues interrupts after key USB events to direct firmware to appropriate tasks:

- Fill and empty the USB data buffers in USB SRAM.
- Enable PMA channels appropriately.
- Coordinate enumeration by decoding USB device requests.
- Suspend and resume coordination.
- Verify and select data toggle values.

Table 20-1. Mode Encoding for Control and Non-Control Endpoints

Mode	Encoding	SETUP	IN	OUT	Comments
Disable	0000	Ignore	Ignore	Ignore	Ignore all USB traffic to this endpoint.
NAK IN/OUT	0001	Accept	NAK	NAK	NAK IN and OUT token.
Status OUT Only	0010	Accept	STALL	Check	For control endpoint, STALL IN and ACK zero byte OUT.
STALL IN/OUT	0011	Accept	STALL	STALL	For control endpoint, STALL IN and OUT token.
Reserved	0100	Ignore	Ignore	Ignore	
ISO OUT	0101	Ignore	Ignore	Always	Isochronous OUT.
Status IN Only	0110	Accept	TX 0 Byte	STALL	For control endpoint, STALL OUT and send zero byte data for IN token.
ISO IN	0111	Ignore	TX Count	Ignore	Isochronous IN.
NAK OUT	1000	Ignore	Ignore	NAK	Send NAK handshake to OUT token.
ACK OUT (STALL = 0)	1001	Ignore	Ignore	ACK	This mode is changed by the SIE to mode 1000 on issuance of ACK handshake to an OUT.
ACK OUT (STALL = 1)	1001	Ignore	Ignore	STALL	STALL the OUT transfer.
Reserved	1010	Ignore	Ignore	Ignore	
ACK OUT – STATUS IN	1011	Accept	TX0 Byte	ACK	ACK the OUT token or send zero byte data for IN token.
NAK IN	1100	Ignore	NAK	Ignore	Send NAK handshake for IN token.
ACK IN (STALL = 0)	1101	Ignore	TX Count	Ignore	This mode is changed by the SIE to mode 1100 after receiving ACK handshake to an IN data.
ACK IN (STALL = 1)	1101	Ignore	STALL	Ignore	STALL the IN transfer.
Reserved	1110	Ignore	Ignore	Ignore	
ACK IN – Status OUT	1111	Accept	TX Count	Check	Respond to IN data or Status OUT.

20.2.2 USB SRAM

The PSoC USB System Resource contains dedicated 512 bytes SRAM. This SRAM is identical to 2 SRAM pages used in the PSoC Core; however, it is not accessible by way of the M8C memory access instructions. The USB's dedicated SRAM may only be accessed by way of the PMA registers. For more information on how to use the USB's dedicated SRAM, see the next section, PSoC Memory Arbiter (PMA).

The USB SRAM contents are not directly affected by any reset, but must be treated as unknown after any POR, WDR, and XRES.

20.2.2.1 PSoC Memory Arbiter

The PSoC Memory Arbiter (PMA) is the interface between the USB's dedicated SRAM and the two blocks that access the SRAM: the M8C and the USB SIE. The PMA provides 16 channels to manage data with four Endpoints/8 channels mapped to Page 0 of USB dedicated SRAM and other four Endpoints/8 channels mapped to Page 1 of USB dedicated SRAM. All of the channel registers may be used by the M8C, but the eight non-control USB endpoints are each allocated to a specific set of PMA channel registers. It is the responsibility of the firmware to ensure that the M8C is not accessing a set of channel registers that are in use by the USB SIE.

If the M8C wants to access the same data that an SIE channel is using, two channels must be configured to access the same SRAM address ranges. [Table 20-2](#) shows the mapping between PMA channels and which blocks can use them.

Table 20-2. PMA Channel Assignments

PMA Channel	USB SIE	M8C	Channel Registers (PMAx_xx)
0		Page 0	PMA0_DR, PMA0_RA, PMA0_WA
1	EP1	Page 0	PMA1_DR, PMA1_RA, PMA1_WA
2	EP2	Page 0	PMA2_DR, PMA2_RA, PMA2_WA
3	EP3	Page 0	PMA3_DR, PMA3_RA, PMA3_WA
4	EP4	Page 0	PMA4_DR, PMA4_RA, PMA4_WA
5		Page 0	PMA5_DR, PMA5_RA, PMA5_WA
6		Page 0	PMA6_DR, PMA6_RA, PMA6_WA
7		Page 0	PMA7_DR, PMA7_RA, PMA7_WA
8		Page 1	PMA8_DR, PMA8_RA, PMA8_WA
9	EP5	Page 1	PMA9_DR, PMA9_RA, PMA9_WA
10	EP6	Page 1	PMA10_DR, PMA10_RA, PMA10_WA
11	EP7	Page 1	PMA11_DR, PMA11_RA, PMA11_WA
12	EP8	Page 1	PMA12_DR, PMA12_RA, PMA12_WA
13		Page 1	PMA13_DR, PMA13_RA, PMA13_WA
14		Page 1	PMA14_DR, PMA14_RA, PMA14_WA
15		Page 1	PMA15_DR, PMA15_RA, PMA15_WA

The PMA's purpose is to manage the potentially conflicting SRAM access requests from the M8C and the USB SIE. From a performance standpoint, the PMA guarantees that a continuous stream of move instructions (see ahead), are serviced by the PMA without delay even while the USB SIE is transferring data at its maximum rate in to or out of the dedicated USB SRAM.

When servicing a request, the PMA is in one of two addressing modes. For M8C access the PMA always uses Post-Increment Addressing. After a read or write request is made to the channel's PMAx_DR register, the PMA automatically increments the pointer into SRAM. For a read access the next value is also automatically prefetched. For USB SIE accesses, the PMA uses an offset addressing mode. In this mode the channel's base address, as stored in the PMAx_WA and PMAx_RA registers, is added to the byte count value provided by the USB SIE.

A PMA channel does not have a defined upper limit. It is the responsibility of the firmware to ensure that channels do not access memory outside of the range defined by the application.

During SIE writes to the USB SRAM, the maximum number of bytes written is limited to the count value in the respective endpoint's count registers. This value must be loaded by firmware before data is received.

The rest of the description of the PMA is broken into two parts: the M8C interface and the USB SIE interface, which are described as follows.

PMA to M8C Interface

The M8C accesses the PMA, and thus the USB's dedicated SRAM, by way of a register interface. Each PMA channel has three registers associated with it as shown in [Table 20-2 on page 166](#). Only the following basic M8C register access instructions may be used with these registers.

```
MOV A, reg[expr]
MOV A, reg[X+expr]
MOV [expr], [expr]
MOV reg[expr], A
MOV reg[X+expr], A
MOV reg[expr], expr
MOV reg[X+expr], expr
```

When the M8C uses a PMA channel to write data into SRAM, follow these steps:

1. Choose a PMA channel that is not allocated to a USB endpoint, or choose a channel where the endpoint is inactive.
2. Write the channel's PMAx_WA register with the first address in SRAM that must be used by this channel.
3. Write data to the channel's PMAx_DR register. The PMA logic automatically increments the PMAx_WA address after each write.

While these steps are executed by the M8C, the USB SIE may be fully active on any other PMA channel.

The M8C may also service another channel and come back to the channel being serviced by the previous steps. To determine the next address that is used when data is written to the channel's PMAx_DR register, the PMAx_WA register may be read.

When the M8C uses a PMA channel to read data from SRAM, follow these steps:

1. Choose a PMA channel that is not allocated to a USB endpoint, or choose a channel where the endpoint is inactive.
2. Write the channel's PMAx_RA register with the first address in SRAM that must be read by this channel.
3. Read data from the channel's PMAx_DR register. The PMA logic automatically increments the PMAx_RA address after each read.

When data is read from a PMA channel the data is prefetched; therefore, the channel must be pre-loaded prior to the first M8C read that expects to get actual data. This pre-loading is taken care of automatically when the PMAx_RA register is written. This pre-loading mechanism is actually the only difference between the PMAx_RA and PMAx_WA registers.

PMA to USB SIE Interface

The USB SIE accesses the PMA, and thus the dedicated USB SRAM, by way of a private interface and does not affect the PSoC Core address or data bus. The only area of contention that is not automatically arbitrated between the M8C, PMA, and USB SIE are the PMAx_xx registers. When the USB SIE is actively using a PMA channel, the M8C must not attempt to access that channel's PMA registers. If the M8C wants to access the same data as an active USB endpoint, the M8C must use a PMA channel separate from the PMA channel that is permanently allocated to that endpoint.

Just as the M8C has two uses for PMA channels, read or write, the USB SIE has two uses for a PMA channel. The USB SIEs use of a channel may be thought of as read or write; but, in USB terms the USB SIEs need to read data would be associated with an IN transaction and the need to write data with an OUT transaction.

For a USB IN transaction, the USB SIE is reading data from the PMA and sending the data to the USB host. The following steps must be used to set up a PMA channel for a USB IN transaction. These steps assume that the data has already been written by the M8C into the dedicated USB SRAM.

1. Select the PMA channel whose number matches the endpoint number that is handling the IN transaction.
2. Write the PMA channel's PMAx_RA register with the address of the first byte in SRAM that is used for the IN transaction.
3. Configure the USB endpoint registers with the proper byte count and enable the endpoint to send data when the IN transaction occurs.

Because the PMA prefetches data for M8C and USB SIE reads, step two above is very important. This step not only sets the first address from which data is read by the USB SIE; but, it also triggers a read operation on the dedicated USB SRAM and stores the result of that read in the PMAx_DR ready for the USB SIE to read. When the USB SIE begins the IN transaction for the endpoint, it uses its byte counter to tell the PMA which byte is needed next. Therefore, when the first byte of the transaction is read by the SIE, the PMA automatically fetches the next byte in preparation for the USB SIEs next byte request.

For a USB OUT transaction, the USB SIE is writing data to the PMA that was received from the USB host. The following steps must be used to set up a PMA channel for a USB OUT transaction.

1. Select the PMA channel whose number matches the endpoint number that is handling the OUT transaction.
2. Write the PMA channel's PMAx_WA register with the address of the first byte in SRAM that is used for the OUT transaction.
3. Configure the USB endpoint with the proper maximum receive byte count and enable the endpoint to receive the OUT transaction.

As with the IN transaction, the PMA uses the byte counter from the SIE as an offset to the value of the PMAx_WA register. As the USB SIE sends bytes to the PMA, the counter is added to the base address and the data byte is written into the dedicated USB SRAM. Should an error occur in the OUT transaction and the packet be resent by the USB host, the byte count is reset to zero and the PMA writes the new data over top of the potentially corrupt data from the previous failed transaction.

If the number of bytes received exceeds the count in the endpoint's count register, the extra bytes are not written into the USB SRAM, but the received byte count reported in the endpoint count registers includes the ignored bytes.

20.2.3 Oscillator Lock

The PSoC device can operate without using any external components, such as a crystal, and still achieve the clock accuracy required for full-speed USB. It does this by locking its internal oscillator to the incoming USB traffic. Therefore, the initial accuracy of the oscillator may not meet the required accuracy ($\pm 0.25\%$), but it self-tunes to this precision before the device needs to transmit USB data.

This oscillator locking feature is disabled by default and must be enabled by firmware. In USB systems, this feature must always be enabled unless the device is being used with an accurate external clock. The EnableLock bit in the USB_CR1 register is used to turn on the locking feature.

20.2.4 Transceiver

The internal USB transceiver interfaces to the external USB bus to transmit and receive signals according to the USB 2.0 Specification. In normal USB operation, the transceiver interfaces directly to the SIE and no user interaction is needed after initialization. The USB Enable bit in the USB_CR0 register must be set to enable the transceiver for USB operation. The I/O Mode bit in USBIO_CR1 must not be set during normal USB mode of operation. The transceiver can also be used in non-USB modes, since the D+ and D- pins can be read and written through register control bits. This enables multi-purpose use of these pins (for example, in a system that supports both USB and PS/2 signaling). Clearing the USB Enable bit of USB_CR0 powers down the USB differential receiver and disables USB communication.

For USB operation, the transceiver contains an internal 1.5 k Ω pull up resistor on the D+ line. This resistor is isolated from the D+ pin at reset and is attached under firmware control through the USBPUEN bit in the USBIO_CR1 register. After the D+ pull up resistor is connected to the D+ line, the system normally detects that as an attach and begins the USB enumeration process. No additional external pull up resistor must be added to the D+ line, since the transceiver signaling is optimized for use with the internal D+ pull up resistor. However, low value series resistors (24 Ω) must be added externally to meet the driving impedance requirement for full-speed USB.

The transceiver also includes 5 k Ω pull up resistors on both the D+ and D- pins for communication at PS/2 or similar signaling levels. These resistors are disconnected at reset and can be connected with the PS2PUEN bit in the USBIO_CR1 register. The D+ and D- pins can also be driven individually high and low in both USB and non-USB modes. The state of those pins can be read in any mode. Refer to the description of the [USBIO_CR0](#) and [USBIO_CR1](#) registers for more detail.

20.2.5 USB Suspend

Loss of USB activity, while the USB VBus is still asserted, indicates that the device must enter USB Suspend mode.

(Self-powered devices do not need to go into Suspend mode.) This condition is detected by monitoring the Bus Activity bit in the USB_CR1 register. This bit must be polled periodically. If it reads high (bus activity present), it must be cleared by firmware. If no activity is detected for the desired time (for example, 3 ms), then the device must enter Suspend mode.

Not all sleep modes preserve the USB configuration register states during sleep. The Standby I2C-USB mode is the preferred sleep mode for USB operation because the state of all USB registers is maintained during sleep. The other sleep modes do not preserve all of these registers to save sleep power.

The USB regulator settings must not be changed when entering sleep state, since the regulator automatically enters a low power state for the given regulator mode (pass-through or regulating).

20.2.5.1 *Using Standby I2C-USB Sleep Mode for USB Suspend*

To enter Standby I2C-USB mode, firmware powers down the desired functions, as it would to enter a Standby I2C-USB mode sleep state as documented in [10.1.1 Sleep Control Implementation Logic on page 74](#), including writing the SLEEP bit of the CPU_SCR0 register. In this mode, USB configuration registers and data are preserved during sleep.

20.2.5.2 *Using Standby or Deep Sleep Modes for USB Suspend*

The Standby and Deep Sleep modes do not have hardware supported for USB suspend operation because not all USB registers are preserved in these modes. In USB parts, if there is a need to enter Standby or Deep Sleep mode, then user firmware must save the non-retained registers to SRAM before entering into sleep and restore these registers once the device wakes up. The list of these registers is given below. Some configuration registers retain their values in all sleep modes: USB_CR0, USB_CR1, USBIO_CR0, USBIO_CR1, IMO_TR, IMO_TR1. In addition, the USB SRAM contents are preserved in all sleep modes.

The USB registers that retain state in Standby I2C-USB mode but not in other sleep modes are: the endpoint control registers (EPx_CRx), endpoint PMA write address (PMAx_WA)/read address (PMAx_RA) registers, PMA data registers (PMAx_DR), endpoint count registers (EPx_CNTx), endpoint 0 data register (EP0_DRx) and start of frame registers (USB_SOFx). These registers are reset after the device comes out of sleep.

(An alternative is to simply disconnect from the USB bus before going into one of these sleep modes, and then re-connect to re-initialize the USB system after waking up.)

20.2.5.3 *Wakeup from Suspend*

The USB wake interrupt must be enabled to allow the device to exit the sleep state when there is activity on the USB bus. This interrupt can be enabled at any time since it only asserts when the device is in the sleep state. Other interrupts may be optionally enabled, such as the sleep interrupt, GPIO, I2C, to periodically wake the device while in USB suspend state. The USB wake interrupt can wake up the device from all the three sleep states (Standby sleep, I2C-USB sleep and Deep Sleep). If D+ is low when the SLEEP bit is being set, the device briefly enters sleep state, and then exits sleep due to the USB wake interrupt.

By carefully using a sleep timer interrupt, the device can wake periodically, monitor the environment, and return to sleep while maintaining a low average current that meets the USB suspend current specification.

If the device needs to issue a resume signal to the USB system, firmware can write to the TEN and TD bits in the USBIO_CR0 register to manually force a K state on the bus. Using these bits produces signaling that meets the USB timing specifications.

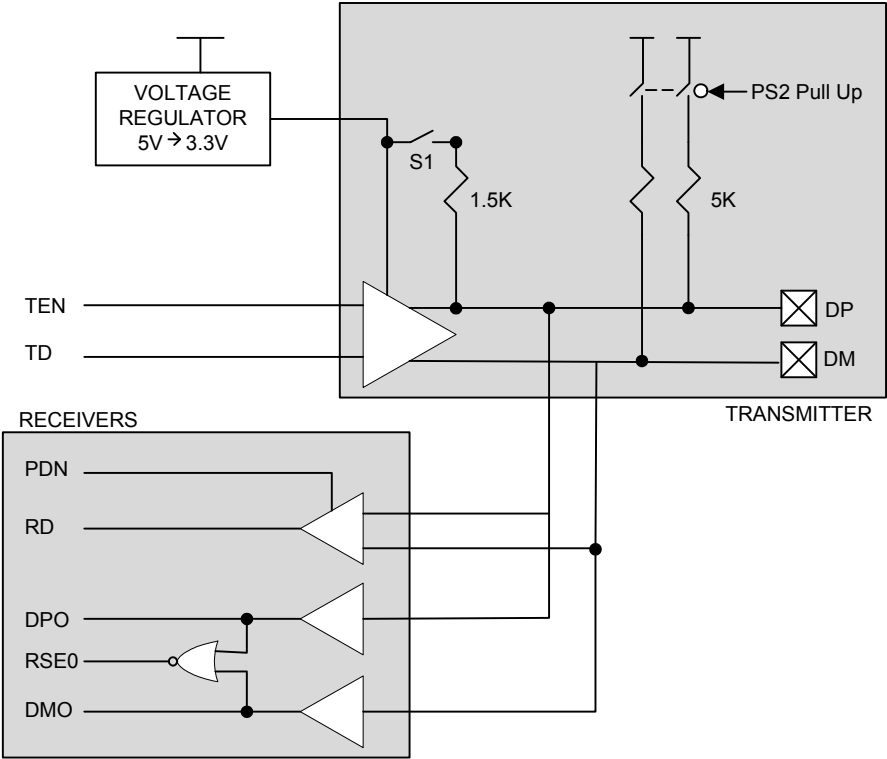
When driving a resume, the J state (TD=1) must be driven briefly before driving the K state (TD=0). The steps are summarized as follows:

1. Drive the J state (TEN=1, TD=1) for one instruction.
2. Drive the resume, or K state (TEN=1, TD=0) for the proper time (1 ms to 15 ms).
3. Stop driving the USB bus manually (TEN=0).

20.2.6 Regulator

The transceiver contains a built-in regulator that can be used to power the transceiver from the USB bus voltage or other supply around 5V. The regulator supplies the proper levels for USB signals, which switch between 0V and 3.3V nominally. If the PSoC device is operating with a Vdd supply near 3.3V, then the regulator must be placed into a pass-through mode so that the Vdd voltage is directly supplied to the transceiver, without regulation. The RegEnable bit (bit 0 in the USB_CR1 register) is used to pick between the regulating mode (5V supply) or the passthrough mode (3.3V supply). At power up, the regulator is automatically held in pass-through mode, but the USB transceiver pins are tristated.

Figure 20-2. Transceiver and Regulator Block Diagram



20.3 Register Definitions

The following registers are related to Full-Speed USB in the PSoC device. For a complete table of the Full-Speed USB registers, refer to the Registers table [Summary Table of the System Resource Registers on page 106](#). Register bits that are grayed out in this document are reserved bits and are not detailed in the register descriptions that follow. Always write reserved bits with a value of '0'.

20.3.1 USB_SOF0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,31h	USB_SOF0	Frame Number[7:0]								R : 00
0,32h	USB_SOF1					Frame Number[10:8]				R : 0

The USB Start of Frame Registers (USB_SOF0 and USB_SOF1) provide access to the 11-bit SOF frame number. Start of frame packets are sent from the host (for example, the PC) every 1 ms. For more information, see the *Universal Serial Bus Specification, revision 2.0*.

Bits 7 to 0: Frame Number. The USB_SOF0 register has the lower 8 bits [7:0] and the USB_SOF1 register has the upper 3 bits [10:8] of the SOF frame number.

For additional information, refer to the [USB_SOF0 register on page 193](#) and the [USB_SOF1 register on page 194](#).

20.3.2 USB_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,33h	USB_CR0	USB Enable	Device Address[6:0]							RW : 00

The USB Control Register 0 (USB_CR0) is used to set the PSoC's USB address and enable the USB system resource.

All bits in this register are reset to zero when a USB bus reset interrupt occurs.

Note Set the IMO frequency to 24 MHz and enable the 48 MHz clock in the OSC_CR2 register before USB is enabled. See IMO_TR and CPU_SCR1 registers for selecting IMO frequency as 24 MHz.

Bit 7: USB Enable. This bit enables the PSoC device to respond to USB traffic. '0' is USB disabled. The device does not respond to USB traffic. '1' is USB enabled.

Bits 6 to 0: Device Address[6:0]. These bits specify the USB device address to which the SIE responds. This address must be set by firmware and is specified by the USB host with a SET ADDRESS command during USB enumeration. This value must be programmed by firmware when assigned during enumeration. It is not set automatically by the hardware.

For additional information, refer to the [USB_CR0 register on page 195](#).

20.3.3 USBIO_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,34h	USBIO_CR0	TEN	TSE0	TD					RD	# : 0

The USB I/O Control Register 0 (USBIO_CR0) is used for manually transmitting on the USB D+ and D- pins, or reading the differential receiver.

Bit 7: TEN. This is used to manually transmit on the D+ and D- pins. Normally, this bit must be cleared to allow the internal SIE to drive the pins. The most common reason for manually transmitting is to force a resume state on the bus. '0' is manual transmission off (TSE0 and TD have no effect). '1' is manual transmission enabled (TSE0 and TD determine the state of the D+ and D- pins).

Bit 6: TSE0. Transmit Single-Ended Zero. SE0: both D+ and D- low. No effect if TEN=0. '0' is do not force SE0. '1' is force SE0 on D+ and D-.

Bit 5: TD. Transmit a USB J or K state on the USB bus. No effect if TEN=0 or TSE0=1. '0' forces USB K state (D+ is low, D- is high). '1' forces USB J state (D+ is high, D- is low).

Bit 0: RD. This read only bit gives the state of the USB differential receiver. '0' is D+ < D- or D+ = D- = 0. '1' is D+ > D-.

For additional information, refer to the [USBIO_CR0 register on page 196](#).

20.3.4 USBIO_CR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,35h	USBIO_CR1	IOMode	Drive Mode	DPI	DMI	PS2PUEN	USBPUEEN	DPO	DMO	# : 03

The USB I/O Control Register 1 (USBIO_CR1) is used to manually read or write the D+ and D- pins, and to configure internal pull up resistors on those pins.

Bit 7: IOMode. This bit allows the D+ and D- pins to be configured for either USB mode or bit banded modes. If this bit is set, the DMI and DPI bits are used to drive the D- and D+ pins. '0' is USB mode. Drive mode has no effect. '1' is drive mode, DMI and DPI determine state of the D+ and D- pins.

Bit 6: Drive Mode. If the IOMode bit is set, this bit configures the D- and D+ pins for either CMOS drive or open drain drive. If IOMode is cleared, this bit has no effect. Note that in open drain mode, 5 kΩ pull up resistors can be connected internally with the PS2PUEN bit. '0' is D+ and D- are in open drain mode. If the DPI or DMI bits are set high, the corresponding D+ or D- pad is high impedance. '1' is D+ and D- are in CMOS drive mode. D+ follows DPI and D- follows DMI.

Bit 5: DPI. This bit is used to drive the D+ pin if IOMode=1. Refer to the Drive Mode bit for drive state of pad. '0' is drive D+ pad low. '1' is drive D+ pad high (unless Drive Mode=0).

Bit 4: DMI. This bit is used to drive the D- pin if IOMode=1. Refer to the Drive Mode bit for drive state of pad. '0' is drive D- pad low. '1' is drive D- pad high (unless Drive Mode=0).

Bit 3: PS2PUEN. This bit controls the connection of the two internal 5 kΩ pull up resistors to the D+ and D- pins. '0' is no effect. '1' is apply 5k pull ups between Vdd and both D+ and D- pads, independent of the IOMode and Drive Mode bits.

Bit 2: USBPUEEN. This bit controls the connection of the internal 1.5 kΩ pull up resistor on the D+ pin. '0' is no effect. '1' is apply internal USB pull up resistor to D+ pad.

Bit 1: DPO. This read only bit gives the state of the D+ pin.

Bit 0: DMO. This read only bit gives the state of the D- pin.

For additional information, refer to the [USBIO_CR1 register on page 197](#).

20.3.5 EP0_CR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,36h	EP0_CR	Setup Received	IN Received	OUT Received	ACK'ed Transaction	Mode[3:0]				# : 00

The Endpoint Control Register (EP0_CR) is used to configure endpoint 0.

Because both firmware and the SIE are allowed to write to the Endpoint 0 Control and Count registers, the SIE provides an interlocking mechanism to prevent accidental overwriting of data. When the SIE writes to these registers they are locked and the processor cannot write to them until after reading the EP0_CR register. Writing to this register clears the upper four bits regardless of the value written.

Note The register lock is removed when the following M8C instructions are used to write the EP0_CR register as these instructions generate a read (IOR_) signal, which causes the lock to be removed even without the firmware actually reading the EP0_CR register.

```
mov reg[expr], expr
mov reg[X+expr], expr
```

Bit 7: Setup Received. When set, this bit indicates a valid setup packet was received and ACK'ed. This bit is forced high from the start of the data packet phase of the setup transaction, until the start of the ACK packet returned by the SIE. The CPU is prevented from clearing this bit during this interval. After this interval, the bit remains set until cleared by firmware. While this bit is set to '1', the CPU cannot write to the EP0_DRx registers. This prevents firmware from overwriting an incoming setup transaction before firmware has a chance to read the setup data. This bit is cleared by any non-locked writes to the register.

Bit 6: IN Received. When set, this bit indicates a valid IN packet has been received. This bit is updated to '1' after the host acknowledges an IN data packet. When clear, this bit indicates either no IN has been received or that the host did not acknowledge the IN data by sending an ACK handshake. It is cleared by any non-locked writes to the register.

Bit 5: OUT Received. When set, this bit indicates a valid OUT packet has been received and ACK'ed. This bit is updated to '1' after the last received packet in an OUT transaction. When clear, this bit indicates no OUT has been received. It is cleared by any non-locked writes to the register.

Bit 4: ACK'ed Transaction. This bit is set whenever the SIE engages in a transaction to the register's endpoint that completes with an ACK packet. This bit is cleared by any non-locked writes to the register.

Bits 3 to 0: Mode[3:0]. The mode bits control how the USB SIE responds to traffic and how the USB SIE changes the mode of that endpoint as a result of host packets to the endpoint.

For additional information, refer to the [EP0_CR register on page 198](#).

20.3.6 EP0_CNT Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,37h	EP0_CNT	Data Toggle	Data Valid			Byte Count[3:0]				# : 00

The Endpoint 0 Count Register (EP0_CNT) is used to configure endpoint 0.

Whenever the count updates from a setup or OUT transaction, this register locks and cannot be written by the CPU. Reading the EP0_CR register unlocks this register. This prevents firmware from overwriting a status update on incoming setup or OUT transactions, before firmware has a chance to read the data.

Bit 7: Data Toggle. This bit selects the data packet's toggle state. For IN transactions, firmware must set this bit. For OUT or setup transactions, the SIE hardware sets this bit to the state of the received Data Toggle bit. '0' is DATA0. '1' is DATA1.

Bit 6: Data Valid. This bit indicates whether there were errors in OUT or setup transactions. It is cleared to '0' if CRC, bit stuff, or PID errors have occurred. This bit does not update for some endpoint mode settings. This bit may be cleared by writing a zero to it when the register is not locked. '0' is error in data received. '1' is no error.

Bits 3 to 0: Byte Count[3:0]. These bits indicate the number of data bytes in a transaction. For IN transactions, firmware loads the count with the number of bytes to be transmitted to the host from the endpoint FIFO. Valid values are 0 to 8. For OUT or setup transactions, the count is updated by hardware to the number of data bytes received, plus two for the CRC bytes. Valid values are 2 to 10.

For additional information, refer to the [EP0_CNT register on page 199](#).

20.3.7 EP0_DRx Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,38h	EP0_DR0	Data Byte[7:0]								RW : 00
0,39h	EP0_DR1	Data Byte[7:0]								RW : 00
0,3Ah	EP0_DR2	Data Byte[7:0]								RW : 00
0,3Bh	EP0_DR3	Data Byte[7:0]								RW : 00
0,3Ch	EP0_DR4	Data Byte[7:0]								RW : 00
0,3Dh	EP0_DR5	Data Byte[7:0]								RW : 00
0,3Eh	EP0_DR6	Data Byte[7:0]								RW : 00
0,3Fh	EP0_DR7	Data Byte[7:0]								RW : 00

The Endpoint 0 Data Register (EP0_DRx) is used to read and write data to the USB control endpoint.

The EP0_DRx registers have a hardware-locking feature that prevents the CPU write when setup is active. The registers are locked as soon as the setup token is decoded and remain locked throughout the setup transaction and until the EP0_CR register has been read. This is to prevent overwriting new setup data before firmware knows it has arrived.

All other endpoint data buffers do not have this locking feature.

Bits 7 to 0: Data Byte[7:0]. These registers are shared for both transmit and receive. The count in the EP0_CNT register determines the number of bytes received or to be transferred.

For additional information, refer to the [EP0_DRx register on page 200](#).

20.3.8 EPx_CNT1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,41h	EP1_CNT1	Data Count[7:0]								RW : 00
0,43h	EP2_CNT1	Data Count[7:0]								RW : 00
0,45h	EP3_CNT1	Data Count[7:0]								RW : 00
0,47h	EP4_CNT1	Data Count[7:0]								RW : 00
0,49h	EP5_CNT1	Data Count[7:0]								RW : 00
0,4Bh	EP6_CNT1	Data Count[7:0]								RW : 00
0,4Dh	EP7_CNT1	Data Count[7:0]								RW : 00
0,4Fh	EP8_CNT1	Data Count[7:0]								RW : 00

The Endpoint Count Register 1 (EPx_CNT1) sets or reports the number of bytes in a USB data transfer to the non-control endpoints.

Bit 7 to 0: Data Count. These bits are the eight LSb of a 9-bit counter. The MSb is the Count MSb bit of the EPx_CNT0 register. The 9-bit count indicates the number of data bytes in a transaction. For IN transactions, firmware loads the count with the number of data bytes to be transmitted to the host. Valid values are 0 to 256. The 9-bit count also sets the limit for the number of bytes that are received for an OUT transaction. Before an OUT transaction is received for an endpoint, this count value must be set to the maximum number of data bytes to receive. If this count value is set to a value greater than the number of bytes (Data + CRC) received, both the data from the USB packet and the two-byte CRC are written to the USB's dedicated SRAM.

If the number of data bytes received is exactly the same as the 9-bit count, then only the data is updated into the USB SRAM and the CRC is discarded but the OUT transaction is completed according to the Mode bits of the EPx Control Register. If the number of data bytes received is more than the 9-bit count, then the OUT transaction is ignored.

After the OUT transaction is complete, the full 9-bit count is updated by the SIE to the actual number of data bytes received by the SIE plus two for the packet's CRC. Valid values are 2 to 258.

To get the actual number of bytes received, firmware must decrement the 9-bit count by two.

For additional information, refer to the [EPx_CNT1 register on page 202](#).

20.3.9 EPx_CNT0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,40h	EP1_CNT0	Data Toggle	Data Valid						Count MSB	# : 0
0,42h	EP2_CNT0	Data Toggle	Data Valid						Count MSB	# : 0
0,44h	EP3_CNT0	Data Toggle	Data Valid						Count MSB	# : 0
0,46h	EP4_CNT0	Data Toggle	Data Valid						Count MSB	# : 0
0,48h	EP5_CNT0	Data Toggle	Data Valid						Count MSB	# : 0
0,4Ah	EP6_CNT0	Data Toggle	Data Valid						Count MSB	# : 0
0,4Ch	EP7_CNT0	Data Toggle	Data Valid						Count MSB	# : 0
0,4Eh	EP8_CNT0	Data Toggle	Data Valid						Count MSB	# : 0

The Endpoint Count Register 0 (EPx_CNT0) is used for configuring endpoints 1 through 8.

Bit 7: Data Toggle. This bit selects the data packet's toggle state. For IN transactions, firmware must set this bit to the expected state. For OUT transactions, the hardware sets this bit to the state of the received Data Toggle bit. '0' is DATA0. '1' is DATA1.

Bit 6: Data Valid. This bit is used for OUT transactions only and is read only. It is cleared to '0' if CRC, bit stuffing errors, or PID errors occur. This bit does not update for some endpoint mode settings. '0' is error in data received. '1' is no error.

Bit 0: Count MSB. This bit is the one MSb of a 9-bit counter. The LSb are the Data Count[7:0] bits of the EPx_CNT1 register. Refer to the [EPx_CNT1 Register](#) for more information.

For additional information, refer to the [EPx_CNT0 register on page 201](#).

20.3.10 EPx_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,54h	EP1_CR0	Stall		NAK_INT_EN	ACK'ed Tx	Mode[3:0]				# : 00
1,55h	EP2_CR0	Stall		NAK_INT_EN	ACK'ed Tx	Mode[3:0]				# : 00
1,56h	EP3_CR0	Stall		NAK_INT_EN	ACK'ed Tx	Mode[3:0]				# : 00
1,57h	EP4_CR0	Stall		NAK_INT_EN	ACK'ed Tx	Mode[3:0]				# : 00
1,58h	EP5_CR0	Stall		NAK_INT_EN	ACK'ed Tx	Mode[3:0]				# : 00
1,59h	EP6_CR0	Stall		NAK_INT_EN	ACK'ed Tx	Mode[3:0]				# : 00
1,5Ah	EP7_CR0	Stall		NAK_INT_EN	ACK'ed Tx	Mode[3:0]				# : 00
1,5Bh	EP8_CR0	Stall		NAK_INT_EN	ACK'ed Tx	Mode[3:0]				# : 00

The Endpoint Control Register 0 (EPx_CR0) is used for status and configuration of the non-control endpoints 1 to 8.

Bit 7: Stall. When this bit is set, the SIE stalls an OUT packet if the mode bits are set to ACK-OUT. The SIE stalls an IN packet if the mode bits are set to ACK-IN. This bit must be cleared for all other modes. '0' is do not issue a stall. '1' is stall an OUT packet if mode bits are set to ACK-OUT, or stall an IN packet if mode bits are set to ACK-IN.

Bit 5: NAK Int Enable. When set, this bit causes an endpoint interrupt to be generated even when a transfer completes with a NAK. '0' is do not issue an interrupt after completing the transaction by sending NAK. '1' is interrupt after transaction is complete by sending NAK.

Bit 4: ACK'ed Transaction. The ACK'ed Transaction bit is set whenever the SIE engages in a transaction to the register's endpoint that completes with an ACK packet. This bit is cleared by any writes to the register. '0' is no ACK'ed transactions since bit was last cleared. '1' indicates a transaction ended with an ACK.

Bits 3 to 0: Mode[3:0]. The mode controls how the USB SIE responds to traffic and how the USB SIE changes the mode of that endpoint as a result of host packets to the endpoint. Refer to ["Mode Encoding for Control and Non-Control Endpoints"](#) on page 166.

For additional information, refer to the [EPx_CR0 register](#) on page 265.

20.3.11 PMAx_WA Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,34h	PMA0_WA	Write Address[7:0]								RW : 00
1,35h	PMA1_WA	Write Address[7:0]								RW : 00
1,36h	PMA2_WA	Write Address[7:0]								RW : 00
1,37h	PMA3_WA	Write Address[7:0]								RW : 00
1,38h	PMA4_WA	Write Address[7:0]								RW : 00
1,39h	PMA5_WA	Write Address[7:0]								RW : 00
1,3Ah	PMA6_WA	Write Address[7:0]								RW : 00
1,3Bh	PMA7_WA	Write Address[7:0]								RW : 00
1,44h	PMA8_WA	Write Address[7:0]								RW : 00
1,45h	PMA9_WA	Write Address[7:0]								RW : 00
1,46h	PMA10_WA	Write Address[7:0]								RW : 00
1,47h	PMA11_WA	Write Address[7:0]								RW : 00
1,48h	PMA12_WA	Write Address[7:0]								RW : 00
1,49h	PMA13_WA	Write Address[7:0]								RW : 00
1,4Ah	PMA14_WA	Write Address[7:0]								RW : 00
1,4Bh	PMA15_WA	Write Address[7:0]								RW : 00

The PSoC Memory Arbiter Write Address Register (PMAx_WA) is used to set the beginning SRAM address for the PMA channel. A PMAx_WA register address uses the same physical register as the PMAx_RA register address. Therefore, when the read address is changed, the write address is also changed and the PMAx_RA and PMAx_WA registers always return the same value when read.

Bits 7 to 0: Address[7:0]. The value returned when this register is read depends on whether the PMA channel is being used by the USB SIE or by the M8C. In the USB case, this register always returns the beginning SRAM address for the PMA channel. In the M8C case, this register always returns the next SRAM address that is used by the PMA channel, if a byte is written to the channel's data register (PMAx_DR) by the M8C.

For additional information, refer to the [PMAx_WA register on page 263](#).

20.3.12 PMAx_DR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,58h	PMA0_DR	Data Byte[7:0]								RW : 00
0,59h	PMA1_DR	Data Byte[7:0]								RW : 00
0,5Ah	PMA2_DR	Data Byte[7:0]								RW : 00
0,5Bh	PMA3_DR	Data Byte[7:0]								RW : 00
0,5Ch	PMA4_DR	Data Byte[7:0]								RW : 00
0,5Dh	PMA5_DR	Data Byte[7:0]								RW : 00
0,5Eh	PMA6_DR	Data Byte[7:0]								RW : 00
0,5Fh	PMA7_DR	Data Byte[7:0]								RW : 00
0,64h	PMA8_DR	Data Byte[7:0]								RW : 00
0,65h	PMA9_DR	Data Byte[7:0]								RW : 00
0,66h	PMA10_DR	Data Byte[7:0]								RW : 00
0,67h	PMA11_DR	Data Byte[7:0]								RW : 00
0,68h	PMA12_DR	Data Byte[7:0]								RW : 00
0,69h	PMA13_DR	Data Byte[7:0]								RW : 00
0,6Ah	PMA14_DR	Data Byte[7:0]								RW : 00
0,6Bh	PMA15_DR	Data Byte[7:0]								RW : 00

The PSoC Memory Arbiter Data Register (PMAx_DR) is used to read and write to a particular PMA channel by either the USB SIE or the M8C. Note that a PMA channel may not be used simultaneously by both the USB SIE and the M8C.

Bits 7 to 0: Data Byte[7:0]. When the M8C writes to this register, the PMA registers the byte and then stores the value at the address in SRAM indicated by the PMAx_WA register.

After the value has been written to SRAM, the PMAx_WA register is automatically incremented. When the USB SIE writes to this register, the PMA registers the byte and then stores the value in SRAM using the sum of the value of the PMAx_WA register and the USB SIEs received byte count. When the M8C reads this register, a pre-loaded value is returned and the PMAx_RA value is automatically incremented.

The new PMAx_RA value is used to fetch the next value from the SRAM, to be ready for the next read from the channel's PMAx_DR register. When the USB SIE reads the PMAx_DR register, it also receives a pre-loaded value, which triggers the PMA logic to fetch the next value in SRAM to be ready for the USB SIEs next read request. In all read cases, the initial pre-load of the first address of the channel is triggered by writing the first address of the channel to the channel's PMAx_RA register. Therefore, the PMAx_RA register must be written after data has been stored for the channel.

For additional information, refer to the [PMAx_DR register on page 203](#).

20.3.13 PMAx_RA Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,3Ch	PMA0_RA	Read Address[7:0]								RW : 00
1,3Dh	PMA1_RA	Read Address[7:0]								RW : 00
1,3Eh	PMA2_RA	Read Address[7:0]								RW : 00
1,3Fh	PMA3_RA	Read Address[7:0]								RW : 00
1,40h	PMA4_RA	Read Address[7:0]								RW : 00
1,41h	PMA5_RA	Read Address[7:0]								RW : 00
1,42h	PMA6_RA	Read Address[7:0]								RW : 00
1,43h	PMA7_RA	Read Address[7:0]								RW : 00
1,4Ch	PMA8_RA	Read Address[7:0]								RW : 00
1,4Dh	PMA9_RA	Read Address[7:0]								RW : 00
1,4Eh	PMA10_RA	Read Address[7:0]								RW : 00
1,4Fh	PMA11_RA	Read Address[7:0]								RW : 00
1,50h	PMA12_RA	Read Address[7:0]								RW : 00
1,51h	PMA13_RA	Read Address[7:0]								RW : 00
1,52h	PMA14_RA	Read Address[7:0]								RW : 00
1,53h	PMA15_RA	Read Address[7:0]								RW : 00

The PSoC Memory Arbiter Read Address Register (PMAx_RA) is used to set the beginning address for the PMA channel. A PMAx_RA register address uses the same physical register as the PMAx_WA register address. Therefore, when the read address is changed, the write address is also changed and the PMAx_WA and PMAx_RA registers always return the same value when read. When a PMAx_RA register is written, the address is stored and the value of the corresponding SRAM address is loaded into the channel's PMAx_DR. Therefore, this register must only be written after valid data has been stored in SRAM for the channel.

Bits 7 to 0: Address[7:0]. The value returned when this register is read depends on whether the PMA channel is being used by the USB SIE or by the M8C. In the USB SIE case, this register always returns the beginning SRAM address for the PMA channel. In the M8C case, this register always returns the next SRAM address that is used by the PMA channel, if a byte is read from the channel's data register (PMAx_DR) by the M8C.

For additional information, refer to the [PMAx_RA register on page 264](#).

20.3.14 USB_CR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,30h	USB_CR1						BusActivity	EnableLock	RegEnable	# : 0

The USB Control Register 1 (USB_CR1) is used to configure the internal regulator and the oscillator tuning capability.

Bit 2: BusActivity. The BusActivity bit is a "sticky" bit that detects any non-idle USB event that has occurred on the USB bus. After set to high by the SIE to indicate the bus activity, this bit retains its logical high value until firmware clears it. Writing a '0' to this bit clears it; writing a '1' preserves its value. '0' is no activity. '1' is non-idle activity (D+ = low) was detected since the last time the bit was cleared.

Bit 1: EnableLock. Set this bit to turn on the automatic frequency locking of the internal oscillator to USB traffic. Unless an external clock is being provided, this bit must remain set for proper USB operation. '0' is locking disabled. '1' is locking enabled.

Bit 0: RegEnable. This bit controls the operation of the internal USB regulator. For applications with device supply voltages in the 5V range, set this bit high to enable the internal regulator. For device supply voltages in the 3.3V range, clear this bit to connect the transceiver directly to the supply. '0' is passthrough mode. Use for Vdd = 3.3V range. '1' is regulating mode. Use for Vdd = 5V range.

For additional information, refer to the [USB_CR1 register on page 262](#).

20.3.15 IMO_TR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,FAh	IMO_TR1						Fine Trim[2:0]			RW : 00

INTERNAL Register – The Internal Main Oscillator Trim Register 1 (IMO_TR1) fine tunes the IMO frequency.

For information on the other IMO trim register (IMO_TR) see the Internal Main Oscillator chapter or refer to the [IMO_TR1 register on page 286](#) in the Register Details chapter.

Bits 2 to 0: Fine Trim[2:0]. These bits provide a fine tuning capability to the IMO trim. These three bits are the three LSb of the IMO trim with the IMO_TR register supplying the eight MSb.

For additional information, refer to the [IMO_TR1 register on page 286](#).

20.3.16 Related Registers

- [USB_MISC_CR Register on page 112](#).

Section E: Registers



The Registers section discusses the registers of the PSoC device. It lists all the registers in mapping tables, in address order. For easy reference, each register is linked to the page of a detailed description located in the next chapter. This section encompasses the following chapter:

- [Register Reference chapter on page 187.](#)

Register General Conventions

The register conventions specific to this section and the Register Reference chapter are listed in the following table:

Register Conventions

Convention	Description
Empty, grayed-out table cell	Illustrates a reserved bit or group of bits.
'x' before the comma in an address	Indicates the register exists in register bank 1 and register bank 2.
'x' in a register name	Indicates that there are multiple instances/address ranges of the same register.
R	Read register or bit(s).
W	Write register or bit(s).
O	Only a read/write register or bit(s).
L	Logical register or bit(s).
C	Clearable register or bit(s).
#	Access is bit specific.

Register Mapping Tables

The PSoC device has a total register address space of 512 bytes. The register space is also referred to as I/O space and is broken into two parts: Bank 0 (user space) and Bank 1 (configuration space). The XIO bit in the Flag register (CPU_F) determines which bank the user is currently in. When the XIO bit is set, the user is said to be in the “extended” address space or the “configuration” registers.

Refer to the individual PSoC device data sheets for device-specific register mapping information.

Section E: Registers

Register Map Bank 0 Table: User Space

Name	Addr (0,Hex)	Access	Page	Name	Addr (0,Hex)	Access	Page	Name	Addr (0,Hex)	Access	Page	Name	Addr (0,Hex)	Access	Page
PRT0DR	00	RW	188	EP1_CNT0	40	#	201		80				C0		
PRT0IE	01	RW	189	EP1_CNT1	41	RW	202		81				C1		
	02			EP2_CNT0	42	#	201		82				C2		
	03			EP2_CNT1	43	RW	202		83				C3		
PRT1DR	04	RW	188	EP3_CNT0	44	#	201		84				C4		
PRT1IE	05	RW	189	EP3_CNT1	45	RW	202		85				C5		
	06			EP4_CNT0	46	#	201		86				C6		
	07			EP4_CNT1	47	RW	202		87				C7		
PRT2DR	08	RW	188	EP5_CNT0	48	#	201		88			I2C_XCFG	C8	RW	226
PRT2IE	09	RW	189	EP5_CNT1	49	RW	202		89			I2C_XSTAT	C9	R	227
	0A			EP6_CNT0	4A	#	201		8A			I2C_ADDR	CA	RW	228
	0B			EP6_CNT1	4B	RW	202		8B			I2C_BP	CB	R	229
PRT3DR	0C	RW	188	EP7_CNT0	4C	#	201		8C			I2C_CP	CC	R	230
PRT3IE	0D	RW	189	EP7_CNT1	4D	RW	202		8D			CPU_BP	CD	RW	231
	0E			EP8_CNT0	4E	#	201		8E			CPU_CP	CE	R	232
	0F			EP8_CNT1	4F	RW	202		8F			I2C_BUF	CF	RW	233
PRT4DR	10	RW	188		50				90			CUR_PP	D0	RW	234
PRT4IE	11	RW	189		51				91			STK_PP	D1	RW	235
	12				52				92				D2		
	13				53				93			IDX_PP	D3	RW	236
	14				54				94			MVR_PP	D4	RW	237
	15				55				95			MVW_PP	D5	RW	238
	16				56				96			I2C_CFG	D6	RW	239
	17				57				97			I2C_SCR	D7	#	240
	18			PMA0_DR	58	RW	203		98			I2C_DR	D8	RW	241
	19			PMA1_DR	59	RW	203		99				D9		
	1A			PMA2_DR	5A	RW	203		9A			INT_CLR0	DA	RW	242
	1B			PMA_DR	5B	RW	203		9B			INT_CLR1	DB	RW	244
	1C			PMA4_DR	5C	RW	203		9C			INT_CLR2	DC	RW	246
	1D			PMA5_DR	5D	RW	203		9D				DD		
	1E			PMA6_DR	5E	RW	203		9E			INT_MSK2	DE	RW	248
	1F			PMA7_DR	5F	RW	203		9F			INT_MSK1	DF	RW	249
	20				60			CS_CR0	A0	RW	211	INT_MSK0	E0	RW	250
	21			AMUX_CFG	61	RW	204	CS_CR1	A1	RW	212	INT_SW_EN	E1	RW	251
	22				62			CS_CR2	A2	RW	213	INT_VC	E2	RC	252
	23				63			CS_CR3	A3	RW	214	RES_WDT	E3	W	253
	24			PMA8_DR	64	RW	203	CS_CNTL	A4	RW	215		E4		
	25			PMA9_DR	65	RW	203	CS_CNTH	A5	RW	216		E5		
	26			PMA10_DR	66	RW	203	CS_STAT	A6	#	217		E6		
	27			PMA11_DR	67	RW	203	CS_TIMER	A7	RW	218		E7		
	28			PMA12_DR	68	RW	203	CS_SLEW	A8	RW	219		E8		
SPI_TXR	29	W	190	PMA13_DR	69	RW	203	PRS_CR	A9	RW	220		E9		
SPI_RXR	2A	R	191	PMA14_DR	6A	RW	203		AA				EA		
SPI_CR	2B	#	192	PMA15_DR	6B	RW	203		AB				EB		
	2C			TMP_DR0	6C	RW	266		AC				EC		
	2D			TMP_DR1	6D	RW	266		AD				ED		
	2E			TMP_DR2	6E	RW	266		AE				EE		
	2F			TMP_DR3	6F	RW	266		AF				EF		
	30				70			PT0_CFG	B0	RW	221		F0		
USB_SOF0	31	R	193		71			PT0_DATA1	B1	RW	223		F1		
USB_SOF1	32	R	194		72			PT0_DATA0	B2	RW	223		F2		
USB_CR0	33	RW	195		73			PT1_CFG	B3	RW	224		F3		
USBIO_CR0	34	#	196		74			PT1_DATA1	B4	RW	223		F4		
USBIO_CR1	35	#	197		75			PT1_DATA0	B5	RW	223		F5		
EP0_CR	36	#	198		76			PT2_CFG	B6	RW	225		F6		
EP0_CNT0	37	#	199		77			PT2_DATA1	B7	RW	223	CPU_F	F7	RL	254
EP0_DR0	38	RW	200	CMP_RDC	78	#	205	PT2_DATA0	B8	RW	223		F8		
EP0_DR1	39	RW	200	CMP_MUX	79	RW	206		B9				F9		
EP0_DR2	3A	RW	200	CMP_CR0	7A	RW	207		BA				FA		
EP0_DR3	3B	RW	200	CMP_CR1	7B	RW	208		BB				FB		
EP0_DR4	3C	RW	200	CMP_LUT	7C	RW	210		BC				FC		
EP0_DR5	3D	RW	200		7D				BD			IDAC_D	FD	RW	256
EP0_DR6	3E	RW	200		7E				BE			CPU_SCR1	FE	#	257
EP0_DR7	3F	RW	200		7F				BF			CPU_SCR0	FF	#	258

Gray fields are reserved. # Access is bit specific.

Register Map Bank 1 Table: Configuration Space

Name	Addr (1.Hex)	Access	Page	Name	Addr (1.Hex)	Access	Page	Name	Addr (1.Hex)	Access	Page	Name	Addr (1.Hex)	Access	Page
PRT0DM0	00	RW	259	PMA4_RA	40	RW	264		80				C0		
PRT0DM1	01	RW	260	PMA5_RA	41	RW	264		81				C1		
	02			PMA6_RA	42	RW	264		82				C2		
	03			PMA7_RA	43	RW	264		83				C3		
PRT1DM0	04	RW	259	PMA8_WA	44	RW	263		84				C4		
PRT1DM1	05	RW	260	PMA9_WA	45	RW	263		85				C5		
	06			PMA10_WA	46	RW	263		86				C6		
	07			PMA11_WA	47	RW	263		87				C7		
PRT2DM0	08	RW	259	PMA12_WA	48	RW	263		88				C8		
PRT2DM1	09	RW	260	PMA13_WA	49	RW	263		89				C9		
	0A			PMA14_WA	4A	RW	263		8A				CA		
	0B			PMA15_WA	4B	RW	263		8B				CB		
PRT3DM0	0C	RW	259	PMA8_RA	4C	RW	264		8C				CC		
PRT3DM1	0D	RW	260	PMA9_RA	4D	RW	264		8D				CD		
	0E			PMA10_RA	4E	RW	264		8E				CE		
	0F			PMA11_RA	4F	RW	264		8F				CF		
PRT4DM0	10	RW	260	PMA12_RA	50	RW	264		90				D0		
PRT4DM1	11	RW	260	PMA13_RA	51	RW	264		91			OUT_P0	D1	RW	268
	12			PMA14_RA	52	RW	264		92			ECO_ENBUS	D2	RW	269
	13			PMA15_RA	53	RW	264		93			ECO_TRIM	D3	RW	270
	14			EP1_CR0	54	#	265		94				D4		
	15			EP2_CR0	55	#	265		95				D5		
	16			EP3_CR0	56	#	265		96				D6		
	17			EP4_CR0	57	#	265		97				D7		
	18			EP5_CR0	58	#	265		98			MUX_CR0	D8	RW	271
	19			EP6_CR0	59	#	265		99			MUX_CR1	D9	RW	271
	1A			EP7_CR0	5A	#	265		9A			MUX_CR2	DA	RW	271
	1B			EP8_CR0	5B	#	265		9B			MUX_CR3	DB	RW	271
	1C				5C				9C			IO_CFG1	DC	RW	272
	1D				5D				9D			OUT_P1	DD	RW	273
	1E				5E				9E			IO_CFG2	DE	RW	275
	1F				5F				9F			MUX_CR4	DF	RW	271
	20				60				A0			OSC_CR0	E0	RW	276
	21				61				A1			ECO_CFG	E1	#	277
	22				62				A2			OSC_CR2	E2	RW	278
	23				63				A3			VLT_CR	E3	RW	279
	24				64				A4			VLT_CMP	E4	R	280
	25				65				A5				E5		
	26				66				A6				E6		
	27				67				A7				E7		
	28				68				A8			IMO_TR	E8	W	281
SPI_CFG	29	RW	261		69				A9			ILO_TR	E9	W	282
	2A				6A				AA				EA		
	2B				6B				AB			SLP_CFG	EB	RW	283
	2C			TMP_DR0	6C	RW	266		AC			SLP_CFG2	EC	RW	284
	2D			TMP_DR1	6D	RW	266		AD			SLP_CFG3	ED	RW	285
	2E			TMP_DR2	6E	RW	266		AE				EE		
	2F			TMP_DR3	6F	RW	266		AF				EF		
USB_CR1	30	#	262		70				B0				F0		
	31				71				B1				F1		
	32				72				B2				F2		
	33				73				B3				F3		
PMA0_WA	34	RW	263		74				B4				F4		
PMA1_WA	35	RW	263		75				B5				F5		
PMA2_WA	36	RW	263		76				B6				F6		
PMA3_WA	37	RW	263		77				B7			CPU_F	F7	RL	254
PMA4_WA	38	RW	263		78				B8				F8		
PMA5_WA	39	RW	263		79				B9				F9		
PMA6_WA	3A	RW	263		7A				BA			IMO_TR1	FA	RW	286
PMA7_WA	3B	RW	263		7B				BB				FB		
PMA0_RA	3C	RW	264		7C				BC				FC		
PMA1_RA	3D	RW	264		7D			USB_MISC_CR	BD	RW	267		FD		
PMA2_RA	3E	RW	264		7E				BE				FE		
PMA3_RA	3F	RW	264		7F				BF				FF		

Gray fields are reserved # Access is bit specific.

21. Register Reference



This chapter is a reference for all the PSoC device registers in address order, for Bank 0 and Bank 1. The most detailed descriptions of the PSoC registers are in the Register Definitions section of each chapter. The registers that are in both banks are incorporated with the Bank 0 registers, designated with an 'x', rather than a '0' preceding the comma in the address. Bank 0 registers are listed first and begin on page 188. Bank 1 registers are listed second and begin on page 259. A condensed view of all the registers is shown in the register mapping tables starting on page 183.

21.1 Maneuvering Around the Registers

For ease-of-use, this chapter has been formatted so that there is one register per page, although some registers use two pages. On each page, from top to bottom, there are four sections:

1. Register name and address (from lowest to highest).
2. Register table showing the bit organization, with reserved bits grayed out.
3. Written description of register specifics or links to additional register information.
4. Detailed register bit descriptions.

Use the register tables, in addition to the detailed register bit descriptions, to determine which bits are reserved. Reserved bits are grayed table cells and are not described in the bit description section. Reserved bits must always be written with a value of '0'. For all registers, an 'x' before the comma in the address field indicates that the register can be accessed or written to no matter what bank is used. For example, the M8C flag register's (CPU_F) address is 'x,F7h' meaning it is located in bank 0 and bank 1 at F7h.

21.2 Register Conventions

The following table lists the register conventions that are specific to this chapter.

Register Conventions

Convention	Example	Description
'x' in a register name	PRTxIE	Multiple instances/address ranges of the same register.
R	R : 00	Read register or bit(s).
W	W : 00	Write register or bit(s).
O	RO : 00	Only a read/write register or bit(s).
L	RL : 00	Logical register or bit(s).
C	RC : 00	Clearable register or bit(s).
00	RW : 00	Reset value is 0x00 or 00h.
XX	RW : XX	Register is not reset.
0,	0,04h	Register is in bank 0.
1,	1,23h	Register is in bank 1.
x,	x,F7h	Register exists in register bank 0 and register bank 1.
Empty, grayed-out table cell		Reserved bit or group of bits, unless otherwise stated.

21.3 Bank 0 Registers

The following registers are all in bank 0 and are listed in address order. An 'x' before the comma in the register's address indicates that the register can be accessed in Bank 0 and Bank 1, independent of the XIO bit in the CPU_F register. Registers that are in both Bank 0 and Bank 1 are listed in address order in Bank 0. For example, the RDIxLT1 register has an address of x,B4h and is listed only in Bank 0 but is accessed in both Bank 0 and Bank 1.

21.3.1 PRTxDR

Port Data Registers

Individual Register Names and Addresses:

0,00h

PRT0DR : 0,00h
PRT4DR : 0,10h

PRT1DR : 0,04h

PRT2DR : 0,08h

PRT3DR : 0,0Ch

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

These registers allow for write or read access, or the current logical equivalent, of pin voltage.

The upper nibble of the PRT4DR register returns the last data bus value when read. You need to mask it off before using this information. For additional information, refer to the [Register Definitions on page 59](#) in the GPIO chapter.

Bit	Name	Description
7:0	Data[7:0]	Write value to port or read value from port. Reads return the state of the pin, not the value in the PRTxDR register.

21.3.2 PRTxIE

Port Interrupt Enable Registers

Individual Register Names and Addresses:

PRT0IE : 0,01h
PRT4IE : 0,11h

PRT1IE : 0,05h

PRT2IE : 0,09h

PRT3IE : 0,0Dh

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

These registers enable or disable interrupts from individual GPIO pins.

The upper nibble of the PRT4IE register returns the last data bus value when read and must be masked off before using this information. For additional information, refer to the [Register Definitions on page 59](#) in the GPIO chapter.

Bit	Name	Description
7:0	Interrupt Enables[7:0]	These bits enable the corresponding port pin interrupt. Only four LSB are used since this port has four pins. 0 Port pin interrupt disabled for the corresponding pin. 1 Port pin interrupt enabled for the corresponding pin. Interrupt mode is determined by the IOINT bit in the IO_CFG1 register.

21.3.3 SPI_TXR

SPI Transmit Data Register

Individual Register Names and Addresses:

0,29h

SPI_TXR : 0,29h

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

This register is the SPI's transmit data register.

For additional information, refer to the [Register Definitions on page 147](#) in the SPI chapter.

Bit	Name	Description
7:0	Data[7:0]	Data for selected function.

21.3.4 SPI_RXR

SPI Receive Data Register

Individual Register Names and Addresses:

0,2Ah

SPI_RXR : 0,2Ah

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

This register is the SPI's receive data register.

For additional information, refer to the [Register Definitions on page 147](#) in the SPI chapter.

Bit	Name	Description
7:0	Data[7:0]	Data for selected function.

21.3.5 SPI_CR

SPI Control Register

Individual Register Names and Addresses:

SPI_CR : 0,2Bh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	R : 0	R : 0	R : 1	R : 0	RW : 0	RW : 0	RW : 0
Bit Name	LSb First	Overrun	SPI Complete	TX Reg Empty	RX Reg Full	Clock Phase	Clock Polarity	Enable

This register is the SPI control register.

The LSb First, Clock Phase, and Clock Polarity bits are configuration bits. Do not change them once the block is enabled. These bits can be set at the same time that the block is enabled. For additional information, refer to the [Register Definitions on page 147](#) in the SPI chapter.

Bit	Name	Description
7	LSb First	Do not change this bit during an SPI transfer. 0 Data is shifted out MSb first. 1 Data is shifted out LSb first.
6	Overrun	0 No overrun has occurred. 1 Overrun has occurred. Indicates that a new byte is received and loaded into the RX Buffer before the previous one is read. It is cleared on a read of this (CR0) register.
5	SPI Complete	0 Indicates that a byte may still be in the process of shifting out, or no transmission is active. 1 Indicates that a byte is shifted out and all associated clocks are generated. It is cleared on a read of this (CR0) register. Optional interrupt.
4	TX Reg Empty	Reset state and the state when the block is disabled is '1'. 0 Indicates that a byte is currently buffered in the TX register. 1 Indicates that a byte is written to the TX register and cleared on write of the TX Buffer (DR1) register. This is the default interrupt. This status is initially asserted on block enable; however, the TX Reg Empty interrupt occurs only after the first data byte is written and transferred into the shifter.
3	RX Reg Full	0 RX register is empty. 1 A byte is received and loaded into the RX register. It is cleared on a read of the RX Buffer (DR2) register.
2	Clock Phase	0 Data is latched on the leading clock edge. Data changes on the trailing edge (modes 0, 1). 1 Data changes on the leading clock edge. Data is latched on the trailing edge (modes 2, 3).
1	Clock Polarity	0 Non-inverted, clock idles low (modes 0, 2). 1 Inverted, clock idles high (modes 1, 3).
0	Enable	0 SPI function is not enabled. 1 SPI function is enabled.

21.3.6 USB_SOF0

USB Start-of-Frame Register 0

Individual Register Names and Addresses:

0,31h

USB_SOF0 : 0,31h

	7	6	5	4	3	2	1	0
Access : POR	R : 00							
Bit Name	Frame Number[7:0]							

This register is a USB Start-of-Frame register 0.

For additional information, refer to the [Register Definitions on page 171](#) in the Full-Speed USB chapter.

Bit	Name	Description
7:0	Frame Number[7:0]	Contains the lower eight bits of the frame number.

21.3.7 USB_SOF1

USB Start-of-Frame Register 1

Individual Register Names and Addresses:

0,32h

USB_SOF1 : 0,32h

	7	6	5	4	3	2	1	0
Access : POR								R : 0
Bit Name								Frame Number[10:8]

This register is a USB Start-of-Frame register 1.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 171](#) in the Full-Speed USB chapter.

Bit	Name	Description
2:0	Frame Number[10:8]	Contains the upper three bits of the frame number.

21.3.8 USB_CR0

USB Control Register 0

Individual Register Names and Addresses:

0,33h

USB_CR0 : 0,33h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 00						
Bit Name	USB Enable	Device Address[6:0]						

This register is a USB control register 0.

For additional information, refer to the [Register Definitions on page 171](#) in the Full-Speed USB chapter.

Bit	Name	Description
7	USB Enable	This bit enables the PSoC device to respond to USB traffic. 0 USB disabled. 1 USB enabled.
6:0	Device Address	These bits specify the USB address to which the SIE responds.

21.3.9 USBIO_CR0

USB I/O Control Register 0

Individual Register Names and Addresses:

USBIO_CR0 : 0,34h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0					R : 0
Bit Name	TEN	TSE0	TD					RD

This register is a USBIO manual control register 0.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 171](#) in the Full-Speed USB chapter.

Bit	Name	Description
7	TEN	Transmit Enable. This bit is used to manually transmit on D+, D- pins. Normally, this bit must be cleared to allow the internal SIE to drive the pins. The most common reason for manually transmitting is to force a resume state on the bus. 0 Manual transmission off. 1 Manual transmission enabled.
6	TSE0	Both D+ and D- are low. There is no effect if TEN=0.
5	TD	This bit transmits a USB J or K state on the USB bus. There is no effect if TEN=0 or TSE0=1. 0 Force USB K state. 1 Force USB J state.
0	RD	This read only bit gives the state of the USB differential receiver. 0 D+ < D- or D+ = D- = 0. 1 D+ > D-.

21.3.10 USBIO_CR1

USB I/O Control Register 1

Individual Register Names and Addresses:

0,35h

USBIO_CR1 : 0,35h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	R : 1	R : 1
Bit Name	IOMode	Drive Mode	DPI	DMI	PS2PUEN	USBPUEN	DPO	DMO

This register is a USBIO manual control register 1.

For additional information, refer to the [Register Definitions on page 171](#) in the Full-Speed USB chapter.

Bit	Name	Description
7	IOMode	This bit allows the configuration of D+ and D- pins.
6	Drive Mode	This bit configures D+ and D- pins for either CMOS drive or open drain drive.
5	DPI	This bit drives the D+ pin if IOMode=1.
4	DMI	This pin drives the D- pin if IOMode=1.
3	PS2PUEN	This bit controls the connection of the two internal 5 k Ω pull up resistors to the D+ and D- pins.
2	USBPUEN	This bit controls the connection of the internal 1.5 k Ω pull up resistor on the D+ pin.
1	DPO	This read only bit provides the D+ pin status.
0	DMO	This read only bit provides the D- pin status.

21.3.11 EP0_CR

Endpoint 0 Control Register

Individual Register Names and Addresses:

EP0_CR : 0,36h

	7	6	5	4	3	2	1	0
Access : POR	RC : 0	RC : 0	RC : 0	RC : 0				RW : 0
Bit Name	Setup Received	IN Received	OUT Received	ACK'ed Transaction				Mode[3:0]

This register is an endpoint 0 control register.

For additional information, refer to the [Register Definitions on page 171](#) in the Full-Speed USB chapter.

Bit	Name	Description
7	Setup Received	When set, this bit indicates a valid setup packet was received and ACK'ed.
6	IN Received	When set, this bit indicates a valid IN packet was received.
5	OUT Received	When set, this bit indicates an OUT packet was received.
4	ACK'ed Transaction	When set, this bit indicates a valid OUT packet has been received and ACK'ed.
3:0	Mode[3:0]	The mode bits control how the USB SIE responds to traffic and how the USB SIE changes the mode of that endpoint as a result of host packets to the endpoint.

21.3.12 EP0_CNT

Endpoint 0 Count Register

Individual Register Names and Addresses:

0,37h

EP0_CNT : 0,37h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RC : 0					RW : 0	
Bit Name	Data Toggle	Data Valid					Byte Count[3:0]	

The Endpoint 0 Count register (EP0_CNT) configures endpoint 0.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 171](#) in the Full-Speed USB chapter.

Bit	Name	Description
7	Data Toggle	This bit selects the data packet's toggle state.
6	Data Valid	This bit indicates whether there were errors in OUT or setup transactions.
3:0	Byte Count[3:0]	These bits indicate the number of data bytes in a transaction.

21.3.13 EP0_DRx

Endpoint 0 Data Registers

Individual Register Names and Addresses:

0,38h

EP0_DR0 : 0,38h
EP0_DR4 : 0,3Ch

EP0_DR1 : 0,39h
EP0_DR5 : 0,3Dh

EP0_DR2 : 0,3Ah
EP0_DR6 : 0,3Eh

EP0_DR3 : 0,3Bh
EP0_DR7 : 0,3Fh

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

These registers are endpoint 0 data registers.

For additional information, refer to the [Register Definitions on page 171](#) in the Full-Speed USB chapter.

Bit	Name	Description
-----	------	-------------

7:0	Data Byte[7:0]	These registers are shared for both transmit and receive.
-----	----------------	---

21.3.14 EPx_CNT0

Endpoint Count 0 Registers

Individual Register Names and Addresses:

EP1_CNT0 : 0,40h EP2_CNT0 : 0,42h EP3_CNT0 : 0,44h EP4_CNT0 : 0,46h
 EP5_CNT0 : 0,48h EP6_CNT0 : 0,4Ah EP7_CNT0 : 0,4Ch EP8_CNT0 : 0,4Eh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RC : 0						# : 0
Bit Name	Data Toggle	Data Valid						Count MSB

These registers are endpoint count 0 registers.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 171](#) in the Full-Speed USB chapter.

Bit	Name	Description
7	Data Toggle	This bit selects the data packet's toggle state.
6	Data Valid	This bit is used for OUT transactions only and is read only.
0	Count MSB	This bit is the 1 MSb of a 9-bit counter.

21.3.15 EPx_CNT1

Endpoint Count 1 Registers

Individual Register Names and Addresses:

EP1_CNT1 : 0,41h EP2_CNT1 : 0,43h EP3_CNT1 : 0,45h EP4_CNT1 : 0,47h
EP5_CNT1 : 0,49h EP6_CNT1 : 0,4Bh EP7_CNT1 : 0,4Dh EP8_CNT1 : 0,4Fh

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

These registers are endpoint count 1 registers.

For additional information, refer to the [Register Definitions on page 171](#) in the Full-Speed USB chapter.

Bit	Name	Description
7:0	Data Count[7:0]	These bits are the eight LSb of a 9-bit counter. The MSb is the Count MSb of the EPx_CNT0 register.

21.3.16 PMAx_DR

PSoC Memory Arbiter Data Registers

Individual Register Names and Addresses:

0,58h

PMA0_DR : 0,58h	PMA1_DR : 0,59h	PMA2_DR : 0,5Ah	PMA3_DR : 0,5Bh
PMA4_DR : 0,5Ch	PMA5_DR : 0,5Dh	PMA6_DR : 0,5Eh	PMA7_DR : 0,5Fh
PMA8_DR : 0,64h	PMA9_DR : 0,65h	PMA10_DR : 0,66h	PMA11_DR : 0,67h
PMA12_DR : 0,68h	PMA13_DR : 0,69h	PMA14_DR : 0,6Ah	PMA15_DR : 0,6Bh

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

These registers are PSoC Memory Arbiter write address registers.

For additional information, refer to the [Register Definitions on page 171](#) in the Full-Speed USB chapter.

Bit	Name	Description
7:0	Data Byte[7:0]	When the M8C writes to this register, the PMA registers the byte and then stores the value at the address in SRAM indicated by the PMAx_WA register.

21.3.17 AMUX_CFG

Analog Mux Configuration Register

Individual Register Names and Addresses:

AMUX_CFG : 0,61h

	7	6	5	4	3	2	1	0
Access : POR			RW : 0			RW : 0		RW : 0
Bit Name			PRX_MODE			ICAPEN[1:0]		INTCAP[1:0]

This register is used to configure the integration capacitor pin connections to the analog global bus.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 100](#) in the I/O Analog Mux chapter.

Bits	Name	Description
5	PRX_MODE	When this bit is set to 1, CS_CLK from TrueTouch counter logic block toggles regardless of the "EN" bit setting in CS_CR0 register. When this bit is 0 CS_CLK is gated by "EN" bit in CS_CR0 register. This bit is typically used in Proximity detection mode.
3:2	ICAPEN[1:0]	Bits connect internal capacitance to the analog global bus. 00b No capacitance 01b Approximately 25 pF connected 10b Approximately 50 pF connected 11b Approximately 100 pF connected
1:0	INTCAP[1:0]	Select pins to enable connection of external integration capacitor in the charge integration mode. 00b Neither P0[3] or P0[1] enabled 01b P0[1] pin enabled 10b P0[3] pin enabled 11b Both P0[3] and P0[1] pins enabled

21.3.18 CMP_RDC

Comparator Read/Clear Register

Individual Register Names and Addresses:

CMP_RDC : 0,78h

	7	6	5	4	3	2	1	0
Access : POR			R : 0	R : 0			RC : 0	RC : 0
Bit Name			CMP1D	CMP0D			CMP1L	CMP0L

This register is used to read the state of the comparator data signal and the latched state of the comparator.

In the table above, reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 103](#) in the Comparators chapter.

Bit	Name	Description
5	CMP1D	Read only bit that returns the dynamically changing state of comparator 1. This bit reads zero whenever the comparator is disabled.
4	CMP0D	Read only bit that returns the dynamically changing state of comparator 0. This bit reads zero whenever the comparator is disabled.
1	CMP1L	This bit reads the latch output for comparator 1. It is cleared by either a write of '0' to this bit, or by a rising edge of the comparator 0 LUT, depending on the state of the CRST1 bit in the CMP_CR1 register.
0	CMP0L	This bit reads the latch output for comparator 0. It is cleared by either a write of '0' to this bit, or by a rising edge of the comparator 1 LUT, depending on the state of the CRST0 bit in the CMP_CR1 register.

21.3.19 CMP_MUX

Comparator Multiplexer Register

Individual Register Names and Addresses:

CMP_MUX : 0,79h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0		RW : 0		RW : 0		RW : 0	
Bit Name	INP1[1:0]		INN1[1:0]		INP0[1:0]		INN0[1:0]	

This register contains control bits for input selection of comparators 0 and 1.

For additional information, refer to the [Register Definitions on page 103](#) in the Comparators chapter.

Bit	Name	Description
7:6	INP1[1:0]	Comparator 1 Positive Input Select 00b Analog Global Mux Bus 01b Reserved 10b P0[1] pin 11b P0[3] pin
5:4	INN1[1:0]	Comparator 1 Negative Input Select 00b VREF (1.0V) 01b Ref Lo (approximately 0.6V) 10b Ref Hi (approximately 1.2V) 11b Reserved
3:2	INP0[1:0]	Comparator 0 Positive Input Select 00b Analog Global Mux Bus 01b Reserved 10b P0[1] pin 11b P0[3] pin
1:0	INN0[1:0]	Comparator 0 Negative Input Select 00b VREF (1.0V) 01b Ref Lo (approximately 0.6V) 10b Ref Hi (approximately 1.2V) 11b Reserved

21.3.20 CMP_CR0

Comparator Control Register 0

Individual Register Names and Addresses:

0,7Ah

CMP_CR0 : 0,7Ah

	7	6	5	4	3	2	1	0
Access : POR				RW : 0				RW : 0
Bit Name				CMP1EN				CMP0EN

This register is used to enable and configure the input range of the comparators.

In the table above, reserved bits are grayed table cells and are not described in the bit description section below. Always write reserved bits with a value of '0'. For additional information, refer to the [Register Definitions on page 103](#) in the Comparators chapter.

Bit	Name	Description
4	CMP1EN	0 Comparator 1 disabled, powered off. 1 Comparator 1 enabled.
0	CMP0EN	0 Comparator 0 disabled, powered off. 1 Comparator 0 enabled.

21.3.21 CMP_CR1

Comparator Control Register 1

Individual Register Names and Addresses:

CMP_CR1 : 0,7Bh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	CINT1	CPIN1	CRST1	CDS1	CINT0	CPIN0	CRST0	CDS0

This register is used to configure the comparator output options.

For additional information, refer to the [Register Definitions on page 103](#) in the Comparators chapter.

Bit	Name	Description
7	CINT1	This bit selects comparator 1 for input to the analog interrupt. Note that if both CINT1 and CINT0 are set high, a rising edge on either comparator output may cause an interrupt. 0 Comparator 1 does not connect to the analog interrupt. 1 Comparator 1 connects to the analog interrupt. A rising edge asserts that interrupt, if it is enabled in the INT_MSK0 register.
6	CPIN1	This bit selects the comparator 1 signal for possible connection to the GPIO pin. Connection to the pin also depends on the configuration of the OUT_P1 register. 0 Select comparator 1 LUT output. 1 Select comparator 1 Latch output.
5	CRST1	This bit selects the source for resetting the comparator 1 latch. 0 Reset by writing a '0' to the CMP_RDC register's CMP1L bit. 1 Reset by rising edge of comparator 0 LUT output.
4	CDS1	This bit selects the data output for the comparator 1 channel, for routing to the capacitive sense logic and comparator 1 interrupt. 0 Select the comparator 1 LUT output. 1 Select the comparator 1 latch output.
3	CINT0	This bit selects comparator 0 for input to the analog interrupt. Note that if both CINT1 and CINT0 are set high, a rising edge on either comparator output may cause an interrupt. 0 Comparator 0 does not connect to the analog interrupt. 1 Comparator 0 connects to the analog interrupt. A rising edge asserts that interrupt, if it is enabled in the INT_MSK0 register.
2	CPIN0	This bit selects the comparator 0 signal for possible connection to the GPIO pin. Connection to the pin also depends on the configuration of the OUT_P1 register. 0 Select comparator 0 LUT output. 1 Select comparator 0 Latch output.
1	CRST0	This bit selects the source for resetting the comparator 0 latch. 0 Reset by writing a '0' to the CMP_RDC register's CMP0L bit. 1 Reset by rising edge of comparator 1 LUT output.

(continued on next page)

21.3.21 CMP_CR1 (continued)

0	CDS0	This bit selects the data output for the comparator 0 channel, for routing to the capacitive sense logic and comparator 0 interrupt. 0 Select the comparator 0 LUT output. 1 Select the comparator 0 latch output.
---	-------------	--

21.3.22 CMP_LUT

Comparator LUT Register

Individual Register Names and Addresses:

0,7Ch

CMP_LUT : 0,7Ch

	7	6	5	4	3	2	1	0
Access : POR	RW : 0				RW : 0			
Bit Name	LUT1[3:0]				LUT0[3:0]			

This register is used to select the logic function.

For additional information, refer to the [Register Definitions on page 103](#) in the Comparators chapter.

Bits	Name	Description
7:4	LUT1[3:0]	Select 1 of 16 logic functions for output of comparator bus 1. A=Comp1 output, B=Comp0 output. Function 0h FALSE 1h A AND B 2h A AND $\overline{B}$ 3h $\overline{A}$ 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 $\overline{A}$ OR $\overline{B}$ Ch $\overline{A}$ Dh $\overline{A}$ OR B Eh A NAND B Fh TRUE
3:0	LUT0[3:0]	Select 1 of 16 logic functions for output of comparator bus 0. A=Comp0 output, B=Comp1 output. Function 0h FALSE 1h A AND B 2h A AND $\overline{B}$ 3h $\overline{A}$ 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 $\overline{A}$ OR $\overline{B}$ Ch $\overline{A}$ Dh $\overline{A}$ OR B Eh A NAND B Fh TRUE

21.3.23 CS_CR0

TrueTouch Control Register 0

Individual Register Names and Addresses:

CS_CR0 : 0,A0h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	CSOUT[1:0]	CSD_PRSCCLK	CSD_CS_CLK	CSD_MODE	MODE[1:0]	EN		

This register controls the operation of the TrueTouch counters.

Do not write bits [7:1] while the block is enabled. For additional information, refer to the [Register Definitions on page 92](#) in the TrueTouch Module chapter.

Bit	Name	Description
7:6	CSOUT[1:0]	These bits select between a number of TrueTouch signals that can be driven to an output pin. 00b IN. 01b CS_INT. 10b COL. 11b COH.
5	CSD_PRSCCLK	This bit selects between IMO-P or the PRS output as a clock source to drive the main capacitor switch. 0 Select IMO-P. 1 Select PRS Output.
4	CSD_CS_CLK	This bit selects between IMO or IMO-P for the TrueTouch counters to work. Depending on this bit selection either IMO or IMO-P is sent as the source clock to the clock dividers which generate CS_CLK as shown in Figure 11-13 on page 91 . 0 Select IMO. 1 Select IMO-P.
3	CSD_MODE	This bit enables the CSD mode. When this bit is enabled, the TIMER1 block works on IMO-P (pre-scaled IMO) clock. This is also an enable for TrueTouch counters to toggle. Note: Once the CSD_MODE bit is enabled, the IMO-P clock is a free running divider clock that cannot be stopped and re-started. The IMO-P and the CPU clock are both derived from the IMO clock but the phase relationship between them is nondeterministic. 0 Disable CSD mode. Programmable Timer1 works on either CPUCLK/CLK32, (depends on CLKSEL bit selection in PT1_CFG (0, B3h) register). 1 Enable CSD mode. When this bit is set to 1, Programmable Timer1 works on IMO-P.
2:1	MODE[1:0]	TrueTouch Counter Mode. 00b Event Mode. Start in enable, stop on interrupt event. 01b Pulse Width Mode. Start on positive edge of next input. Stop on negative edge of input. 10b Period Mode. Start on positive edge of input. Stop on next positive edge of input. 11b Start in enable, continuous operation until disable.
0	EN	0 Counting is stopped and all counter values are reset to zero. 1 Counters are enabled for counting.

21.3.24 CS_CR1

TrueTouch Control Register 1

Individual Register Names and Addresses:

CS_CR1 : 0,A1h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	CHAIN	CLKSEL[1:0]	RLOCK	INV	INSEL[2:0]			

This register contains additional TrueTouch system control options.

Never write to this register while the block is enabled. For additional information, refer to the [Register Definitions on page 92](#) in the TrueTouch Module chapter.

Bit	Name	Description
7	CHAIN	Counter chain control. 0 8-bit high/low counters operate independently. 1 High/low counters operate as a 16-bit synchronous block.
6:5	CLKSEL[1:0]	TrueTouch clock (CSCLK) selection. 00b IMO. 01b IMO/2. 10b IMO/4. 11b IMO/8.
4	RLOCK	Relaxation oscillator clock (RLO) select. 0 High byte counter runs on the selected IMO-based frequency. 1 High byte counter runs on the RLO clock frequency.
3	INV	Input invert. 0 Selected input is not inverted. 1 Selected input is inverted.
2:0	INSEL[2:0]	Input selection. 000b Comparator 0. 001b ILO. 010b Comparator 1. 011b RLO Timer Terminal Count. 100b Interval Timer. 101b RLO Timer IRQ. 110b Analog Global Mux Bus. 111b '0'.

21.3.25 CS_CR2

TrueTouch Control Register 2

Individual Register Names and Addresses:

CS_CR2 : 0,A2h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	IRANGE	IDACDIR	IDAC_EN	CIN_EN	PXD_EN	CIP_EN	RO_EN	

This register contains additional TrueTouch system control options.

For additional information, refer to the [Register Definitions on page 92](#) in the TrueTouch Module chapter.

Bit	Name	Description
7:6	IRANGE	Bits scale the IDAC current output. The IDAC_D register sets the base current in the IDAC. 00b IDAC output scaled to 1X range. 01b IDAC output scaled to 2X range. 10b IDAC output scaled to 4X range. 11b IDAC output scaled to 8X range.
5	IDACDIR	Bit determines the source/sink state of the IDAC when enabled (IDAC_EN = 1 or PXD_EN = 1 or CSD_MODE = 1). 0 IDAC sources current to analog global bus. 1 IDAC sinks current from analog global bus.
4	IDAC_EN	Bit provides manual connection of the IDAC to the analog global bus. The IDAC is automatically connected when RO_EN = 1 or PXD_EN = 1. 0 No manual connection. 1 IDAC is connected to analog global bus.
3	CIN_EN	0 Negative charge integration disabled. 1 Negative charge integration enabled. Selected sense pin(s) alternately connect to the analog global bus and ground. Clock rate is selected by the CLKSEL bits in the CS_CR1 register.
2	PXD_EN	0 No clock to I/O pins. 1 Enabled pins switch between ground and the analog global bus. Clock rate selected by the CLKSEL bits in the CS_CR1 register. Selected clock drives TrueTouch timer.
1	CIP_EN	0 Positive charge integration disabled. 1 Positive charge integration enabled. Reference buffer and integration capacitor pins alternately connect to analog global bus. Clock rate selected by the CLKSEL bits in the CS_CR1 register.
0	RO_EN	0 Relaxation oscillator disabled. 1 Relaxation oscillator enabled. Charging currents are set by the IRANGE bits and the IDAC_D register value.

21.3.26 CS_CR3

TrueTouch Control Register 3

Individual Register Names and Addresses:

0,A3h

CS_CR3 : 0,A3h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name		REFMUX	REFMODE	REF_EN	LPFilt[1:0]		LPF_EN[1:0]	

This register contains control bits primarily for the proximity detection algorithm.

For additional information, refer to the [Register Definitions on page 92](#) in the TrueTouch Module chapter.

Bit	Name	Description
6	REFMUX	This bit selects the reference voltage for the input of the reference buffer. 0 Select REFHI (1.2V). 1 Select VREF (1.0V).
5	REFMODE	This bit allows manual connection of the reference buffer output to the analog global bus. If either CI_EN = 1 or RO_EN = 1 in the CS_CR2 register, this bit has no effect (reference buffer connection is off or controlled by other settings). 0 No connection. 1 Reference buffer connected to the analog global bus.
4	REF_EN	This bit enables the reference buffer to drive the analog global bus. 0 Reference buffer is disabled, powered down. 1 Reference buffer is enabled. Connection to the analog global bus is controlled by the REF-MODE bit in this register, and by the CI_EN and RO_EN bits in the CS_CR2 register.
3:2	LPFilt[1:0]	Low pass filter approximate time constant. 00b 1 μ s 01b 2 μ s 10b 5 μ s 11b 10 μ s
1:0	LPF_EN[1:0]	Enable this bit for the low pass filter. 00b No connection of either comparator channel to low pass filter. 01b Connect comparator channel 0 through the low pass filter. 10b Connect comparator channel 1 through the low pass filter. 11b Connect both comparator channel inputs together, and through the low pass filter.

21.3.27 CS_CNTL

TrueTouch Counter Low Byte Register

Individual Register Names and Addresses:

0,A4h

CS_CNTL : 0,A4h

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

This register contains the current count for the low byte counter and is read only.

For additional information, refer to the [Register Definitions on page 92](#) in the TrueTouch Module chapter.

Bit	Name	Description
7:0	Data[7:0]	On a read of this register, the current count is returned. It may only be read when the counter is stopped. Note The counter must be stopped by the configured event. When the counter is disabled, the count is reset to 00h.

21.3.28 CS_CNTH

TrueTouch Counter High Byte Register

Individual Register Names and Addresses:

CS_CNTH : 0,A5h

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

This register contains the current count value for the high byte counter and is read only.

For additional information, refer to the [Register Definitions on page 92](#) in the TrueTouch Module chapter.

Bit	Name	Description
7:0	Data[7:0]	On a read of this register, the current count is returned. It is only read when the counter is stopped. Note The counter must be stopped by the configured event. When the counter is disabled, the count is reset to 00h.

21.3.29 CS_STAT

TrueTouch Status Register

Individual Register Names and Addresses:

0,A6h

CS_STA : 0,A6h

T

	7	6	5	4	3	2	1	0
Access : POR	RC : 0	RC : 0	RC : 0	RC : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	INS	COLS	COHS	PPS	INM	COLM	COHM	PPM

This register controls the TrueTouch counter options.

Never modify the interrupt mask bits while the block is enabled. If a modification to bits 3 to 0 is necessary while the block is enabled, then pay close attention to ensure that the status bits 7 to 4, are not accidentally cleared. You do this by writing a '1' to all of the status bits when writing to the mask bits. For additional information, refer to the [Register Definitions on page 92](#) in the TrueTouch Module chapter.

Bit	Name	Description
7	INS	Input Status. 0 No event detected. 1 A rising edge on the selected input was detected. Cleared by writing a '0' to this bit.
6	COLS	Counter Carry Out Low Status. 0 No event detected. 1 A carry out from low byte counter was detected. Cleared by writing a '0' back to this bit.
5	COHS	Counter Carry Out High Status. 0 No event detected. 1 A carry out from high byte counter was detected. Cleared by writing a '0' back to this bit.
4	PPS	Pulse Width/Period Measurement Status. 0 No event detected. 1 A pulse width or period measurement was completed. Cleared by writing a '0' back to this bit.
3	INM	Input Interrupt/Mask. 0 Disabled. 1 Input event is enabled to assert the block interrupt.
2	COLM	Counter Carry Out Low Interrupt Mask. 0 Disabled. 1 Counter carry out low is enabled to assert the block interrupt.
1	COHM	Counter Carry Out High Interrupt Mask. 0 Disabled. 1 Counter carry out high is enabled to assert the block interrupt.
0	PPM	Pulse Width/Period Measurement Interrupt Mask. 0 Disabled. 1 Completion of a pulse width or period measurement is enabled to assert the block interrupt.

21.3.30 CS_TIMER

TrueTouch Timer Register

Individual Register Names and Addresses:

CS_TIMER : 0,A7h

	7	6	5	4	3	2	1	0
Access : POR								
Bit Name								

This register sets the timer count value.

For additional information, refer to the [Register Definitions on page 92](#) in the TrueTouch Module chapter.

Bit	Name	Description
6:0	Timer Count Value[6:0]	Holds the timer count value.

21.3.31 CS_SLEW

TrueTouch Slew Control Register

Individual Register Names and Addresses:

CS_SLEW : 0,A8h

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

This register enables and controls a fast slewing mode for the relaxation oscillator.

For additional information, refer to the [Register Definitions on page 92](#) in the TrueTouch Module chapter.

Bit	Name	Description
7:1	FastSlew[6:0]	This 7-bit value sets a counter, clocked at the IMO frequency. While the counter is counting down from this value, the relaxation oscillator edge slews at the maximum gain setting. During this interval, the IRANGE bits in the CS_CR2 register are internally set to maximum (11b). At the end of the interval, the user-defined IRANGE level is restored so that the relaxation oscillator continues slewing with a slower edge rate to the target voltage threshold. If the FS_EN bit is low, the FastSlew setting has no effect. After each edge of the relaxation oscillator, the counter is re-loaded and the fast slewing interval re-occurs, followed by the slower edge rate at the end of the count down. Note that the IRANGE bits in the CS_CR2 register always read the user-defined setting. Because the IRANGE value is forced to maximum during this interval, the increase in the edge rate can be 1X, 2X, 4X, or 8X, depending on the programmed value of the IRANGE bits. 0000000b No fast edge rate interval. 0000001b Minimum fast edge rate interval (1 IMO period). ... 1111111b Maximum fast edge rate interval (127 IMO period).
0	FS_EN	Enable bit for the Fast Slew mode. 0 Fast slew mode disabled. 1 Fast slew mode enabled. After each relaxation oscillator transition, the relaxation oscillator runs with a higher current for a time controlled by the FastSlew bits.

21.3.32 PRS_CR

Pseudo Ransom Sequence and Prescaler Control Register

Individual Register Names and Addresses:

PRS_CR : 0,A9h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0'	RW : 0'	RW : 0	RW : 0	RW : 0		
Bit Name	CS_CLK_OUT	CS_CLK_INV	PRS_12BIT	PRS_EN	PRESCALE-BYP	PRESCALE_CLK_DIV[2:0]		

This register controls the Prescaler and Pseudo Random Sequence generator output. For additional information, refer to the [Register Definitions on page 92](#) in the TrueTouch Module chapter.

Bit	Name	Description
7	CS_CLK_OUT	This bit selects the TrueTouch clock (inversion or non-inversion depending on bit 6 setting) to be routed onto primary pin depending on the OUT_P1/OUT_P0 register selection. 1 Route prescaler output or PRS output depending on CSD_PRSClk in CS_CR0 when in CSD mode, or route clock based on CLKSEL bits in CS_CR1 register in normal TrueTouch mode to primary pins P1[2]/P0[7] depending on OUT_P1/OUT_P0 bit selections instead of normal CSOUT[selections. See OUT_P1/OUT_P0 registers for details. 0 Use CSOUT [1:0] selections as normal.
6	CS_CLK_INV	This bit allows you to route either TrueTouch clock or inversion of the TrueTouch clock onto pin. 1 Inverted TrueTouch clock routes to pin (see bit 7). 0 Non-inverted TrueTouch clock routes to pin (see bit 7).
5	PRS_12BIT	This bit allows selection between 8-bit PRS or 12-bit PRS output. 0 MSB of 8-bit PRS is sent out. 1 MSB of 12-bit PRS is sent out
4	PRS_EN	This bit is used to enable or disable the PRS block. 0 PRS is disabled. PRS block output is '0'. 1 PRS is enabled and the bit 5 decides whether MSB of 12-bit PRS is sent out or MSB of 8-bit PRS is sent out.
3	PRESCALEBYP	This bit is used to bypass the prescaler and pass the input clock undivided onto the output. The output of the prescaler feeds the clock input to the PRS block. 0 Divided clock is sent out of prescaler depending on bit [2:0] setting. 1 Incoming IMO clock is sent out of prescaler without any division.
2:0	PRESCALE_CLK_DIV[2:0]	These bits allow for one of eight frequencies of incoming IMO clock to be fed as input to PRS. 000 Divides the input IMO clock by 2. 001 Divides the input IMO clock by 4. 010 Divides the input IMO clock by 8. 011 Divides the input IMO clock by 16. 100 Divides the input IMO clock by 32. 101 Divides the input IMO clock by 64. 110 Divides the input IMO clock by 128. 111 Divides the input IMO clock by 256.

21.3.33 PT0_CFG

Programmable Timer 0 Configuration Register

Individual Register Names and Addresses:

0,B0h

PT0_CFG : 0,B0h

	7	6	5	4	3	2	1	0
Access : POR						RW : 0	RW : 0	RW : 0
Bit Name						CLKSEL	One Shot	START

This register configures the programmable timer 0.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 163](#) in the Programmable Timer chapter.

Bit	Name	Description
2	CLKSEL	This bit determines if the timer runs on the 32 kHz clock or CPU clock. If the bit is set to 1'b1, the timer runs on the CPU clock, otherwise, the timer runs on the 32 kHz clock.
1	One Shot	0 Continuous count mode. Timer reloads the count value from the data registers upon each terminal count, and continues counting. 1 One-shot mode. Timer goes through one complete count period and then stops. Upon completion, the START bit in this register is cleared.
0	START	0 Timer held in reset. 1 Timer counts down from a full count determined from its data registers (PT_DATA1, PT_DATA0). When complete, it either stops or reloads and continues, based on the One Shot bit in this register.

21.3.34 PTx_DATA1

Programmable Timers Data Register 1

Individual Register Names and Addresses:

PT0_DATA1 : 0,B1h PT1_DATA1 : 0,B4h PT2_DATA : 0,B7h
1

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

These registers hold the eight bits of the programmable timer's count value for the device.

For additional information, refer to the [Register Definitions on page 163](#) in the Programmable Timer chapter.

Bit	Name	Description
7:0	DATA[7:0]	This is the upper byte of a 16-bit timer. The lower byte is in the corresponding PTx_DATA0 register.

21.3.35 PTx_DATA0

Programmable Timers Data Register 0

Individual Register Names and Addresses:

PT0_DATA : 0,B2h
0

PT1_DATA0 : 0,B5h

PT2_DATA0 : 0,B8h

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

These registers provide the programmable timer with its lower eight bits of the count value.

For additional information, refer to the [Register Definitions on page 163](#) in the Programmable Timer chapter.

Bit	Name	Description
7:0	DATA[7:0]	This is the lower byte of a 16-bit timer. The upper byte is in the corresponding PTxDATA1 register.

21.3.36 PT1_CFG

Programmable Timer 1 Configuration Register

Individual Register Names and Addresses:

PT1_CFG : 0,B3h

	7	6	5	4	3	2	1	0
Access : POR						RW : 0	RW : 0	RW : 0
Bit Name						CLKSEL	One Shot	START

This register configures the programmable timer 1.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 163](#) in the Programmable Timer chapter.

Bit	Name	Description
2	CLKSEL	This bit determines if the timer runs on the 32 kHz clock or CPU clock. If the bit is set to 1'b1, the timer runs on the CPU clock, otherwise, the timer runs on the 32 kHz clock.
1	One Shot	0 Continuous count mode. Timer reloads the count value from the data registers upon each terminal count, and continues counting. 1 One-shot mode. Timer goes through one complete count period and then stops. Upon completion, the START bit in this register is cleared.
0	START	0 Timer held in reset. 1 Timer counts down from a full count determined from its data registers (PT_DATA1, PT_DATA0). When complete, it either stops or reloads and continues, based on the One Shot bit in this register.

21.3.37 PT2_CFG

Programmable Timer 2 Configuration Register

Individual Register Names and Addresses:

0,B6h

PT2_CFG : 0,B6h

	7	6	5	4	3	2	1	0
Access : POR						RW : 0	RW : 0	RW : 0
Bit Name						CLKSEL	One Shot	START

This register configures the programmable timer 2.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 163](#) in the Programmable Timer chapter.

Bit	Name	Description
2	CLKSEL	This bit determines if the timer runs on the 32 kHz clock or CPU clock. If the bit is set to 1'b1, the timer runs on the CPU clock, otherwise, the timer runs on the 32 kHz clock.
1	One Shot	0 Continuous count mode. Timer reloads the count value from the data registers upon each terminal count, and continues counting. 1 One-shot mode. Timer goes through one complete count period and then stops. Upon completion, the START bit in this register is cleared.
0	START	0 Timer held in reset. 1 Timer counts down from a full count determined from its data registers (PT_DATA1, PT_DATA0). When complete, it either stops or reloads and continues, based on the One Shot bit in this register.

21.3.38 I2C_XCFG

I²C Extended Configuration Register

Individual Register Names and Addresses:

0,C8h

I2C_XCFG: 0,C8h

	7	6	5	4	3	2	1	0
Access : POR					RW : 0		RW : 0	RW : 0
Bit Name					No BC Int		Buffer Mode	HW Addr En

This register configures enhanced features. The Enable bit (bit 0) of the [I2C_CFG](#) (0,D6h) register should be set to 1'b1 for the I2C enhanced features to work.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Always write reserved bits with a value of '0'. For additional information, refer to the [Register Definitions on page 122](#) in the I2C Slave chapter.

Bit	Name	Description									
3	No BC Int	In compatibility mode, every received or transmitted byte generates a Byte Complete Interrupt. This is also true in buffered mode regardless of whether the bus is stalled or not. Note When this bit is set to a '1', A BC interrupt is not enabled for any data byte that is automatically ACK'ed (i.e., does not require the bus to stall). A BC interrupt is always generated upon any stall so the CPU takes the appropriate action. When the bit is set, it is possible to implement packet transfers without CPU intervention by enabling an interrupt upon the Stop detect.									
1	Buffer Mode	This bit determines the operation mode of the enhanced buffer module. The following table describes the available modes. <table data-bbox="480 1297 1123 1663"> <tr> <th>Buffer Mode</th><th>Name</th><th>Description</th></tr> <tr> <td>0</td><td>Compatible</td><td>There is no buffering in the default compatibility mode. The I2C bus is stalled upon every received or transmitted byte, including address bytes. The CPU is required to process the interrupt and write or read the data and status as required to continue the operation.</td></tr> <tr> <td>1</td><td>EZI2C</td><td>The I2C slave appears as a 32-byte RAM interface to an external master. A specific protocol must be followed, in which the master controls the RAM pointer for both read and write operations. The I2C bus is never stalled. Receive ACKs are automatically generated. The CPU is responsible for putting valid data into the RAM for external reads, and for reading received data.</td></tr> </table>	Buffer Mode	Name	Description	0	Compatible	There is no buffering in the default compatibility mode. The I2C bus is stalled upon every received or transmitted byte, including address bytes. The CPU is required to process the interrupt and write or read the data and status as required to continue the operation.	1	EZI2C	The I2C slave appears as a 32-byte RAM interface to an external master. A specific protocol must be followed, in which the master controls the RAM pointer for both read and write operations. The I2C bus is never stalled. Receive ACKs are automatically generated. The CPU is responsible for putting valid data into the RAM for external reads, and for reading received data.
Buffer Mode	Name	Description									
0	Compatible	There is no buffering in the default compatibility mode. The I2C bus is stalled upon every received or transmitted byte, including address bytes. The CPU is required to process the interrupt and write or read the data and status as required to continue the operation.									
1	EZI2C	The I2C slave appears as a 32-byte RAM interface to an external master. A specific protocol must be followed, in which the master controls the RAM pointer for both read and write operations. The I2C bus is never stalled. Receive ACKs are automatically generated. The CPU is responsible for putting valid data into the RAM for external reads, and for reading received data.									
0	HW Addr En	When this bit is set to a '1', hardware address compare is enabled. When enabled, bit 3 in the I2C_SCR register is not set. Upon a compare, the address is automatically ACK'ed, and upon a mismatch, the address is automatically NACK'ed and the hardware reverts to an idle state waiting for the next Start detection. You must configure the compare address in the I2C_ADDR register. When this bit is a '0', bit 3 of the I2C_SCR register is set and the bus stalls, and the received address is available in the I2C_DR register to enable the CPU to do a firmware address compare. The functionality of this bit is independent of the data buffering mode.									

21.3.39 I2C_XSTAT

I²C Extended Status Register

Individual Register Names and Addresses:

0,C0h

0,C0h

I2C_XSTAT : 0,C9h

	7	6	5	4	3	2	1	0
Access : POR							R : 0	R : 0
Bit Name							Dir	Slave Busy

This register reads enhanced feature status.

When the bits of the [I2C_XCFG](#) register are left in their reset state, the block is in compatibility mode and this register is not in use. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Always write reserved bits with a value of '0'. For additional information, refer to the [Register Definitions on page 122](#) in the I2C Slave chapter.

Bit	Name	Description
1	Dir	This bit indicates the direction of the current transfer. A '1' indicates a master read, and a '0' indicates a master write. It is only valid when the Slave Busy bit (bit 0) is set to a '1'.
0	Slave Busy	This bit is set upon a hardware compare and is reset upon the following stop signal. Poll this bit to determine when the slave is busy and the buffer module is being accessed.

21.3.40 I2C_ADDR

I²C Slave Address Register

Individual Register Names and Addresses:

0,CAh

0,00h

I2C_ADDR : 0,CAh

	7	6	5	4	3	2	1	0
Access : POR								
Bit Name								

RW : 00

Slave Address[6:0]

This register holds the slave's 7-bit address.

When hardware address compare mode is not enabled in the [I2C_XCFG](#) register, this register is not in use. In the table above, note that the reserved bit is a grayed table cell and not described in the bit description section below. Always write reserved bits with a value of '0'. For additional information, refer to the [Register Definitions on page 122](#) in the I2C Slave chapter.

Bit	Name	Description
6:0	Slave Address[6:0]	These seven bits hold the slave's own device address.

21.3.41 I2C_BP

I²C Base Address Pointer Register

Individual Register Names and Addresses:

I2C_BP : 0,CBh

	7	6	5	4	3	2	1	0
Access : POR								R : 00
Bit Name								I2C Base Pointer[4:0]

This register contains the base address value of the RAM data buffer and is read only.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Always write reserved bits with a value of '0'. For additional information, refer to the [Register Definitions on page 122](#) in the I2C Slave chapter.

Bit	Name	Description
4:0	I2C Base Pointer[4:0]	In the EZI2C protocol, the first data byte after the slave address transaction in write mode is the base address for subsequent reads and writes and it is transferred directly into this register. If the desired transaction is a master write to the slave, subsequent bytes are written to the RAM buffer starting with this address and auto incremented (see I2C_CP register). In case of a read, a Start or Restart must be issued and the read location starts with this address and again subsequent read addresses are auto incremented as pointed to by the I2C_CP register value. The value of this register is modified only at the beginning of every I2C write transaction. The I2C master must always supply a value for this register in the first byte of data after the slave's address in a given write transaction. If performing reads, the master need not set the value of this register. The current value of this register is also used directly for reads.

21.3.42 I2C_CP

I²C Current Address Pointer Register

Individual Register Names and Addresses: 0,CCh

I2C_CP : 0,CCh

	7	6	5	4	3	2	1	0
Access : POR								R : 00
Bit Name								I2C Current Pointer[4:0]

This register contains the current address value of the RAM data buffer and is read only.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Always write reserved bits with a value of '0'. For additional information, refer to the [Register Definitions on page 122](#) in the I2C Slave chapter.

Bit	Name	Description
4:0	I2C Current Pointer[4:0]	This register is set at the same time and with the same value to which the I2C_BP register is set. After each completed data byte of the current I2C transaction, the value of this register is incremented by one. The value of this register always determines the location that read or write data comes from or is written to. This register is set to the value contained in the I2C_BP Register on every start condition detected in the bus.

21.3.43 CPU_BP

CPU Base Address Pointer Register

Individual Register Names and Addresses: 0,CDh

CPU_BP : 0,CDh

	7	6	5	4	3	2	1	0
Access : POR								
Bit Name								

This register contains the base address value of the RAM data buffer.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Always write reserved bits with a value of '0'. For additional information, refer to the [Register Definitions on page 122](#) in the I2C Slave chapter.

Bit	Name	Description
4:0	CPU Base Pointer[4:0]	This register value is completely controlled by I/O writes by the CPU. Firmware routines must set this register. Similar to the I2C_BP_WR, the value of this register sets the starting address for the data location being written to or read from. When this register is written, the current address pointer CPU_CP is also updated with the same value. The first read or write from/to the I2C_BUF register starts at this address. The location of the data in subsequent read or writes is determined by the CPU_CP register value, which auto increments after each read or write. Firmware makes certain that the slave device always has valid data or the data is read before overwritten.

21.3.44 CPU_CP

CPU Current Address Pointer Register

Individual Register Names and Addresses: 0,CEh

CPU_CP : 0,CEh

	7	6	5	4	3	2	1	0
Access : POR								R : 00
Bit Name								CPU Current Pointer [4:0]

This register is a pointer into the RAM buffer for CPU reads and writes and is read only.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Always write reserved bits with a value of '0'. For additional information, refer to the [Register Definitions on page 122](#) in the I2C Slave chapter.

Bit	Name	Description
4:0	CPU Current Pointer[4:0]	This register is set at the same time and with the same value to which the CPU_BP register is set. Whenever the I2C_BUF register is written to or read from, the CPU_CP automatically increments.

21.3.45 I2C_BUF

I²C Data Buffer Register

Individual Register Names and Addresses: 0,CFh

I2C_BUF : 0,CFh

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

This register is the CPU read/write interface to the data buffer.

For additional information, refer to the [Register Definitions on page 122](#) in the I2C Slave chapter.

Bit	Name	Description
7:0	Data Buffer[7:0]	The I2C Data Buffer Register (I2C_BUF) is the CPU read/write interface to the data buffer. Whenever this register is read, the data at the location pointed to by CPU current pointer (CPU_CP) is returned. Similarly, whenever this register is written, the data is transferred to the buffer and written at the location pointed to by the CPU current pointer (CPU_CP). Whenever this register is read without initializing the RAM contents either through the I2C or CPU interface, no valid value is returned.

21.3.46 CUR_PP

Current Page Pointer Register

Individual Register Names and Addresses:

CUR_PP : 0,D0h

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

This register is used to set the effective SRAM page for normal memory accesses in a multi-SRAM page device. It is only used when a device has more than one SRAM page.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 42](#) in the RAM Paging chapter.

Bit	Name	Description
2:0	Page Bits[2:0]	Bits determine which SRAM page is used for generic SRAM access. See the RAM Paging chapter on page 39 for more information.
	000b	SRAM Page 0
	001b	SRAM Page 1
	010b	SRAM Page 2
	011b	SRAM Page 3
	100b	SRAM Page 4
	101b	SRAM Page 5
	110b	SRAM Page 6
	111b	SRAM Page 7

21.3.47 STK_PP

Stack Page Pointer Register

Individual Register Names and Addresses:

0,D1h

STK_PP : 0,D1h

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

This register is used to set the effective SRAM page for stack memory accesses in a multi-SRAM page PSoC device. It is only used when a device has more than one SRAM page.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 42](#) in the RAM Paging chapter.

Bit	Name	Description
2:0	Page Bits[2:0]	Bits determine which SRAM page is used to hold the stack. See the RAM Paging chapter on page 39 for more information.
	000b	SRAM Page 0
	001b	SRAM Page 1
	010b	SRAM Page 2
	011b	SRAM Page 3
	100b	SRAM Page 4
	101b	SRAM Page 5
	110b	SRAM Page 6
	111b	SRAM Page 7

21.3.48 IDX_PP

Indexed Memory Access Page Pointer Register

Individual Register Names and Addresses:

0,D3h

IDX_PP : 0,D3h

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

This register is used to set the effective SRAM page for indexed memory accesses in a multi-SRAM page PSoC device.

This register is only used when a device has more than one page of SRAM. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 42](#) in the RAM Paging chapter.

Bit	Name	Description
2:0	Page Bits[2:0]	Bits determine which SRAM page an indexed memory access operates on. See the Register Definitions on page 42 for more information on when this register is active.
	000b	SRAM Page 0
	001b	SRAM Page 1
	010b	SRAM Page 2
	011b	SRAM Page 3
	100b	SRAM Page 4
	101b	SRAM Page 5
	110b	SRAM Page 6
	111b	SRAM Page 7

21.3.49 MVR_PP

MVI Read Page Pointer Register

Individual Register Names and Addresses:

0,D4h

MVR_PP : 0,D4h

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

This register is used to set the effective SRAM page for MVI read memory accesses in a multi-SRAM page PSoC device.

This register is only used when a device has more than one page of SRAM. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 42](#) in the RAM Paging chapter.

Bit	Name	Description
2:0	Page Bits[2:0]	Bits determine which SRAM page an MVI Read instruction operates on.
	000b	SRAM Page 0
	001b	SRAM Page 1
	010b	SRAM Page 2
	011b	SRAM Page 3
	100b	SRAM Page 4
	101b	SRAM Page 5
	110b	SRAM Page 6
	111b	SRAM Page 7

21.3.50 MVW_PP

MVI Write Page Pointer Register

Individual Register Names and Addresses: 0,D5h

MVW_PP : 0,D5h

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

This register is used to set the effective SRAM page for MVI write memory accesses in a multi-SRAM page PSoC device.

This register is only used when a device has more than one page of SRAM. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 42](#) in the RAM Paging chapter.

Bit	Name	Description
2:0	Page Bits[2:0]	Bits determine which SRAM page an MVI Write instruction operates on.
	000b	SRAM Page 0
	001b	SRAM Page 1
	010b	SRAM Page 2
	011b	SRAM Page 3
	100b	SRAM Page 4
	101b	SRAM Page 5
	110b	SRAM Page 6
	111b	SRAM Page 7

21.3.51 I2C_CFG

I²C Configuration Register

Individual Register Names and Addresses: 0,D6h

I2C_CFG : 0,D6h

	7	6	5	4	3	2	1	0
Access : POR		RW : 0		RW : 0	RW : 0			RW : 0
Bit Name		PSelect		Stop IE	Clock Rate[1:0]			Enable

This register is used to set the basic operating modes, baud rate, and interrupt selection.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Always write reserved bits with a value of '0'. For additional information, refer to the [Register Definitions on page 122](#) in the I2C Slave chapter.

Bit	Name	Description
6	P Select	I2C Pin Select. 0 P1[5] and P1[7]. 1 P1[0] and P1[1]. Note Read the I2C Slave chapter for a discussion of the side effects of choosing the P1[0] and P1[1] pair of pins.
4	Stop IE	Stop Interrupt Enable. 0 Disabled. 1 Enabled. An interrupt is generated on the detection of a Stop condition.
3:2	Clock Rate[1:0]	00b 100K Standard Mode. 01b 400K Fast Mode. 10b 50K Standard Mode. 11b Reserved..
0	Enable	0 Disabled. 1 Enabled.

21.3.52 I2C_SCR

I²C Status and Control Register

Individual Register Names and Addresses: 0,D7h

I2C_SCR : 0,D7h

	7	6	5	4	3	2	1	0
Access : POR	RC : 0		RC : 0	RW : 0	RC : 0	RW : 0	RC : 0	RC : 0
Bit Name	Bus Error		Stop Status	ACK	Address	Transmit	LRB	Byte Complete

This register is used by the slave to control the flow of data bytes and to keep track of the bus state during a transfer.

Bits in this register are held in reset until one of the enable bits in I2C_CFG is set. In the table above, note that the reserved bit is a grayed table cell and not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 122](#) in the I2C Slave chapter.

Bit	Name	Description
7	Bus Error	0 Status bit. It must be cleared by firmware by writing a '0' to the bit position. It is never cleared by the hardware. 1 A misplaced Start or Stop condition was detected.
5	Stop Status	0 Status bit. It must be cleared by firmware with a write of '0' to the bit position. It is never cleared by the hardware. 1 A Stop condition was detected.
4	ACK	Acknowledge Out. Bit is automatically cleared by hardware upon a Byte Complete event. 0 NACK the last received byte. 1 ACK the last received byte
3	Address	0 Status bit. It must be cleared by firmware with a write of '0' to the bit position. 1 The received byte is a slave address.
2	Transmit	Bit is set by firmware to define the direction of the byte transfer. Any Start detect or a write to the Start or Restart generate bits when operating in master mode also clears the bit. 0 Receive mode. 1 Transmit mode.
1	LRB	Last Received Bit. The value of the 9 th bit in a Transmit sequence, which is the acknowledge bit from the receiver. Any Start detect or a write to the Start or Restart generate bits when operating in master mode also clears the bit. 0 Last transmitted byte was ACK'ed by the receiver. 1 Last transmitted byte was NACK'ed by the receiver.
0	Byte Complete	Transmit/Receive Mode: 0 No completed transmit/receive since last cleared by firmware. Any Start detect or a write to the Start or Restart generate bits when operating in master mode also clears the bit. Transmit Mode: 1 Eight bits of data have been transmitted and an ACK or NACK has been received. Receive Mode: 1 Eight bits of data have been received.

21.3.53 I2C_DR

I²C Data Register

Individual Register Names and Addresses: 0,D8h

I2C_DR : 0,D8h

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

This register provides read/write access to the Shift register.

This register is read only for received data and write only for transmitted data. For additional information, refer to the [Register Definitions on page 122](#) in the I2C Slave chapter.

Bit	Name	Description
7:0	Data[7:0]	Read received data or write data to transmit.

21.3.54 INT_CLR0

Interrupt Clear Register 0

Individual Register Names and Addresses:

INT_CLR0 : 0,DAh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	I2C	Sleep	SPI	GPIO	Timer0	TrueTouch	Analog	V Monitor

This register enables the individual interrupt sources' ability to clear posted interrupts.

When bits in this register are read, a '1' is returned for every bit position that has a corresponding posted interrupt. When bits in this register are written with a '0' and ENSWINT is not set, posted interrupts are cleared at the corresponding bit positions. If there is no posted interrupt, there is no effect. When bits in this register are written with a '1' and ENSWINT is set, an interrupt is posted in the interrupt controller.

For additional information, refer to the [Register Definitions on page 48](#) in the Interrupt Controller chapter.

Bit	Name	Description
7	I2C	Read 0 No posted interrupt for I2C. Read 1 Posted interrupt present for I2C. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for I2C.
6	Sleep	Read 0 No posted interrupt for sleep timer. Read 1 Posted interrupt present for sleep timer. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for sleep timer.
5	SPI	Read 0 No posted interrupt for SPI. Read 1 Posted interrupt present for SPI. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for SPI.
4	GPIO	Read 0 No posted interrupt for general purpose inputs and outputs (GPIO) (pins). Read 1 Posted interrupt present for GPIO (pins). Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for general purpose inputs and outputs (pins).

(continued on next page)

21.3.54 INT_CLR0 (continued)

3	Timer0	Read 0 No posted interrupt for Timer. Read 1 Posted interrupt present for Timer. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for Timer.
2	TrueTouch	Read 0 No posted interrupt for TrueTouch. Read 1 Posted interrupt present for TrueTouch. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for TrueTouch.
1	Analog	Read 0 No posted interrupt for Analog. Read 1 Posted interrupt present for Analog. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for Analog.
0	V Monitor	Read 0 No posted interrupt for Supply Voltage Monitor. Read 1 Posted interrupt present for Supply Voltage Monitor. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for Supply Voltage Monitor.

21.3.55 INT_CLR1

Interrupt Clear Register 1

Individual Register Names and Addresses: 0,DBh

INT_CLR1 : 0,DBh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	Endpoint3	Endpoint2	Endpoint1	Endpoint0	USB_SOF	USB_BUS_RST	Timer2	Timer1

This register is used to enable the individual interrupt sources' ability to clear posted interrupts.

When bits in this register are read, a '1' is returned for every bit position that has a corresponding posted interrupt. When bits in this register are written with a '0' and ENSWINT is not set, posted interrupts are cleared at the corresponding bit positions. If there is no posted interrupt, there is no effect. When bits in this register are written with a '1' and ENSWINT is set, an interrupt is posted in the interrupt controller.

For additional information, refer to the [Register Definitions on page 48](#) in the Interrupt Controller chapter.

Bit	Name	Description
7	Endpoint3	Read 0 No posted interrupt for USB Endpoint3. Read 1 Posted interrupt present for USB Endpoint3. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for USB Endpoint3.
6	Endpoint2	Read 0 No posted interrupt for USB Endpoint2. Read 1 Posted interrupt present for USB Endpoint2. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for USB Endpoint2.
5	Endpoint1	Read 0 No posted interrupt for USB Endpoint1. Read 1 Posted interrupt present for USB Endpoint1. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for USB Endpoint1.

(continued on next page)

21.3.55 INT_CLR1 (continued)

4	Endpoint0	Read 0 No posted interrupt for USB Endpoint0. Read 1 Posted interrupt present for USB Endpoint0. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for USB Endpoint0.
3	USB_SOF	Read 0 No posted interrupt for USB Start of Frame (SOF). Read 1 Posted interrupt present for USB Start of Frame (SOF). Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for USB Start of Frame(SOF).
2	USB_BUS_RST	Read 0 No posted interrupt for USB Bus Reset. Read 1 Posted interrupt present for USB Bus Reset. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for USB Bus Reset.
1	Timer2	Read 0 No posted interrupt for Timer2. Read 1 Posted interrupt present for Timer2. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for Timer2.
0	Timer1	Read 0 No posted interrupt for Timer1. Read 1 Posted interrupt present for Timer1. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for Timer1.

21.3.56 INT_CLR2

Interrupt Clear Register 2

Individual Register Names and Addresses: 0,DCh

INT_CLR2 : 0,DCh

	7	6	5	4	3	2	1	0
Access : RW:00			RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name			USB_WAKE	Endpoint8	Endpoint7	Endpoint6	Endpoint5	Endpoint4

This register is used to enable the individual interrupt sources' ability to clear posted interrupts.

When bits in this register are read, a '1' is returned for every bit position that has a corresponding posted interrupt. When bits in this register are written with a '0' and ENSWINT is not set, posted interrupts are cleared at the corresponding bit positions. If there is no posted interrupt, there is no effect. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. When bits in this register are written with a '1' and ENSWINT is set, an interrupt is posted in the interrupt controller. For additional information, refer to the [Register Definitions on page 48](#) in the Interrupt Controller chapter.

Bit	Name	Description
5	USB_WAKE	Read 0 No posted interrupt for USB Wake. Read 1 Posted interrupt present for USB Wake. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for USB Wake.
4	Endpoint8	Read 0 No posted interrupt for USB Endpoint8. Read 1 Posted interrupt present for USB Endpoint8. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for USB Endpoint8.

(continued on next page)

21.3.56 INT_CLR2 (continued)

3	Endpoint7	Read 0 No posted interrupt for USB Endpoint7. Read 1 Posted interrupt present for USB Endpoint7. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for USB Endpoint7.
2	Endpoint6	Read 0 No posted interrupt for USB Endpoint6. Read 1 Posted interrupt present for USB Endpoint6. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for USB Endpoint6.
1	Endpoint5	Read 0 No posted interrupt for USB Endpoint5. Read 1 Posted interrupt present for USB Endpoint5. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for USB Endpoint5
0	Endpoint4	Read 0 No posted interrupt for USB Endpoint4. Read 1 Posted interrupt present for USB Endpoint4. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt for USB Endpoint4.

21.3.57 INT_MSK2

Interrupt Mask Register 2

Individual Register Names and Addresses:

0,DEh

INT_MSK2 : 0,DEh

	7	6	5	4	3	2	1	0
Access : POR			RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name			USB Wakeup	Endpoint8	Endpoint7	Endpoint6	Endpoint5	Endpoint4

This register enables the individual sources' ability to create pending interrupts.

When an interrupt is masked off, the mask bit is '0'. The interrupt continues to post in the interrupt controller. Clearing the mask bit only prevents a posted interrupt from becoming a pending interrupt. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 48](#) in the Interrupt Controller chapter.

Bit	Name	Description
5	USB Wakeup	0 Mask USB Wakeup interrupt. 1 Unmask USB Wakeup interrupt.
4	Endpoint8	0 Mask USB Endpoint8 interrupt. 1 Unmask USB Endpoint8 interrupt.
3	Endpoint7	0 Mask USB Endpoint7 interrupt. 1 Unmask USB Endpoint7 interrupt.
2	Endpoint6	0 Mask USB Endpoint6 interrupt. 1 Unmask USB Endpoint6 interrupt.
1	Endpoint5	0 Mask USB Endpoint5 interrupt. 1 Unmask USB Endpoint5 interrupt.
0	Endpoint4	0 Mask USB Endpoint4 interrupt. 1 Unmask USB Endpoint4 interrupt.

21.3.58 INT_MSK1

Interrupt Mask Register 1

Individual Register Names and Addresses: 0,DFh

INT_MSK1 : 0,DFh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	Endpoint3	Endpoint2	Endpoint1	Endpoint0	USB SOF	USB Bus Reset	Timer2	Timer1

This register enables the individual sources' ability to create pending interrupts.

When an interrupt is masked off, the mask bit is '0'. The interrupt continues to post in the interrupt controller. Clearing the mask bit only prevents a posted interrupt from becoming a pending interrupt.

For additional information, refer to the [Register Definitions on page 48](#) in the Interrupt Controller chapter.

Bit	Name	Description
7	Endpoint3	0 Mask USB Endpoint3 interrupt. 1 Unmask USB Endpoint3 interrupt.
6	Endpoint2	0 Mask USB Endpoint2 interrupt. 1 Unmask USB Endpoint2 interrupt.
5	Endpoint1	0 Mask USB Endpoint1 interrupt. 1 Unmask USB Endpoint1 interrupt.
4	Endpoint0	0 Mask USB Endpoint0 interrupt. 1 Unmask USB Endpoint0 interrupt.
3	USB SOF	0 Mask USB SOF interrupt. 1 Unmask USB SOF interrupt.
2	USB Bus Reset	0 Mask USB Bus Reset interrupt. 1 Unmask USB Bus Reset interrupt.
1	Timer2	0 Mask Timer2 interrupt. 1 Unmask Timer2 interrupt.
0	Timer1	0 Mask Timer1 interrupt. 1 Unmask Timer1 interrupt.

21.3.59 INT_MSK0

Interrupt Mask Register 0

Individual Register Names and Addresses:

0,E0h

INT_MSK0 : 0,E0h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	I2C	Sleep	SPI	GPIO	Timer0	TrueTouch	Analog	V Monitor

This register enables the individual sources' ability to create pending interrupts.

When an interrupt is masked off, the mask bit is '0'. The interrupt continues to post in the interrupt controller. Clearing the mask bit only prevents a posted interrupt from becoming a pending interrupt. For additional information, refer to the [Register Definitions on page 48](#) in the Interrupt Controller chapter.

Bit	Name	Description
7	I2C	0 Mask I2C interrupt.
		1 Unmask I2C interrupt.
6	Sleep	0 Mask Sleep interrupt.
		1 Unmask Sleep interrupt.
5	SPI	0 Mask SPI interrupt.
		1 Unmask SPI interrupt.
4	GPIO	0 Mask GPIO interrupt.
		1 Unmask GPIO interrupt.
3	Timer0	0 Mask Timer0 interrupt.
		1 Unmask Timer0 interrupt.
2	TrueTouch	0 Mask TrueTouch interrupt.
		1 Unmask TrueTouch interrupt.
1	Analog	0 Mask Analog interrupt.
		1 Unmask Analog interrupt.
0	V Monitor	0 Mask Voltage Monitor interrupt.
		1 Unmask Voltage Monitor interrupt.

21.3.60 INT_SW_EN

Interrupt Software Enable Register

Individual Register Names and Addresses:

INT_SW_EN : 0,E1h

	7	6	5	4	3	2	1	0
Access : POR								RW : 0
Bit Name								ENSWINT

This register is used to enable software interrupts.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 48](#) in the Interrupt Controller chapter.

Bit	Name	Description
0	ENSWINT	0 Disable software interrupts. 1 Enable software interrupts.

21.3.61 INT_VC

Interrupt Vector Clear Register

Individual Register Names and Addresses: 0,E2h

INT_VC : 0,E2h

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

This register returns the next pending interrupt and clears all pending interrupts when written.

For additional information, refer to the [Register Definitions on page 48](#) 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.

21.3.62 RES_WDT

Reset Watchdog Timer Register

Individual Register Names and Addresses:

RES_WDT : 0,E3h

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

This register is used to clear the watchdog timer alone, or clear both the watchdog timer and the sleep timer together.

For additional information, refer to the [Register Definitions on page 77](#) in the Sleep and Watchdog chapter.

Bit	Name	Description
7:0	WDSL_Clear[7:0]	Any write clears the watchdog timer. A write of 38h clears both the watchdog and sleep timers.

21.3.63 CPU_F

M8C Flag Register

Individual Register Names and Addresses:

CPU_F : x,F7h

	7	6	5	4	3	2	1	0
Access : POR	RL : 0	RL : 0	RL : 0			RL : 0	RL : 0	RL : 0
Bit Name	PgMode[1:0]	BINC	XIO			Carry	Zero	GIE

This register provides read access to the M8C flags.

The AND f, expr; OR f, expr; and XOR f, expr flag instructions are used to modify this register. In the table above, note that the reserved bit is a grayed table cell and is not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 32](#) in the M8C chapter and the [Register Definitions on page 48](#) in the Interrupt Controller chapter.

Bit	Name	Description
7:6	PgMode[1:0]	00b Direct Address mode and Indexed Address mode operands are referred to RAM Page 0, regardless of the values of CUR_PP and IDX_PP. Note that this condition prevails upon entry to an Interrupt Service Routine when the CPU_F register is cleared.
		01b Direct Address mode instructions are referred to Page 0. Indexed Address mode instructions are referred to the RAM page specified by the stack page pointer, STK_PP.
		10b Direct Address mode instructions are referred to the RAM page specified by the current page pointer, CUR_PP. Indexed Address mode instructions are referred to the RAM page specified by the index page pointer, IDX_PP.
		11b Direct Address mode instructions are referred to the RAM page specified by the current page pointer, CUR_PP. Indexed Address mode instructions are referred to the RAM page specified by the stack page pointer, STK_PP.
5	BINC	Bit Implemented Not Connected.
4	XIO	0 Normal register address space.
		1 Extended register address space. Primarily used for configuration.
2	Carry	Set by the M8C CPU Core to indicate whether there has been a carry in the previous logical/arithmetic operation.
		0 No carry.
		1 Carry.

(continued on next page)

21.3.63 CPU_F (continued)

1	Zero		Set by the M8C CPU Core to indicate whether there was a zero result in the previous logical/arithmetic operation.
		0	Not equal to zero.
		1	Equal to zero.
0	GIE	0	M8C does not process any interrupts.
		1	Interrupt processing enabled.

21.3.64 IDAC_D

Current DAC Data Register

Individual Register Names and Addresses:

IDAC_D : 0,FDh

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

This register specifies the 8-bit multiplying factor that determines the output IDAC current.

For additional information, refer to the [Register Definitions on page 92](#) in the TrueTouch Module chapter.

Bits	Name	Description
7:0	IDACDATA[7:0]	This is an 8-bit value that selects the number of current units that combine to form the IDAC current. This current then drives the analog mux bus when IDAC mode is enabled. For example, a setting of 80h means that the charging current is 128 current units. The current size also depends on the IRANGE setting in the CS_CR2 register. This setting supplies the charging current for the relaxation oscillator. This current and the external capacitance connected to the analog global bus determines the RO frequency. This register is also used to set the charging current in the proximity detect mode. Step size is approximately 330 nA/bit for default IRANGE state 00b. 00h Smallest current. · · · FFh Largest current.

21.3.65 CPU_SCR1

System Status and Control Register 1

Individual Register Names and Addresses:

CPU_SCR1: x,FEh

	7	6	5	4	3	2	1	0
Access : POR	R : 0				RW : 0			RW : 0
Bit Name	IRESS			SLIMO[1:0]				IRAMDIS

This register is used to convey the status and control of events related to internal resets and watchdog reset.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 137](#) in the System Resets chapter.

Bit	Name	Description
7	IRESS	This bit is read only. 0 Boot phase only executed once. 1 Boot phase occurred multiple times.
4:3	SLIMO[1:0]	These bits set the frequency range for the IMO. Note When changing from the default setting, the corresponding trim value must be loaded into the IMO_TR register for highest frequency accuracy. SLIMO CY8CTMG20x/ CY8CTST200 00 12 01 6 10 24 11 Reserved
0	IRAMDIS	0 SRAM is initialized to 00h after POR, XRES, and WDR. 1 Addresses 03h - D7h of SRAM Page 0 are not modified by WDR.

21.3.66 CPU_SCR0

System Status and Control Register 0

Individual Register Names and Addresses:

CPU_SCR0 : x,FFh

	7	6	5	4	3	2	1	0
Access : POR	R : 0		RC : 0	RC : 1	RW : 0			RW : 0
Bit Name	GIES		WDRS	PORS	Sleep			STOP

This register is used to convey the status and control of events for various functions of a PSoC device.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 137](#) in the System Resets chapter.

Bit	Name	Description
7	GIES	Global Interrupt Enable Status. It is recommended that the user read the Global Interrupt Enable Flag bit from the CPU_F register on page 254 . This bit is read only for GIES. Its use is discouraged, as the Flag register is now readable at address x,F7h (read only).
5	WDRS	Watchdog Reset Status. This bit may not be set by user code; however, it may be cleared by writing a '0'. 0 No watchdog reset has occurred. 1 Watchdog reset has occurred.
4	PORS	Power On Reset Status. This bit may not be set by user code; however, it may be cleared by writing a '0'. 0 Power on reset has not occurred and watchdog timer is enabled. 1 Is set after external reset or power on reset.
3	Sleep	Set by the user to enable the CPU sleep state. CPU remains in Sleep mode until any interrupt is pending. 0 Normal operation. 1 Sleep.
0	STOP	0 M8C is free to execute code. 1 M8C is halted and is only cleared by POR, XRES, or WDR.

21.4 Bank 1 Registers

The following registers are all in bank 1 and are listed in address order. Registers that are in both Bank 0 and Bank 1 are listed in address order in the section titled [Bank 0 Registers on page 188](#).

21.4.1 PRTxDM0

Port Drive Mode Bit Registers 0

Individual Register Names and Addresses:

PRT0DM0 : 1,00h PRT1DM0 : 1,04h PRT2DM0 : 1,08h PRT3DM0 : 1,0Ch
PRT4DM0 : 1,10h

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

This register is one of two registers where the combined value determines the unique drive mode of each bit in a GPIO port.

In register PRTxDM0 there are four possible drive 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 (PRTxDM0 and [PRTxDM1 on page 260](#)). The bit position of the effected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the two Drive Mode register bits that control the drive mode for that pin (for example, bit[2] in PRT0DM0 and bit[2] in PRT0DM1). The two bits from the two registers are treated as a group. These are referred to as DM1 and DM0, or together as DM[1:0].

All drive mode bits are shown in the sub-table below ([10] refers to the combination (in order) of bits in a given bit position); however, this register only controls the **least significant bit (LSb)** of the drive mode.

The upper nibble of the PRT4DM0 register returns the last data bus value when read. You need to mask it off prior to using this information. For additional information, refer to the [Register Definitions on page 59](#) in the GPIO chapter.

Bit	Name	Description																				
7:0	Drive Mode 0[7:0]	Bit 0 of the drive mode, for each of 8-port pins, for a GPIO port.																				
		<table><tr><th>[10]</th><th>Pin Output High</th><th>Pin Output Low</th><th>Notes</th></tr><tr><td>00b</td><td>Resistive</td><td>Strong</td><td></td></tr><tr><td>01b</td><td>Strong</td><td>Strong</td><td></td></tr><tr><td>10b</td><td>High Z</td><td>High Z</td><td>Reset state. Digital input disabled for zero power.</td></tr><tr><td>11b</td><td>High Z</td><td>Strong</td><td>I2C compatible mode. For digital inputs, use this</td></tr></table>	[10]	Pin Output High	Pin Output Low	Notes	00b	Resistive	Strong		01b	Strong	Strong		10b	High Z	High Z	Reset state. Digital input disabled for zero power.	11b	High Z	Strong	I2C compatible mode. For digital inputs, use this
[10]	Pin Output High	Pin Output Low	Notes																			
00b	Resistive	Strong																				
01b	Strong	Strong																				
10b	High Z	High Z	Reset state. Digital input disabled for zero power.																			
11b	High Z	Strong	I2C compatible mode. For digital inputs, use this																			

Note A bold digit in the table above signifies that the digit is used in this register.

21.4.2 PRTxDM1

Port Drive Mode Bit Registers 1

Individual Register Names and Addresses:

PRT0DM1 : 1,01h PRT1DM1 : 1,05h PRT2DM1 : 1,09h PRT3DM1 : 1,0Dh
PRT4DM1 : 1,11h

	7	6	5	4	3	2	1	0
Access : POR	RW : FF							
Bit Name	Drive Mode 1[7:0]							

This register is one of three registers where the combined value determines the unique drive mode of each bit in a GPIO port.

In register PRTxDM1 there are four possible drive 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 (PRTxDM1 and [PRTxDM0 on page 259](#)). The bit position of the effected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the two Drive Mode register bits that control the drive mode for that pin (for example, bit[2] in PRT0DM0 and bit[2] in PRT0DM1). The two bits from the two registers are treated as a group. These are referred to as DM1 and DM0, or together as DM[1:0].

All drive mode bits are shown in the sub-table below ([10] refers to the combination (in order) of bits in a given bit position); however, this register only controls the most significant bit (MSb) of the drive mode.

The upper nibble of the PRT4DM1 register returns the last data bus value when read. You need to mask it off before using this information. For additional information, refer to the [Register Definitions on page 59](#) in the GPIO chapter.

Bit	Name	Description																				
7:0	Drive Mode 1[7:0]	Bit 1 of the drive mode, for each of 8-port pins, for a GPIO port. <table><tr><th>[10]</th><th>Pin Output High</th><th>Pin Output Low</th><th>Notes</th></tr><tr><td>00b</td><td>Resistive</td><td>Strong</td><td></td></tr><tr><td>01b</td><td>Strong</td><td>Strong</td><td></td></tr><tr><td>10b</td><td>High Z</td><td>High Z</td><td>Reset state. Digital input disabled for zero power.</td></tr><tr><td>11b</td><td>High Z</td><td>Strong</td><td>I2C compatible mode. For digital inputs, use this mode with data bit (PRTxDR register) set high.</td></tr></table> Note A bold digit in the table above signifies that the digit is used in this register.	[10]	Pin Output High	Pin Output Low	Notes	00b	Resistive	Strong		01b	Strong	Strong		10b	High Z	High Z	Reset state. Digital input disabled for zero power.	11b	High Z	Strong	I2C compatible mode. For digital inputs, use this mode with data bit (PRTxDR register) set high.
[10]	Pin Output High	Pin Output Low	Notes																			
00b	Resistive	Strong																				
01b	Strong	Strong																				
10b	High Z	High Z	Reset state. Digital input disabled for zero power.																			
11b	High Z	Strong	I2C compatible mode. For digital inputs, use this mode with data bit (PRTxDR register) set high.																			

21.4.3 SPI_CFG

SPI Configuration Register

Individual Register Names and Addresses:

SPI_CFG : 1,29h

	7	6	5	4	3	2	1	0
Access : POR		RW : 0		RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name		Clock Sel [2:0]		Bypass	SS_	SS_EN_	Int Sel	Slave

This register is used to configure the SPI.

Do not change the values in this register while the block is enabled. For additional information, refer to the [Register Definitions on page 147](#) in the SPI chapter.

Bit	Name	Description
7:5	Clock Sel [2:0]	SYSCLK in Master mode. 000b / 2 001b / 4 010b / 8 011b / 16 100b / 32 101b / 64 110b / 128 111b / 256
4	Bypass	Bypass Synchronization. 0 All pin inputs are doubled and synchronized. 1 Input synchronization is bypassed.
3	SS_	Slave Select in Slave mode. 0 Slave selected. 1 Slave not selected.
2	SS_EN_	Internal Slave Select Enable. 0 Slave selection determined from SS_ bit. 1 Slave selection determined from external SS_ pin.
1	Int Sel	Interrupt Select. 0 Interrupt on TX Reg Empty. 1 Interrupt on SPI Complete.
0	Slave	0 Operates as a master. 1 Operates as a slave.

21.4.4 USB_CR1

USB Control Register 1

Individual Register Names and Addresses:

USB_CR1 : 1,30h

	7	6	5	4	3	2	1	0
Access : POR						RC : 0	RW : 0	RW : 0
Bit Name						BusActivity	EnableLock	RegEnable

This register is used to configure the internal regulator and the oscillator tuning capability.

This register is only used by the CY8CTMG20x, CY8CTST200 PSoC devices. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 171](#) in the Full-Speed USB chapter.

Bit	Name	Description
2	BusActivity	Monitors activity on USB bus. This bit is only set by the hardware. Writing a '0' clears this bit. Writing a '1' preserves its present state. 0 No activity. 1 Non-idle activity (D+ = Low) was detected since the last time the bit was cleared.
1	EnableLock	Controls the automatic tuning of the internal oscillator. Hardware locks the internal oscillator based on the frequency of incoming USB data when this bit is set. Normally, this is set when the device is used in a USB application. 0 Locking disabled. 1 Locking enabled.
0	RegEnable	Configures USB regulator for appropriate power supply range. 0 Pass through mode. Use for Vdd = 3.3V range. 1 Regulating mode. Use for Vdd = 5V range.

21.4.5 PMAx_WA

PSoC Memory Arbiter Write Address Registers

Individual Register Names and Addresses:

PMA0_WA : 1,34h PMA1_WA : 1,35h PMA2_WA : 1,36h PMA3_WA : 1,37h
 PMA4_WA : 1,38h PMA5_WA : 1,39h PMA6_WA : 1,3Ah PMA7_WA : 1,3Bh
 PMA8_WA : 1,44H PMA9_WA : 1,45H PMA10_WA : 1,46H PMA11_WA : 1,47H
 PMA12_W : 1, 48H PMA13_WA : 1,49h PMA14_WA : 1,4Ah PMA15_WA : 1,4Bh

A

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

These registers are PSoC Memory Arbiter write address registers.

For additional information, refer to the [Register Definitions on page 171](#) in the Full-Speed USB chapter.

Bit	Name	Description
7:0	Write Address[7:0]	The value returned when this register is read depends on whether the PMA channel is being used by the USB SIE or by the M8C.

21.4.6 PMAx_RA

PSoC Memory Arbiter Read Address Registers

Individual Register Names and Addresses:

PMA0_RA : 1,3Ch	PMA1_RA : 1,3Dh	PMA2_RA : 1,3Eh	PMA3_RA : 1,3Fh
PMA4_RA : 1,40h	PMA5_RA : 1,41h	PMA6_RA : 1,42h	PMA7_RA : 1,43h
PMA8_RA : 1,4Ch	PMA9_RA : 1,4Dh	PMA10_RA : 1,4Eh	PMA11_RA : 1,4Fh
PMA12_RA : 1, 50h	PMA13_RA : 1,51h	PMA14_RA : 1,52h	PMA15_RA : 1,53h

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

These registers are PSoC Memory Arbiter read address registers.

For additional information, refer to the [Register Definitions on page 171](#) in the Full-Speed USB chapter.

Bit	Name	Description
7:0	ReadAddress[7:0]	The value returned when this register is read depends on whether the PMA channel is being used by the USB SIE or by the M8C. In the USB case, this register always returns the beginning SRAM address for the PMA channel.

21.4.7 EPx_CR0

Endpoint Control Registers 0

Individual Register Names and Addresses:

EP1_CR0 : 1,54h EP2_CR0 : 1,55h EP3_CR0 : 1,56h EP4_CR0 : 1,57h
 EP5_CR0 : 1,58h EP6_CR0 : 1,59h EP7_CR0 : 1,5Ah EP7_CR0 : 1,5Bh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0		RW : 0	RC : 0				RW : 0
Bit Name	Stall		NAK_INT_EN	ACK'ed Tx				Mode[3:0]

These registers endpoint control registers.

In the table above, note that the reserved bit is a grayed table cell and is not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 171](#) in the Full-Speed USB chapter.

Bit	Name	Description
7	Stall	When this bit is set, the SIE stalls an OUT packet if the Mode bits are set to ACK-OUT. The SIE stalls an IN packet if the Mode bits are set to ACK-IN. This bit must be clear for all other modes.
5	NAK_INT_EN	When set, this bit causes an endpoint interrupt to be generated even when a transfer completes with a NAK.
4	ACKed Tx	The ACK'ed transaction bit is set whenever the SIE engages in a transaction to the register's endpoint that completes with an ACK packet.
3:0	Mode[3:0]	The mode controls how the USB SIE responds to traffic and how the USB SIE changes the mode of that endpoint as a result of host packets to the endpoint.

21.4.8 TMP_DRx

Temporary Data Registers

Individual Register Names and Addresses:

TMP_DR0 : x,6Ch TMP_DR1 : x,6Dh TMP_DR2 : x,6Eh TMP_DR3 : x,6Fh

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

These registers enhance the performance in multiple SRAM page PSoC devices.

All bits in this register are reserved for PSoC devices with 256 bytes of SRAM. For additional information, refer to the [Register Definitions on page 42](#) in the RAM Paging chapter.

Bit	Name	Description
7:0	Data[7:0]	General purpose register space

21.4.9 USB_MISC_CR

USB Miscellaneous Control Register

Individual Register Names and Addresses:

USB_MISC_CR: 1,BDh

	7	6	5	4	3	2	1	0
Access : POR						RW : 0	RW : 0	RW : 0
Bit Name						USB_SE_EN	USB_ON	USB_CLK_ON

The USB Miscellaneous Control Register controls the clocks to the USB block to make IMO work with better accuracy for the USB part and to disable the single ended input of USBIO in the case of a non-USB part.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 112](#) in the Digital Clocks chapter.

Bit	Name	Description
2	USE_SE_EN	The single ended outputs of USBIO is enabled or disabled based upon this bit setting. Set this bit to '1' when using this part as a USB part. 0 The single ended outputs of USBIO are disabled. The DPO, DMO is held at logic high state and RSEO is held at a low state. 1 The single ended output of USBIO is enabled and USB transactions can occur. Note Bit [1:0] of the USBIO_CR1 register is also affected by this register setting. When this bit is '0' (default) regardless of the DP and DM state, the DPO and DMO bits of USBIO_CR1 are '11b'.
1	USB_ON	This bit is used by the IMO DAC block to either work with better DNL consuming higher power, or with sacrificed DNL consuming lower power. Set this bit to '1' when the part is used as a USB part. 0 The IMO runs with sacrificed DNL by consuming less power. 1 The IMO runs with better DNL by consuming more power.
0	USB_CLK_ON	This bit either enables or disables the clocks to the USB block. It is used to save power in cases when the device need not respond to USB traffic. Set this bit to '1' when the device is used as a USB part. 0 All clocks to the USB block are driven as '0'. The device does not respond to USB traffic and none of the USB registers, except IMO_TR, IMO_TR1 and USBIO_CR1, listed in the Register Definitions on page 171 are writable. 1 Clocks are not blocked to the USB block. The device responds to USB traffic depending upon the other register settings mentioned in Register Definitions on page 171 in the Full-Speed USB chapter.

21.4.10 OUT_P0

Output Override to Port 0 Register

Individual Register Names and Addresses:

OUT_P0: 1,D1h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0				
Bit Name	P0P7D	P0P7EN	P0P4D	P0P4EN				

This register enables specific internal signals to output to Port 0 pins.

The GPIO drive modes must be specified to support the desired output mode (registers PRT1DM1 and PRT1DM0). If a pin is enabled for output by a bit in this register, the corresponding signal has priority over any other internal function that may be configured to output to that pin.

For additional information, refer to the [Register Definitions on page 112](#) in the Digital Clocks chapter.

Bit	Name	Description
7	P0P7D	0 Select Main system clock (SYSCLK).
		1 Select either TrueTouch Output signal (CS_OUT is selected by CS_OUT[1:0] bits in CS_CR0 register) or TrueTouch clock depending upon bit 7 in PRS_CR register.
6	P0P7EN	This bit enables pin P0 [7] for output of the signal selected by P0P7D.
		0 No internal signal output to P0 [7].
		1 Output the signal selected by P0P7D on to P0[7].
5	P0P4D	0 Select Timer Output (TIMEROUT)
		1 Select CLK32
		Note: During sleep mode these outputs will not be passed through P0[4].
4	P0P4EN	This bit enables pin P0 [4] for output of the signal selected by P0P4D.
		0 No internal signal output to P0 [4].
		1 Output the signal selected by P0P4D on to P0[4].

21.4.11 ECO_ENBUS

External Oscillator ENBUS Register

Individual Register Names and Addresses:

ECO_ENBUS : 1,D2h

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

The ECO_ENBUS register is used to disable and enable the external crystal oscillator (ECO). In the table above, note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. See the [Application Overview on page 70](#) for the proper sequence for enabling the ECO.

Bits	Name	Description
2:0	ECO_ENBUS[2:0]	These bits should be written with a value of 011b to allow the ECO to be enabled by bits in the ECO_CFG register, or a value of 111b (default) to disable the ECO. Other values in this register are reserved.

External Oscillator Trim Register

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

Bits	Name	Description
4:2	ECO_XGM[2:0]	These bits set the amplifier gain. The high power mode setup size is approximately 220 nA. The low power step size is approximately 5% lower than the 111 setting. 000 Lowest power setting. 111 Highest power setting.
1:0	ECO_LP[1:0]	These regulate low power mode settings. 00 Highest Power Setting. 11 Lowest power setting (30% power reduction).

21.4.13 MUX_CRx

Analog Mux Port Bit Enable Registers

Individual Register Names and Addresses:

MUX_CR0 : 1,D8h MUX_CR1 : 1,D9h MUX_CR2 : 1,DAh MUX_CR3 : 1,DBh
MUX_CR4 : 1,DFh

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

This register is used to control the connection between the analog mux bus and the corresponding pin.

Port 4 is a 4-bit port, so the upper 4 bits of the MUX_CR4 register are reserved and return zeros when read.

For additional information, refer to the [Register Definitions on page 100](#) in the I/O Analog Multiplexer chapter.

Bits	Name	Description
7:0	ENABLE[7:0]	Each bit controls the connection between the analog mux bus and the corresponding port pin. For example, MUX_CR2[3] controls the connection to bit 3 in Port 2. Any number of pins may be connected at the same time. 0 No connection between port pin and analog mux bus. 1 Connect port pin to analog mux bus.

21.4.14 IO_CFG1

Input/Output Configuration Register 1

Individual Register Names and Addresses:

IO_CFG1 : 1,DCh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0		RW : 0		RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	StrongP		Range[1:0]		P1_LOW_THRS	SPICLK_ON_P10	REG_EN	IO INT

This register is used to configure the Port 1 output regulator and set the interrupt mode for all GPIO.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 59](#) in the GPIO chapter.

Bits	Name	Description
7	StrongP	Setting this bit increases the drive strength and edge ratio for high outputs.
5:4	Range[1:0]	Selects the high output level for Port 1 outputs. 00 3.0 volts 01 3.0 volts 10 1.8 volts 11 2.5 volts
3	P1-LOW_THRS	This bit reduces the threshold voltage of the P1 port input buffers so that there are no compatibility issues when Port 1 is communicating at regulated voltage levels. 0 Standard threshold of VIH, VIL 1 Reduce threshold of VIH, VIL
2	SPICLK_ON_P10	When set to '1', the SPI clock is mapped to Port 1 pin 0. Otherwise, it is mapped to Port 1 pin 3.
1	REG_EN	Controls the regulator on Port 1 outputs. 0 Regulator disabled, so Port 1 strong outputs drive to Vdd. 1 Regulator enabled, so Port 1 strong outputs drive to approximately 3V (for Vdd > 3V).
0	IO INT	Sets the GPIO interrupt mode for all pins in the PSoC device. GPIO interrupts are also controlled at each pin by the PRTxIE registers, and by the global GPIO bit in the INT_MSK0 register. 0 GPIO interrupt configured for interrupt when pin is low. 1 GPIO interrupt configured for interrupt when pin state changes from last time port was read.

21.4.15 OUT_P1

Output Override to Port 1 Register

Individual Register Names and Addresses:

OUT_P1: 1,DDh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	P16D	P16EN	P14D	P14EN	P12D	P12EN	P10D	P10EN

This register enables specific internal signals to be output to Port 1 pins.

The GPIO drive modes must be specified to support the desired output mode (registers PRT1DM1 and PRT1DM0). If a pin is enabled for output by a bit in this register, the corresponding signal has priority over any other internal function that may be configured to output to that pin.

For additional information, refer to the [Register Definitions on page 112](#) in the Digital Clocks chapter.

Bit	Name	Description
7	P16D	Bit selects the data output to P1[6] when P16EN is high. Timer Output and CLK32 are available on P0[4] as well, but outputs on P0[4] will not be passed through during sleep modes. Please refer to the OUT_P0 register for more details. 0 Select Timer output (TIMEROUT) 1 Select CLK32
6	P16EN	Bit enables pin P1[6] for output of the signal selected by the P16D bit. 0 No internal signal output to P1[6] 1 Output the signal selected by P16D to P1[6]
5	P14D	Bit selects the data output to P1[4] when P14EN is high. 0 Select Relaxation Oscillator (RO) 1 Select Comparator 1 Output (CMP1)
4	P14EN	Bit enables pin P1[4] for output of the signal selected by the P14D bit. 0 No internal signal output to P1[4] 1 Output the signal selected by P14D to P1[4]
3	P12D	Bit selects the data output to P1[2] when P12EN is high. 0 Select Main System Clock (SYSClk) 1 Select either TrueTouch Output signal (CS_OUT is selected by CS_OUT[1:0] bits in CS_CR0 register) or TrueTouch clock depending upon bit 7 in PRS_CR register.
2	P12EN	Bit enables pin P1[2] for output of the signal selected by the P12D bit. 0 No internal signal output to P1[2] 1 Output the signal selected by P12D to P1[2]

(continued on next page)

21.4.15 OUT_P1 (continued)

1	P10D	Bit selects the data output to P1[0] when P10EN is high. 0 Select Sleep Interrupt (SLPINT) 1 Select Comparator 0 Output (CMP0)
0	P10EN	Bit enables pin P1[0] for output of the signal selected by the P10D bit. 0 No internal signal output to P1[0] 1 Output the signal selected by P10D to P1[0]

21.4.16 IO_CFG2

Input/Output Configuration Register 2

Individual Register Names and Addresses:

IO_CFG2 : 1,DEh

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

The Input/Output Configuration 2 Register (IO_CFG2) selects output regulated supply and clock rates.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the [IO_CFG2 Register on page 61](#) in the GPIO chapter.

Bits	Name	Description
5:3	REG_LEVEL[2:0]	These bits select output regulated supply

REG_LEVEL[2:0]	Approx. Regulated Supply (V)		
000	3	2.5	1.8
001	3.1	2.6	1.9
010	3.2	2.7	2.0
011	3.3	2.8	2.1
100	3.4	2.9	2.2
101	3.5	3.0	2.3
110	3.6	3.1	2.4
111	3.7	3.2	2.5

1:0	REG_CLOCK[1:0]	The Regulated I/O charge pump can operate with a maximum clock speed of 12 MHz. The REG_CLOCK[1:0] bits select clocking options for the regulator. Setting REG_CLOCK[1:0] to '10' should be used with 24 MHz SYSCLK and '01' should be used with 6/12 MHz SYSCLK.
-----	-----------------------	---

REG_CLOCK[1:0]	SYSCLK Clock Rate
10	24 MHz
01	6/12 MHz

21.4.17 OSC_CR0

Oscillator Control Register 0

Individual Register Names and Addresses:

OSC_CR0: 1,E0h

	7	6	5	4	3	2	1	0
Access : POR	RW: 0	RW: 0	RW: 0		RW: 0			RW: 010b
Bit Name	X32ON	Disable Buzz	No Buzz		Sleep[1:0]			CPU Speed[2:0]

This register is used to configure various features of internal clock sources and clock nets.

In the table above, note that the reserved bit is a grayed table cell and is not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 112](#) in the Digital Clocks chapter.

Bit	Name	Description																																													
7	X32ON	Select bit for the external 32 kHz external crystal oscillator (ECO). See the Application Overview on page 70 for the proper sequence for enabling the ECO. 0 The internal 32 kHz oscillator is the source of the 32K clock. 1 The external crystal oscillator is the source of the 32K clock.																																													
6	Disable Buzz	Option to disable buzz during sleep. This bit has lower priority than the No Buzz bit. Therefore, if No Buzz = 1, the Disable Buzz bit has no effect. 0 No effect on buzz modes. 1 Buzz is disabled during sleep, with bandgap powered down. No periodic wakeup of the bandgap during sleep.																																													
5	No Buzz	This bit allows the bandgap to stay powered during sleep. 0 Buzz bandgap during power down. 1 Bandgap is always powered even during sleep.																																													
4:3	Sleep[1:0]	Sleep interval. 00b 1.95 ms (512 Hz) 01b 15.6 ms (64 Hz) 10b 125 ms (8 Hz) 11b 1s (1 Hz)																																													
2:0	CPU Speed[2:0]	These bits set the CPU clock speed, based on the system clock (SYSCLK). SYSCLK is 12 MHz by default, but it can also be set to other frequencies (6 and 24 MHz), or driven from an external clock. Note During USB operation, the CPU speed can be set to any setting. Be aware that USB throughput decreases with a decrease in CPU speed. For maximum throughput, the CPU clock should be made equal to the system clock. The system clock must be 24 MHz for USB operation.																																													
		<table><tr><th></th><th>6 MHz IMO</th><th>12 MHz IMO</th><th>24 MHz IMO</th><th>External Clock</th></tr><tr><td>000b</td><td>750 kHz</td><td>1.5 MHz</td><td>3 MHz</td><td>EXTCLK/8</td></tr><tr><td>001b</td><td>1.5 MHz</td><td>3 MHz</td><td>6 MHz</td><td>EXTCLK/4</td></tr><tr><td>010b</td><td>3 MHz</td><td>6 MHz</td><td>12 MHz</td><td>EXTCLK/2 (Reset State)</td></tr><tr><td>011b</td><td>6 MHz</td><td>12 MHz</td><td>24 MHz</td><td>EXTCLK/1</td></tr><tr><td>100b</td><td>375 kHz</td><td>750 kHz</td><td>1.5 MHz</td><td>EXTCLK/16</td></tr><tr><td>101b</td><td>187.5 kHz</td><td>375 kHz</td><td>750 kHz</td><td>EXTCLK/32</td></tr><tr><td>110b</td><td>46.9 kHz</td><td>93.7 kHz</td><td>187.5 kHz</td><td>EXTCLK/128</td></tr><tr><td>111b</td><td>23.4 kHz</td><td>46.8 kHz</td><td>93.7 kHz</td><td>EXTCLK/256</td></tr></table>		6 MHz IMO	12 MHz IMO	24 MHz IMO	External Clock	000b	750 kHz	1.5 MHz	3 MHz	EXTCLK/8	001b	1.5 MHz	3 MHz	6 MHz	EXTCLK/4	010b	3 MHz	6 MHz	12 MHz	EXTCLK/2 (Reset State)	011b	6 MHz	12 MHz	24 MHz	EXTCLK/1	100b	375 kHz	750 kHz	1.5 MHz	EXTCLK/16	101b	187.5 kHz	375 kHz	750 kHz	EXTCLK/32	110b	46.9 kHz	93.7 kHz	187.5 kHz	EXTCLK/128	111b	23.4 kHz	46.8 kHz	93.7 kHz	EXTCLK/256
	6 MHz IMO	12 MHz IMO	24 MHz IMO	External Clock																																											
000b	750 kHz	1.5 MHz	3 MHz	EXTCLK/8																																											
001b	1.5 MHz	3 MHz	6 MHz	EXTCLK/4																																											
010b	3 MHz	6 MHz	12 MHz	EXTCLK/2 (Reset State)																																											
011b	6 MHz	12 MHz	24 MHz	EXTCLK/1																																											
100b	375 kHz	750 kHz	1.5 MHz	EXTCLK/16																																											
101b	187.5 kHz	375 kHz	750 kHz	EXTCLK/32																																											
110b	46.9 kHz	93.7 kHz	187.5 kHz	EXTCLK/128																																											
111b	23.4 kHz	46.8 kHz	93.7 kHz	EXTCLK/256																																											

21.4.18 ECO_CFG

External Oscillator Trim Configuration Register

Individual Register Names and Addresses:

ECO_CFG : 1,E1h

	7	6	5	4	3	2	1	0
Access : POR						RW : 0	RW : 0	RW : 0
Bit Name						ECO_LPM	ECO_EXW	ECO_EX

This register provides ECO status and control information.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [ECO_CFG Register on page 71](#) in the External Crystal Oscillator chapter.

Bits	Name	Description
2	ECO_LPM	This bit enables the ECO lower power mode.
1	ECO_EXW	This is a status bit that indicates that the ECO_EX bit was previously written to. When this bit is a '1', this indicates that the ECO_CONFIG register was written to and is now locked. When this bit is a '0', the register was not written to since the last reset event.
0	ECO_EX	The ECO Exists bit is a flag to the hardware indicating that an external crystal oscillator exists in the system.If the bit is '0', a switch-over to the ECO is locked out by hardware. If the bit is '1', hardware allows the firmware to freely switch between the ECO and ILO. It must be written as early as possible after a Power On Reset (POR) or External Reset (XRES) event.

21.4.19 OSC_CR2

Oscillator Control Register 2

Individual Register Names and Addresses:

OSC_CR2 : 1,E2h

	7	6	5	4	3	2	1	0
Access : POR				RW : 0		RW : 0	RW : 0	
Bit Name				CLK48MEN		EXTCLKEN	IMODIS	

This register is used to configure various features of internal clock sources and clock nets.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 112](#) in the Digital Clocks chapter.

Bit	Name	Description
4	CLK48MEN	This is the 48 MHz clock enable bit. 0 Disables the 48 MHz clock. 1 Enables the 48 MHz clock.
2	EXTCLKEN	External Clock Mode Enable. 0 Disabled. Operate from internal main oscillator. 1 Enabled. Operate from the clock supplied at P1[4] or P1[1] based upon the TSYNC bit in CPU_SCR1.
1	IMODIS	Internal Oscillator Disable. This bit can be set to save power when using an external clock on P1[4]. 0 Enabled. Internal oscillator enabled. 1 Disabled. Note This bit must not be set high in the same instruction that sets EXTCLKEN high, but it can be set in the next instruction. Also, this bit must not be set high if the external clock frequency is less than 6 MHz. When switching from external clock to internal clock, the IMO must be enabled for at least 10 μs before the transition to internal clock. Refer to Switch Operation on page 110.

21.4.20 VLT_CR

Voltage Monitor Control Register

Individual Register Names and Addresses:

VLT_CR: 1,E3h

	7	6	5	4	3	2	1	0
Access : POR				RW : 0	RW : 0		RW : 0	
Bit Name				PORLEV[1:0]	LVDTBEN		VM[2:0]	

This register is used to set the trip points for POR and LVD.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 144](#) in the POR chapter.

Bit	Name	Description
5:4	PORLEV[1:0]	Sets the POR level according to the DC Electrical Specifications in the PSoC device data sheet.
3	LVDTBEN	Enables reset of the CPU speed register by LVD comparator output.
2:0	VM[2:0]	Sets the LVD levels per the DC Electrical Specifications in the PSoC device data sheet, for those devices with this feature. 000b Lowest voltage setting. 001b 010b 011b 100b 101b 110b 111b Highest voltage setting.

21.4.21 VLT_CMP

Voltage Monitor Comparators Register

Individual Register Names and Addresses:

VLT_CMP : 1,E4h

	7	6	5	4	3	2	1	0
Access : POR							R : 0	
Bit Name							LVD	

This register reads the state of the internal supply voltage monitors.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 144](#) in the POR chapter.

Bit	Name	Description
1	LVD	This bit reads the state of the LVD comparator. 0 Vdd is above LVD trip point. 1 Vdd is below LVD trip point.

21.4.22 IMO_TR

Internal Main Oscillator Trim Register

Individual Register Names and Addresses:

IMO_TR : 1,E8h

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

This register is used to manually center the Internal Main Oscillator's (IMO) output to a target frequency.

It is strongly recommended that you do not alter this register's values except to load factory trim settings when changing IMO range.

When changing ranges, the new trim value for this range must be read from Flash using a Table Read operation. The new value must be written at the lower frequency range. That is, when moving to a higher frequency range, change the IMO_TR value and then change the range (SLIMO[1:0] in CPU_SCR1). When moving to a lower frequency, change the range first and then update IMO_TR.

For additional information, refer to the [Register Definitions on page 64](#) in the Internal Main Oscillator chapter.

Bit	Name	Description
7:0	Trim[7:0]	The value of this register is used to trim the Internal Main Oscillator. Its value is set to the best value for the device during boot.

21.4.23 ILO_TR

Internal Low Speed Oscillator Trim Register

Individual Register Names and Addresses:

ILO_TR : 1,E9h

	7	6	5	4	3	2	1	0
Access : POR		RW : 0	RW : 0					RW : 08
Bit Name		PD_MODE	ILOFREQ					Freq Trim[3:0]

This register sets the adjustment for the Internal Low Speed Oscillator (ILO).

It is strongly recommended that you do not alter this register's Freq Trim[3:0] values. The trim bits are set to factory specifications and must not be changed.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Always write reserved bits with a value of '0'. For additional information, refer to the [Register Definitions on page 68](#) in the Internal Low Speed Oscillator chapter.

Bit	Name	Description
6	PD_MODE	This bit selects power down mode. Setting this bit high disables oscillator and current bias, which results in slower startup time. Power down mode: 0 Partial oscillator power down for faster startup (100 nA nominal). 1 Full oscillator power down for lower power (0 nA nominal).
5	ILOFREQ	Selects oscillator nominal frequency. 0 32 kHz 1 1 kHz
3:0	Freq Trim[3:0]	These bits trim the oscillator frequency.

21.4.24 SLP_CFG

Sleep Configuration Register

Individual Register Names and Addresses:

SLP_CFG : 1,EBh

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

This register sets up the sleep duty cycle.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 77](#) in the Sleep and Watchdog chapter.

Bit	Name	Description
7:6	PSSDC[1:0]	Sleep Duty Cycle. Controls the ratios (in numbers of 32.768 kHz clock periods) of "on" time versus "off" time for PORLVD, bandgap reference, and pspump. 00b 1 / 256 (8 ms). 01b 1 / 1024 (31.2 ms). 10b 1 / 64 (2 ms). 11b 1 / 16 (500 μs).

21.4.25 SLP_CFG2

Sleep Configuration Register 2

Individual Register Names and Addresses:

SLP_CFG2 : 1,ECh

	7	6	5	4	3	2	1	0
Access : POR						RW : 0	RW : 0	RW : 0
Bit Name						ALT_Buzz[1:0]	I2C_ON	LSO_OFF

This register holds the configuration for I2C sleep, deep sleep, and buzz.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 77](#) in the Sleep and Watchdog chapter.

Bit	Name	Description
3:2	ALT_Buzz[1:0]	These bits control additional selections for POR/LVD buzz rates. 00 Compatibility mode, buzz rate is determined by PSSDC bits. 01 Duty cycle is 1/32768. 10 Duty cycle is 1/8192. 11 Reserved.
1	I2C_ON	This bit enables the standby regulator in I2C sleep mode at a level sufficient to supply the I2C circuitry.
0	LSO_OFF	This bit disables the LSO oscillator when in sleep state.

21.4.26 SLP_CFG3

Sleep Configuration Register 3

Individual Register Names and Addresses:

SLP_CFG3 : 1,EDh

	7	6	5	4	3	2	1	0
Access : POR		RW : 1	RW : 11	RW : 11	RW : 11	RW : 11	RW : 11	
Bit Name		DBL_TAPS	T2TAP[1:0]	T1TAP[1:0]	T0TAP[1:0]			

This register holds the configuration of the wakeup sequence taps.

It is strongly recommended to not alter this register setting.

In the table above, note that the reserved bit is a grayed table cell and is not described in the bit description section below. Reserved bits must always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 77](#) in the Sleep and Watchdog chapter.

Bit	Name	Description								
6	DBL_TAPS	When this bit is set all the tap values (T0, T1, and T2) are doubled for the wakeup sequence.								
5:4	T2TAP[1:0]	These bits control the duration of the T2-T4 sequence (see Figure 10-2 on page 75) by selecting a tap from the WakeupTimer. Note: The T2 delay is only valid for the wakeup sequence. It is not used for the buzz sequence. <table><tr><td>00</td><td>1 μs</td></tr><tr><td>01</td><td>2 μs</td></tr><tr><td>10</td><td>5 μs</td></tr><tr><td>11</td><td>10 μs</td></tr></table>	00	1 μ s	01	2 μ s	10	5 μ s	11	10 μ s
00	1 μ s									
01	2 μ s									
10	5 μ s									
11	10 μ s									
3:2	T1TAP[1:0]	These bits control the duration of the T1-T2 sequence (see Figure 10-2 on page 75) by selecting a tap from the Wakeup Timer. <table><tr><td>00</td><td>3 μs</td></tr><tr><td>01</td><td>4 μs</td></tr><tr><td>10</td><td>5 μs</td></tr><tr><td>11</td><td>10 μs</td></tr></table>	00	3 μ s	01	4 μ s	10	5 μ s	11	10 μ s
00	3 μ s									
01	4 μ s									
10	5 μ s									
11	10 μ s									
1:0	T0TAP[1:0]	These bits control the duration of the T0-T1 sequence (see Figure 10-2 on page 75) by selecting a tap from the Wakeup Timer. <table><tr><td>00</td><td>10 μs</td></tr><tr><td>01</td><td>14 μs</td></tr><tr><td>10</td><td>20 μs</td></tr><tr><td>11</td><td>30 μs</td></tr></table>	00	10 μ s	01	14 μ s	10	20 μ s	11	30 μ s
00	10 μ s									
01	14 μ s									
10	20 μ s									
11	30 μ s									

21.4.27 IMO_TR1

Internal Main Oscillator Trim Register 1

Individual Register Names and Addresses:

1,FAh

IMO_TR1 : 1,FAh

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

This register is used to fine tune the IMO frequency.

It is strongly recommended that the user not alter this register's values.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the [Register Definitions on page 64](#) in the Internal Main Oscillator chapter.

Bit	Name	Description
7:0	FineTrim[2:0]	These bits provide ability to fine tune the IMO frequency. These values are normally only changed by the oscillator-locking function. These are the lower 3 bits of the 11-bit oscillator trim. IMO_TR holds the MSb.

Section F: Glossary



The Glossary section explains the terminology used in this technical reference manual. Glossary terms are characterized in **bold, italic font** throughout the text of this manual.

A

<i>accumulator</i>	In a CPU, a register in which intermediate results are stored. Without an accumulator, it would be necessary to write the result of each calculation (addition, subtraction, shift, and so on) to main memory and read them back. Access to main memory is slower than access to the accumulator, which usually has direct paths to and from the arithmetic and logic unit (ALU).
<i>active high</i>	1. A logic signal having its asserted state as the logic 1 state. 2. A logic signal having the logic 1 state as the higher voltage of the two states.
<i>active low</i>	1. A logic signal having its asserted state as the logic 0 state. 2. A logic signal having its logic 1 state as the lower voltage of the two states: inverted logic.
<i>address</i>	The label or number identifying the memory location (RAM, ROM, or register) where a unit of information is stored.
<i>algorithm</i>	A procedure for solving a mathematical problem in a finite number of steps that frequently involve repetition of an operation.
<i>ambient temperature</i>	The temperature of the air in a designated area, particularly the area surrounding the PSoC device.
<i>analog</i>	As opposed to digital, signals that are on or off or '1' or '0'. Analog signals vary in a continuous manner. See also <i>analog signals</i> .
<i>analog blocks</i>	The basic programmable opamp circuits. These are SC (switched capacitor) and CT (continuous time) blocks. These blocks can be interconnected to provide ADCs, DACs, multi-pole filters, gain stages, and much more.
<i>analog output</i>	An output that is capable of driving any voltage between the supply rails, instead of just a logic 1 or logic 0.
<i>analog signals</i>	A signal represented in a continuous form with respect to continuous times, as contrasted with a digital signal represented in a discrete (discontinuous) form in a sequence of time.
<i>analog-to-digital (ADC)</i>	A device that changes an analog signal to a digital signal of corresponding magnitude. Typically, an ADC converts a voltage to a digital number. The <i>digital-to-analog (DAC)</i> converter performs the reverse operation.
<i>AND</i>	See <i>Boolean Algebra</i> .

API (Application Programming Interface)	A series of software routines that comprise an interface between a computer application and lower-level services and functions (for example, user modules and libraries). APIs serve as building blocks for programmers that create software applications.
array	An array, also known as a vector or list, is one of the simplest data structures in computer programming. Arrays hold a fixed number of equally-sized data elements, generally of the same data type. Individual elements are accessed by index using a consecutive range of integers, as opposed to an associative array. Most high level programming languages have arrays as a built-in data type. Some arrays are multi-dimensional, meaning they are indexed by a fixed number of integers; for example, by a group of two integers. One- and two-dimensional arrays are the most common. Also, an array can be a group of capacitors or resistors connected in some common form.
assembly	A symbolic representation of the machine language of a specific processor. Assembly language is converted to machine code by an assembler. Usually, each line of assembly code produces one machine instruction, though the use of macros is common. Assembly languages are considered low level languages; where as C is considered a high level language.
asynchronous	A signal whose data is acknowledged or acted upon immediately, irrespective of any clock signal.
attenuation	The decrease in intensity of a signal as a result of absorption of energy and of scattering out of the path to the detector, but not including the reduction due to geometric spreading. Attenuation is usually expressed in dB.
B	
bandgap reference	A stable voltage reference design that matches the positive temperature coefficient of V_T with the negative temperature coefficient of V_{BE} , to produce a zero temperature coefficient (ideally) reference.
bandwidth	1. The frequency range of a message or information processing system measured in hertz. 2. The width of the spectral region over which an amplifier (or absorber) has substantial gain (or loss). It is sometimes represented more specifically as, for example, full width at half maximum.
bias	1. A systematic deviation of a value from a reference value. 2. The amount by which the average of a set of values departs from a reference value. 3. The electrical, mechanical, magnetic, or other force (field) applied to a device to establish a reference level to operate the device.
bias current	The constant low level DC current that is used to produce a stable operation in amplifiers. This current can sometimes be changed to alter the bandwidth of an amplifier.
binary	The name for the base 2 numbering system. The most common numbering system is the base 10 numbering system. The base of a numbering system indicates the number of values that may exist for a particular positioning within a number for that system. For example, in base 2, binary, each position may have one of two values (0 or 1). In the base 10, decimal, each position may have one of ten values (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9).
bit	A single digit of a binary number. Therefore, a bit may only have a value of '0' or '1'. A group of 8 bits is called a byte. Because the PSoC M8C is an 8-bit microcontroller, the PSoC native data chunk size is a byte.

bit rate (BR)	The number of bits occurring per unit of time in a bit stream, usually expressed in bits per second (bps).
block	1. A functional unit that performs a single function, such as an oscillator. 2. A functional unit that may be configured to perform one of several functions, such as a digital block or an analog block.
Boolean Algebra	In mathematics and computer science, Boolean algebras or Boolean lattices, are algebraic structures which "capture the essence" of the logical operations AND, OR and NOT as well as set the theoretic operations union, intersection, and complement. Boolean algebra also defines a set of theorems that describe how Boolean equations can be manipulated. For example, these theorems are used to simplify Boolean equations which reduces the number of logic elements needed to implement the equation. The operators of Boolean algebra may be represented in various ways. Often they are simply written as AND, OR, and NOT. In describing circuits, NAND (NOT AND), NOR (NOT OR), XNOR (exclusive NOT OR), and XOR (exclusive OR) may also be used. Mathematicians often use + (for example, A+B) for OR and • for AND (for example, A*B) (since in some ways those operations are analogous to addition and multiplication in other algebraic structures) and represent NOT by a line drawn above the expression being negated (for example, $\sim A$, $A_{\sim}$, !A).
break-before-make	The elements involved go through a disconnected state entering ("break") before the new connected state ("make").
broadcast net	A signal that is routed throughout the microcontroller and is accessible by many blocks or systems.
buffer	1. A storage area for data that is used to compensate for a speed difference, when transferring data from one device to another. Usually refers to an area reserved for I/O operations into which data is read or from which data is written. 2. A portion of memory set aside to store data, often before it is sent to an external device or as it is received from an external device. 3. An amplifier used to lower the output impedance of a system.
bus	1. A named connection of nets. Bundling nets together in a bus makes it easier to route nets with similar routing patterns. 2. A set of signals performing a common function and carrying similar data. Typically represented using vector notation; for example, address[7:0]. 3. One or more conductors that serve as a common connection for a group of related devices.
byte	A digital storage unit consisting of 8 bits.
C	
C	A high level programming language.
capacitance	A measure of the ability of two adjacent conductors, separated by an insulator, to hold a charge when a voltage differential is applied between them. Capacitance is measured in units of Farads.
capture	To extract information automatically through the use of software or hardware, as opposed to hand-entering of data into a computer file.

chaining	Connecting two or more 8-bit digital blocks to form 16-, 24-, and even 32-bit functions. Chaining allows certain signals such as Compare, Carry, Enable, Capture, and Gate to be produced from one block to another.
checksum	The checksum of a set of data is generated by adding the value of each data word to a sum. The actual checksum can simply be the result sum or a value that must be added to the sum to generate a pre-determined value.
chip	A single monolithic Integrated Circuit (IC). See also integrated circuit (IC) .
clear	To force a bit/register to a value of logic 0.
clock	The device that generates a periodic signal with a fixed frequency and duty cycle. A clock is sometimes used to synchronize different logic blocks.
clock generator	A circuit that is used to generate a clock signal.
CMOS	The logic gates constructed using MOS transistors connected in a complementary manner. CMOS is an acronym for complementary metal-oxide semiconductor.
comparator	An electronic circuit that produces an output voltage or current whenever two input levels simultaneously satisfy pre-determined amplitude requirements.
compiler	A program that translates a high level language, such as C, into machine language.
configuration	In a computer system, an arrangement of functional units according to their nature, number, and chief characteristics. Configuration pertains to hardware, software, firmware, and documentation. The configuration affects system performance.
configuration space	In PSoC devices, the register space accessed when the XIO bit in the CPU_F register is set to '1'.
crowbar	A type of over-voltage protection that rapidly places a low resistance shunt (typically an SCR) from the signal to one of the power supply rails, when the output voltage exceeds a pre-determined value.
crystal oscillator	An oscillator in which the frequency is controlled by a piezoelectric crystal. Typically a piezoelectric crystal is less sensitive to ambient temperature than other circuit components.
cyclic redundancy check (CRC)	A calculation used to detect errors in data communications, typically performed using a linear feedback shift register. Similar calculations may be used for a variety of other purposes such as data compression.

D

data bus	A bidirectional set of signals used by a computer to convey information from a memory location to the central processing unit and vice versa. More generally, a set of signals used to convey data between digital functions.
data stream	A sequence of digitally encoded signals used to represent information in transmission.
data transmission	The sending of data from one place to another by means of signals over a channel.

debugger	A hardware and software system that allows the user to analyze the operation of the system under development. A debugger usually allows the developer to step through the firmware one step at a time, set break points, and analyze memory.
dead band	A period of time when neither of two or more signals are in their active state or in transition.
decimal	A base 10 numbering system, which uses the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 (called digits) together with the decimal point and the sign symbols + (plus) and - (minus) to represent numbers.
default value	Pertaining to the pre-defined initial, original, or specific setting, condition, value, or action a system assumes, uses, or takes in the absence of instructions from the user.
device	The device referred to in this manual is the PSoC chip, unless otherwise specified.
die	An unpackaged Integrated Circuit (IC), normally cut from a wafer.
digital	A signal or function, the amplitude of which is characterized by one of two discrete values: '0' or '1'.
digital blocks	The 8-bit logic blocks that can act as a counter, timer, serial receiver, serial transmitter, CRC generator, pseudo-random number generator, or SPI.
digital logic	A methodology for dealing with expressions containing two-state variables that describe the behavior of a circuit or system.
digital-to-analog (DAC)	A device that changes a digital signal to an analog signal of corresponding magnitude. The analog-to-digital (ADC) converter performs the reverse operation.
direct access	The capability to obtain data from a storage device, or to enter data into a storage device, in a sequence independent of their relative positions by means of addresses that indicate the physical location of the data.
duty cycle	The relationship of a clock period high time to its low time , expressed as a percent.

E

emulator	Duplicates (provides an emulation of) the functions of one system with a different system, so that the second system appears to behave similar to the first system.
External Reset (XRES)	An active high signal that is driven into the PSoC device. It causes all operation of the CPU and blocks to stop and return to a pre-defined state.

F

falling edge	A transition from a logic 1 to a logic 0. Also known as a negative edge.
feedback	The return of a portion of the output, or processed portion of the output, of a (usually active) device to the input.
filter	A device or process by which certain frequency components of a signal are attenuated.

firmware	The software that is embedded in a hardware device and executed by the CPU. The software may be executed by the end user but it may not be modified.
flag	Any of various types of indicators used for identification of a condition or event (for example, a character that signals the termination of a transmission).
Flash	An electrically programmable and erasable, non volatile technology that provides users with the programmability and data storage of EPROMs, plus in-system erasability. Nonvolatile means that the data is retained when power is off.
Flash bank	A group of Flash ROM blocks where Flash block numbers always begin with '0' in an individual Flash bank. A Flash bank also has its own block level protection information.
Flash block	The smallest amount of Flash ROM space that may be programmed at one time and the smallest amount of Flash space that may be protected. A Flash block holds 64 bytes.
flip-flop	A device having two stable states and two input terminals (or types of input signals) each of which corresponds with one of the two states. The circuit remains in either state until it is made to change to the other state by application of the corresponding signal.
frequency	The number of cycles or events per unit of time, for a periodic function.

G

gain	The ratio of output current, voltage, or power to input current, voltage, or power, respectively. Gain is usually expressed in dB.
ground	1. The electrical neutral line having the same potential as the surrounding earth. 2. The negative side of DC power supply. 3. The reference point for an electrical system. 4. The conducting paths between an electric circuit or equipment and the earth, or some conducting body serving in place of the earth.

H

hardware	A comprehensive term for all of the physical parts of a computer or embedded system, as distinguished from the data it contains or operates on, and the software that provides instructions for the hardware to accomplish tasks.
hardware reset	A reset that is caused by a circuit, such as a POR, watchdog reset, or external reset. A hardware reset restores the state of the device as it was when it was first powered up. Therefore, all registers are set to the POR value as indicated in register tables throughout this manual.

hexadecimal

A base 16 numeral system (often abbreviated and called hex), usually written using the symbols 0-9 and A-F. It is a useful system in computers because there is an easy mapping from four bits to a single hex digit. Thus, one can represent every byte as two consecutive hexadecimal digits. Compare the binary, hex, and decimal representations:

bin	=	hex	=	dec
0000b	=	0x0	=	0
0001b	=	0x1	=	1
0010b	=	0x2	=	2
...				
1001b	=	0x9	=	9
1010b	=	0xA	=	10
1011b	=	0xB	=	11
...				
1111b	=	0xF	=	15

So the decimal numeral 79 whose binary representation is 0100 1111b can be written as 4Fh in hexadecimal (0x4F).

high time

The amount of time the signal has a value of '1' in one period, for a periodic digital signal.

I

I2C

A two-wire serial computer bus by Philips Semiconductors (now NXP Semiconductors). I2C is an inter-integrated circuit. It is used to connect low-speed peripherals in an embedded system. The original system was created in the early 1980s as a battery control interface, but it was later used as a simple internal bus system for building control electronics. I2C uses only two bidirectional pins, clock and data, both running at +5V and pulled high with resistors. The bus operates at 100 kbits/second in standard mode and 400 kbits/second in fast mode.

ICE

The in-circuit emulator that allows users to test the project in a hardware environment, while viewing the debugging device activity in a software environment (PSoC Designer™).

idle state

A condition that exists whenever user messages are not being transmitted, but the service is immediately available for use.

impedance

1. The resistance to the flow of current caused by resistive, capacitive, or inductive devices in a circuit.
2. The total passive opposition offered to the flow of electric current. Note the impedance is determined by the particular combination of resistance, inductive reactance, and capacitive reactance in a given circuit.

input

A point that accepts data in a device, process, or channel.

input/output (I/O)

A device that introduces data into or extracts data from a system.

instruction

An expression that specifies one operation and identifies its operands, if any, in a programming language such as C or assembly.

integrated circuit (IC)

A device in which components such as resistors, capacitors, diodes, and **transistors** are formed on the surface of a single piece of semiconductor.

interface

The means by which two systems or devices are connected and interact with each other.

interrupt A suspension of a process, such as the execution of a computer program, caused by an event external to that process and performed in such a way that the process can be resumed.

interrupt service routine (ISR) A block of code that normal code execution is diverted to when the M8C receives a hardware interrupt. Many interrupt sources may each exist with its own priority and individual ISR code block. Each ISR code block ends with the RETI instruction, returning the device to the point in the program where it left normal program execution.

J

jitter

1. A misplacement of the timing of a transition from its ideal position. A typical form of corruption that occurs on serial data streams.
2. The abrupt and unwanted variations of one or more signal characteristics, such as the interval between successive pulses, the amplitude of successive cycles, or the frequency or phase of successive cycles.

K

keeper A circuit that holds a signal to the last driven value, even when the signal becomes un-driven.

L

latency The time or delay that it takes for a signal to pass through a given circuit or network.

least significant bit (LSb) The binary digit, or bit, in a binary number that represents the least significant value (typically the right-hand bit). The bit versus byte distinction is made by using a lower case "b" for bit in LSb.

least significant byte (LSB) The byte in a multi-byte word that represents the least significant value (typically the right-hand byte). The byte versus bit distinction is made by using an upper case "B" for byte in LSB.

Linear Feedback Shift Register (LFSR) A shift register whose data input is generated as an **XOR** of two or more elements in the register chain.

load The electrical demand of a process expressed as power (watts), current (amps), or resistance (ohms).

logic function A mathematical function that performs a digital operation on digital data and returns a digital value.

lookup table (LUT) A logic block that implements several logic functions. The logic function is selected by means of select lines and is applied to the inputs of the block. For example: A 2 input LUT with 4 select lines can be used to perform any one of 16 logic functions on the two inputs resulting in a single logic output. The LUT is a combinational device; therefore, the input/output relationship is continuous, that is, not sampled.

low time The amount of time the signal has a value of '0' in one period, for a periodic digital signal.

low voltage detect (LVD) A circuit that senses Vdd and provides an interrupt to the system when Vdd falls below a selected threshold.

M

M8C	An 8-bit Harvard Architecture microprocessor. The microprocessor coordinates all activity inside the PSoC device by interfacing to the Flash, SRAM, and register space.
macro	A programming language macro is an abstraction whereby a certain textual pattern is replaced according to a defined set of rules. The interpreter or compiler automatically replaces the macro instance with the macro contents when an instance of the macro is encountered. Therefore, if a macro is used 5 times and the macro definition required 10 bytes of code space, 50 bytes of code space are needed in total.
mask	1. To obscure, hide, or otherwise prevent information from being derived from a signal. It is usually the result of interaction with another signal, such as noise, static, jamming, or other forms of interference. 2. A pattern of bits that can be used to retain or suppress segments of another pattern of bits in computing and data processing systems.
master device	A device that controls the timing for data exchanges between two devices. Or when devices are cascaded in width, the master device is the one that controls the timing for data exchanges between the cascaded devices and an external interface. The controlled device is called the slave device .
microcontroller	An integrated circuit chip that is designed primarily for control systems and products. In addition to a CPU, a microcontroller typically includes memory, timing circuits, and I/O circuitry. The reason for this is to permit the realization of a controller with a minimal quantity of chips, thus achieving maximal possible miniaturization. This, in turn, reduces the volume and the cost of the controller. The microcontroller is normally not used for general-purpose computation as is a microprocessor.
mixed signal	The reference to a circuit containing both analog and digital techniques and components.
mnemonic	1. A tool intended to assist the memory. Mnemonics rely on not only repetition to remember facts, but also on creating associations between easy-to-remember constructs and lists of data. 2. A two to four character string representing a microprocessor instruction.
mode	A distinct method of operation for software or hardware. For example, the digital block may be in either counter mode or timer mode.
modulation	A range of techniques for encoding information on a carrier signal, typically a sine-wave signal. A device that performs modulation is known as a modulator.
Modulator	A device that imposes a signal on a carrier.
MOS	An acronym for metal-oxide semiconductor.
most significant bit (MSb)	The binary digit, or bit, in a binary number that represents the most significant value (typically the left-hand bit). The bit versus byte distinction is made by using a lower case "b" for bit in MSb.
most significant byte (MSB)	The byte in a multi-byte word that represents the most significant value (typically the left-hand byte). The byte versus bit distinction is made by using an upper case "B" for byte in MSB.

- multiplexer (mux)**
1. A logic function that uses a binary value, or address, to select between a number of inputs and conveys the data from the selected input to the output.
 2. A technique which allows different input (or output) signals to use the same lines at different times, controlled by an external signal. Multiplexing is used to save on wiring and I/O ports.

N

NAND See **Boolean Algebra**.

negative edge A transition from a logic 1 to a logic 0. Also known as a falling edge.

net The routing between devices.

nibble A group of four bits, which is one-half of a byte.

noise

1. A disturbance that affects a signal and that may distort the information carried by the signal.
2. The random variations of one or more characteristics of any entity such as voltage, current, or data.

NOR See **Boolean Algebra**.

NOT See **Boolean Algebra**.

O

OR See **Boolean Algebra**.

oscillator A circuit that may be crystal controlled and is used to generate a clock frequency.

output The electrical signal or signals which are produced by an analog or digital block.

P

parallel The means of communication in which digital data is sent multiple bits at a time, with each simultaneous bit being sent over a separate line.

parameter Characteristics for a given block that have either been characterized or may be defined by the designer.

parameter block A location in memory where parameters for the SSC instruction are placed prior to execution.

parity A technique for testing transmitting data. Typically, a binary digit is added to the data to make the sum of all the digits of the binary data either always even (even parity) or always odd (odd parity).

path

1. The logical sequence of instructions executed by a computer.
2. The flow of an electrical signal through a circuit.

pending interrupts An interrupt that has been triggered but has not been serviced, either because the processor is busy servicing another interrupt or global interrupts are disabled.

phase	The relationship between two signals, usually the same frequency, that determines the delay between them. This delay between signals is either measured by time or angle (degrees).
Phase-Locked Loop (PLL)	An electronic circuit that controls an oscillator so that it maintains a constant phase angle relative to a reference signal.
pin	A terminal on a hardware component. Also called lead.
pinouts	The pin number assignment: the relation between the logical inputs and outputs of the PSoC device and their physical counterparts in the printed circuit board (PCB) package. Pinouts involves pin numbers as a link between schematic and PCB design (both being computer generated files) and may also involve pin names.
port	A group of pins, usually eight.
positive edge	A transition from a logic 0 to a logic 1. Also known as a rising edge.
posted interrupts	An interrupt that has been detected by the hardware but may or may not be enabled by its mask bit. Posted interrupts that are not masked become pending interrupts.
Power On Reset (POR)	A circuit that forces the PSoC device to reset when the voltage is below a pre-set level. This is one type of hardware reset .
program counter	The instruction pointer (also called the program counter) is a register in a computer processor that indicates where in memory the CPU is executing instructions. Depending on the details of the particular machine, it holds either the address of the instruction being executed or the address of the next instruction to be executed.
protocol	A set of rules. Particularly the rules that govern networked communications.
PSoC®	Cypress Semiconductor's Programmable System-on-Chip (PSoC) mixed-signal array. PSoC® is a trademark of Cypress.
PSoC blocks	See analog blocks and digital blocks .
PSoC Designer™	The software for Cypress' Programmable System-on-Chip technology.
pulse	A rapid change in some characteristic of a signal (for example, phase or frequency) from a baseline value to a higher or lower value, followed by a rapid return to the baseline value.
pulse width modulator (PWM)	An output in the form of duty cycle which varies as a function of the applied measure.

R

RAM	An acronym for random access memory. A data-storage device from which data can be read out and new data can be written in.
register	A storage device with a specific capacity, such as a bit or byte.
reset	A means of bringing a system back to a known state. See hardware reset and software reset .
resistance	The resistance to the flow of electric current measured in ohms for a conductor.

revision ID	A unique identifier of the PSoC device.
ripple divider	An asynchronous ripple counter constructed of flip-flops. The clock is fed to the first stage of the counter. An n-bit binary counter consisting of n flip-flops that can count in binary from 0 to $2^n - 1$.
rising edge	See positive edge .
ROM	An acronym for read only memory. A data-storage device from which data can be read out, but new data cannot be written in.
routine	A block of code, called by another block of code, that may have some general or frequent use.
routing	Physically connecting objects in a design according to design rules set in the reference library.
runt pulses	In digital circuits, narrow pulses that, due to non-zero rise and fall times of the signal, do not reach a valid high or low level. For example, a runt pulse may occur when switching between asynchronous clocks or as the result of a race condition in which a signal takes two separate paths through a circuit. These race conditions may have different delays and are then recombined to form a glitch or when the output of a flip-flop becomes metastable.
S	
sampling	The process of converting an analog signal into a series of digital values or reversed.
schematic	A diagram, drawing, or sketch that details the elements of a system, such as the elements of an electrical circuit or the elements of a logic diagram for a computer.
seed value	An initial value loaded into a linear feedback shift register or random number generator.
serial	1. Pertaining to a process in which all events occur one after the other. 2. Pertaining to the sequential or consecutive occurrence of two or more related activities in a single device or channel.
set	To force a bit/register to a value of logic 1.
settling time	The time it takes for an output signal or value to stabilize after the input has changed from one value to another.
shift	The movement of each bit in a word, one position to either the left or right. For example, if the hex value 0x24 is shifted one place to the left, it becomes 0x48. If the hex value 0x24 is shifted one place to the right, it becomes 0x12.
shift register	A memory storage device that sequentially shifts a word either left or right to output a stream of serial data.
sign bit	The most significant binary digit, or bit, of a signed binary number. If set to a logic 1, this bit represents a negative quantity.
signal	A detectable transmitted energy that can be used to carry information. As applied to electronics, any transmitted electrical impulse.
silicon ID	A unique identifier of the PSoC silicon.

skew	The difference in arrival time of bits transmitted at the same time, in parallel transmission.
slave device	A device that allows another device to control the timing for data exchanges between two devices. Or when devices are cascaded in width, the slave device is the one that allows another device to control the timing of data exchanges between the cascaded devices and an external interface. The controlling device is called the master device.
software	A set of computer programs, procedures, and associated documentation concerned with the operation of a data processing system (for example, compilers, library routines, manuals, and circuit diagrams). Software is often written first as source code and then converted to a binary format that is specific to the device on which the code is executed.
software reset	A partial reset executed by software to bring part of the system back to a known state. A software reset restores the M8C to a known state but not blocks, systems, peripherals, or registers. For a software reset, the CPU registers (CPU_A, CPU_F, CPU_PC, CPU_SP, and CPU_X) are set to 0x00. Therefore, code execution begins at Flash address 0x0000.
SRAM	An acronym for static random access memory. A memory device allowing users to store and retrieve data at a high rate of speed. The term static is used because, once a value has been loaded into an SRAM cell, it remains unchanged until it is explicitly altered or until power is removed from the device.
SROM	An acronym for supervisory read only memory. The SROM holds code that is used to boot the device, calibrate circuitry, and perform Flash operations. The functions of the SROM may be accessed in normal user code, operating from Flash.
stack	A stack is a data structure that works on the principle of Last In First Out (LIFO). This means that the last item put on the stack is the first item that can be taken off.
stack pointer	A stack may be represented in a computer's inside blocks of memory cells, with the bottom at a fixed location and a variable stack pointer to the current top cell.
state machine	The actual implementation (in hardware or software) of a function that can be considered to consist of a set of states through which it sequences.
sticky	A bit in a register that maintains its value past the time of the event that caused its transition has passed.
stop bit	A signal following a character or block that prepares the receiving device to receive the next character or block.
switching	The controlling or routing of signals in circuits to execute logical or arithmetic operations, or to transmit data between specific points in a network.
switch phasing	The clock that controls a given switch, PHI1 or PHI2, in respect to the switched capacitor (SC) blocks. The SC blocks have two groups of switches. One group of these switches is normally closed during PHI1 and open during PHI2. The other group is open during PHI1 and closed during PHI2. These switches can be controlled in the normal operation, or in reverse mode if the PHI1 and PHI2 clocks are reversed.
synchronous	1. A signal whose data is not acknowledged or acted upon until the next active edge of a clock signal. 2. A system whose operation is synchronized by a clock signal.

T

tap	The connection between two blocks of a device created by connecting several blocks/components in a series, such as a shift register or resistive voltage divider.
terminal count	The state at which a counter is counted down to zero.
threshold	The minimum value of a signal that can be detected by the system or sensor under consideration.
transistors	The transistor is a solid-state semiconductor device used for amplification and switching, and has three terminals. A small current or voltage applied to one terminal controls the current through the other two. It is the key component in all modern electronics. In digital circuits, transistors are used as very fast electrical switches, and arrangements of transistors can function as logic gates, RAM-type memory, and other devices. In analog circuits, transistors are essentially used as amplifiers.
tri-state	A function whose output can adopt three states: 0, 1, and Z (high-impedance). The function does not drive any value in the Z state and, in many respects, may be considered to be disconnected from the rest of the circuit, allowing another output to drive the same net .

U

UART	A UART or universal asynchronous receiver-transmitter translates between parallel bits of data and serial bits.
user	The person using the PSoC device and reading this manual.
user modules	Pre-build, pre-tested hardware/firmware peripheral functions that take care of managing and configuring the lower level analog and digital blocks. User modules also provide high level API (Application Programming Interface) for the peripheral function.
user space	The bank 0 space of the register map. The registers in this bank are more likely to be modified during normal program execution and not just during initialization. Registers in bank 1 are most likely to be modified only during the initialization phase of the program.

V

Vdd	A name for a power net meaning "voltage drain." The most positive power supply signal. Usually 5 or 3.3 volts.
volatile	Not guaranteed to stay the same value or level when not in scope.
Vss	A name for a power net meaning "voltage source." The most negative power supply signal.

W

watchdog timer A timer that must be serviced periodically. If it is not serviced, the CPU resets after a specified period of time.

waveform The representation of a signal as a plot of amplitude versus time.

X

XOR See ***Boolean Algebra***.

Index



Numerics

16-Pin Part Pinout 19
24-Pin Part Pinout 20
32 kHz clock selection 67
32-Pin Part Pinout 21
48-Pin Part Pinout 22

A

ACK bit 240
acronyms 18
Address bits
 in I2C_SCR register 240
address spaces, CPU core 27
AMUX_CFG register 204
Analog bit
 in INT_CLR0 register 243
 in INT_MSK0 register 250
analog input, GPIO 56
architecture
 PSoC core 23
 system resources 105
 top level 14
 TrueTouch system 83

B

bank 0 registers 188
 register mapping table 184
bank 1 registers 259
 register mapping table 185
basic paging in RAM paging 39
Bias Trim bits in ILO_TR register 282
Bus Error bit 240
Bypass bit 261
Byte Complete bit 240

C

Calibrate0 function in SROM 37
Calibrate1 function in SROM 37
Carry bit 254
CDSx bits 208
CHAIN bit 212
Checksum function in SROM 37
CIN_EN bit 213

CINTx bits 208
CIP_EN bit 213
CLKSEL bits 212
Clock Phase bit in SPI_CR 192
Clock Polarity bit 192
Clock Rate bits 239
Clock Sel bit 261
clock, external digital 110
 switch operation 110
clocks digital, See digital clocks
CMP_CR0 register 104, 207
CMP_CR1 register 104, 208
CMP_LUT register 104, 210
CMP_MUX register 103, 206
CMP_RDC register 103, 205
CMP0D bit 205
CMP0L bit 205
CMP1D bit 205
CMP1L bit 205
CMPxEN bits 207
COHM bit 217
COHS bit 217
COLM bit 217
COLS bit 217
comparators 101
 architecture 101
 register definitions 103
configuration register in SPI
 SPI_CFG register 150
control register in SPI
 SPI_CR register 149
conventions, documentation
 acronyms 18
 numeric naming 17
 register conventions 183
core, See PSoC core
CPINx bits 208
CPU core 27
 instruction formats 30
 instruction set summary 28–29
 overview 27
 register definitions 32
CPU Speed bits 276
CPU_F register 32, 254
CPU_SCR0 register 138, 258
CPU_SCR1 register 65, 257
CRSTx bits 208
CS_CNTH register 94, 216

CS_CNTL register 94, 215
 CS_CR0 register 92, 211
 CS_CR1 register 93, 212
 CS_CR2 register 93, 213
 CS_CR3 register 94, 214
 CS_SLEW register 219, 220
 CS_STAT register 95, 217
 CS_TIMER register 95, 218
 CSD_CS_CLK bit in CS_CR0 register 211
 CSD_MODE bit in CS_CR0 register 211
 CSD_PRCLK bit in CS_CR0 register 211
 CSOUT bits in CS_CR0 register 211
 CUR_PP register 42, 234
 current page pointer in RAM paging 40

D

DATA 222, 223
 DATA bits
 in PT_DATA0 register 222, 223
 Data bits
 in CS_CNTH register 94, 216
 in CS_CNTL register 94, 215
 in I2C_DR register 233, 241
 in PRTxDR register 59, 188
 in SPI_RXR register 148, 191
 in SPI_TXR register 147, 190
 in TMP_DRx register 42, 266
 data bypass in GPIO 58
 data registers in SPI 147
 development kits 16
 digital clocks 109
 architecture 109
 internal low speed oscillator 110
 internal main oscillator 109
 register definitions 112
 system clocking signals 109
 digital IO, GPIO 56
 documentation
 history 16
 overview 13
 Drive Mode 0 bits 259
 Drive Mode 1 bits 260

E

EN bit in CS_CR0 register 211
 Enable bit 239
 ENABLE bits
 in MUX_CRx registers 271
 in SPI_CR register 192
 ENSWINT bit 52, 251
 EP0_CNT register 174, 199
 EP0_CR register 173, 177
 EP0_DRx register 174, 200
 EPx_CNT1 register 175, 176
 EraseAll function in SROM 36
 EraseBlock function in SROM 36

EXTCLKEN bit 278
 external digital clock 110
 external reset 139

F

FastSlew bits 219
 Flash
 memory organization 35
 Freq Trim bits for ILO_TR 282
 FS_EN bit 219
 full-speed USB 165
 architecture 165
 memory arbiter 166
 register definitions 171
 suspend mode 168
 USB SIE 165
 USB SRAM 166

G

general purpose IO 55
 analog and digital input 56
 architecture 55
 block interrupts 57
 data bypass 58
 digital IO 56
 drive modes 60
 interrupt modes 57
 port 1 distinctions 56
 register definitions 59
 GIE bit 255
 GIES bit 258
 GPIO bit
 in INT_CLR0 register 242
 in INT_MSK0 register 250
 GPIO block interrupts 57
 GPIO, *See* general purpose IO

H

help, getting
 development kits 16
 support 16
 upgrades 16

I

I2C bit
 in INT_CLR0 register 242
 in INT_MSK0 register 250
 I2C slave 117
 application overview 118
 architecture 117
 basic data transfer 118
 basic IO timing 130
 clock generation timing 130
 operation 118
 register definitions 122

- stall timing 132
 - status timing 131
- I2C_CFG register 126, 239
- I2C_DR register 129, 241
- I2C_SCR register 128, 240
- ICAPEN bit 204
- IDAC_D register 97, 256
- IDACDATA bits 256
- IDACEN bit 213
- IDX_PP register 43, 236
- ILO, *See* internal low speed oscillator
- IMO, *See* internal main oscillator
- IMO_TR register 281
- IMO_TR1 register 181
- IMODIS bit 278
- index memory page pointer in RAM paging 41
- INM bit 217
- INNx bits 206
- INPx bits 206
- INS bit 217
- INSEL bit 212
- instruction formats
 - 1-byte instructions 30
 - 2-byte instructions 30
 - 3-byte instructions 31
- instruction set summary 28–29
- Int Sel bit 261
- INT_CLR0 register 48, 49, 50, 71, 242, 244, 245
- INT_MSK0 register 51, 52, 248, 249, 250
- INT_SW_EN register 52, 251
- INT_VC register 52, 252
- INTCAP bits 204
- internal low speed oscillator 67
 - 32 kHz clock selection 67
 - architecture 67
 - in digital clocks 110
 - register definitions 68
- internal M8C registers 27
- internal main oscillator 63
 - architecture 63
 - in digital clocks 109
 - register definitions 64
- interrupt controller 45
 - application overview 46
 - architecture 45
 - interrupt table 47
 - latency and priority 46
 - posted vs pending interrupts 46
 - register definitions 48
- Interrupt Enables bits 189
- interrupt modes in GPIO 57
- interrupt table 47
- interrupts in RAM paging 40
- INV bit 212
- IO analog multiplexer 99
 - architecture 99
 - register definitions 100
- IO_CFG register 61, 272
- IRAMDIS bit 257
- IRANGE bit 213

IRESS bit 257

L

- low voltage detect (LVD)
 - See* POR and LVD
- LRB bit 240
- LSb First bit 192
- LUTx bits 210
- LVD bit 280
- LVDTBEN bits 279

M

- M8C, *See* CPU core
- mapping tables, registers 183
- master function for SPI 146
- measurement units 17
- memory arbiter in USB 166
- MODE bits in CS_CR0 register 211
- MUX_CRx register 100, 270, 271, 277
- MVI instructions in RAM paging 40
- MVR_PP register 43, 237
- MVW_PP register 44, 238

N

- No Buzz bit 276
- numeric naming conventions 17

O

- One Shot bit 221, 224, 225
- OSC_CR0 register 276
- OSC_CR2 register 65, 278
- OUT_P1 register 113, 273, 274
- Overrun bit 192
- overviews
 - PSoC core 23
 - register tables 183
 - system resources 105
 - TrueTouch system 83

P

- P10D bit 274
- P10EN bit 274
- P12D bit 273
- P12EN bit 273
- P14D bit 273
- P14EN bit 273
- P16D bit 273
- P16EN bit 273
- Page bits
 - in CUR_PP register 234
 - in IDX_PP register 236

- in MVR_PP register 237
 - in MVW_PP register 238
 - in STK_PP register 235
- Pending Interrupt bits 252
- PgMode bits 254
- pin behavior during reset 135
- pin information, *See* pinouts
- pinouts
 - 16-pin part 19
 - 24-pin part 20
 - 32-pin part 21
 - 48-pin part 22
- PMAx_CUR register 180
- POR and LVD 143
 - architecture 143
 - register definitions 144
- PORLEV bits 279
- PORS bit 258
- power modes
 - system resets 141
- power on reset (POR)
 - See* POR and LVD
- power on reset in system resets 139
- PPM bit 217
- PPS bit 217
- product upgrades 16
- programmable timer 161
 - architecture 161
 - register definitions 163
- ProtectBlock function in SROM 36
- protocol function for SPI 145
- PRTxDM0 register 60, 259
- PRTxDM1 register 60, 260
- PRTxDR register 59, 188
- PRTxIE register 59, 189
- PSelect bit 239
- PSoC core
 - architecture 23
 - overview 14
 - register summary 24
 - See also* CPU core
- PSSDC bits 283, 284, 285
- PT_CFG register 163, 164, 221
- PT_DATA1 register 164
- PTx_DATA0 register 164, 223
- PTx_DATA1 register 164, 222
- PXD_EN bit 213

R

- RAM paging 39
 - architecture 39
 - basic paging 39
 - current page pointer 40
 - index memory page pointer 41
 - interrupts 40
 - MVI instructions 40
 - register definitions 42
 - stack operations 40
- ReadBlock function in SROM 35

- reference of all registers 187
- register conventions 17, 187
- register definitions
 - comparators 103
 - CPU core 32
 - digital clocks 112
 - general purpose IO 59
 - I2C slave 122
 - internal low speed oscillator 68
 - internal main oscillator 64
 - interrupt controller 48
 - IO analog multiplexer 100
 - POR and LVD 144
 - programmable timer 163
 - RAM paging 42
 - sleep and watchdog 77
 - SPI 147
 - supervisory ROM 38
 - system resets 137
 - TrueTouch module 92
 - USB, full-speed 171
- register mapping tables
 - bank 0 registers 184
 - bank 1 registers 185
- registers
 - bank 0 registers 188
 - bank 1 registers 259
 - core register summary 24
 - internal M8C registers 27
 - maneuvering around 187
 - mapping tables 183
 - reference of all registers 187
 - system resources register summary 106
 - TrueTouch register summary 84
- regulated IO
 - application overview 70
 - architecture 69
- RES_WDT register 77, 253
- RLOCK bit 212
- RX Reg Full bit 192

S

- serial peripheral interconnect, *See* SPI
- Slave bit 261
- slave function for SPI 146
- slave operation, I2C 118
- sleep and watchdog 73
 - application overview 76
 - architecture 73
 - bandgap refresh 80
 - register definitions 77
 - sleep sequence 79
 - sleep timer 76
 - timing diagrams 79
 - wake up sequence 80
 - watchdog timer 81
- Sleep bit
 - in CPU_SCR0 register 258
 - in INT_CLR0 register 242
 - in INT_MSK0 register 250

Sleep bits 276
 sleep timer 76
 SLIMO bit 257
 SLP_CFG register 77, 283, 284, 285
 SPI 145
 architecture 145
 configuration register 150
 control register 149
 data registers 147
 master data register definitions 148
 master function 146
 protocol function 145
 register definitions 147
 slave data register definitions 148
 slave function 146
 timing diagrams 151
 SPI bit
 in INT_CLR0 register 242
 in INT_MSK0 register 250
 SPI Complete bit 192
 SPI timing diagrams
 SPI mode timing 151
 SPIM timing 152
 SPIS timing 157
 SPI_CFG register 150, 261
 SPI_CR register 149, 192
 SPI_RXR register 148, 191
 SPI_TXR register 147, 190
 SRAM with USB 166
 SROM, *See* supervisory ROM
 SS_bit 261
 SS_EN_bit 261
 stack operations in RAM paging 40
 START bit 221, 224, 225
 start, how to 16
 STK_PP register 43, 235
 STOP bit 258
 Stop IE bit 239
 Stop Status bit 240
 summary of registers
 mapping tables 183
 PSoC core 24
 system resources 106
 TrueTouch system 84
 supervisory ROM
 architecture 33
 Calibrate0 function 37
 Calibrate1 function 37
 EraseAll function 36
 EraseBlock function 36
 function descriptions 34
 ProtectBlock function 36
 ReadBlock function 35
 SWBootReset function 34
 TableRead function 36
 WriteBlock function 35
 SWBootReset function in SROM 34
 switch operation in digital clocks 110
 system resets 135
 architecture 135
 external reset 139

 functional details 141
 pin behavior 135
 power modes 141
 power on reset 139
 register definitions 137
 timing diagrams 139
 watchdog timer reset 139
 system resources
 architecture 105
 overview 14, 105
 register summary 106
T
 TableRead function in SROM 36
 technical support 16
 Timer1/0 bits
 in INT_CLR0 register 243
 in INT_MSK0 register 250
 timing diagrams
 I2C slave 130
 sleep and watchdog 79
 SPI 151
 system resets 139
 TrueTouch module 97
 TMP_DRx register 42, 205
 Transmit bit 240
 Trim bits in IMO_TR register 281, 286
 TrueTouch counter 90
 TrueTouch module 85
 architecture 85
 register definitions 92
 timing diagram 97
 types of approaches 85
 TrueTouch system
 architecture 83
 overview 14, 83
 register summary 84
 TX Reg Empty bit 192

U
 units of measure 17
 upgrades 16
 USB 195
 USB, *See* full-speed USB
 USB_CR0 register 171, 195
 USB_CR1 register 180, 262
 USB_SOFx register 171, 193
 USBIO_CR0 register 172, 196
 USBIO_CR1 register 172, 197

V
 V Monitor bit
 in INT_CLR0 register 243, 245
 in INT_MSK0 register 250
 VLT_CMP register 144, 280
 VLT_CR register 144, 279

Index

VM bits 279

W

watchdog timer reset 139

WDRS bit 258

WDSL_Clear bits 253

WriteAndVerify function in SROM 37

WriteBlock function in SROM 35

X

XIO bit 254

XRES reset 139

Z

Zero bit 255