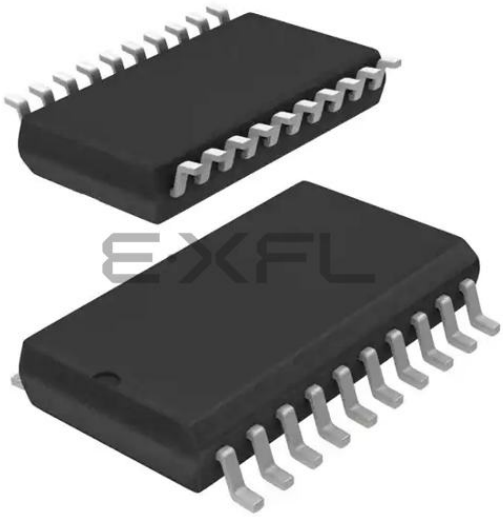

Welcome to E-XFL.COM

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	16
Program Memory Size	8KB (8K 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	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy8c26233-24si

This page has intentionally been left blank.

1.2 Pin-out Descriptions

Table 2: Pin-out 8 Pin

Name	I/O	Pin	Description
P0[7]	I/O	1	Port 0[7] (Analog Input)
P0[5]	I/O	2	Port 0[5] (Analog Input/Output)
P1[1]	I/O	3	Port 1[1] / XtalIn / SCLK
Vss	Power	4	Ground
P1[0]	I/O	5	Port 1[0] / XtalOut / SDATA
P0[2]	I/O	6	Port 0[2] (Analog Input/Output)
P0[4]	I/O	7	Port 0[4] (Analog Input/Output)
Vcc	Power	8	Supply Voltage

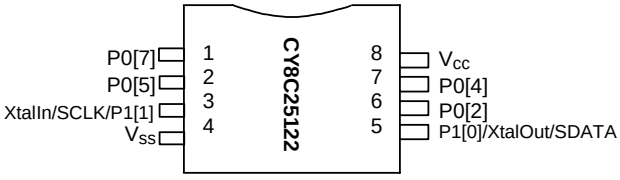

Figure 2: CY8C25122

Table 3: Pin-out 20 Pin

Name	I/O	Pin	Description
P0[7]	I/O	1	Port 0[7] (Analog Input)
P0[5]	I/O	2	Port 0[5] (Analog Input/Output)
P0[3]	I/O	3	Port 0[3] (Analog Input/Output)
P0[1]	I/O	4	Port 0[1] (Analog Input)
SMP	O	5	Switch Mode Pump
P1[7]	I/O	6	Port 1[7]
P1[5]	I/O	7	Port 1[5]
P1[3]	I/O	8	Port 1[3]
P1[1]	I/O	9	Port 1[1] / XtalIn / SCLK
Vss	Power	10	Ground
P1[0]	I/O	11	Port 1[0] / XtalOut / SDATA
P1[2]	I/O	12	Port 1[2]
P1[4]	I/O	13	Port 1[4]
P1[6]	I/O	14	Port 1[6]
XRES	I	15	External Reset
P0[0]	I/O	16	Port 0[0] (Analog Input)
P0[2]	I/O	17	Port 0[2] (Analog Input/Output)
P0[4]	I/O	18	Port 0[4] (Analog Input/Output)
P0[6]	I/O	19	Port 0[6] (Analog Input)
Vcc	Power	20	Supply Voltage

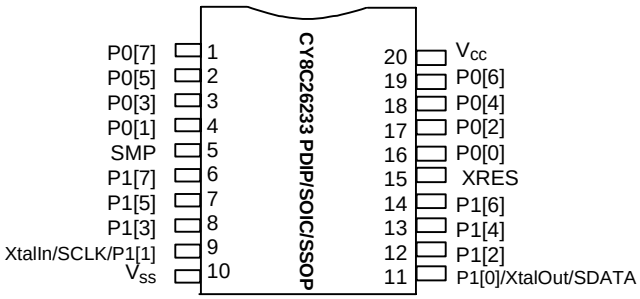

Figure 3: CY8C26233

2.2 CPU Registers

2.2.1 Flags Register

The Flags Register can only be set or reset with logical instruction.

Table 8: Flags Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	1	0
Read/ Write	--	--	--	RW	R	RW	RW	RW
Bit Name	Reserved	Reserved	Reserved	XIO	Super	Carry	Zero	Global IE

Bit 7: **Reserved**
Bit 6: **Reserved**
Bit 5: **Reserved**

Bit 4: **XIO** Set by the user to select between the register banks
0 = Bank 0
1 = Bank 1

Bit 3: **Super** Indicates whether the CPU is executing user code or Supervisor Code. (This code cannot be accessed directly by the user and is not displayed in the ICE debugger.)
0 = User Code
1 = Supervisor Code

Bit 2: **Carry** Set by CPU to indicate whether there has been a carry in the previous logical/arithmetic operation
0 = No Carry
1 = Carry

Bit 1: **Zero** Set by CPU to indicate whether there has been a zero result in the previous logical/arithmetic operation
0 = Not Equal to Zero
1 = Equal to Zero

Bit 0: **Global IE** Determines whether all interrupts are enabled or disabled
0 = Disabled
1 = Enabled

2.2.2 Accumulator Register

Table 9: Accumulator Register (CPU_A)

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 the result of any logical/arithmetic instruction that uses a source addressing mode

1. System - not directly accessible by the user

Examples:

```
                ;In this case, the immediate
                ;value of 7 is added with the
ADD   A,   7    ;Accumulator, and the result
                ;is placed in the
                ;Accumulator.
                ;In this case, the immediate
MOV   X,   8    ;value of 8 is moved to the X
                ;register.
                ;In this case, the immediate
                ;value of 9 is logically
AND   F,   9    ;ANDed with the F register
                ;and the result is placed in
                ;the F register.
```

2.3.2 Source Direct

The result of an instruction using this addressing mode is placed in either the A register or the X register, which is specified as part of the instruction opcode. Operand 1 is an address that points to a location in either the RAM memory space or the register space that is the source for the instruction. Arithmetic instructions require two sources, the second source is the A register or X register specified in the opcode. Instructions using this addressing mode are two bytes in length.

Table 14: Source Direct

Opcode	Operand 1
Instruction	Source Address

Examples:

```
                ;In this case, the
                ;value in the RAM
                ;memory location at
                ;address 7 is added
ADD   A,   [7]  ;with the Accumulator,
                ;and the result is
                ;placed in the
                ;Accumulator.
                ;In this case, the
                ;value in the register
MOV   X,   REG[8] ;space at address 8 is
                ;moved to the X
                ;register.
```

2.3.3 Source Indexed

The result of an instruction using this addressing mode is placed in either the A register or the X register, which is specified as part of the instruction opcode. Operand 1 is added to the X register forming an address that points to a location in either the RAM memory space or the register space that is the source for the instruction. Arithmetic instructions require two sources, the second source is the A register or X register specified in the opcode. Instructions using this addressing mode are two bytes.

Table 15: Source Indexed

Opcode	Operand 1
Instruction	Source Index

Examples:

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

2.3.4 Destination Direct

The result of an instruction using this addressing mode is placed within either the RAM memory space or the register space. Operand 1 is an address that points to the location of the result. The source for the instruction is either the A register or the X register, which is specified as part of the instruction opcode. Arithmetic instructions require two sources, the second source is the location specified by Operand 1. Instructions using this addressing mode are two bytes in length.

Table 16: Destination Direct

Opcode	Operand 1
Instruction	Destination Address

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
MVI   A,   [8]  ;location pointed to by
                ;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.
```

4.3 Register Bank 1 Map

Table 27: Bank 1

Register Name	Address	Data Sheet Page	Access	Register Name	Address	Data Sheet Page	Access	Register Name	Address	Data Sheet Page	Access	Register Name	Address	Data Sheet Page	Access
PRT0DM0	00h	32	W	Reserved	40h			ASA10CR0	80h	88	RW	Reserved	C0h		
PRT0DM1	01h	33	W		41h			ASA10CR1	81h	90	RW		C1h		
PRT0IC0	02h	33	W		42h			ASA10CR2	82h	92	RW		C2h		
PRT0IC1	03h	34	W		43h			ASA10CR3	83h	93	RW		C3h		
PRT1DM0	04h	32	W		44h			ASB11CR0	84h	95	RW		C4h		
PRT1DM1	05h	33	W		45h			ASB11CR1	85h	97	RW		C5h		
PRT1IC0	06h	33	W		46h			ASB11CR2	86h	99	RW		C6h		
PRT1IC1	07h	34	W		47h			ASB11CR3	87h	100	RW		C7h		
PRT2DM0	08h	32	W		48h			ASA12CR0	88h	88	RW		C8h		
PRT2DM1	09h	33	W		49h			ASA12CR1	89h	90	RW		C9h		
PRT2IC0	0Ah	33	W		4Ah			ASA12CR2	8Ah	92	RW		CAh		
PRT2IC1	0Bh	34	W		4Bh			ASA12CR3	8Bh	93	RW		CBh		
PRT3DM0	0Ch	32	W		4Ch			ASB13CR0	8Ch	95	RW		CCh		
PRT3DM1	0Dh	33	W		4Dh			ASB13CR1	8Dh	97	RW		CDh		
PRT3IC0	0Eh	33	W		4Eh			ASB13CR2	8Eh	99	RW		CEh		
PRT3IC1	0Fh	34	W		4Fh			ASB13CR3	8Fh	100	RW		CFh		
PRT4DM0	10h	32	W		50h			ASB20CR0	90h	95	RW		D0h		
PRT4DM1	11h	33	W		51h			ASB20CR1	91h	97	RW		D1h		
PRT4IC0	12h	33	W		52h			ASB20CR2	92h	99	RW		D2h		
PRT4IC1	13h	34	W		53h			ASB20CR3	93h	100	RW		D3h		
PRT5DM0	14h	32	W		54h			ASA21CR0	94h	88	RW		D4h		
PRT5DM1	15h	33	W		55h			ASA21CR1	95h	90	RW		D5h		
PRT5IC0	16h	33	W		56h			ASA21CR2	96h	92	RW		D6h		
PRT5IC1	17h	34	W		57h			ASA21CR3	97h	93	RW		D7h		
Reserved	18h				58h			ASB22CR0	98h	95	RW		D8h		
	19h				59h			ASB22CR1	99h	97	RW		D9h		
	1Ah				5Ah			ASB22CR2	9Ah	99	RW		DAh		
	1Bh				5Bh			ASB22CR3	9Bh	100	RW		DBh		
	1Ch				5Ch			ASA23CR0	9Ch	88	RW		DCh		
	1Dh				5Dh			ASA23CR1	9Dh	90	RW		DDh		
	1Eh				5Eh			ASA23CR2	9Eh	92	RW		DEh		
	1Fh				5Fh			ASA23CR3	9Fh	93	RW		DFh		
DBA00FN	20h	50	RW	CLK_CR0	60h	76	RW	Reserved	A0h			OSC_CR0	E0h	40	RW
DBA00IN	21h	51	RW	CLK_CR1	61h	77	RW		A1h			OSC_CR1	E1h	40	RW
DBA00OU	22h	53	RW	ABF_CR	62h	106	W		A2h			Reserved	E2h		
Reserved	23h			AMD_CR	63h	107	RW		A3h			VLT_CR	E3h	118	RW
DBA01FN	24h	50	RW	Reserved	64h				A4h			Reserved	E4h		
DBA01IN	25h	51	RW		65h				A5h			Reserved	E5h		
DBA01OU	26h	53	RW		66h				A6h			Reserved	E6h		
Reserved	27h				67h				A7h			Reserved	E7h		
DBA02FN	28h	50	RW		68h				A8h			IMO_TR	E8h	35	W
DBA02IN	29h	51	RW		69h				A9h			ILO_TR	E9h	36	W
DBA02OU	2Ah	53	RW		6Ah				AAh			BDG_TR	EAh	120	W
Reserved	2Bh				6Bh				ABh			ECO_TR	EBh	37	W
DBA03FN	2Ch	50	RW		6Ch				ACh			Reserved	ECh		
DBA03IN	2Dh	51	RW		6Dh				ADh				EDh		
DBA03OU	2Eh	53	RW		6Eh				AEh				EEh		
Reserved	2Fh				6Fh				AFh				EFh		
DCA04FN	30h	50	RW		70h				B0h				F0h		
DCA04IN	31h	51	RW	ACA00CR0	71h	82	RW		B1h				F1h		
DCA04OU	32h	53	RW	ACA00CR1	72h	83	RW		B2h				F2h		
Reserved	33h			ACA00CR2	73h	84	RW		B3h				F3h		
DCA05FN	34h	50	RW	Reserved	74h				B4h				F4h		
DCA05IN	35h	51	RW	ACA01CR0	75h	82	RW		B5h				F5h		
DCA05OU	36h	53	RW	ACA01CR1	76h	83	RW		B6h				F6h		
Reserved	37h			ACA01CR2	77h	84	RW		B7h				F7h		
DCA06FN	38h	50	RW	Reserved	78h				B8h				F8h		
DCA06IN	39h	51	RW	ACA02CR0	79h	82	RW		B9h				F9h		
DCA06OU	3Ah	53	RW	ACA02CR1	7Ah	83	RW		BAh				FAh		
Reserved	3Bh			ACA02CR2	7Bh	84	RW		BBh				FBh		
DCA07FN	3Ch	50	RW	Reserved	7Ch				BCh				FCh		
DCA07IN	3Dh	51	RW	ACA03CR0	7Dh	82	RW		BDh				FDh		
DCA07OU	3Eh	53	RW	ACA03CR1	7Eh	83	RW		BEh				FEh		
Reserved	3Fh			ACA03CR2	7Fh	84	RW		BFh			CPU_SCR	FFh	114	[1]

1. Read/Write access is bit-specific or varies by function. See register.

6.3 Port Global Select Registers

Table 30: Port Global Select 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	GlobSel [7]	GlobSel [6]	GlobSel [5]	GlobSel [4]	GlobSel [3]	GlobSel [2]	GlobSel [1]	GlobSel [0]

Bit [7:0]: **Global Select [7:0]** When written determines whether a pin is connected to the Global Input Bus and Global Output Bus
0 = Not Connected
1 = Connected

Drive Mode xx = Global Select Register 0 = Standard CPU controlled port (Default)
Drive Mode 1 0 (High Z) = Global Select Register 1 = Direct Drive of associated Global Input line
Drive Mode 0 0, 0 1, 1 1 = Global Select Register 1 = Direct Receive from associated Global Output line

Port 0 Global Select Register (PRT0GS, Address = Bank 0, 02h)
Port 1 Global Select Register (PRT1GS, Address = Bank 0, 06h)
Port 2 Global Select Register (PRT2GS, Address = Bank 0, 0Ah)
Port 3 Global Select Register (PRT3GS, Address = Bank 0, 0Eh)
Port 4 Global Select Register (PRT4GS, Address = Bank 0, 12h)
Port 5 Global Select Register (PRT5GS, Address = Bank 0, 16h) Note: If implemented, Port 5 is 4-bits wide

6.3.1 Port Drive Mode 0 Registers

Table 31: Port Drive Mode 0 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	DM0 [7]	DM0 [6]	DM0 [5]	DM0 [4]	DM0 [3]	DM0 [2]	DM0 [1]	DM0 [0]

Bit [7:0]: **DM0 [7:0]** The two Drive Mode bits that control a particular port pin are treated as a pair and are decoded as follows:

Port Data Register Bit 0 = Drive Mode 0 0 = 0 Resistive (Default)
Port Data Register Bit 0 = Drive Mode 0 1 = 0 Strong
Port Data Register Bit 0 = Drive Mode 1 0 = High Z
Port Data Register Bit 0 = Drive Mode 1 1 = 0 Strong
Port Data Register Bit 1 = Drive Mode 0 0 = 1 Strong
Port Data Register Bit 1 = Drive Mode 0 1 = 1 Strong
Port Data Register Bit 1 = Drive Mode 1 0 = High Z
Port Data Register Bit 1 = Drive Mode 1 1 = 1 Resistive

Port 0 Drive Mode 0 Register (PRT0DM0, Address = Bank 1, 00h)
Port 1 Drive Mode 0 Register (PRT1DM0, Address = Bank 1, 04h)
Port 2 Drive Mode 0 Register (PRT2DM0, Address = Bank 1, 08h)
Port 3 Drive Mode 0 Register (PRT3DM0, Address = Bank 1, 0Ch)
Port 4 Drive Mode 0 Register (PRT4DM0, Address = Bank 1, 10h)
Port 5 Drive Mode 0 Register (PRT5DM0, Address = Bank 1, 14h) **Note**: Port 5 is 4-bits wide

Table 36: Internal Low Speed Oscillator Trim Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	FS[1]	FS[1]	FS[1]	FS[1]	FS[1]	FS[1]
Read/ Write	--	W	W	W	W	W	W	W
Bit Name	Reserved	Disable	ILO Trim [5]	ILO Trim [4]	ILO Trim [3]	ILO Trim [2]	ILO Trim [1]	ILO Trim [0]

Bit 7: **Reserved**

Bit 6: **Disable**
0 = Low Speed Oscillator is on
1 = Low Speed Oscillator is off (minimum power state)

Bit [5:0]: **ILO Trim [5:0]** Data value stored will alter the trimmed frequency of the Internal Low Speed Oscillator. (Not recommended for customer alteration)

1. FS = Factory set trim value

Internal Low Speed Oscillator Trim Register (ILO_TR, Address = Bank 1, E9h)

7.1.3 External Crystal Oscillator

The XtalIn and XtalOut pins support connection of a 32.768 kHz watch crystal to drive the 32K clock. To connect to the external crystal, the XtalIn and XtalOut pins' drive modes must be set to High Z. To enable the external crystal oscillator, bit 7 of the Oscillator Control 0 Register (OSC_CR0) must be set (default is off). Note that the Internal Low Speed Oscillator continues to run when this external function is selected. It runs until the oscillator is automatically switched over when the sleep timer reaches terminal count. External feedback capacitors to V_{cc} are required.

The firmware steps involved in switching between the Internal Low Speed Oscillator and External Crystal Oscillator are as follows:

1. At reset, the chip begins operation using the Internal Low Speed Oscillator.
2. User immediately selects a sleep interval of 1 second in the Oscillator Control 0 Register (OSC_CR0), as the oscillator stabilization interval.
3. User selects External Crystal Oscillator by setting bit [7] in Oscillator Control 0 Register (OSC_CR0) to 1.
4. The External Crystal Oscillator becomes the selected 32.768 kHz source at the end of the 1-second interval, created by the Sleep Interrupt logic. The 1-second interval gives the oscillator time to stabilize before it becomes the active source. The Sleep Interrupt need not be enabled for the switch over to occur. The user may want to reset the sleep timer (if this does not interfere with any ongoing real-time clock operation), to guarantee the interval length.
5. The user must wait the 1-second stabilization period prior to engaging the PLL mode to lock the Internal Main Oscillator frequency to the External Crystal Oscillator frequency.

If the proper settings are selected in PSoC Designer, the above steps are automatically done in *boot.asm*.

Note: Transitions between oscillator domains may produce glitches on the 32K clock bus. Functions that require accuracy on the 32K clock should be enabled after the transition in oscillator domains.

The External Crystal Oscillator Trim Register (ECO_TR) sets the adjustment for the External Crystal Oscillator. The value placed in this register at reset is based on factory testing. This register does not adjust the frequency of the External Crystal Oscillator. It is recommended that the user not alter this value.

8.4 Interrupt Masks

Table 44: General Interrupt Mask 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	Sleep	GPIO	Acolumn3	Acolumn2	Acolumn1	Acolumn0	Voltage Monitor

Bit 7: **Reserved**

Bit 6: **Sleep Interrupt Enable Bit (see 11.4)**
0 = Disabled
1 = Enabled

Bit 5: **GPIO Interrupt Enable Bit (see 8.6)**
0 = Disabled
1 = Enabled

Bit [4]: **Acolumn 3 Interrupt Enable Bit (see 10.0)**
0 = Disabled
1 = Enabled

Bit [3]: **Acolumn 2 Interrupt Enable Bit (see 10.0)**
0 = Disabled
1 = Enabled

Bit [2]: **Acolumn 1 Interrupt Enable Bit (see 10.0)**
0 = Disabled
1 = Enabled

Bit [1]: **Acolumn 0 Interrupt Enable Bit (see 10.0)**
0 = Disabled
1 = Enabled

Bit 0: **Voltage Monitor Interrupt Enable Bit (see 11.5)**
0 = Disabled
1 = Enabled

General Interrupt Mask Register (INT_MSK0, Address = Bank 0, E0h)

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)

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:

9.3.3 Digital Basic Type A/Communications Type A Block xx Control Register 0 When Used as Timer, Counter, CRC, and Deadband

Note that the data in this register, as well as the following three registers, are a mapping of the functions of the variables selected in the associated Digital Basic Type A/ Communications Type A Block xx Control Register 0.

Table 55: Digital Basic Type A/Communications Type A Block xx Control Register 0...

Bit #	7	6	5	4	3	2	1	0
POR	--	--	--	--	--	--	--	0
Read/Write	--	--	--	--	--	--	--	RW
Bit Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Enable

Bit 7: **Reserved**
Bit 6: **Reserved**
Bit 5: **Reserved**
Bit 4: **Reserved**
Bit 3: **Reserved**
Bit 2: **Reserved**
Bit 1: **Reserved**

Bit 0: **Enable**
0 = Function Disabled
1 = Function Enabled

Digital Basic Type A Block 00 Control Register 0 (DBA00CR0, Address = Bank 0, 23h)
Digital Basic Type A Block 01 Control Register 0 (DBA01CR0, Address = Bank 0, 27h)
Digital Basic Type A Block 02 Control Register 0 (DBA02CR0, Address = Bank 0, 2Bh)
Digital Basic Type A Block 03 Control Register 0 (DBA03CR0, Address = Bank 0, 2Fh)
Digital Communications Type A Block 04 Control Register 0 (DCA04CR0, Address = Bank 0, 33h)
Digital Communications Type A Block 05 Control Register 0 (DCA05CR0, Address = Bank 0, 37h)
Digital Communications Type A Block 06 Control Register 0 (DCA06CR0, Address = Bank 0, 3Bh)
Digital Communications Type A Block 07 Control Register 0 (DCA07CR0, Address = Bank 0, 3Fh)

9.4 Global Inputs and Outputs

Global Inputs and Outputs provide additional capability to route clock and data signals to the digital PSoC blocks. Digital PSoC blocks are connected to the global input and output lines by configuring the PSoC block Input and Output registers (DBA00IN-DCA07IN, DBA00OU-DCA07OU). These global input and output lines form an 8-bit global input bus and an 8-bit global output bus. Four Digital PSoC blocks have access to the upper half of these buses, while the other four access the lower half, per the configuration register. These global input/output buses may be connected to the I/O pins on a per-pin basis using the pin configuration registers. This allows digital PSoC blocks to route their inputs and outputs to pins using the global I/O buses.

9.4.1 Input Assignments

The PSoC block Input Register defines the selection of Global Inputs to digital PSoC blocks. Only 4 of the Global Inputs bus lines are available as selections to a given digital PSoC block as shown in the table below. Once the Global Input has been selected using the PSoC block Input Register selection bits, a GPIO pin must be configured to drive the selected Global Input. This configuration may be set in the Port Global Select Register. The GPIO direction must also be set to input mode by configuring the Port Drive Mode Registers to select High Z.

Table 59: Global Input Assignments

Global Input [7]	Global Input [6]	Global Input [5]	Global Input [4]	Global Input [3]	Global Input [2]	Global Input [1]	Global Input [0]
Port x[7]	Port x[6]	Port x[5]	Port x[4]	Port x[3]	Port x[2]	Port x[1]	Port x[0]
PSoC Block 04 PSoC Block 05 PSoC Block 06 PSoC Block 07	PSoC Block 04 PSoC Block 05 PSoC Block 06 PSoC Block 07	PSoC Block 04 PSoC Block 05 PSoC Block 06 PSoC Block 07	PSoC Block 04 PSoC Block 05 PSoC Block 06 PSoC Block 07	PSoC Block 00 PSoC Block 01 PSoC Block 02 PSoC Block 03	PSoC Block 00 PSoC Block 01 PSoC Block 02 PSoC Block 03	PSoC Block 00 PSoC Block 01 PSoC Block 02 PSoC Block 03	PSoC Block 00 PSoC Block 01 PSoC Block 02 PSoC Block 03

9.4.2 Output Assignments

The PSoC block Output Register defines the selection of the Global Output bus line to be driven by the digital PSoC blocks. Only 4 of the Global Output bus lines are available as selections to a given digital PSoC block as shown in the table below. The Global Output bus has two functions. Since Global Outputs are also selectable as inputs to digital PSoC blocks, signals can be routed between blocks using this bus. In addition, Global Outputs may drive out to GPIO pins. In this case, once the Global Output has been selected using the PSoC block Output Register selection bits, a GPIO pin must be configured to select the Global Output to drive to the pin. This configuration may be set in the Port Global Select Register. The GPIO direction must also be set to output mode (which is the default) by configuring the Port Drive Mode Registers to one of the available driving strengths.

Table 60: Global Output Assignments

Global Output [7]	Global Output [6]	Global Output [5]	Global Output [4]	Global Output [3]	Global Output [2]	Global Output [1]	Global Output [0]
Port x[7]	Port x[6]	Port x[5]	Port x[4]	Port x[3]	Port x[2]	Port x[1]	Port x[0]
PSoC Block 04 PSoC Block 05 PSoC Block 06 PSoC Block 07	PSoC Block 04 PSoC Block 05 PSoC Block 06 PSoC Block 07	PSoC Block 04 PSoC Block 05 PSoC Block 06 PSoC Block 07	PSoC Block 04 PSoC Block 05 PSoC Block 06 PSoC Block 07	PSoC Block 00 PSoC Block 01 PSoC Block 02 PSoC Block 03	PSoC Block 00 PSoC Block 01 PSoC Block 02 PSoC Block 03	PSoC Block 00 PSoC Block 01 PSoC Block 02 PSoC Block 03	PSoC Block 00 PSoC Block 01 PSoC Block 02 PSoC Block 03

9.5 Available Programmed Digital Functionality

9.5.1 Timer with Optional Capture

9.5.1.1 Summary

The timer function continuously measures the amount of time in “ticks” between two events, and provides a rate generator. A down counter lies at the heart of the timer functions. Rate generators divide their clock source by an integer value. Hardware or software generated events

only be input from GPIO input pins (Global Input Bus). There is no way to enable the SS_internally. In SPI modes 2 & 3, where SS is not required between each byte, the external pin may be grounded.

Important: The AUX Out Enable bit (bit 5) of the Output Register (DCA04OU-DCA07OU) must be set to 0 to disable it.

9.5.9.4 Outputs

The function output is the MISO (master-in, slave-out) signal, which may be driven on the Global Output bus and is selected by Output Register (DCA04OU-DCA07OU).

9.5.9.5 Interrupts

When enabled, the function generates an interrupt on RX Reg Full status (Data Register 2 full). If Mode[1] of the Function Register is set, the interrupt will be generated on SPI Complete status.

9.5.9.6 Usage Notes

1. Reading the Status

   Reading Control Register 0, which contains the status bits, automatically resets the status bits to 0 with the exception of TX Reg Empty, which is cleared when a byte is written to the TX Data Register (Data Register 1), and the RX Reg Full, which is cleared when a byte is read from the RX Data Register (Data Register 2).

2. Multi-Slave Environment

   The SS_ signal does not have any affect on the output from the slave. The output of the slave at the end of a reception/transmission is always the first bit sent (the MSB, unless LSBF option is selected, then it's the LSB). To implement a multi-slave environment, a GPIO interrupt may be configured on the SS_ input, and the Slave output strength may be toggled between driving and High Z in firmware.

3. Using Interrupts

   RX Reg Full status or SPI Complete status generates an interrupt. Executing the interrupt routine does not automatically clear status. If SPI Complete is selected as the interrupt source, Control Register 0 (status) must be read in the interrupt routine to clear the status. If RX Reg Full status is selected, a byte must be read from the RX Data Register (Data Register 2) to clear the status. If the interrupting status is not cleared further interrupts will be suppressed.

4. Synchronization of CPU Interaction

   Because the SPI Slave is clocked asynchronously by the master SCLK, transfer of data between the TX Register to shifter and shifter to RX Register occurs asynchronously.

   Either polling or interrupts can be used to detect that a byte has been received and is ready to read. However, on the TX side, the user is responsible for implementing a protocol that ensures there is enough set-up time from the TX Data Register write to the first clock (mode 2, 3) or SS_ (mode 0, 1) from the master.

divided into distinct bit fields. Some bit fields set the PSoC block's resistor ratios or capacitor values. Others configure switches and multiplexers that form connections between internal block nodes. Additionally, a block may be connected via local interconnection resources to neighboring analog PSoC blocks, reference voltage sources, input multiplexers and output busses. Specific advantages and applications of each type are treated separately below.

10.6.1 Local Interconnect

Analog continuous-time PSoC blocks occupy the top row, (row 0) of the analog array. Designated ACA for analog continuous-time subtype "A," each connects to its neighbors by means of three multiplexers. (Note that unlike the switched capacitor blocks, the continuous time blocks in the current family of parts only have one subtype.) The three are the non-inverting input multiplexer, "PMux," the inverting input multiplexer, "NMux," and the "RBotMux" which controls the node at the bottom of the resistor string. The bit fields, which control these multiplexers, are named PMux, NMux, and RBotMux, respectively. The following diagrams show how each multiplexer connects its ACA block connect to its neighbors. Each arrow points from an input source, either a PSoC block, bus or reference voltage to the block where it is used. Each arrow is labeled with the value to which the bit-field must be set to select that input source.

10.6.1.1 NMux

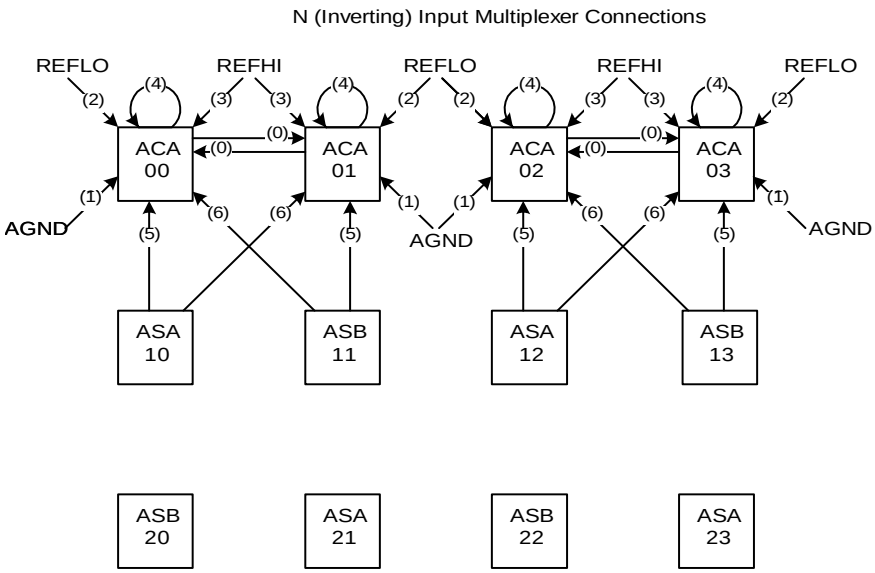

Figure 18: NMux Connections

10.6.1.2 PMux

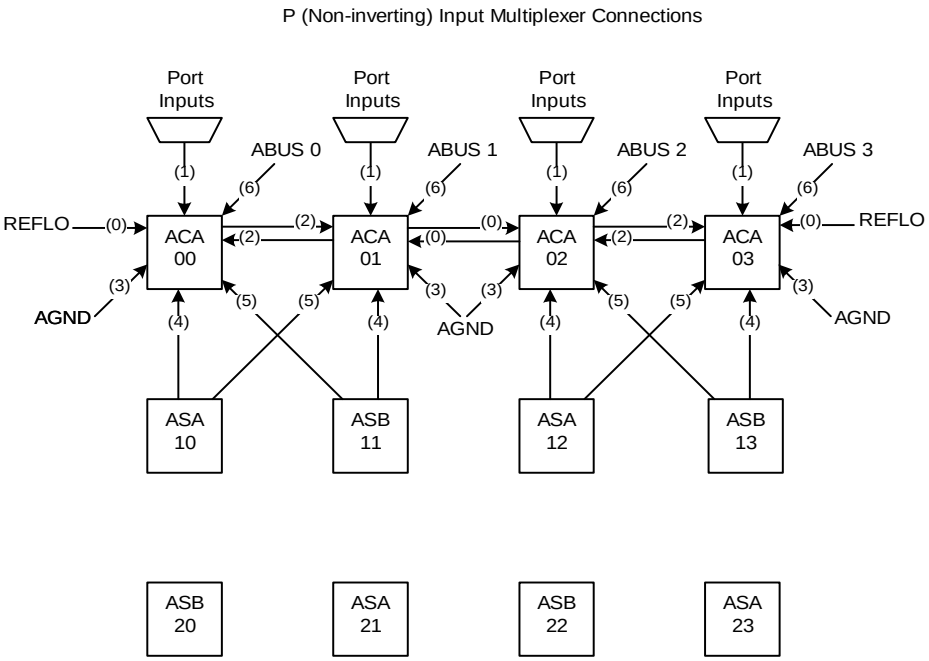

Figure 19: PMux Connections

10.6.1.3 RBotMux

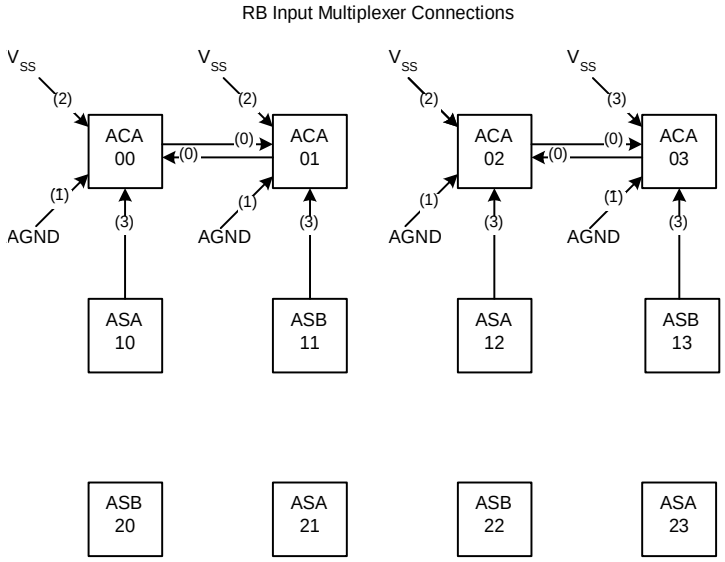

Figure 20: RBotMux Connections

10.9 Analog Switch Cap Type B PSoC Blocks

10.9.1 Introduction

The Analog Switch Cap Type B PSoC blocks are built around an operational amplifier. There are several analog muxes that are controlled by register-bit settings in the control registers that determine the signal topology inside the block. There are also four arrays of unit value capacitors that are located in the feedback path for the op-amp, and are switched by two phase clocks, PHI1 and PHI2. These four capacitor arrays are labeled A Cap Array, B Cap Array, C Cap Array, and F Cap Array. There is also an analog comparator connected to the output OUT, which converts analog comparisons into digital signals.

There are three discrete outputs from this block. These outputs are:

1. The analog output bus (ABUS), which is an analog bus resource that is shared by all of the analog blocks in the analog column for that block.

2. The comparator bus (CBUS), which is a digital bus that is a resource that is shared by all of the analog blocks in a column for that block.

3. The output bus (OUT), which is an analog bus resource that is shared by all of the analog blocks in a column and connects to one of the analog output buffers, to send a signal externally to the device.

The SCB block also supports Delta-Sigma, Successive Approximation and Incremental A/D Conversion, Capacitor DACs, and SC filters. It has two input arrays of switched capacitors, and a Non-Switched capacitor feedback array from the output. When preceded by an SC Block A Integrator, the combination can be used to provide a full Switched Capacitor Biquad.

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)

10.10 Analog Comparator Bus

Each analog column has a dedicated comparator bus associated with it. Every analog PSoC block has a comparator output that can drive out on this bus, but the comparator output from only one analog block in a column can be actively driving the comparator bus for that column at any one time. The output on the comparator bus can drive into the digital blocks, and is also available to be read in the Analog Comparator Control Register (CMP_CR, Address = Bank 0,64H).

The comparator bus is latched before it is available to either drive the digital blocks, or be read in the Analog Comparator Control Register. The latch for each comparator bus is transparent (the output tracks the input) during the high period of PHI2. During the low period of PHI2 the latch retains the value on the comparator bus during the high to low transition of PHI2.

The output from the analog block that is actively driving the bus may also be latched internal to the analog block itself.

In the Continuous Time analog blocks, the CPhase and CLatch bits inside the Analog Continuous Time Type A Block xx Control Register 2 determine whether the output signal on the comparator bus is latched inside the block, and if it is, which clock phase it is latched on.

In the Switched Capacitor analog blocks, the output on the comparator bus is always latched. The ClockPhase bit in the Analog SwitchCap Type A Block xx Control Register 0 or the Analog SwitchCap Type B Block xx Control Register 0 determines the phase on which this data is latched and available.

Table 77: Analog Comparator Control 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	RW	RW	RW	RW
Bit Name	COMP 3	COMP 2	COMP 1	COMP 0	AINT 3	AINT 2	AINT 1	AINT 0

Bit 7: COMP 3 COMP 3 bit [0] indicates the state of the analog comparator bus for the Analog Column x
Bit 6: COMP 2 COMP 2 bit [0] indicates the state of the analog comparator bus for the Analog Column x
Bit 5: COMP 1 COMP 1 bit [0] indicates the state of the analog comparator bus for the Analog Column x
Bit 4: COMP 0 COMP 0 bit [0] indicates the state of the analog comparator bus for the Analog Column x

Bit 3: AINT 3 AINT 3 bit [0] or [1] (as defined below) selects the Analog Interrupt Source for the Analog Column x
Bit 2: AINT 2 AINT 2 bit [0] or [1] (as defined below) selects the Analog Interrupt Source for the Analog Column x
Bit 1: AINT 1 AINT 1 bit [0] or [1] (as defined below) selects the Analog Interrupt Source for the Analog Column x
Bit 0: AINT 0 AINT 0 bit [0] or [1] (as defined below) selects the Analog Interrupt Source for the Analog Column x

0 = Comparator bus
1 = PHI2 (Falling edge of PHI2 causes an interrupt)

Analog Comparator Control Register (CMP_CR, Address = Bank 0, 64h)

10.11 Analog Synchronization

For high precision analog operation, it may be necessary to precisely time when updated register values are available to the analog PSOC blocks. The optimum time to update values in Switch Cap registers is at the beginning of the PHI1 active period. The SYNCEN bit in the Analog Synchronization Control Register is designed to address this. (The AINT bits of the Analog Comparator Register (CMP_CR) are another way to address it with interrupts.) When the SYNCEN bit is set, a subsequent write instruction to any register in a Switch Cap block will cause the CPU to stall until the rising edge of PHI1. This mode is in effect until the SYNCEN bit is cleared.

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)