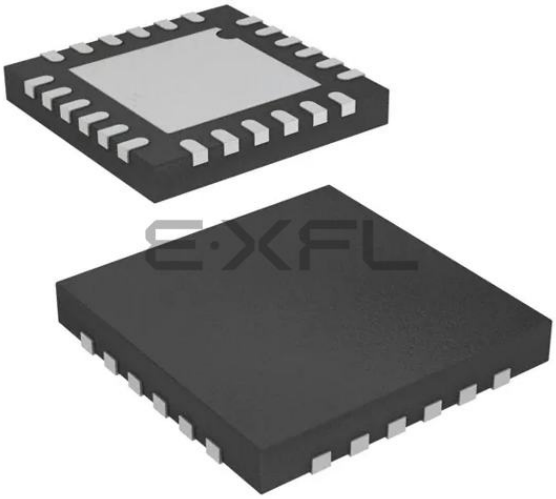


Welcome to E-XFL.COM

Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

What Are Embedded - Microcontrollers - Application Specific?

Application-specific microcontrollers are engineered to

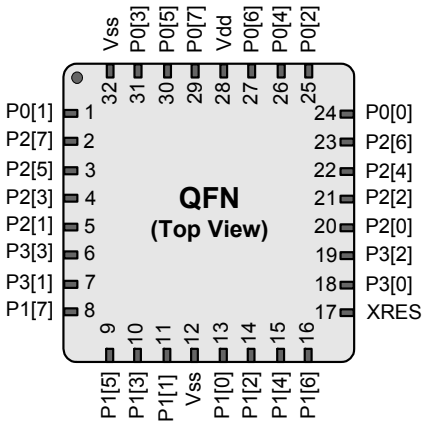
Details	
Product Status	Obsolete
Applications	Touchscreen Controller
Core Processor	M8C
Program Memory Type	FLASH (32kB)
Controller Series	CY8CT
RAM Size	2K x 8
Interface	I²C, SPI, UART/USART, USB
Number of I/O	20
Voltage - Supply	1.8V
Operating Temperature	-40°C ~ 85°C
Mounting Type	Surface Mount
Package / Case	24-UFQFN Exposed Pad
Supplier Device Package	24-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy8ctst200a-24lqxit

1.1.3 CY8CTMG200-32LQXI, CY8CTMG200A-32LQXI, CY8CTST200-32LQXI, CY8CTST200A-32LQXI, CY8CTMG201-32LQXI, CY8CTMG201A-32LQXI PSoC 32-Pin Part Pinout

Table 1-3. 32-Pin QFN Part Pinout **

Pin No.	Digital	Analog	Name	Description
1	IOH	I	P0[1]	Integrating input
2	IO	I	P2[7]	
3	IO	I	P2[5]	XTAL Out
4	IO	I	P2[3]	XTAL In
5	IO	I	P2[1]	
6	IO	I	P3[3]	
7	IO	I	P3[1]	
8	IOHR	I	P1[7]	I2C SCL, SPI SS
9	IOHR	I	P1[5]	I2C SDA, SPI MISO
10	IOHR	I	P1[3]	SPI CLK
11	IOHR	I	P1[1]	TC CLK*, I2C SCL, SPI MOSI
12	Power		Vss	Ground pin
13	IOHR	I	P1[0]	TC DATA*, I2C SDA, SPI CLK
14	IOHR	I	P1[2]	
15	IOHR	I	P1[4]	EXTCLK
16	IOHR	I	P1[6]	
17	Input		XRES	Active high external reset with internal pull down
18	IO	I	P3[0]	
19	IO	I	P3[2]	
20	IO	I	P2[0]	
21	IO	I	P2[2]	
22	IO	I	P2[4]	
23	IO	I	P2[6]	
24	IOH	I	P0[0]	
25	IOH	I	P0[2]	
26	IOH	I	P0[4]	
27	IOH	I	P0[6]	
28	Power		Vdd	Power pin
29	IOH	I	P0[7]	
30	IOH	I	P0[5]	
31	IOH	I	P0[3]	
32	Power		Vss	Ground pin

CY8CTMG200-32LQXI, CY8CTMG200A-32LQXI, CY8CTST200-32LQXI, CY8CTST200A-32LQXI, CY8CTMG201-32LQXI, CY8CTMG201A-32LQXI PSoC Devices



LEGEND A = Analog, I = Input, O = Output, NC = No Connection, H = 5 mA High Output Drive, R = Regulated Output Option.
* ISSP pin which is not High Z at POR (Power On Reset).
** The center pad on the QFN package must be connected to ground (Vss) for best mechanical, thermal, and electrical performance. If not connected to ground, it must be electrically floated and not connected to any other signal.

An `MVI A, [expr]` instruction is used to move data from SRAM into Flash. Therefore, use the `MVI` read pointer (MVR_PP register) to specify which SRAM page from which data is pulled. Using the `MVI` read pointer and the parameter blocks POINTER value allows the SROM WriteBlock function to move data from any SRAM page into any Flash block.

The SRAM address, the first of the 128 bytes to store in Flash, is indicated using the POINTER variable in the parameter block (SRAM address FBh).

Table 3-8. WriteBlock Parameters (02h)

Name	Address	Type	Description
MVR_PP	0,D4h	Register	`MVI` read page pointer register.
KEY1	0,F8h	RAM	3Ah.
KEY2	0,F9h	RAM	Stack Pointer value+3, when `SSC` is executed.
BLOCKID	0,FAh	RAM	Flash block number.
POINTER	0,FBh	RAM	First of 128 addresses in SRAM, where the data to be stored in Flash, is located before calling WriteBlock.

3.1.2.4 EraseBlock Function

The EraseBlock function is not recommended for use. The functionality is redundant with the WriteBlock and WriteAndVerify functions. The only practical use is for clearing all data in a 128 byte block of contiguous bytes in Flash to 0x00. If used, it should not be called repeatedly on the same block. It may be used between WriteAndVerify or WriteBlock operations.

If write protection is turned on, then the EraseBlock function exits, setting the accumulator and KEY2 back to 00h. KEY1 has a value of 01h, indicating a write failure.

To set up the parameter block for the EraseBlock function, store the correct key values in KEY1 and KEY2. The block number to erase must be stored in the BLOCKID variable.

Table 3-9. EraseBlock Parameters (03h)

Name	Address	Type	Description
KEY1	0,F8h	RAM	3Ah.
KEY2	0,F9h	RAM	Stack Pointer value+3, when SSC is executed.
BLOCKID	0,FAh	RAM	Flash block number.

3.1.2.5 ProtectBlock Function

The PSoC devices offer Flash protection on a block-by-block basis. Table 3-10 lists the protection modes available. In the table, ER and EW indicate the ability to perform external reads and writes (that is, by an external programmer). For internal writes, IW is used. Internal reading is always permitted by way of the `ROMX` instruction. An SR indicates the ability to read by way of the SROM ReadBlock function.

In this table, note that all protection is removed by EraseAll.

Table 3-10. Protect Block Modes

Mode	Settings	Description	In PSoC Designer
00b	SR ER EW IW	Unprotected	U = Unprotected
01b	$\overline{SR}$ ER EW IW	Read protect	F = Factory upgrade
10b	$\overline{SR}$ $\overline{ER}$ $\overline{EW}$ IW	Disable external write	R = Field upgrade
11b	$\overline{SR}$ $\overline{ER}$ $\overline{EW}$ $\overline{IW}$	Disable internal write	W = Full protection

Table 3-11. Protection Level Bit Packing

7	6	5	4	3	2	1	0
Block n+3		Block n+2		Block n+1		Block n	

3.1.2.6 TableRead Function

The TableRead function gives the user access to part-specific data stored in the Flash during manufacturing. The Flash for these tables is separate from the program Flash and is not directly accessible. It also returns a revision ID for the die (do not confuse this with the silicon ID stored in the Table 0 row in Table 3-14).

There are four 8-byte tables in the CY8CTMG20x, CY8CTST200 devices.

Table 3-12. TableRead Parameters (06h)

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

3.1.2.7 EraseAll Function

The EraseAll function performs a series of steps that destroys the user data in the Flash banks and resets the protection block in each Flash bank to all zeros (the unprotected state). This function is only executed by an external programmer. If EraseAll is executed from code, the M8C HALTs without touching the Flash or protections. See Table 3-13. The three other hidden blocks above the protection block, in each Flash bank, are not affected by the EraseAll.

Table 3-13. EraseAll Parameters (05h)

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

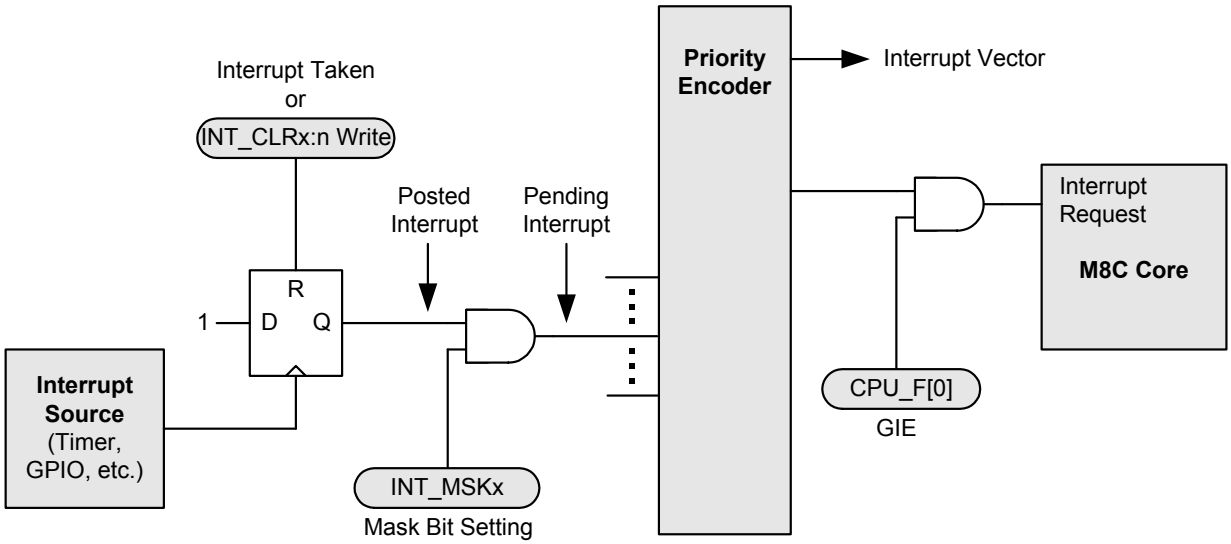
5. Interrupt Controller

This chapter presents the Interrupt Controller and its associated registers. The interrupt controller provides a mechanism for a hardware resource in PSoC devices to change program execution to a new address without regard to the current task being performed by the code being executed. For a quick reference of all PSoC registers in address order, refer to the Register Reference chapter on page 187.

5.1 Architectural Description

A block diagram of the Interrupt Controller is shown in Figure 5-1, illustrating the concepts of ***posted interrupts*** and ***pending interrupts***.

Figure 5-1. Interrupt Controller Block Diagram



This is the sequence of events that occur during interrupt processing.

1. An interrupt becomes active, either because (a) the interrupt condition occurs (for example, a timer expires), (b) a previously posted interrupt is enabled through an update of an interrupt ***mask*** register, or (c) an interrupt is pending and GIE is set from '0' to '1' in the CPU Flag register.
2. The current executing instruction finishes.
3. The internal interrupt service routine (ISR) executes, taking 13 cycles. During this time, the following actions occur:

- The PCH, PCL, and Flag register (CPU_F) are pushed onto the stack (in that order).
- The CPU_F register clears. Since this clears the GIE bit to '0', additional interrupts are temporarily disabled.
- The PCH (PC[15:8]) is cleared to zero.
- The interrupt vector is read from the interrupt controller and its value is placed into PCL (PC[7:0]). This sets the program counter to point to the appropriate address in the interrupt table (for example, 0014h for the GPIO interrupt).

7.3 Register Definitions

The following registers are associated with the Internal Main Oscillator (IMO). The register descriptions have associated register tables 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 showing all oscillator registers, refer to the Summary Table of the Core Registers on page 24.

7.3.1 IMO_TR Register

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

The Internal Main Oscillator Trim Register (IMO_TR) manually centers the oscillator's output to a target frequency.

This register is loaded with a factory trim value at boot. When changing frequency ranges, the matching frequency trim value must be loaded into this register.

A TableRead command to the Supervisory ROM returns the trim values to the SRAM. EraseAll Parameters (05h), on page 36 has information on the location of various trim settings stored in Flash tables. Firmware needs to read the right trim value for desired frequency and update the IMO_TR register. The IMO_TR register must be changed at the lower frequency range setting.

For additional information, refer to the IMO_TR register on page 281

7.3.2 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 : 0

The Internal Main Oscillator Trim Register 1 (IMO_TR1) adjusts the IMO frequency .

Bits 2 to 0: Fine Trim[2:0]. These bits provide a fine tuning capability to the IMO trim. These three bits are the 3 LSB of the IMO trim with the IMO_TR register supplying the 8 MSB. A larger value in this register will increase the speed of the oscillator. The value in these bits varies the IMO frequency: approximately 7.5 kHz/step. When the EnableLock bit is set in the USB_CR1 register, firmware writes to this register are disabled.

For additional information, refer to the IMO_TR1 register on page 286.

9.3 Register Definitions

These registers are associated with the external crystal oscillator.

9.3.1 ECO_ENBUS Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,D2h	ECO_ENBUS						ECO_ENBUS[2:0]			RW : 07

The ECO_ENBUS register is used to disable and enable the external crystal oscillator (ECO).

Bits 2 to 0 ECO_ENBUS[2:0]. 111b – Default. Disables the external crystal oscillator (ECO).

011b – Allows the ECO to be enabled by bits in the ECO_CFG register.

Other values are reserved. See the Application Overview on page 70 for the proper sequence for enabling the ECO.

For additional information, refer to the ECO_ENBUS register on page 269.

9.3.2 ECO_TRIM Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,D3h	ECO_TRIM				ECO_XGM[2:0]			ECO_LP[1:0]		RW : 11

The ECO TRIM Register (ECO_TRIM) controls gain and power settings for the 32 kHz crystal oscillator.

These settings should not be changed from their default state.

Bits 4 to 2: ECO_XGM[2:0]. These bits set the amplifier gain. The high power mode step size is approximately 220 nA. The low power step size is approximately 5% lower than the ‘111’ setting.

’000’ is the lowest power setting.

’111’ is the highest power setting (30% power reduction).

Bits 1 to 0: ECO_LP[1:0]. These bits set the gain mode.

’00’ is the highest power setting.

’11’ is the lowest power setting. (30% power reduction).

For additional information, refer to the ECO_TRIM register on page 270.

9.3.3 ECO_CFG Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E1h	ECO_CFG						ECO_LPM	ECO_EXW	ECO_EX	RW : 00

The ECO Configuration Register provides status and control for the ECO.

Bit 2 ECO_LPM. This bit enables the ECO low power mode when high. This is recommended for use only during sleep mode.

Bit 1 ECO_EXW. The ECO Exists Written bit is used as a status bit to indicate that the ECO EX bit was previously written to. It is read only. When this bit is a '1' indicates that the ECO_CFG register was written to and is now locked.

Bit 0 ECO_EX. The ECO Exists bit serves as a flag to the hardware, to indicate that an external crystal oscillator exists in the system. Just after boot, it may be written only once to a value of '1' (crystal exists) or '0' (crystal does not exist).

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 should be written as early as possible after a Power On Reset (POR) or External Reset (XRES) event.

For additional information, refer to the ECO_CFG register on page 277

10.1.1 Sleep Control Implementation Logic

This section details the sleep mode logic implementation.

Conditions for entering the sleep modes:

- Standby Mode: Set the SLEEP bit in the CPU_SCR0 register. This asserts the "sleep" signal for the sleep controller.
- I2C_USB Mode: Set the I2C_ON bit in the SLP_CFG2 register and then set the SLEEP bit in the CPU_SCR0 register. Another way to enter I2C_USB sleep mode is to set the USB Enable bit in the USB_CR0 register and then set the SLEEP bit in the CPU_SCR0 register. This asserts the sleep signal for the sleep controller and also the I2CEnable signal to the power system.

  The I2C block works in I2C_USB sleep mode only to wake up the system. That is, when the device is in sleep, I2C can detect a start condition and receive an address. If the address matches, I2C generates an interrupt and wakes the system (refer to Power Modes on page 141). If you put the device to sleep again while these transactions are ocurring (i.e., when you are in the middle of I2C transactions), I2C does not work and will send NACKs. I2C can only detect a start condition and collect an 8-bit address then wake the system through an interrupt during I2C sleep mode. Therefore it is recommended to check the bus status in the I2C_XSTAT register before putting the device to sleep if there is any I2C data transfer.
- Deep Sleep Mode: Configure the I2C_ON bit in the SLP_CFG2 register to 0, then USB Enable bit in the USB_CR0 register to '0' and the X32ON bit in OSC_CR0 to '0'. Set the LSO_OFF bit in the SLP_CFG2 register and then set the "SLEEP" bit in the CPU_SCR0 register. This enables the LSO_OFF signal to power down the LSO. The system enters into deep sleep mode. One point to note here is to not set the X32ON bit to '1' without setting the ECO_EX (ECO exists) bit in the ECO_CFG (1,E1h) register to a '1'. If you do so, the deep sleep mode is not entered, but clk32K is also not running. This implies that the sleep timer interrupt or the programmable timer interrupt cannot occur.

10.1.1.1 Wakeup Logic

- Waking up from standby mode is by an interrupt, which can be a sleep timer interrupt, a GPIO interrupt, a 16-bit programmable timer 0 interrupt, or a USB interrupt.
- For the device, the wakeup from I2C_USB sleep mode can be by an I2C interrupt in addition to a sleep timer interrupt, a programmable timer 0 interrupt, a GPIO interrupt, or a USB interrupt.
- For the device, the wakeup from deep sleep mode can be by either a GPIO interrupt or a USB interrupt.
- In standby mode during buzz, if the external supply falls below the LVD limit, an LVD interrupt occurs and initiates the wakeup sequence.
- In standby mode, if watchdog reset occurs, it first initiates the wakeup sequence. Once the wakeup is done, it resets the system.

Section D: System Resources

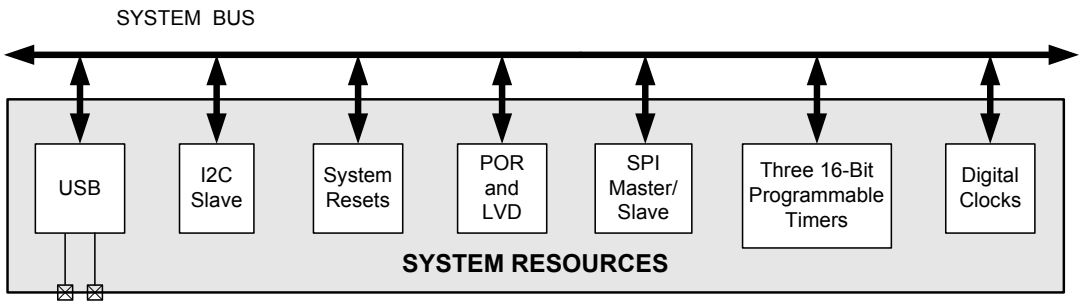
The System Resources section discusses the system resources that are available for the PSoC devices and the registers associated with those resources. This section encompasses the following chapters:

- Digital Clocks on page 109.
- I2C Slave on page 117.
- System Resets on page 135.
- POR and LVD on page 143.
- SPI on page 145.
- Programmable Timer on page 161.
- Full-Speed USB on page 165.

Top-Level System Resources Architecture

The figure below displays the top-level architecture of the PSoC system resources. Each component of the figure is discussed at length in the chapters that follow.

PSoC System Resources



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 I^2C 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 I^2C transfer, and control bits to determine the actions for the next byte transfer. At the end of each byte transfer, the I^2C hardware interrupts the M8C microcontroller and stalls the I^2C 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 I^2C bus traffic. According to the I^2C 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 I^2C 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.

20. Full-Speed USB

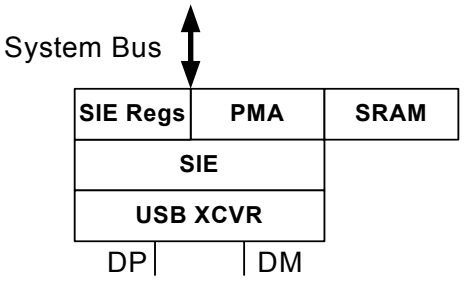
This chapter explains the Full-Speed USB (Universal Serial Bus) resource and its associated registers. For a quick reference of all PSoC registers in address order, refer to the Register Reference chapter on page 187.

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 in 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.

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.

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:

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:

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.

21.3.16 PMAx_DR

PSoC Memory Arbiter Data Registers

Individual Register Names and Addresses:

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.

21.3.29 CS_STAT

TrueTouch Status Register

Individual Register Names and Addresses:

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.

21.3.40 I2C_ADDR

I²C Slave Address Register

Individual Register Names and Addresses:

I2C_ADDR : 0,CAh

	7	6	5	4	3	2	1	0
Access : POR		RW : 00						
Bit Name		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.

21.4.5 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_WA : 1, 48H	PMA13_WA : 1,49h	PMA14_WA : 1,4Ah	PMA15_WA : 1,4Bh

	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.

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.