Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	M8C
Core Size	8-Bit
Speed	24MHz
Connectivity	SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	44
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.25V
Data Converters	A/D 1x8b, 1x11b, 1x12b; D/A 1x9b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-BSSOP (0.295", 7.50mm Width)
Supplier Device Package	48-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy8c26643-24pvxi

List of Figures

2.2.3 Index Register

Table 10: Index Register (CPU_X)

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/ Write	System[1]	System[1]	System[1]	System[1]	System[1]	System[1]	System[1]	System[1]
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]
Bit [7:0]: **Data [7:0]** 8-bit data value holds an index for any instruction that uses an indexed addressing mode								

1. System - not directly accessible by the user

2.2.4 Stack Pointer Register

Table 11: Stack Pointer Register (CPU_SP)

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/ Write	System[1]	System[1]	System[1]	System[1]	System[1]	System[1]	System[1]	System[1]
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]
Bit [7:0]: **Data [7:0]** 8-bit data value holds a pointer to the current top-of-stack								

1. System - not directly accessible by the user

2.2.5 Program Counter Register

Table 12: Program Counter Register (CPU_PC)

Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/ Write	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit Name	Data [15]	Data [14]	Data [13]	Data [12]	Data [11]	Data [10]	Data [9]	Data [8]	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]
Bit [15:0]: **Data [15:0]** 16-bit data value is the low-order/high-order byte of the Program Counter																

1. System - not directly accessible by the user

2.3 Addressing Modes

2.3.1 Source Immediate

The result of an instruction using this addressing mode is placed in the A register, the F register, the SP register, or the X register, which is specified as part of the instruction opcode. Operand 1 is an immediate value that serves as a source for the instruction. Arithmetic instructions require two sources. Instructions using this addressing mode are two bytes in length.

Table 13: Source Immediate

Opcode	Operand 1
Instruction	Immediate Value

6.0 I/O Registers

6.1 Port Data Registers

Table 28: Port Data Registers

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]
Bit [7:0]: **Data [7:0]** When written is the bits for output on port pins. When read is the state of the port pins								

Port 0 Data Register (PRT0DR, Address = Bank 0, 00h)
Port 1 Data Register (PRT1DR, Address = Bank 0, 04h)
Port 2 Data Register (PRT2DR, Address = Bank 0, 08h)
Port 3 Data Register (PRT3DR, Address = Bank 0, 0Ch)
Port 4 Data Register (PRT4DR, Address = Bank 0, 10h)
Port 5 Data Register (PRT5DR, Address = Bank 0, 14h) **Note**: Port 5 is 4-bits wide, Bit [3:0]

6.2 Port Interrupt Enable Registers

Table 29: Port Interrupt Enable Registers

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W
Bit Name	Int En [7]	Int En [6]	Int En [5]	Int En [4]	Int En [3]	Int En [2]	Int En [1]	Int En [0]
Bit [7:0]: **Int En [7:0]** When written sets the pin interrupt state 0 = Interrupt disabled for pin 1 = Interrupt enabled for pin								

Port 0 Interrupt Enable Register (PRT0IE, Address = Bank 0, 01h)
Port 1 Interrupt Enable Register (PRT1IE, Address = Bank 0, 05h)
Port 2 Interrupt Enable Register (PRT2IE, Address = Bank 0, 09h)
Port 3 Interrupt Enable Register (PRT3IE, Address = Bank 0, 0Dh)
Port 4 Interrupt Enable Register (PRT4IE, Address = Bank 0, 11h)
Port 5 Interrupt Enable Register (PRT5IE, Address = Bank 0, 15h) **Note**: Port 5 is 4-bits wide

7.0 Clocking

7.1 Oscillator Options

7.1.1 Internal Main Oscillator

The internal main oscillator outputs two frequencies, 48 MHz and 24 MHz. In the absence of a high-precision input source from the external oscillator, the accuracy of this circuit is +/- 2.5% (between 0°C and +85°C). No external components are required to achieve this level of accuracy. The Internal Main Oscillator Trim Register (IMO_TR) is used to calibrate this oscillator into specified tolerance. Factory-programmed trim values are available for 5.0V and 3.3V operation. The 5.0V value is loaded in the IMO_TR register upon reset. This register must be adjusted when the operating voltage is outside the range for which factory calibration was set. The factory-programmed trim value is selected using the Table Read Supervisor Call, and is documented in 11.8.

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

Table 35: Internal Main Oscillator Trim Register

Bit #	7	6	5	4	3	2	1	0
POR	FS[1]	FS[1]	FS[1]	FS[1]	FS[1]	FS[1]	FS[1]	FS[1]
Read/Write	W	W	W	W	W	W	W	W
Bit Name	IMO Trim [7]	IMO Trim [6]	IMO Trim [5]	IMO Trim [4]	IMO Trim [3]	IMO Trim [2]	IMO Trim [1]	IMO Trim [0]
Bit [7:0]: **IMO Trim [7:0]** Data value stored will alter the trimmed frequency of the Internal Main Oscillator. A larger value in this register will increase the speed of the Internal Main Oscillator								

1. FS = Factory set trim value

Internal Main Oscillator Trim Register (IMO_TR, Address = Bank 1, E8h)

7.1.2 Internal Low Speed Oscillator

An internal low speed oscillator of nominally 32 kHz is available to generate sleep wake-up interrupts and Watchdog resets if the user does not want to attach a 32.768 kHz watch crystal. This oscillator can also be used as a clocking source for the digital PSoC blocks.

The oscillator operates in two different modes. A trim value is written to the Internal Low Speed Oscillator Trim Register (ILO_TR), shown below, upon reset. See section 13.0 for accuracy information. When the IC is put into sleep mode this oscillator drops into an ultra low current state and the accuracy is reduced.

This register sets the adjustment for the Internal Low Speed Oscillator. The value placed in this register is based on factory testing. It is recommended that the user not alter this value.

Table 40: Oscillator Control 0 Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/ Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	32k Select	PLL Mode	Reserved	Sleep [1]	Sleep [0]	CPU [2]	CPU [1]	CPU [0]

Bit 7: **32k Select**
0 = Internal low precision 32 kHz oscillator
1 = External Crystal Oscillator

Bit 6: **PLL Mode**
0 = Disabled
1 = Enabled, Internal Main Oscillator is locked to External Crystal Oscillator

Bit 5: **Reserved**

Bit [4:3]: **Sleep [1:0]**
0 0 = 512 Hz or 1.95 ms period
0 1 = 64 Hz or 15.6 ms period
1 0 = 8 Hz or 125 ms period
1 1 = 1 Hz or 1 s period

Bit [2:0]: **CPU [2:0]**
0 0 0 = 3 MHz
0 0 1 = 6 MHz
0 1 0 = 12 MHz
0 1 1 = 24 MHz
1 0 0 = 1.5 MHz
1 0 1 = 750 kHz
1 1 0 = 187.5 kHz
1 1 1 = 93.7 kHz

Oscillator Control 0 Register (OSC_CR0, Address = Bank 1, E0h)

Table 41: Oscillator Control 1 Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/ Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	24V1 [3]	24V1 [2]	24V1 [1]	24V1 [0]	24V2 [3]	24V2 [2]	24V2 [1]	24V2 [0]

Bit [7:4]: **24V1 [3:0]** 4-bit data value determines the divider value for the **24V1** system-clocking signal. Note that the 4-bit data value equals n-1, where n is the desired divider value, as illustrated in PSoC MCU Clock Tree of Signals. See Table 42 on page 41.

Bit [3:0]: **24V2 [3:0]** 4-bit data value determines the divider value for the **24V2** system-clocking signal. Note that the 4-bit data value equals n-1, where n is the desired divider value, as illustrated in the PSoC MCU Clock Tree of Signals. See Table 42 on page 41.

Oscillator Control 1 Register (OSC_CR1, Address = Bank 1, E1h)

8.6 GPIO Interrupt

GPIO Interrupts are polarity configurable and pin-wise maskable (within each Port's pin configuration registers). They all share the same interrupt priority and vector.

Any general purpose I/O can be used as an interrupt source. The GPIO bit in the General Interrupt Mask Register (INT_MSK0) must be set to enable pin interrupts, as well as the enable bits for each pin, which are located in the Port x Interrupt Enable Registers (PRTxIE). There are user selectable options to generate an interrupt on 1) any change from the last read state, 2) rising edge, and 3) falling edge.

When Interrupt on Change is selected, the state of the GPIO pin is stored when the port is read. Changes from this state will then assert the interrupt, if enabled.

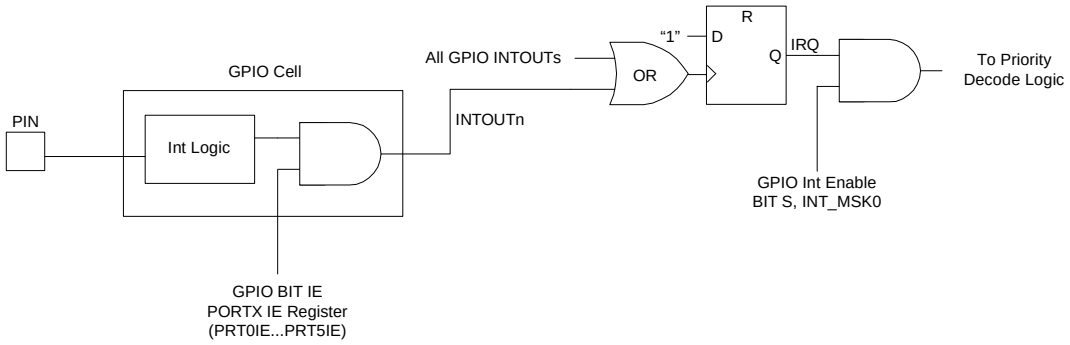

Figure 11: GPIO Interrupt Enable Diagram

For a GPIO interrupt to occur, the following steps must be taken:

1. The pin Drive Mode must be set so the pin can be an input.
2. The pin must be enabled to generate an interrupt by setting the appropriate bit in the Port interrupt Enable Register (PRTxIE).
3. The edge type for the interrupt must be set in the Port Interrupt Control 0 and Control 1 Registers (PRTxIC0 and PRTxIC1). Edge type must be set to a value other than 00.
4. The GPIO bit must be set in the General Interrupt Mask Register (INT_MSK0).
5. The Global Interrupt Enable bit must be set.
6. Because the GPIO interrupts all share the same interrupt vector, the source for the GPIO interrupt must be cleared before any other GPIO interrupt will occur (i.e., the OR gate in Figure 11: "ors" all of the INTOUTn signals together). If any of the INTOUTn signals are high, the flip-flop in Figure 11: will not see a rising edge and no IRQ will occur.

9.0 Digital PSoC Blocks

9.1 Introduction

PSoC blocks are user configurable system resources. On-chip digital PSoC blocks reduce the need for many MCU part types and external peripheral components. Digital PSoC blocks can be configured to provide a wide variety of peripheral functions. PSoC Designer Software Integrated Development Environment provides automated configuration of PSoC blocks by simply selecting the desired functions. PSoC Designer then generates the proper configuration information and can print a device data sheet unique to that configuration.

Digital PSoC blocks provide up to eight, 8-bit multipurpose timers/counters supporting multiple event timers, real-time clocks, Pulse Width Modulators (PWM), and CRCs. In addition to all PSoC block functions, communication PSoC blocks support full-duplex UARTs and SPI master or slave functions.

As shown in Figure 12:, there are a total of eight 8-bit digital PSoC blocks in this device family configured as a linear array. Four of these are the Digital Basic Type A blocks and four are the Digital Communications Type A blocks. Each of these digital PSoC blocks can be configured independently, or used in combination.

Each digital PSoC block has a unique Interrupt Vector and Interrupt Enable bit. Functions can be stopped or started with a user-accessible Enable bit.

The Timer/Counter/CRC/PRS/Deadband functions are available on the Digital Basic Type A blocks and also the Digital Communications Type A blocks. The UART and SPI communications functions are only available on the Digital Communications Type A blocks.

There are three configuration registers: the Function Register (DBA00FN-DCA07FN) to select the block function and mode, the Input Register (DBA00IN-DCA07IN) to select data input and clock selection, and the Output Register (DBA00OU-DCA07OU) to select and enable function outputs.

The three data registers are designated Data 0 (DBA00DR0-DCA07DR0), Data 1 (DBA00DR1-DCA07DR1), and Data 2 (DBA00DR2-DCA07DR2). The function of these registers and their bit mapping is dependent on the overall block function selected by the user.

The one Control Register (DBA00CR0-DCA07CR0) is designated Control 0. The function of this register and its bit mapping is dependent on the overall block function selected by the user.

If the CPU frequency is 24 MHz and a PSoC timer/counter of 24-bits or longer is operating at 48 MHz, a write to the block Control Register to enable it (for example, a call to Timer_1_Start) may not start the block properly. In the failure case, the first count will typically be indeterminate as the upper bytes fail to make the first count correctly. However, on the first terminal count, the correct period will be loaded and counted thereafter.

Table 47: Digital Basic Type A/ Communications Type A Block xx Function Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Reserved	Reserved	End	Mode 1	Mode 0	Function [2]	Function [1]	Function [0]

Bit 7: **Reserved**
Bit 6: **Reserved**

Bit 5: **End**
0 = PSoC block is not the end of a chained function (End should not be set to 0 in block DCA07)
1 = PSoC block is the end of a chained function, or is an unchained PSoC block

Bit 4: **Mode 1** The definition of the Mode [1] bit depends on the block function selected
Timer: The Mode [1] bit signifies the Compare Type
0 = Less Than or Equal
1 = Less Than
Counter: The Mode [1] bit signifies the Compare Type
0 = Less Than or Equal
1 = Less Than
CRC/PRS: The Mode [1] bit is unused in this function
Deadband: The Mode [1] bit is unused in this function
UART: The Mode[1] bit signifies the Interrupt Type (Transmitter only)
0 = Transmit: Interrupt on TX_Reg Empty
1 = Transmit: Interrupt on TX Complete
SPI: The Mode[1] bit signifies the Interrupt Type
0 = Master: Interrupt on TX Reg Empty, Slave: Interrupt on RX Reg Full
1 = Master: Interrupt on SPI Complete, Slave: Interrupt on SPI Complete

Bit 3: **Mode 0** The definition of the Mode [0] bit depends on the block function selected
Timer: The Mode [0] bit signifies Interrupt Type
0 = Terminal Count
1 = Compare True
Counter: The Mode [0] bit signifies Interrupt Type
0 = Terminal Count
1 = Compare True
CRC/PRS: The Mode [0] bit is unused in this function
Deadband: The Mode [0] bit is unused in this function
UART: The Mode [0] bit signifies the Direction
0 = Receive
1 = Transmit
SPI: The Mode [0] bit signifies the Type
0 = Master
1 = Slave

Bit [2:0]: **Function [2:0]** The Function [2:0] bits select the block function which determines the basic hardware configuration
0 0 0 = Timer (chainable)
0 0 1 = Counter (chainable)
0 1 0 = CRC/PRS (Cyclical Redundancy Checker or Pseudo Random Sequencer) (chainable)
0 1 1 = Reserved
1 0 0 = Deadband for Pulse Width Modulator
1 0 1 = UART (function only available on DCA type blocks)
1 1 0 = SPI (function only available on DCA type blocks)
1 1 1 = Reserved

Digital Basic Type A Block 00 Function Register (DBA00FN, Address = Bank 1, 20h)
Digital Basic Type A Block 01 Function Register (DBA01FN, Address = Bank 1, 24h)
Digital Basic Type A Block 02 Function Register (DBA02FN, Address = Bank 1, 28h)
Digital Basic Type A Block 03 Function Register (DBA03FN, Address = Bank 1, 2Ch)
Digital Communications Type A Block 04 Function Register (DCA04FN, Address = Bank 1, 30h)

Table 50: Digital Basic Type A / Communications Type A Block xx Output Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/ Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Reserved	Reserved	AUX Out Enable	AUX IO Sel [1]	AUX IO Sel [0]	Out Enable	Out Sel [1]	Out Sel [0]

Bit 7: **Reserved**
Bit 6: **Reserved**

Bit 5: **AUX Out Enable**
0 = Disable Auxiliary Output
1 = Enable Auxiliary Output (function dependent)

Bit [4:3]: **AUX IO Sel [1:0] Function-dependent selection of auxiliary input or output**
0 0 = Drive Global Output[0] (for Digital Blocks 00 to 03) **or**
Input from Global Input[4] or Drive Global Output [4] (for Digital Blocks 04 to 07)
0 1 = Drive Global Output[1] (for Digital Blocks 00 to 03) **or**
Input from Global Input[5] or Drive Global Output[5] (for Digital Blocks 04 to 07)
1 0 = Drive Global Output[2] (for Digital Blocks 00 to 03) **or**
Input from Global Input[6] or Drive Global Output[6] (for Digital Blocks 04 to 07)
1 1 = Drive Global Output[3] (for Digital Blocks 00 to 03) **or**
Input from Global Input[7] or Drive Global Output[7] (for Digital Blocks 04 to 07)

Bit 2: **Out Enable**
0 = Disable Primary Output
1 = Enable Primary Output (function dependant)

Bit [1:0]: **Out Sel [1:0]** Primary Output
0 0 = Drive Global Output[0] (for Digital Blocks 00 to 03) **or** Drive Global Output[4] (for Digital Blocks 04 to 07)
0 1 = Drive Global Output[1] (for Digital Blocks 00 to 03) **or** Drive Global Output[5] (for Digital Blocks 04 to 07)
1 0 = Drive Global Output[2] (for Digital Blocks 00 to 03) **or** Drive Global Output[6] (for Digital Blocks 04 to 07)
1 1 = Drive Global Output[3] (for Digital Blocks 00 to 03) **or** Drive Global Output[7] (for Digital Blocks 04 to 07)

Digital Basic Type A Block 00 Output Register (DBA00OU, Address = Bank 1, 22h)
Digital Basic Type A Block 01 Output Register (DBA01OU, Address = Bank 1, 26h)
Digital Basic Type A Block 02 Output Register (DBA02OU, Address = Bank 1, 2Ah)
Digital Basic Type A Block 03 Output Register (DBA03OU, Address = Bank 1, 2Eh)
Digital Communications Type A Block 04 Output Register (DCA04OU, Address = Bank 1, 32h)
Digital Communications Type A Block 05 Output Register (DCA05OU, Address = Bank 1, 36h)
Digital Communications Type A Block 06 Output Register (DCA06OU, Address = Bank 1, 3Ah)
Digital Communications Type A Block 07 Output Register (DCA07OU, Address = Bank 1, 3Eh)

The Primary Output is the source for “Previous Digital PSoC Block” or “Digital Block 03,” selections for the “Clock Source Select” in the Digital Basic Type A/Communications Type A Block xx Input Register (Table 48 on page 51).

A digital PSoC block may have 0, 1, or 2 outputs depending on its function, as shown in the following table:

9.5.6.3 Inputs

A baud-rate clock running at 8 times the desired input bit rate is selected by the clock-input multiplexer The serial data input and clock input are controlled by the Input Register (DCA04IN-DCA07IN).

9.5.6.4 Outputs

None.

9.5.6.5 Interrupts

The function can be configured to generate an interrupt on RXREGFULL (Receive Register Full) status (Data Register 2 is full)

9.5.6.6 Usage Notes

1. Reading the Status

   Reading Control Register 0, which contains the status bits, automatically resets all status bits to 0 with the exception of RX Reg Full. Reading Data Register 2 (Receive Data Register) clears the RX Reg Full status.

2. Using Interrupts

   RX Reg Full status generates an interrupt but the Receive Data Register (Data Register 2) must be read to clear the RX Reg Full status. If this registers is not read in the interrupt routine, the status will not be cleared and further interrupts will be suppressed.

   If the stop bit in a transmitted byte is missing, the receiver will declare a framing error. Once this occurs, this missing stop bit can be interpreted as the start bit of the next byte, which will produce another framing error.

9.5.7 Universal Asynchronous Transmitter

9.5.7.1 Summary

The Universal Asynchronous Transmitter implements the output half of a basic 8-bit UART. Start and Stop bits are generated. Parity bit generation and type are configurable features. This function requires a Digital Communications Type PSoC block. It cannot be chained for longer data words.

9.5.7.2 Registers

When Data Register 0 is empty and a new byte has been written to Data Register 1, the function transfers the byte to Data Register 0 and shifts it out along with a start bit, optionally a parity bit and a stop bit. Once Data Register 0 is loaded with the byte to shift out, Data Register 0 can be immediately loaded with the next byte to transmit, acting as a 1 byte transmit buffer. Data Register 2 is not used by this function. The PSoC block's Control Register 0 (DCA04CR0-DCA07CR0) configures the parity type and enable. It also provides status information to enable detection of transmission complete.

9.5.7.3 Inputs

A baud-rate clock running at 8 times the desired output bit rate is selected by the clock-input multiplexer controlled by the PSoC block Input Register (DCA04IN-DCA07IN). The Data Input multiplexer is ignored by this function.

9.5.7.4 Outputs

The transmitter's serial data output appears at the PSoC block output and may be driven onto one of the Global Output bus lines. The PSoC block Output Register (DCA04OU-DCA07OU) controls output options.

9.5.7.5 Interrupts

If enabled, the function will generate an interrupt when the TX Reg Empty status is set (Data Register 1 is empty). Optionally, the interrupt can be set to TX Complete status, which indicates all bits of a given byte have been sent, including framing bits. This option is selected based on the Mode[1] bit in the Function Register.

9.5.7.6 Usage Notes

1. TX Reg Empty Interrupt

   An initial byte must be written to the TX Data Register (Data Register 1) to enable subsequent TX Reg Empty status interrupts. This does not apply if the TX Complete interrupt source is selected.

2. Reading the Status

   Reading Control Register 0, which contains the status bits, automatically resets the status bits to 0,

except for TX Reg Empty. TX Reg Empty is automatically cleared when a byte is written to the TX Data Register (Data Register 1).

3. Using CPU Interrupts

   TX Reg Empty status or optionally TX Complete status generates the block interrupt. Executing the interrupt routine does not automatically clear status. If TX Complete is selected as the interrupt source, Control Register 0 (status) must be read in the interrupt routine to clear the status. If TX Reg Empty is selected, a byte must be written to the TX Data Register (Data Register 1) to clear the status. If the status is not cleared, further interrupts will be suppressed.

9.5.8 SPI Master - Serial Peripheral Interface (SPIM)

9.5.8.1 Summary

The SPI Master function provides a full-duplex synchronous data transceiver that also generates a bit clock for the data. This function requires a Digital Communications Type PSoC block. It cannot be chained for longer data words. This Digital Communications Type PSoC block supports SPI modes for 0, 1, 2, and 3. See Figure 15: for waveforms of the Clock Phase modes.

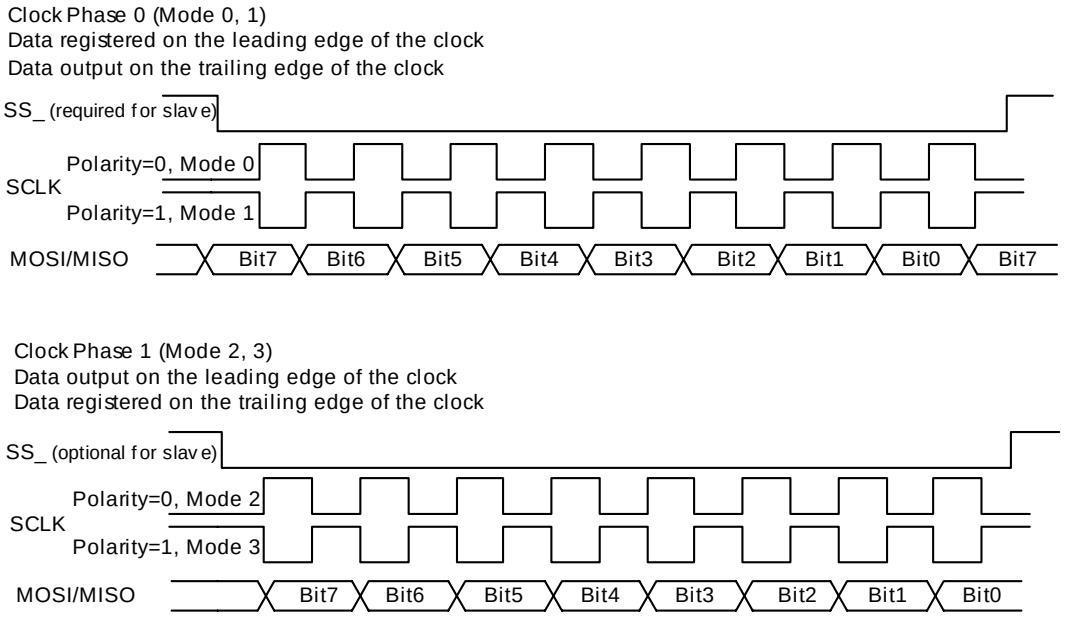

Figure 15: SPI Waveforms

9.5.8.2 Registers

Data Register 0 provides a shift register for both incoming and outgoing data. Output data is written to Data Register 1 (TX Data Register). When this block is idle, a write to the TX Data Register will initiate a transmission. Input data is read from Data Register 2 (RX Data Register). When Data Register 0 is empty, its value is updated from Data Register 1, if new data is available. As data bits are shifted in, the transmit bits are shifted out. After the 8 bits are transmitted and received by Data Register 0, the received byte is transferred into Data Register 2 from where it can be read. Simultaneously, the next byte to transmit, if available, is transferred from Data Register 1 into Data Register 0. Control Register 0 (DCA04CR0-DCA07CR0) provides status information and configures the function for one of the four standard modes, which configure the interface based on clock polarity and phase with respect to data.

10.7.2.2 Analog Continuous Time Block xx Control 1 Register

The PMux bits control the multiplexing of inputs to the non-inverting input of the op-amp. There are physically only 7 inputs.

The 8th code (111) will leave the input floating. This is not desirable, and should be avoided.

The NMux bits control the multiplexing of inputs to the inverting input of the op-amp. There are physically only 7 inputs.

CompBus controls a tri-state buffer that drives the comparator logic. If no PSoC block in the analog column is driving the comparator bus, it will be driven low externally to the blocks.

AnalogBus controls the analog output bus. A CMOS switch connects the op-amp output to the analog bus.

Table 67: Analog Continuous Time Block xx Control 1 Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/ Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	AnalogBus	CompBus	NMux2	NMux1	NMux0	PMux2	PMux1	PMux0

Bit 7: **AnalogBus** Enable output to the analog bus
0 = Disable analog bus driven by this block
1 = Enable analog bus driven by this block
Bit 6: **CompBus** Enable output to the comparator bus
0 = Disable comparator bus driven by this block
1 = Enable comparator bus driven by this block

Bit [5:3]: **NMux [2:0]** Encoding for negative input select

	ACA00	ACA01	ACA02	ACA03
0 0 0 =	ACA01	ACA00	ACA03	ACA02
0 0 1 =	AGND	AGND	AGND	AGND
0 1 0 =	REFLO	REFLO	REFLO	REFLO
0 1 1 =	REFHI	REFHI	REFHI	REFHI
1 0 0[1] =	ACA00	ACA01	ACA02	ACA03
1 0 1 =	ASA10	ASB11	ASA12	ASB13
1 1 0 =	ASB11	ASA10	ASB13	ASA12
1 1 1 =	Reserved	Reserved	Reserved	Reserved

Bit [2:0]: **PMux [2:0]** Encoding for positive input select

	ACA00	ACA01	*ACA02*	ACA03
0 0 0 =	REFLO	ACA02	ACA01	REFLO
0 0 1 =	Port Inputs	Port Inputs	Port Inputs	Port Inputs
0 1 0 =	ACA01	ACA00	ACA03	ACA02
0 1 1 =	AGND	AGND	AGND	AGND
1 0 0 =	ASA10	ASB11	ASA12	ASB13
1 0 1 =	ASB11	ASA10	ASB13	ASA12
1 1 0 =	ABUS0	ABUS1	ABUS2	ABUS3
1 1 1 =	Reserved	Reserved	Reserved	Reserved

1. This in fact is the feedback input of the MUX.

Analog Continuous Time Block 00 Control 1 Register (ACA00CR1, Address = Bank 0/1, 72h)
Analog Continuous Time Block 01 Control 1 Register (ACA01CR1, Address = Bank 0/1, 76h)
Analog Continuous Time Block 02 Control 1 Register (ACA02CR1, Address = Bank 0/1, 7Ah)
Analog Continuous Time Block 03 Control 1 Register (ACA03CR1, Address = Bank 0/1, 7Eh)

Table 69: Analog Switch Cap Type A Block xx Control 0 Register, continued

Bit 7: **FCap** F Capacitor value selection bit
0 = 16 capacitor units
1 = 32 capacitor units

Bit 6: **ClockPhase** Clock phase select, will invert clocks internal to the blocks. During normal operation of an SC block for the amplifier of a column enabled to drive the output bus, the connection is only made for the last half of PHI2 (during PHI1 and for the first half of PHI2, the output bus floats at the last voltage to which it was driven). This forms a sample and hold operation using the output bus and its associated capacitance. This design prevents the output bus from being perturbed by the intermediate states of the SC operation (often a reset state for PHI1 and settling to the valid state during PHI2)

Following are the exceptions: 1) If the ClockPhase bit in CR0 (for the SC block in question) is set to 1, then the output is enabled for the whole of PHI2. 2) If the SHDIS signal is set in bit 6 of the Analog Clock Select Register, then sample and hold operation is disabled for all columns and all enabled outputs of SC blocks are connected to their respective output busses for the entire period of their respective PHI2s

0 = Internal PHI1 = External PHI1
1 = Internal PHI1 = External PHI2

This bit also affects the latching of the comparator output (CBUS). Both clock phases, PHI1 and PHI2, are involved in the output latching mechanism. The capture of the next value to be output from the latch (capture point event) happens during the falling edge of one clock phase, and the rising edge of the other clock phase will cause the value to come out (output point event). This bit determines which clock phase triggers the capture point event, and the other clock will trigger the output point event. The value output to the comparator bus will remain stable between output point events.

0 = Capture Point Event triggered by Falling PHI2, Output Point Event triggered by Rising PHI1
1 = Capture Point Event triggered by Falling PHI1, Output Point Event triggered by Rising PHI2

Bit 5: **ASign**
0 = Input sampled on Internal PHI1, Reference Input sampled on internal PHI2
1 = Input sampled on Internal PHI2, Reference Input sampled on internal PHI1

Bit [4:0]: **ACap [4:0]** Binary encoding for 32 possible capacitor sizes for A Capacitor:

0 0 0 0 0 = 0 Capacitor units in array	1 0 0 0 0 = 16 Capacitor units in array
0 0 0 0 1 = 1 Capacitor units in array	1 0 0 0 1 = 17 Capacitor units in array
0 0 0 1 0 = 2 Capacitor units in array	1 0 0 1 0 = 18 Capacitor units in array
0 0 0 1 1 = 3 Capacitor units in array	1 0 0 1 1 = 19 Capacitor units in array
0 0 1 0 0 = 4 Capacitor units in array	1 0 1 0 0 = 20 Capacitor units in array
0 0 1 0 1 = 5 Capacitor units in array	1 0 1 0 1 = 21 Capacitor units in array
0 0 1 1 0 = 6 Capacitor units in array	1 0 1 1 0 = 22 Capacitor units in array
0 0 1 1 1 = 7 Capacitor units in array	1 0 1 1 1 = 23 Capacitor units in array
0 1 0 0 0 = 8 Capacitor units in array	1 1 0 0 0 = 24 Capacitor units in array
0 1 0 0 1 = 9 Capacitor units in array	1 1 0 0 1 = 25 Capacitor units in array
0 1 0 1 0 = 10 Capacitor units in array	1 1 0 1 0 = 26 Capacitor units in array
0 1 0 1 1 = 11 Capacitor units in array	1 1 0 1 1 = 27 Capacitor units in array
0 1 1 0 0 = 12 Capacitor units in array	1 1 1 0 0 = 28 Capacitor units in array
0 1 1 0 1 = 13 Capacitor units in array	1 1 1 0 1 = 29 Capacitor units in array
0 1 1 1 0 = 14 Capacitor units in array	1 1 1 1 0 = 30 Capacitor units in array
0 1 1 1 1 = 15 Capacitor units in array	1 1 1 1 1 = 31 Capacitor units in array

Analog Switch Cap Type A Block 10 Control 0 Register (ASA10CR0, Address = Bank 0/1, 80h)
Analog Switch Cap Type A Block 12 Control 0 Register (ASA12CR0, Address = Bank 0/1, 88h)
Analog Switch Cap Type A Block 21 Control 0 Register (ASA21CR0, Address = Bank 0/1, 94h)
Analog Switch Cap Type A Block 23 Control 0 Register (ASA23CR0, Address = Bank 0/1, 9Ch)

10.8.3.4 Analog Switch Cap Type A Block xx Control 3 Register

ARefMux selects the reference input of the A capacitor branch.

FSW1 is used to control a switch in the integrator capacitor path. It connects the output of the op-amp to the integrating cap. The state of the switch is affected by the state of the AutoZero bit in Control 2 Register (ASA10CR2, ASA12CR2, ASA21CR2, ASA23CR2). If the FSW1 bit is set to 0, the switch is always disabled. If the FSW1 bit is set to 1, the AutoZero bit determines the state of the switch. If the AutoZero bit is 0, the switch is enabled at all times. If the AutoZero bit is 1, the switch is enabled only when the internal PHI2 is high.

FSW0 is used to control a switch in the integrator capacitor path. It connects the output of the op-amp to analog ground.

BMuxSCA controls the muxing to the input of the B capacitor branch.

Power – encoding for selecting 1 of 4 power levels. The block always powers up in the off state.

Table 72: Analog Switch Cap Type A Block xx Control 3 Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/ Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	ARefMux[1]	ARefMux[0]	FSW[1]	FSW[0]	BMuxSCA[1]	BMuxSCA[0]	Power[1]	Power[0]

Bit [7:6]: **ARefMux [1:0]** Encoding for selecting reference input
0 0 = Analog ground is selected
0 1 = REFHI input selected (This is usually the high reference)
1 0 = REFLO input selected (This is usually the low reference)
1 1 = Reference selection is driven by the comparator (When output comparator node is set high, the input is set to REFHI. When set low, the input is set to REFLO)

Bit 5: **FSW1** Bit for controlling gated switches
0 = Switch is disabled
1 = If the FSW1 bit is set to 1, the state of the switch is determined by the AutoZero bit. If the AutoZero bit is 0, the switch is enabled at all times. If the AutoZero bit is 1, the switch is enabled only when the internal PHI2 is high

Bit 4: **FSW0** Bits for controlling gated switches
0 = Switch is disabled
1 = Switch is enabled when PHI1 is high

Bit [3:2] BMuxSCA [1:0] Encoding for selecting B inputs. (Note that the available mux inputs vary by individual PSoC block.)

ASA10	ASA21	ASA12	ASA23
0 0 = ACA00	ASB11	ACA02	ASB13
0 1 = ASB11	ASB20	ASB13	ASB22
1 0 = P2.3	ASB22	ASB11	P2.0
1 1 = ASB20	T_{ref}GND	ASB22	ABUS3

Bit [1:0]: **Power [1:0]** Encoding for selecting 1 of 4 power levels
0 0 = Off
0 1 = 10 µA, typical
1 0 = 50 µA, typical
1 1 = 200 µA, typical

Analog Switch Cap Type A Block 10 Control 3 Register (ASA10CR3, Address = Bank 0/1, 83h)
Analog Switch Cap Type A Block 12 Control 3 Register (ASA12CR3, Address = Bank 0/1, 8Bh)
Analog Switch Cap Type A Block 21 Control 3 Register (ASA21CR3, Address = Bank 0/1, 97h)
Analog Switch Cap Type A Block 23 Control 3 Register (ASA23CR3, Address = Bank 0/1, 9Fh)

Table 75: Analog Switch Cap Type B Block xx Control 2 Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/ Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	AnalogBus	CompBus	AutoZero	CCap[4]	CCap[3]	CCap[2]	CCap[1]	CCap[0]

Bit 7: **AnalogBus** Enable output to the analog bus
0 = Disable output to analog column bus
1 = Enable output to analog column bus
(The output on the analog column bus is affected by the state of the ClockPhase bit in Control 0 Register (ASB11CR0, ASB13CR0, ASB20CR0, ASB22CR0). If AnalogBus is set to 0, the output to the analog column bus is tri-stated. If AnalogBus is set to 1, the ClockPhase bit selects the signal that is output to the analog column bus. If the ClockPhase bit is 0, the block output is gated by sampling clock on last part of PHI2. If the ClockPhase bit is 1, the block output continuously drives the analog column bus)

Bit 6: **CompBus** Enable output to the comparator bus
0 = Disable output to comparator bus
1 = Enable output to comparator bus

Bit 5: **AutoZero** Bit for controlling gated switches
0 = Shorting switch is not active. Input cap branches shorted to op-amp input
1 = Shorting switch is enabled during internal PHI1. Input cap branches shorted to analog ground during internal PHI1 and to op-amp input during internal PHI2.

Bit [4:0]: **CCap [4:0]** Binary encoding for 32 possible capacitor sizes for C Capacitor:

0 0 0 0 0 = 0 Capacitor units in array	1 0 0 0 0 = 16 Capacitor units in array
0 0 0 0 1 = 1 Capacitor units in array	1 0 0 0 1 = 17 Capacitor units in array
0 0 0 1 0 = 2 Capacitor units in array	1 0 0 1 0 = 18 Capacitor units in array
0 0 0 1 1 = 3 Capacitor units in array	1 0 0 1 1 = 19 Capacitor units in array
0 0 1 0 0 = 4 Capacitor units in array	1 0 1 0 0 = 20 Capacitor units in array
0 0 1 0 1 = 5 Capacitor units in array	1 0 1 0 1 = 21 Capacitor units in array
0 0 1 1 0 = 6 Capacitor units in array	1 0 1 1 0 = 22 Capacitor units in array
0 0 1 1 1 = 7 Capacitor units in array	1 0 1 1 1 = 23 Capacitor units in array
0 1 0 0 0 = 8 Capacitor units in array	1 1 0 0 0 = 24 Capacitor units in array
0 1 0 0 1 = 9 Capacitor units in array	1 1 0 0 1 = 25 Capacitor units in array
0 1 0 1 0 = 10 Capacitor units in array	1 1 0 1 0 = 26 Capacitor units in array
0 1 0 1 1 = 11 Capacitor units in array	1 1 0 1 1 = 27 Capacitor units in array
0 1 1 0 0 = 12 Capacitor units in array	1 1 1 0 0 = 28 Capacitor units in array
0 1 1 0 1 = 13 Capacitor units in array	1 1 1 0 1 = 29 Capacitor units in array
0 1 1 1 0 = 14 Capacitor units in array	1 1 1 1 0 = 30 Capacitor units in array
0 1 1 1 1 = 15 Capacitor units in array	1 1 1 1 1 = 31 Capacitor units in array

Analog Switch Cap Type B Block 11 Control 2 Register (ASB11CR2, Address = Bank 0/1, 86h)
Analog Switch Cap Type B Block 13 Control 2 Register (ASB13CR2, Address = Bank 0/1, 8Eh)
Analog Switch Cap Type B Block 20 Control 2 Register (ASB20CR2, Address = Bank 0/1, 92h)
Analog Switch Cap Type B Block 22 Control 2 Register (ASB22CR2, Address = Bank 0/1, 9Ah)

11.0 Special Features of the CPU

11.1 Multiplier/Accumulator

A fast, on-chip signed 2's complement MAC (Multiply/Accumulate) function is provided to assist the main CPU with digital signal processing applications. Multiply results, as well as the lower 2 bytes of the Accumulator, are available immediately after the input registers are written. The upper 2 bytes require a single instruction delay before reading. The MAC function is tied directly on the internal data bus, and is mapped into the register space. The following MAC block diagram provides data flow information. The user has the choice to either cause a multiply/accumulate function to take place, or a multiply only function. The user selects which operation is performed by the choice of input register. The multiply function occurs immediately whenever the MUL_X or the MUL_Y multiplier input registers are written, and the result is available in the MUL_DH and MUL_DL multiplier result registers. The Multiply/Accumulate function is executed whenever there is a write to the MAC_X or the MAC_Y Multiply/Accumulate input registers, and the result is available in the ACC_DR3, ACC_DR2, ACC_DR1, and ACC_DR0 accumulator result registers. A write to MUL_X or MAC_X is input as the X value to both the multiply and Multiply/Accumulate functions. A write to MUL_Y or MAC_Y is input as the Y value to both the multiply and Multiply/Accumulate functions. A write to the MAC_CL0 or MAC_CL1 registers will clear the value in the four accumulate registers.

Operation of the Multiply/Accumulate function relies on proper multiplicand input. The first value of each multiplicand must be placed into MUL_X (or MUL_Y) register to avoid causing a Multiply/Accumulate to occur. The second multiplicand must be placed into MAC_Y (or MAC_X) thereby triggering the Multiply/Accumulate function.

MUL_X, MUL_Y, MAC_X, and MAC_Y are 8-bit signed input registers. MUL_DL and MUL_DH form a 16-bit signed output. ACC_DR0, ACC_DR1, ACC_DR2 and ACC_DR3 form a 32-bit signed output.

An extra instruction must be inserted between the following sequences of MAC operations to provide extra delay. If this is not done, the Accumulator results will be inaccurate.

a. Two MAC instructions in succession:

```
mov reg[MAC_X],a
nop //add nop or any other instruction
mov reg[MAC_X],a
```

For sequence a., there is no workaround, the nop or other instruction must be inserted.

b. A MAC instruction followed by a read of the most significant Accumulator bytes:

```
mov reg[MAC_X],a
nop //add nop or any other instruction
mov a,[ACC_DR2] // or ACC_DR3
```

For sequence b., the least significant Accumulator bytes (ACC_DR0, ACC_DR1) may be reliably read directly after the MAC instruction.

Writing to the multiplier registers (MUL_X, MUL_Y), and reading the result back from the multiplier product registers (MUL_DH, MUL_DL), is not affected by this problem and does not have any restrictions.

Table 85: Multiply Result High Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]** 8-bit data value is the high order result of the multiply function

Multiply Result High Register (MUL_DH, Address = Bank 0, EAh)

Table 86: Multiply Result Low Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]** 8-bit data value is the low order result of the multiply function

Multiply Result Low Register (MUL_DL, Address = Bank 0, EBh)

Table 87: Accumulator Result 1 / Multiply/Accumulator Input X Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]**
8-bit data value when read is the next to lowest order result of the multiply/accumulate function
8-bit data value when written is the X multiplier input to the multiply/accumulate function

Accumulator Result 1 / Multiply/Accumulator Input X Register (ACC_DR1 / MAC_X, Address = Bank 0, ECh)

Table 88: Accumulator Result 0 / Multiply/Accumulator Input Y Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]**
8-bit data value when read is the lowest order result of the multiply/accumulate function
8-bit data value when written is the Y multiplier input to the multiply/accumulate function

Accumulator Result 0 / Multiply/Accumulator Input Y Register (ACC_DR0 / MAC_Y, Address = Bank 0, EDh)

13.3 AC Characteristics

Table 116: AC Operating Specifications

Symbol	AC Operating Specifications	Minimum	Typical	Maximum	Unit
F_{CPU1}	CPU Frequency (5 V Nominal)[1,2,3]	91.35	2,400	2,460	kHz
F_{CPU2}	CPU Frequency (3.3V Nominal)[4,3]	91.35	1,200	1,230	kHz
F_{48M}	Digital PSoC Block Frequency		48	49.2[1,5]	MHz
F_{24M}	Digital PSoC Block Frequency		24	24.6[2,4]	MHz
F_{GPIO}	GPIO Operating Frequency		12		MHz
F_{IMO}	Internal Main Oscillator Frequency (0^oC to $+85^oC$)	23.4	24	24.6	MHz
F_{IMOC}	Internal Main Oscillator Frequency Cold (-40^oC to 0^oC)	22.44	24	24.6	MHz
F_{32K1}	Internal Low Speed Oscillator Frequency (Non Sleep)	15[6]	32	50	kHz
F_{32K2}	Internal Low Speed Oscillator Frequency (Sleep or Halt)	15[7]	32	64	kHz
F_{32K3}	External Crystal Oscillator	-	32.768[8]	-	kHz
F_{pll}	PLL Frequency	-	23.986[9]	-	MHz
T_f	Output Fall Time	2[10]	-	12	ns
T_r	Output Rise Time	3[10]	-	18	ns
$T_{pllslew}$	PLL Lock Time	0.5	-	10	ms
SV_{dd}	V_{dd} Rise Rate at Power Up	80[11]	-	-	mV/ms
T_{os}	External Crystal Oscillator Startup to 1%	-	100	500[12]	ms
T_{osacc}	External Crystal Oscillator Startup to 100 ppm	-	150	600[13]	ms
T_{xrst}	External Reset Pulse Width	10	-	-	µs

1. $4.75V < V_{cc} < 5.25V$.
2. Accuracy derived from Internal Main Oscillator with appropriate trim for V_{cc} range.
3. 0^oC to $+85^oC$.
4. $3.0V < V_{cc} < 3.6V$.
5. See Application Note AN2012 ”Adjusting PSoC Microcontroller Trims for Dual Voltage-Range Operation” for information on maximum frequency for User Modules.
6. Limits are valid only when *not* in sleep mode.
7. Limits are valid only when in sleep mode.
8. Accuracy is capacitor and crystal dependent.
9. Is a multiple (x732) of crystal frequency.
10. Load capacitance = 50 pF.
11. To minimum allowable voltage for desired frequency.
12. The crystal oscillator frequency is guaranteed to be within 1% of its final value by the end of the 1s startup timer period. Timer period may be as short as 640 ms for the case where F_{32K1} is 50 kHz. Correct operation assumes a properly loaded 1uW maximum drive level 32.768 kHz crystal.
13. The crystal oscillator frequency is within 100 ppm of its final value by the end of the T_{osacc} period. Correct operation assumes a properly loaded 1 uW maximum drive level 32.768 kHz crystal. $3.0V <= V_{cc} <= 5.5V$, $-40\ ^oC <= T_A <= 85\ ^oC$.

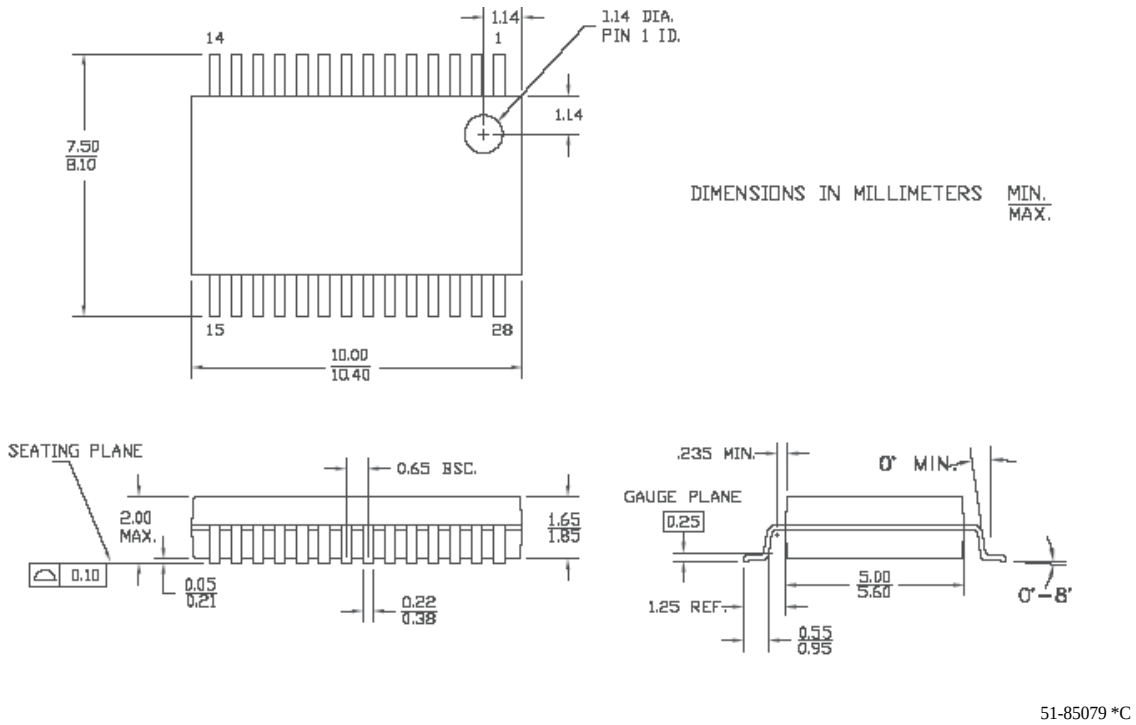

Figure 39: 28-Lead (210-Mil) Shrunk Small Outline Package O28

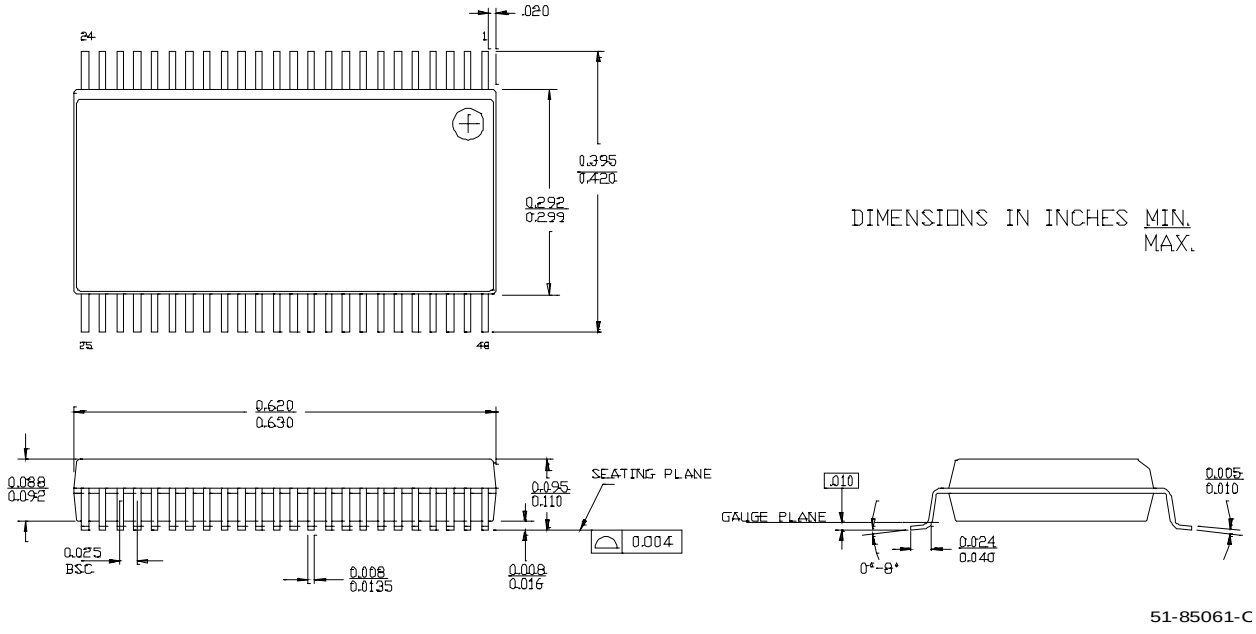

Figure 40: 48-Lead Shrunk Small Outline Package O48

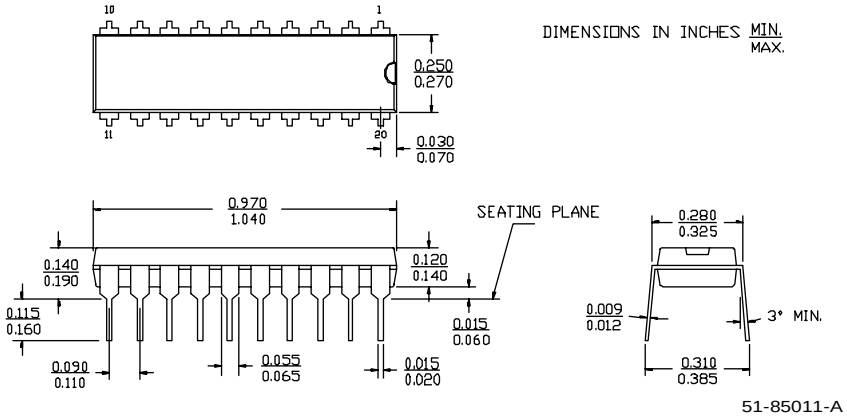

Figure 41: 20-Lead (300-Mil) Molded DIP P5

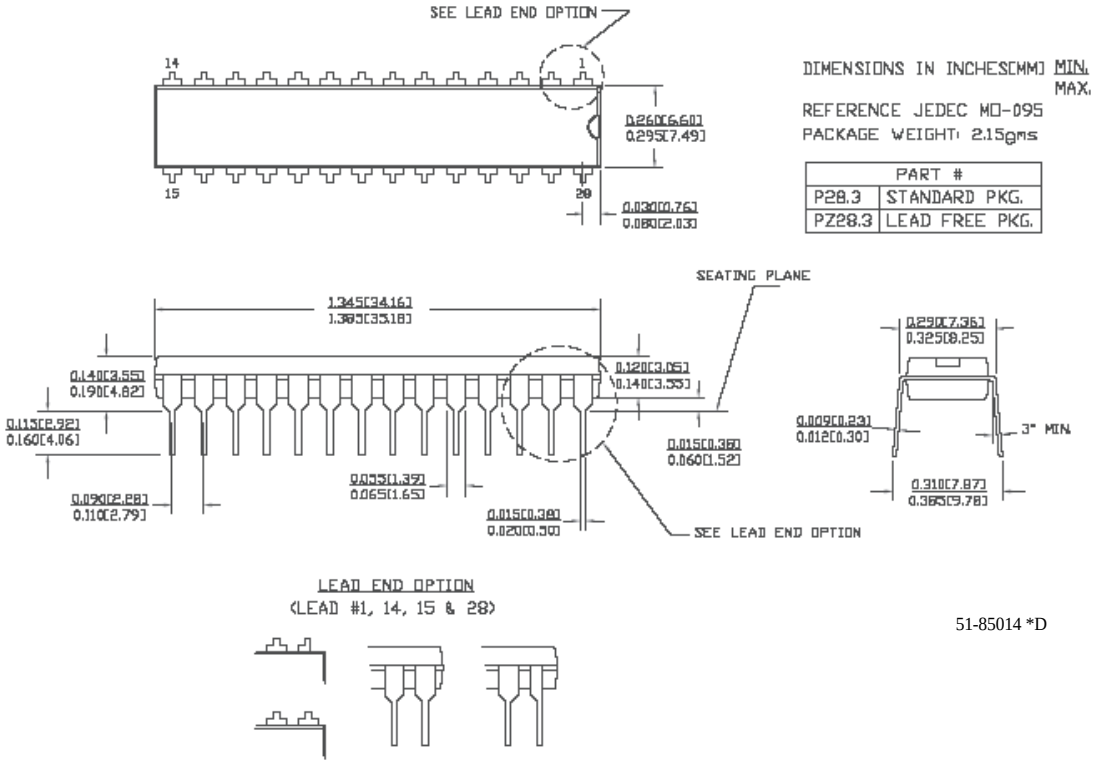

Figure 42: 28-Lead (300-Mil) Molded DIP P21