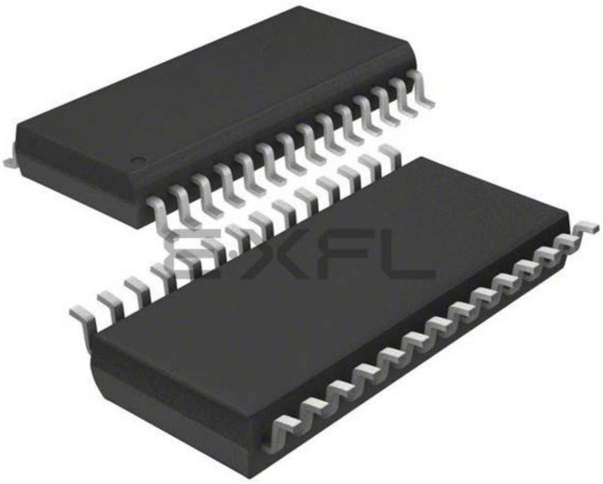

Welcome to E-XFL.COM

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	24
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	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy8c26443-24pvxi

1.0 Functional Overview

The CPU heart of this next generation family of micro-controllers is a high performance, 8-bit, M8C Harvard architecture microprocessor. Separate program and memory busses allow for faster overall throughput. Processor clock speeds to 24 MHz are available. The processor may also be run at lower clock speeds for power-sensitive applications. A rich instruction set allows for efficient low-level language support.

All devices in this family include both analog and digital configurable peripherals (PSoC blocks). These blocks enable the user to define unique functions during configuration of the device. Included are twelve analog PSoC blocks and eight digital PSoC blocks. Potential applications for the digital PSoC blocks are timers, counters, UARTs, CRC generators, PWMs, and other functions. The analog PSoC blocks can be used for SAR ADCs, Multi-slope ADCs, programmable gain amplifiers, programmable filters, DACs, and other functions. Higher order User Modules such as modems, complex motor controllers, and complete sensor signal chains can be created from these building blocks. This allows for an unprecedented level of flexibility and integration in micro-controller-based systems.

A Multiplier/Accumulator (MAC) is available on all devices in this family. The MAC is implemented on this device as a peripheral that is mapped into the register space. When an instruction writes to the MAC input registers, the result of an 8x8 multiply and a 32-bit accumulate are available to be read from the output registers on the next instruction cycle.

The number of general purpose I/Os available in this family of parts range from 6 to 44. Each of these I/O pins has a variety of programmable options. In the output mode, the user can select the drive strength desired. Any pin can serve as an interrupt source, and can be selected to trigger on positive edges, negative edges, or any change. Digital signal sources can be routed directly from a pin to the digital PSoC blocks. Some pins have additional capability to route analog signals to the analog PSoC blocks.

Multiple oscillator options are available for use in clocking the CPU, analog PSoC blocks and digital PSoC blocks. These options include an internal main oscillator running at 48/24 MHz, an external crystal oscillator for use with a 32.768 kHz watch crystal, and an internal low-speed oscillator for use in clocking the PSoC blocks and the Watchdog/Sleep timer. User selectable clock divisors allow for optimizing code execution speed and power trade-offs.

The different device types in this family provide various amounts of code and data memory. The code space ranges in size from 4K to 16K bytes of user programmable Flash memory. This memory can be programmed serially in either a programming Pod or on the user board. The endurance on the Flash memory is 50,000 erase/write cycles. The data space is 256 bytes of user SRAM.

A powerful and flexible protection model secures the user's sensitive information. This model allows the user to selectively lock blocks of memory for read and write protection. This allows partial code updates without exposing proprietary information.

Devices in this family range from 8 pins through 48 pins in PDIP, SOIC and SSOP packages.

1.1 Key Features

Table 1: Device Family Key Features

	CY8C25122	CY8C26233	CY8C26443	CY8C26643
Operating Frequency	93.7kHz - 24MHz	93.7kHz - 24MHz	93.7kHz - 24MHz	93.7kHz - 24MHz
Operating Voltage	3.0 - 5.25V	3.0 - 5.25V	3.0 - 5.25V	3.0 - 5.25V
Program Memory (KBytes)	4	8	16	16
Data Memory (Bytes)	256	256	256	256
Digital PSoC Blocks	8	8	8	8
Analog PSoC Blocks	12	12	12	12
I/O Pins	6	16	24	40/44
External Switch Mode Pump	No	Yes	Yes	Yes
Available Packages	8 PDIP	20 PDIP	28 PDIP	48 PDIP
		20 SOIC	28 SOIC	48 SSOP
		20 SSOP	28 SSOP	44 TQFP

Examples:

```
                        ;In this case, the
                        ;value in the memory
                        ;location at address
                        ;X+7 is added with
ADD   [X+7],    5       ;the immediate value
                        ;of 5, and the result
                        ;is placed in the
                        ;memory location at
                        ;address X+7.
                        ;In this case, the
                        ;immediate value of 6
MOV   REG[X+8], 6       ;is moved into the
                        ;location in the
                        ;register space at
                        ;address X+8.
```

2.3.8 Destination Direct Direct

The result of an instruction using this addressing mode is placed within the RAM memory. Operand 1 is the address of the result. Operand 2 is an address that points to a location in the RAM memory that is the source for the instruction. This addressing mode is only valid on the MOV instruction. The instruction using this addressing mode is three bytes in length.

Table 20: Destination Direct Direct

Opcode	Operand 1	Operand 2
Instruction	Destination Address	Source Address

Example:

```
                 ;In this case, the value
                 ;in the memory location at
MOV   [7], [8]   ;address 8 is moved to the
                 ;memory location at
                 ;address 7.
```

2.3.9 Source Indirect Post Increment

The result of an instruction using this addressing mode is placed in the Accumulator. Operand 1 is an address pointing to a location within the memory space, which contains an address (the indirect address) for the source of the instruction. The indirect address is incremented as part of the instruction execution. This addressing mode is only valid on the MVI instruction. The instruction using this addressing mode is two bytes in length. See **Section 7. Instruction Set** in *PSoC Designer: Assembly Language User Guide* for further details on MVI instruction.

Table 21: Source Indirect Post Increment

Opcode	Operand 1
Instruction	Source Address Address

Example:

```
                 ;In this case, the value
                 ;in the memory location at
                 ;address 8 is an indirect
                 ;address. The memory
                 ;location pointed to by
MVI   A,   [8]   ;the indirect address is
                 ;moved into the
                 ;Accumulator. The
                 ;indirect address is then
                 ;incremented.
```

2.3.10 Destination Indirect Post Increment

The result of an instruction using this addressing mode is placed within the memory space. Operand 1 is an address pointing to a location within the memory space, which contains an address (the indirect address) for the destination of the instruction. The indirect address is incremented as part of the instruction execution. The source for the instruction is the Accumulator. This addressing mode is only valid on the MVI instruction. The instruction using this addressing mode is two bytes in length.

Table 22: Destination Indirect Post Increment

Opcode	Operand 1
Instruction	Destination Address Address

Example:

```
                    ;In this case, the
                    ;value in the memory
                    ;location at address 8
                    ;is an indirect
                    ;address. The
MVI    [8],  A      ;Accumulator is moved
                    ;into the memory
                    ;location pointed to by
                    ;the indirect address.
                    ;The indirect address
                    ;is then incremented.
```

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

7.2.2 24V1/24V2 Frequency Selection

The following table shows the resulting frequencies for 24V1 and 24V2 based on the value written to the OSC_CR1 register.

Table 42: 24V1/24V2 Frequency Selection

Reg. Value	24V1 MHz	24V2 kHz	Reg. Value	24V1 MHz	24V2 kHz	Reg. Value	24V1 MHz	24V2 kHz	Reg. Value	24V1 MHz	24V2 kHz
00	24.00	24000.00	40	4.80	4800.00	80	2.67	2666.67	C0	1.85	1846.15
01	24.00	12000.00	41	4.80	2400.00	81	2.67	1333.33	C1	1.85	923.08
02	24.00	8000.00	42	4.80	1600.00	82	2.67	888.89	C2	1.85	615.38
03	24.00	6000.00	43	4.80	1200.00	83	2.67	666.67	C3	1.85	461.54
04	24.00	4800.00	44	4.80	960.00	84	2.67	533.33	C4	1.85	369.23
05	24.00	4000.00	45	4.80	800.00	85	2.67	444.44	C5	1.85	307.69
06	24.00	3428.57	46	4.80	685.71	86	2.67	380.95	C6	1.85	263.74
07	24.00	3000.00	47	4.80	600.00	87	2.67	333.33	C7	1.85	230.77
08	24.00	2666.67	48	4.80	533.33	88	2.67	296.30	C8	1.85	205.13
09	24.00	2400.00	49	4.80	480.00	89	2.67	266.67	C9	1.85	184.62
0A	24.00	2181.82	4A	4.80	436.36	8A	2.67	242.42	CA	1.85	167.83
0B	24.00	2000.00	4B	4.80	400.00	8B	2.67	222.22	CB	1.85	153.85
0C	24.00	1846.15	4C	4.80	369.23	8C	2.67	205.13	CC	1.85	142.01
0D	24.00	1714.29	4D	4.80	342.86	8D	2.67	190.48	CD	1.85	131.87
0E	24.00	1600.00	4E	4.80	320.00	8E	2.67	177.78	CE	1.85	123.08
0F	24.00	1500.00	4F	4.80	300.00	8F	2.67	166.67	CF	1.85	115.38
10	12.00	12000.00	50	4.00	4000.00	90	2.40	2400.00	D0	1.71	1714.29
11	12.00	6000.00	51	4.00	2000.00	91	2.40	1200.00	D1	1.71	857.14
12	12.00	4000.00	52	4.00	1333.33	92	2.40	800.00	D2	1.71	571.43
13	12.00	3000.00	53	4.00	1000.00	93	2.40	600.00	D3	1.71	428.57
14	12.00	2400.00	54	4.00	800.00	94	2.40	480.00	D4	1.71	342.86
15	12.00	2000.00	55	4.00	666.67	95	2.40	400.00	D5	1.71	285.71
16	12.00	1714.29	56	4.00	571.43	96	2.40	342.86	D6	1.71	244.90
17	12.00	1500.00	57	4.00	500.00	97	2.40	300.00	D7	1.71	214.29
18	12.00	1333.33	58	4.00	444.44	98	2.40	266.67	D8	1.71	190.48
19	12.00	1200.00	59	4.00	400.00	99	2.40	240.00	D9	1.71	171.43
1A	12.00	1090.91	5A	4.00	363.64	9A	2.40	218.18	DA	1.71	155.84
1B	12.00	1000.00	5B	4.00	333.33	9B	2.40	200.00	DB	1.71	142.86
1C	12.00	923.08	5C	4.00	307.69	9C	2.40	184.62	DC	1.71	131.87
1D	12.00	857.14	5D	4.00	285.71	9D	2.40	171.43	DD	1.71	122.45
1E	12.00	800.00	5E	4.00	266.67	9E	2.40	160.00	DE	1.71	114.29
1F	12.00	750.00	5F	4.00	250.00	9F	2.40	150.00	DF	1.71	107.14
20	8.00	8000.00	60	3.43	3428.57	A0	2.18	2181.82	E0	1.60	1600.00
21	8.00	4000.00	61	3.43	1714.29	A1	2.18	1090.91	E1	1.60	800.00
22	8.00	2666.67	62	3.43	1142.86	A2	2.18	727.27	E2	1.60	533.33
23	8.00	2000.00	63	3.43	857.14	A3	2.18	545.45	E3	1.60	400.00
24	8.00	1600.00	64	3.43	685.71	A4	2.18	436.36	E4	1.60	320.00
25	8.00	1333.33	65	3.43	571.43	A5	2.18	363.64	E5	1.60	266.67
26	8.00	1142.86	66	3.43	489.80	A6	2.18	311.69	E6	1.60	228.57
27	8.00	1000.00	67	3.43	428.57	A7	2.18	272.73	E7	1.60	200.00
28	8.00	888.89	68	3.43	380.95	A8	2.18	242.42	E8	1.60	177.78
29	8.00	800.00	69	3.43	342.86	A9	2.18	218.18	E9	1.60	160.00
2A	8.00	727.27	6A	3.43	311.69	AA	2.18	198.35	EA	1.60	145.45
2B	8.00	666.67	6B	3.43	285.71	AB	2.18	181.82	EB	1.60	133.33
2C	8.00	615.38	6C	3.43	263.74	AC	2.18	167.83	EC	1.60	123.08
2D	8.00	571.43	6D	3.43	244.90	AD	2.18	155.84	ED	1.60	114.29
2E	8.00	533.33	6E	3.43	228.57	AE	2.18	145.45	EE	1.60	106.67
2F	8.00	500.00	6F	3.43	214.29	AF	2.18	136.36	EF	1.60	100.00
30	6.00	6000.00	70	3.00	3000.00	B0	2.00	2000.00	F0	1.50	1500.00
31	6.00	3000.00	71	3.00	1500.00	B1	2.00	1000.00	F1	1.50	750.00
32	6.00	2000.00	72	3.00	1000.00	B2	2.00	666.67	F2	1.50	500.00
33	6.00	1500.00	73	3.00	750.00	B3	2.00	500.00	F3	1.50	375.00
34	6.00	1200.00	74	3.00	600.00	B4	2.00	400.00	F4	1.50	300.00
35	6.00	1000.00	75	3.00	500.00	B5	2.00	333.33	F5	1.50	250.00
36	6.00	857.14	76	3.00	428.57	B6	2.00	285.71	F6	1.50	214.29
37	6.00	750.00	77	3.00	375.00	B7	2.00	250.00	F7	1.50	187.50
38	6.00	666.67	78	3.00	333.33	B8	2.00	222.22	F8	1.50	166.67
39	6.00	600.00	79	3.00	300.00	B9	2.00	200.00	F9	1.50	150.00
3A	6.00	545.45	7A	3.00	272.73	BA	2.00	181.82	FA	1.50	136.36
3B	6.00	500.00	7B	3.00	250.00	BB	2.00	166.67	FB	1.50	125.00
3C	6.00	461.54	7C	3.00	230.77	BC	2.00	153.85	FC	1.50	115.38
3D	6.00	428.57	7D	3.00	214.29	BD	2.00	142.86	FD	1.50	107.14
3E	6.00	400.00	7E	3.00	200.00	BE	2.00	133.33	FE	1.50	100.00
3F	6.00	375.00	7F	3.00	187.5	BF	2.00	125.00	FF	1.50	93.75

8.2 Interrupt Control Architecture

The interrupt controller contains a separate flip-flop for each interrupt. When an interrupt is generated, it is registered as a pending interrupt. It will stay pending until it is serviced, a reset occurs, or there is a write to the INT_VC Register. A pending interrupt will only generate an interrupt request when enabled by the appropriate mask bit in the Digital PSoC Block Interrupt Mask Register (INT_MSK1) or General Interrupt Mask Register (INT_MSK0), and the Global IE bit in the CPU_F register is set.

Additionally, for GPIO Interrupts, the appropriate enable and interrupt-type bits for each I/O pin must be set (see section 6.0, Table 29 on page 31, Table 33 on page 33, and Table 34 on page 34). For Analog Column Interrupts, the interrupt source must be set (see section 10.10 and Table 77 on page 101).

During the servicing of any interrupt, the MSB and LSB of Program Counter and Flag registers (CPU_PC and CPU_F) are stored onto the program stack by an automatic `CALL` instruction (13 cycles) generated during the interrupt acknowledge process. The user firmware may preserve and restore processor state during an interrupt using the `PUSH` and `POP` instructions. The memory oriented CPU architecture requires minimal state saving during interrupts, providing very fast interrupt context switching. The Program Counter and Flag registers (CPU_PC and CPU_F) are restored when the `RETI` instruction is executed. If two or more interrupts are pending at the same time, the higher priority interrupt (lower priority number) will be serviced first.

After a copy of the Flag Register is stored on the stack, the Flag Register is automatically cleared. This disables all interrupts, since the Global IE flag bit is now cleared. Executing a `RETI` instruction restores the Flag register, and re-enables the Global Interrupt bit.

Nested interrupts can be accomplished by re-enabling interrupts inside an interrupt service routine. To do this, set the IE bit in the Flag Register. The user must store sufficient information to maintain machine state if this is done.

Each digital PSoC block has its own unique Interrupt Vector and Interrupt Enable bit. There are also individual interrupt vectors for each of the Analog columns, Supply Voltage Monitor, Sleep Timer and General Purpose I/Os.

8.3 Interrupt Vectors

Table 43: Interrupt Vector Table

Address	Interrupt Priority Number	Description
0x0004	1	Supply Monitor Interrupt Vector
0x0008	2	DBA00 PSoC Block Interrupt Vector
0x000C	3	DBA01 PSoC Block Interrupt Vector
0x0010	4	DBA02 PSoC Block Interrupt Vector
0x0014	5	DBA03 PSoC Block Interrupt Vector
0x0018	6	DCA04 PSoC Block Interrupt Vector
0x001C	7	DCA05 PSoC Block Interrupt Vector
0x0020	8	DCA06 PSoC Block Interrupt Vector
0x0024	9	DCA07 PSoC Block Interrupt Vector
0x0028	10	Acolumn 0 Interrupt Vector
0x002C	11	Acolumn 1 Interrupt Vector
0x0030	12	Acolumn 2 Interrupt Vector
0x0034	13	Acolumn 3 Interrupt Vector
0x0038	14	GPIO Interrupt Vector
0x003C	15	Sleep Timer Interrupt Vector
0x0040		On-Chip Program Memory Starts

The interrupt process vectors the Program Counter to the appropriate address in the Interrupt Vector Table. Typically, these addresses contain `JMP` instructions to the start of the interrupt handling routine for the interrupt.

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)

Digital Communications Type A Block 05 Function Register (DCA05FN, Address = Bank 1, 34h)
Digital Communications Type A Block 06 Function Register (DCA06FN, Address = Bank 1, 38h)
Digital Communications Type A Block 07 Function Register (DCA07FN, Address = Bank 1, 3Ch)

9.2.2 Digital Basic Type A / Communications Type A Block xx Input Register

The Digital Basic Type A / Communications Type A Block xx Input Register (DBA00IN-DCA07IN) consists of 4 bits [3:0] to select the block input clock and 4 bits [7:4] to select the primary data/enable input. The actual usage of the input data/enable is function dependent.

Table 48: Digital Basic Type A / Communications Type A Block xx Input 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 [3]	Data [2]	Data [1]	Data [0]	Clock [3]	Clock [2]	Clock [1]	Clock [0]

Bit [7:4]: **Data [3:0] Data Enable Source Select**
0 0 0 0 = Data = 0
0 0 0 1 = Data = 1
0 0 1 0 = Digital Block 03
0 0 1 1 = Chain Function to Previous Block
0 1 0 0 = Analog Column Comparator 0
0 1 0 1 = Analog Column Comparator 1
0 1 1 0 = Analog Column Comparator 2
0 1 1 1 = Analog Column Comparator 3
1 0 0 0 = Global Output[0] (for Digital Blocks 00 to 03) **or** Global Output[4] (for Digital Blocks 04 to 07)
1 0 0 1 = Global Output[1] (for Digital Blocks 00 to 03) **or** Global Output[5] (for Digital Blocks 04 to 07)
1 0 1 0 = Global Output[2] (for Digital Blocks 00 to 03) **or** Global Output[6] (for Digital Blocks 04 to 07)
1 0 1 1 = Global Output[3] (for Digital Blocks 00 to 03) **or** Global Output[7] (for Digital Blocks 04 to 07)
1 1 0 0 = Global Input[0] (for Digital Blocks 00 to 03) **or** Global Input[4] (for Digital Blocks 04 to 07)
1 1 0 1 = Global Input[1] (for Digital Blocks 00 to 03) **or** Global Input[5] (for Digital Blocks 04 to 07)
1 1 1 0 = Global Input[2] (for Digital Blocks 00 to 03) **or** Global Input[6] (for Digital Blocks 04 to 07)
1 1 1 1 = Global Input[3] (for Digital Blocks 00 to 03) **or** Global Input[7] (for Digital Blocks 04 to 07)

Bit [3:0]: **Clock [3:0] Clock Source Select**
0 0 0 0 = Clock Disabled
0 0 0 1 = Global Output[4] (for Digital Blocks 00 to 03) **or** Global Output[0] (for Digital Blocks 04 to 07)
0 0 1 0 = Digital Block 03 (Primary Output)
0 0 1 1 = Previous Digital PSoC block (Primary Output)
0 1 0 0 = 48M
0 1 0 1 = 24V1
0 1 1 0 = 24V2
0 1 1 1 = 32k
1 0 0 0 = Global Output[0] (for Digital Blocks 00 to 03) **or** Global Output[4] (for Digital Blocks 04 to 07)
1 0 0 1 = Global Output[1] (for Digital Blocks 00 to 03) **or** Global Output[5] (for Digital Blocks 04 to 07)
1 0 1 0 = Global Output[2] (for Digital Blocks 00 to 03) **or** Global Output[6] (for Digital Blocks 04 to 07)
1 0 1 1 = Global Output[3] (for Digital Blocks 00 to 03) **or** Global Output[7] (for Digital Blocks 04 to 07)
1 1 0 0 = Global Input[0] (for Digital Blocks 00 to 03) **or** Global Input[4] (for Digital Blocks 04 to 07)
1 1 0 1 = Global Input[1] (for Digital Blocks 00 to 03) **or** Global Input[5] (for Digital Blocks 04 to 07)
1 1 1 0 = Global Input[2] (for Digital Blocks 00 to 03) **or** Global Input[6] (for Digital Blocks 04 to 07)
1 1 1 1 = Global Input[3] (for Digital Blocks 00 to 03) **or** Global Input[7] (for Digital Blocks 04 to 07)

Digital Basic Type A Block 00 Input Register (DBA00IN, Address = Bank 1, 21h)
Digital Basic Type A Block 01 Input Register (DBA01IN, Address = Bank 1, 25h)
Digital Basic Type A Block 02 Input Register (DBA02IN, Address = Bank 1, 29h)
Digital Basic Type A Block 03 Input Register (DBA03IN, Address = Bank 1, 2Dh)
Digital Communications Type A Block 04 Input Register (DCA04IN, Address = Bank 1, 31h)
Digital Communications Type A Block 05 Input Register (DCA05IN, Address = Bank 1, 35h)

2. Disabled State

   When the Control Register Enable bit is set to ‘0’, the internal block clock is turned off. A write to Data Register 1 (Period) is loaded directly into Data Register 0 (Counter) to initialize or reset the count. All outputs are low and the block interrupt is held low. Disabling a counter does not affect the current count value and it may be read by the CPU. Two reads are required to read each byte of a multi-byte register. One to transfer each Data Register 0 count value to the associated Data Register 2 capture register, then one to read the result in Data Register 2.

3. Reading the Count Value

   A CPU read of Data Register 0 (count value) will overwrite Data Register 2 (compare value). Therefore, when reading the current count, a previously written compare value will be overwritten.

4. Extra Count

   In a Counter User Module, the data input is an enable for counting. Normally, when the enable goes low, the counter will hold the current count. However, if the enable happens to go low in the same clock period as Terminal Count (count of all 0's), one additional count will occur that will reload the counter from the Period Register. Once the counter is reloaded from the Period Register, counting will stop.

9.5.3 Deadband Generator

9.5.3.1 Summary

The Deadband function produces two output waveforms, F0 and F1, with the same frequency as the input, but “under-lapped” so they are never both high at the same time. An 8-bit down counter controls the length of the “dead time” during which both output signals are low. When the deadband function detects a rising edge on the input waveform, the F1 output signal goes low and the counter decrements from its initial value to its terminal count. When the down counter reaches zero, the F0 output signal goes high. The process reverses on the falling edge of the input waveform so that after the same dead time, F1 goes high until the input signal transitions again. Dead-band generator PSoC blocks cannot be chained to increase the width of the down counter beyond 8 bits or 256 dead-time “ticks.”

9.5.3.2 Registers

Data Register 1 stores the count that controls the elapsed dead time. Data Register 0 holds the current state of the dead-time down counter. If the function is disabled, writing a period into Data Register 1, will automatically load Data Register 0 with the deadband period. This period is automatically re-loaded into the counter on each edge of the input signal. Data Register 2 is unused. Control Register 0 contains one bit to enable/disable the function.

9.5.3.3 Inputs

The input controls the period and duty cycle of the dead-band generator outputs. This input is fixed to be derived from the primary output of the previous block. If this signal is pulse-width modulated, i.e., if a PWM block is configured as the previous block, the dead-band outputs will be similarly modulated. The F0 output corresponds to the duty cycle of the input (less the dead time) and F1 to the duty cycle of the inverted input (again, less the dead time). The clock input to the dead-band generator controls the rate at which the down counter is decremented. The primary data input is the “Kill” Signal. When this signal is asserted high, both F0 and F1 outputs will go low. The multiplexers selecting these input are controlled by the PSoC block Input Register (DBA00IN-DCA07IN).

9.5.3.4 Outputs

Both the F0 and F1 outputs can be driven onto the Global Output bus. If the next PSoC block selects “Previous PSoC block” for its clock input, it only “sees” the F0 output of the dead-band function. The PSoC block Output Register (DBA00OU-DCA07OU) controls output options.

9.5.3.5 Interrupts

The rising edge of the F0 signal provides the interrupt for this block.

9.5.3.6 Usage Notes

1. Constraints

   The dead time must not exceed the minimum of the input signal’s pulse-width high and pulse-width low time, less two CPU clocks. Dead time equals the period of the input clock times one plus the value written to Data Register 1.

10.5 Analog PSoC Block Clocking Options

All analog PSoC blocks in a particular Analog Column share the same clock signal. Choosing the clocking for an analog PSoC block is a two-step process.

1. First, if the user wants to use the **ACLK0** and **ACLK1** system-clocking signals, the digital PSoC blocks that serve as the source for these signals must be selected. This selection is made in the Analog Clock Select Register (CLK_CR1).
2. Next, the user must select the source for the **Acolumn0**, **Acolumn1**, **Acolumn2**, and **Acolumn3** system-clocking signals. The user will choose the clock for Acolumnx[1:0] bits in the Analog Column Clock Select Register (CLK_CR0). Each analog PSoC block in a particular Analog Column is clocked from the **Acolumn[x]** system-clocking signal for that column. (Note that the Acolumn[x] signals have a 1:4 divider on them.)

10.5.1 Analog Column Clock Select Register

Table 64: Analog Column Clock Select 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	Acolumn3 [1]	Acolumn3 [0]	Acolumn2 [1]	Acolumn2 [0]	Acolumn1 [1]	Acolumn1 [0]	Acolumn0 [1]	Acolumn0 [0]

Bit [7:6]: **Acolumn3 [1:0]**
0 0 = 24V1
0 1 = 24V2
1 0 = ACLK0
1 1 = ACLK1

Bit [5:4]: **Acolumn2 [1:0]**
0 0 = 24V1
0 1 = 24V2
1 0 = ACLK0
1 1 = ACLK1

Bit [3:2]: **Acolumn1 [1:0]**
0 0 = 24V1
0 1 = 24V2
1 0 = ACLK0
1 1 = ACLK1

Bit [1:0]: **Acolumn0 [1:0]**
0 0 = 24V1
0 1 = 24V2
1 0 = ACLK0
1 1 = ACLK1

Analog Column Clock Select Register (CLK_CR0, Address = Bank 1, 60h)

10.9.2.3 Analog Switch Cap Type B Block xx Control 2 Register

AnalogBus gates the output to the 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 ClockPhase continuously drives the analog column bus.

CompBus controls the output to the column comparator bus. Note that if the comparator bus is not driven by anything in the column, it is pulled low. The comparator output is evaluated on the rising edge of internal PHI1 and is latched so it is available during internal PHI2.

AutoZero controls the shorting of the output to the inverting input of the op-amp. When shorted, the op-amp is basically a follower. The output is the op-amp offset. By using the feedback capacitor of the integrator, the block can memorize the offset and create an offset cancellation scheme. AutoZero also controls a pair of switches between the A and B branches and the summing node of the op-amp. If AutoZero is enabled, then the pair of switches is active. AutoZero also affects the function of the FSW1 bit in Control 3 Register.

The CCap bits set the value of the capacitor in the C path.

10.9.2.4 Analog Switch Cap Type B 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 (ASB11CR2, ASB13CR2, ASB20CR2, ASB22CR2). 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.

BSW is used to control switching in the B branch. If disabled, the B capacitor branch is a continuous time branch like the C branch of the SC A Block. If enabled, then on internal PHI1, both ends of the cap are switched to analog ground. On internal PHI2, one end is switched to the B input and the other end is switched to the summing node.

BMuxSCB controls muxing to the input of the B capacitor branch. The B branch can be switched or unswitched.

Table 76: Analog Switch Cap Type B 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]	BSW	BMuxSCB	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
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: **BSW** Enable switching in branch
0 = B branch is a continuous time path
1 = B branch is switched with internal PHI2 sampling

Bit 2: **BMuxSCB** Encoding for selecting B inputs. (Note that the available mux inputs vary by individual PSoC block)
ASB11 ASB13 ASB20 ASB22
0 = ACA00 ACA02 ASB11 ASB13
1 = ACA01 ACA03 ASA10 ASA12

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 B Block 11 Control 3 Register (ASB11CR3, Address = Bank 0/1, 87h)
Analog Switch Cap Type B Block 13 Control 3 Register (ASB13CR3, Address = Bank 0/1, 8Fh)
Analog Switch Cap Type B Block 20 Control 3 Register (ASB20CR3, Address = Bank 0/1, 93h)
Analog Switch Cap Type B Block 22 Control 3 Register (ASB22CR3, Address = Bank 0/1, 9Bh)

10.12.3 Analog Output Buffers

The user has the option to output up to four analog signals on the pins of the device. This is done by enabling the analog output buffers associated with each Analog Column. The enable bits for the analog output buffers are contained in the Analog Output Buffer Control Register (ABF_CR).

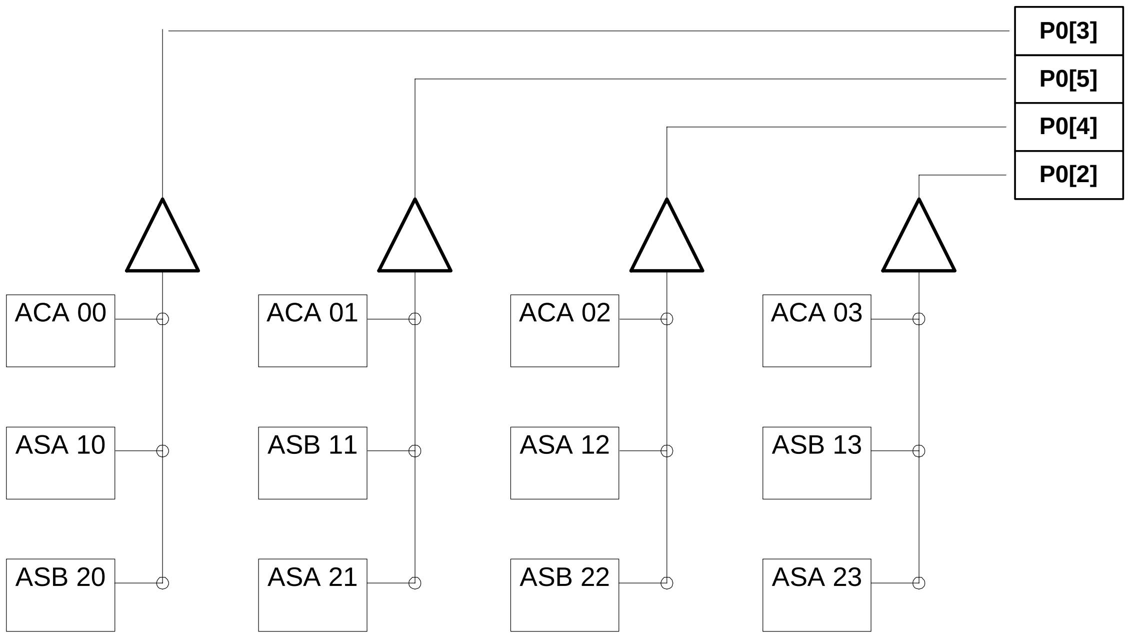

Figure 28: Analog Output Buffers

10.12.4 Analog Output Buffer Control Register

Table 81: Analog Output Buffer Control Register

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
Bit Name	ACol1Mux	ACol2Mux	ABUF1EN	ABUF2EN	ABUF0EN	ABUF3EN	Reserved	PWR

Bit 7: **ACol1Mux**
0 = Set column 1 input to column 1 input mux output
1 = Set column 1 input to column 0 input mux output

Bit 6: **ACol2Mux**
0 = Set column 2 input to column 2 input mux output
1 = Set column 2 input to column 3 input mux output

Bit 5: **ABUF1EN** Enables the analog output buffer for Analog Column 1 (Pin P0[5])
0 = Disable analog output buffer
1 = Enable analog output buffer

Bit 4: **ABUF2EN** Enables the analog output buffer for Analog Column 2 (Pin P0[4])
0 = Disable analog output buffer
1 = Enable analog output buffer

Bit 3: **ABUF0EN** Enables the analog output buffer for Analog Column 0 (Pin P0[3])
0 = Disable analog output buffer
1 = Enable analog output buffer

Bit 2: **ABUF3EN** Enables the analog output buffer for Analog Column 3 (Pin P0[2])
0 = Disable analog output buffer
1 = Enable analog output buffer

Bit [1]: **Reserved** Must be left as 0

Bit [0]: **PWR** Determines power level of all output buffers
0 = Low output power
1 = High output power

Analog Output Buffer Control Register (ABF_CR, Address = Bank 1, 62h)

10.13 Analog Modulator

The user has the capability to use the Analog Switch Cap Type A PSoC Blocks in Columns 0 and 2 as amplitude modulators. The Analog Modulator Control Register (AMD_CR) allows the user to select the appropriate modulating signal. When the modulating signal is low, the polarity follows the setting of the ASign bit set in the Analog Switch Cap Type A Control 0 Register (ASAxxCR0). When this signal is high, the normal gain polarity of the PSoC block is inverted.

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.

This filter is implemented using a combination of hardware and software resources. Hardware is used to accumulate the high-speed in-coming data while the software is used to process the lower speed, enhanced resolution data for output.

Table 91: Decimator/Incremental Control 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	IGEN [3]	IGEN [2]	IGEN [1]	IGEN [0]	ICCKSEL	DCol [1]	DCol [0]	DCLKSEL

Bit [7:4]: **IGEN [3:0]** Individual enables for each analog column that gates the Analog Comparator based on the ICCKSEL input (Bit 3)

Bit 3: **ICCKSEL** Clock select for Incremental gate function
0 = Digital Basic Type A Block 02
1 = Digital Communications Type A Block 06

Bit [2:1]: **DCol [1:0]** Selects Analog Column Comparator source
0 0 = Analog Column Comparator 0
0 1 = Analog Column Comparator 1
1 0 = Analog Column Comparator 2
1 1 = Analog Column Comparator 3

Bit 0: **DCLKSEL** Clock select for Decimator latch
0 = Digital Basic Type A Block 02
1 = Digital Communications Type A Block 06

Decimator Incremental Register (DEC_CR, Address = Bank 0, E6h)

Table 92: Decimator Data High 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 high order byte within the 16-bit decimator data registers
Any 8-bit data value when written will cause both the Decimator Data High (DEC_DH) and Decimator Data Low (DEC_DL) registers to be cleared

Decimator High Register (DEC_DH / DEC_CL, Address = Bank 0, E4h)

Table 93: Decimator Data 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 when read is the low order byte within the 16 bit decimator data registers

Decimator Data Low Register (DEC_DL, Address = Bank 0, E5h)

11.7 Internal Voltage Reference

An internal bandgap voltage reference source is provided on-chip. This reference is used for the Supply Voltage Monitor, and can also be accessed by the user as a reference voltage for analog operations. There is a Bandgap Oscillator Trim Register (BDG_TR) used to calibrate this reference into specified tolerance. Factory-programmed trim values are available for 5.0V and 3.3V operation. The 5.0V value is loaded in the BDG_TR register upon reset. This register must be adjusted when operating voltage outside the range for which factory calibration was set. Changing the factory-programmed trim value is done using the Table Read Supervisor Call routine, and is documented in 11.8.

Table 97: Bandgap 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	FMRD	BGT[2]	BGT[1]	BGT[0]	BGO[3]	BGO[2]	BGO[1]	BGO[0]

Bit 7: **FMRD**
0 = Enable voltage divider between BG and Flash (User must not use other than this setting)
1 = Disable voltage divider between BG and Flash (Test purposes only)

Bit [6:4]: **BGT [2:0]** Provides Temperature Curve compensation

Bit [3:0]: **BGO [3:0]** Provides +/- 5% Offset Trim to center Vbg to 1.30V

1. FS = Factory set trim value

Bandgap Trim Register (BDG_TR, Address = Bank 1, EAh)

11.8 Supervisor ROM/System Supervisor Call Instruction

The parts in this family have a Supervisor ROM to manage the programming, erasure, and protection of the on-chip Flash user program space. The Supervisor ROM also gives the user the capability to read the internal product ID, access factory trim values, as well as calculate checksums on blocks of the Flash memory space.

The System Supervisor Call instruction (SSC, opcode/byte 00h) provides the method for the user to access the pre-existing routines in the Supervisor ROM to implement these functions. This instruction sets the Flags Register (CPU_F) bit 3 to 1 and performs an interrupt to address 0000 into the Supervisory ROM. The flag and old PC are pushed onto the Stack. The fact that the flag pushed has F[3] = 1 is irrelevant as the `RETI` instruction always clears F[3]. The Supervisory code at 0000 does a `JACC` table lookup based on the Accumulator value, which is effectively another level of instruction encoding. This service table implements the vectors to the various supervisory functions. The user must set several parameters when utilizing these functions. The parameters are written to 5 bytes of an 8-byte block near the top of RAM memory space.

Access to these functions must be through the Flash APIs provided in PSoC Designer and described in Application Note AN2015.

The following table documents each function, as well as the required parameter values:

11.10.1 Data File Read

The user's data file should be read into the programmer. The checksum should be calculated by the programmer for each record and compared to the record checksum stored in the file for each record. If there is an error, a message should be sent to the user explaining that the file has a checksum error and the programming should not be allowed to continue.

11.10.2 Programmer Flow

The following sequence (with descriptions) is the main flow used to program the devices: (Note that failure at any step will result in termination of the flow and an error message to the device programmer's operator.)

11.10.2.1 Verify Silicon ID

The silicon ID is read and verified against the expected value. If it is not the expected value, then the device is failed and an error message is sent to the device programmer's operator.

This test will detect a bad connection to the programmer or an incorrect device selection on the programmer.

The silicon ID test is required to be first in the flow and cannot be bypassed. The sequence is as follows:

```
Set Vcc=0V
Set SDATA=HighZ
Set SCLK=VILP
Set Vcc=Vccp
Start the programmer's SCLK driver
"free running"
WAIT-AND-POLL
ID-SETUP
WAIT-AND-POLL
READ-ID-WORD
```

Notes: See "DC Specifications" table in section 13 for value of Vccp and VILP. See "AC Specifications" table in section 13 for value of frequency for the SCLK driver (Fsclk).

11.10.2.2 Erase

The Flash memory is erased. This is accomplished by the following sequence:

```
SET-CLK-FREQ(num_MHz_times_5)
Erase All
WAIT-AND-POLL
```

11.10.2.3 Program

The Flash is programmed with the contents of the user's programming file. This is accomplished by the following sequence:

```
For num_block = 0 to max_data_block
For address =0 to 63
WRITE-BYTE(address,data):
End for address loop
SET-CLK-FREQ(num_MHz_times_5)
SET-BLOCK-NUM(num_block)
PROGRAM-BLOCK
WAIT-AND-POLL
End for num_block loop
```

11.10.2.4 Verify (at Low Vcc and High Vcc)

The device data is read out to compare to the data in the user's programming file. This is accomplished by the following sequence:

```
For num_block = 0 to max_data_block
SET-BLOCK-NUM (num_block)
VERIFY-SETUP
Wait & POLL the SDATA for a high to
low transition
For address =0 to max_byte_per_block
READ-BYTE(address,data)
End for address loop
End for num_block loop
```

Note: This should be done 2 times; once at Vcc=Vcclv and once at Vcc=Vcchv.

11.10.2.5 Set Security

The security operation protects certain blocks from being read or changed. This is done at the end of the flow so that the security does not interfere with the verify step. Security is set with the following sequence:

```
For address =0 to 63
WRITE-SECURITY-BYTE(address,data):
End for address loop
SET-CLK-FREQ(num_MHz_times_5)
SECURE
WAIT-AND-POLL
```

Note: This sequence is done at Vcc=Vccp.

13.2.1.2 3.3V Specifications

The following table lists guaranteed maximum and minimum specifications for the voltage and temperature ranges, 3.3V +/- 10% and -40°C <= T_A <= 85°C. The Operational Amplifier is a component of both the Analog Continuous Time PSoC blocks and the Analog Switch Cap PSoC blocks. The guaranteed specifications are measured in the Analog Continuous Time PSoC block. Typical parameters apply to 5V at 25°C and are for design guidance only. For 5V operation, see Table 105 on page 130.

Table 106: 3.3V DC Operational Amplifier Specifications

Symbol	3.3V DC Operational Amplifier Specifications	Minimum	Typical	Maximum	Unit
	Input Offset Voltage (Absolute Value)	-	7	30	mV
	Average Input Offset Voltage Drift	-	+24	-	µV/°C
	Input Leakage Current[1]	-	2	700	nA
	Input Capacitance[2]	.32	.36	.42	pF
	Common Mode Voltage Range[3]	.5	-	V_{cc} - 1.0	VDC
	Common Mode Rejection Ratio	80	-	-	dB
	Open Loop Gain	80	-	-	dB
	High Output Voltage Swing (Worst Case Internal Load) Bias = Low Bias = Medium Bias = High	 V_{cc} - .4 V_{cc} - .4 V_{cc} - .4	 - - -	 - - -	 V V V
	Low Output Voltage Swing (Worst Case Internal Load) Bias = Low Bias = Medium Bias = High	 - - -	 - - -	 0.1 0.1 0.1	 V V V
	Supply Current (Including Associated AGND Buffer) Bias = Low Bias = Medium Bias = High	 - - -	 80 112 320	 200 300 800	 µA µA µA
	Supply Voltage Rejection Ratio	60	-	-	dB

1. The leakage current includes the Analog Continuous Time PSoC block mux and the analog input mux. The leakage related to the General Purpose I/O pins is not included here.
2. The Input Capacitance includes the Analog Continuous Time PSoC block mux and the analog input mux. The capacitance of the General Purpose I/O pins is not included here.
3. The common-mode input voltage range is measured through an analog output buffer. The specification includes the limitations imposed by the characteristics of the analog output buffer

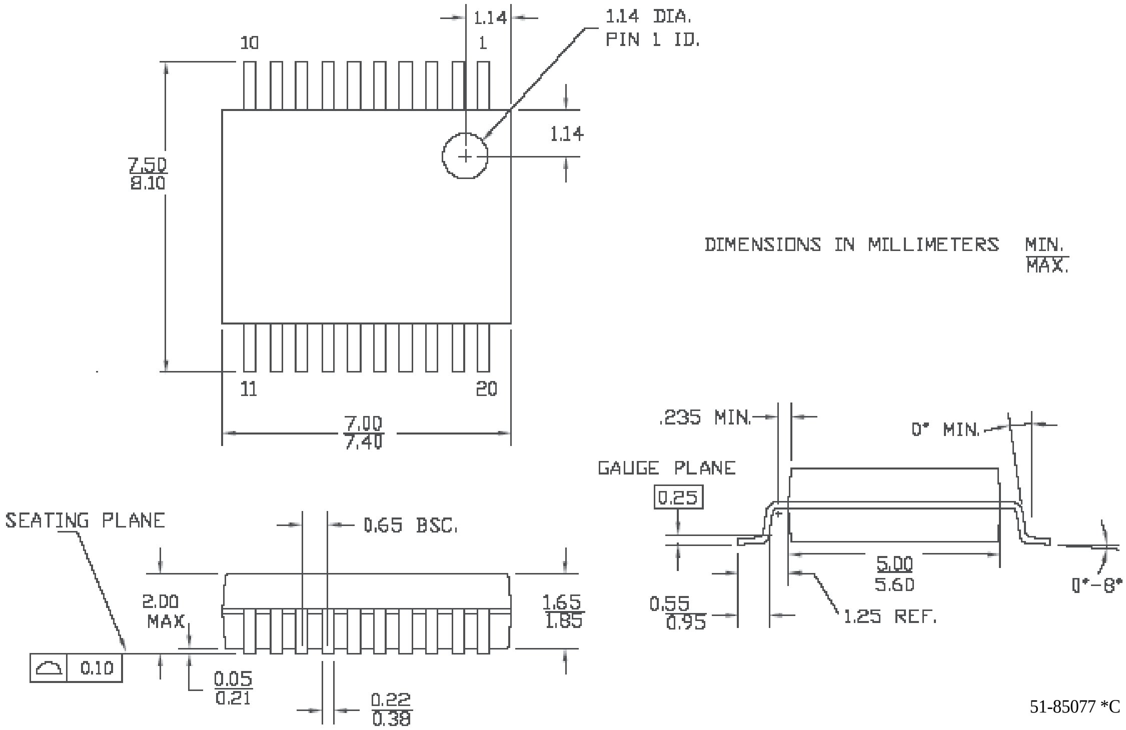

Figure 38: 20-Pin Shrunk Small Outline Package O20

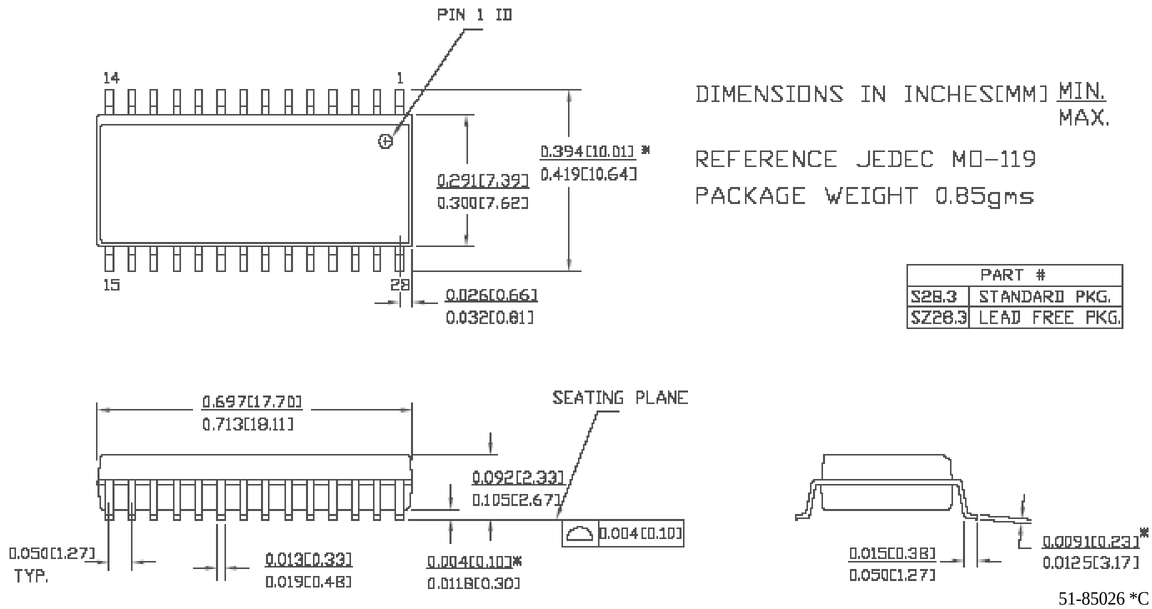

Figure 45: 28-Lead (300-Mil) Molded SOIC S21

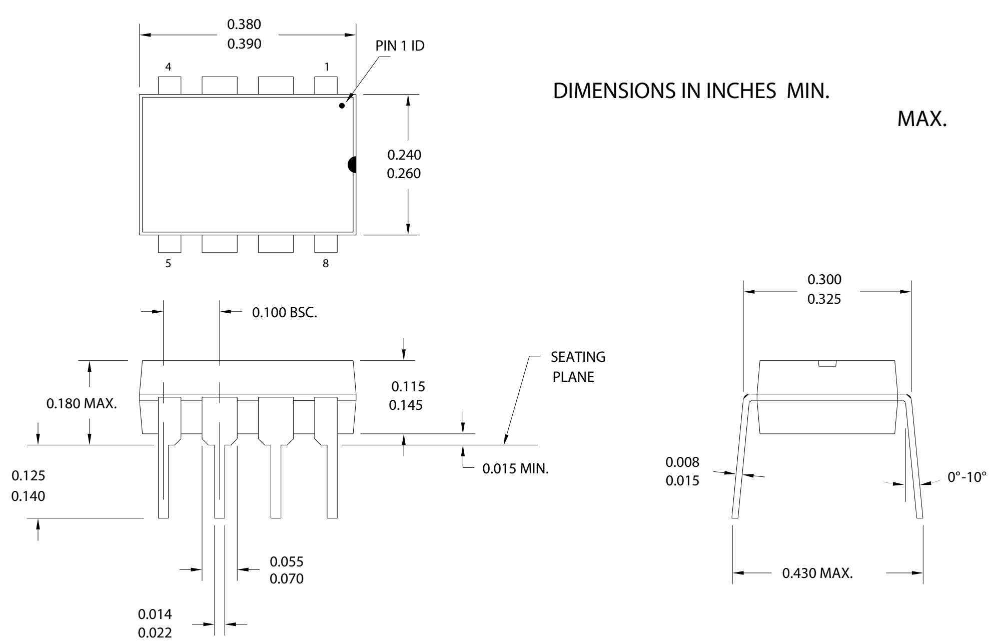

Figure 46: 8-Lead (300-Mil) Molded DIP