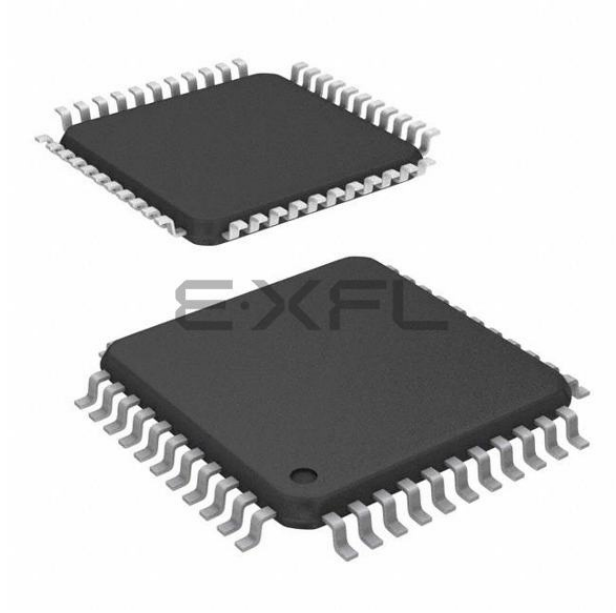

Welcome to E-XFL.COM

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	40
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	44-LQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy8c26643-24axi

Getting Started in the PSoC World!

The award winning PSoC Designer software and PSoC silicon are an integrated unit. The quickest path to understanding the PSoC silicon is through the PSoC Designer software GUI. This data sheet is useful for understanding the details of the PSOC integrated circuit, but is not a good starting point for a new PSoC developer seeking to get a general overview of this new technology.

PSoC developers are NOT required to build their own ADCs, DACs, and other peripherals. Embedded in the PSoC Designer software are the individual data sheets, performance graphs, and PSoC User Modules (graphically selected code packets) for the peripherals, such as the incremental ADCs, DACs, LCD controllers, op amps, low-pass filters, etc. **With simple GUI-based selection, placement, and connection, the basic architecture of a design may be developed within PSoC Designer software without ever writing a single line of code**.

Development Kits are available from the following distributors: Digi-Key, Avnet, Arrow, and Future. The Cypress Online Store also contains development kits, C compilers, and all accessories for PSoC development. Go to the Cypress Online Store web site at http://www.cypress.com, click the Online Store shopping cart icon at the bottom of the web page, and click *PSoC (Programmable System-on-Chip)* to view a current list of available items.

Free PSoC technical training is available for beginners and is taught by a marketing or application engineer over the phone. PSoC training classes cover designing, debugging, advanced analog, as well as application-specific classes covering topics such as PSoC and the LIN bus. Go to http://www.cypress.com, click on Design Support located on the left side of the web page, and select Technical Training for more details.

Certified PSoC Consultants offer everything from technical assistance to completed PSoC designs. To contact or become a PSoC Consultant go to http://www.cypress.com, click on Design Support located on the left side of the web page, and select CYPros Consultants.

PSoC application engineers take pride in fast and accurate response. They can be reached with a 4-hour guaranteed response at http://www.cypress.com/support/login.cfm.

Cypress Semiconductor
2700 162nd Street SW, Building D
Lynnwood, WA 98037
Phone: 425.787.4400
Fax: 425.787.4641
Application Support Hotline: 425.787.4814

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

4.2 Register Bank 0 Map

Table 26: Bank 0

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
PRT0DR	00h	31	RW	Reserved	40h			ASA10CR0	80h	88	RW	Reserved	C0h		
PRT0IE	01h	31	W		41h			ASA10CR1	81h	90	RW		C1h		
PRT0GS	02h	32	W		42h			ASA10CR2	82h	92	RW		C2h		
Reserved	03h				43h			ASA10CR3	83h	93	RW		C3h		
PRT1DR	04h	31	RW		44h			ASB11CR0	84h	95	RW		C4h		
PRT1IE	05h	31	W		45h			ASB11CR1	85h	97	RW		C5h		
PRT1GS	06h	32	W		46h			ASB11CR2	86h	99	RW		C6h		
Reserved	07h				47h			ASB11CR3	87h	100	RW		C7h		
PRT2DR	08h	31	RW		48h			ASA12CR0	88h	88	RW		C8h		
PRT2IE	09h	31	W		49h			ASA12CR1	89h	90	RW		C9h		
PRT2GS	0Ah	32	W		4Ah			ASA12CR2	8Ah	92	RW		CAh		
Reserved	0Bh				4Bh			ASA12CR3	8Bh	93	RW		CBh		
PRT3DR	0Ch	31	RW		4Ch			ASB13CR0	8Ch	95	RW		CCh		
PRT3IE	0Dh	31	W		4Dh			ASB13CR1	8Dh	97	RW		CDh		
PRT3GS	0Eh	32	W		4Eh			ASB13CR2	8Eh	99	RW		CEh		
Reserved	0Fh				4Fh			ASB13CR3	8Fh	100	RW		CFh		
PRT4DR	10h	31	RW		50h			ASB20CR0	90h	95	RW		D0h		
PRT4IE	11h	31	W		51h			ASB20CR1	91h	97	RW		D1h		
PRT4GS	12h	32	W		52h			ASB20CR2	92h	99	RW		D2h		
Reserved	13h				53h			ASB20CR3	93h	100	RW		D3h		
PRT5DR	14h	31	RW		54h			ASA21CR0	94h	88	RW		D4h		
PRT5IE	15h	31	W		55h			ASA21CR1	95h	90	RW		D5h		
PRT5GS	16h	32	W		56h			ASA21CR2	96h	92	RW		D6h		
Reserved	17h				57h			ASA21CR3	97h	93	RW		D7h		
	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		
DBA00DR0	20h	54	1	AMX_IN	60h	104	RW	Reserved	A0h			INT_MSK0	E0h	45	RW
DBA00DR1	21h	54	1	Reserved	61h				A1h			INT_MSK1	E1h	46	RW
DBA00DR2	22h	54	1		62h				A2h			INT_VC	E2h	46	RW
DBA00CR0	23h	55	1	ARF_CR	63h	73	RW		A3h			RES_WDT	E3h	116	RW
DBA01DR0	24h	54	1	CMP_CR	64h	101	1		A4h			DEC_DH/DEC_CL	E4h	113	RW
DBA01DR1	25h	54	1	ASY_CR	65h	102	1		A5h			DEC_DL	E5h	113	R
DBA01DR2	26h	54	1	Reserved	66h				A6h			DEC_CR	E6h	113	RW
DBA01CR0	27h	55	1		67h				A7h			Reserved	E7h		
DBA02DR0	28h	54	1		68h				A8h			MUL_X	E8h	110	W
DBA02DR1	29h	54	1		69h				A9h			MUL_Y	E9h	110	W
DBA02DR2	2Ah	54	1		6Ah				AAh			MUL_DH	EAh	111	R
DBA02CR0	2Bh	55	1		6Bh				ABh			MUL_DL	EBh	111	R
DBA03DR0	2Ch	54	1		6Ch				ACh			ACC_DR1/MAC_X	ECh	111	RW
DBA03DR1	2Dh	54	1		6Dh				ADh			ACC_DR0/MAC_Y	EDh	111	RW
DBA03DR2	2Eh	54	1		6Eh				AEh			ACC_DR3/MAC_CL0	EEh	112	RW
DBA03CR0	2Fh	55	1		6Fh				AFh			ACC_DR2/MAC_CL1	EFh	112	RW
DCA04DR0	30h	54	1		70h				B0h			Reserved	F0h		
DCA04DR1	31h	54	1	ACA00CR0	71h	82	RW		B1h				F1h		
DCA04DR2	32h	54	1	ACA00CR1	72h	83	RW		B2h				F2h		
DCA04CR0	33h	55	1	ACA00CR2	73h	84	RW		B3h				F3h		
DCA05DR0	34h	54	1	Reserved	74h				B4h				F4h		
DCA05DR1	35h	54	1	ACA01CR0	75h	82	RW		B5h				F5h		
DCA05DR2	36h	54	1	ACA01CR1	76h	83	RW		B6h				F6h		
DCA05CR0	37h	55	1	ACA01CR2	77h	84	RW		B7h				F7h		
DCA06DR0	38h	54	1	Reserved	78h				B8h				F8h		
DCA06DR1	39h	54	1	ACA02CR0	79h	82	RW		B9h				F9h		
DCA06DR2	3Ah	54	1	ACA02CR1	7Ah	83	RW		BAh				FAh		
DCA06CR0	3Bh	55	1	ACA02CR2	7Bh	84	RW		BBh				FBh		
DCA07DR0	3Ch	54	1	Reserved	7Ch				BCh				FCh		
DCA07DR1	3Dh	54	1	ACA03CR0	7Dh	82	RW		BDh				FDh		
DCA07DR2	3Eh	54	1	ACA03CR1	7Eh	83	RW		BEh				FEh		
DCA07CR0	3Fh	55	1	ACA03CR2	7Fh	84	RW		BFh			CPU_SCR	FFh	114	1

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.

5.0 I/O Ports

5.1 Introduction

Up to five 8-bit-wide I/O ports (P0-P4) and one 4-bit wide I/O port (P5) are implemented. The number of general purpose I/Os implemented and connected to pins depends on the individual part chosen. All port bits are independently programmable and have the following capabilities:

- General-purpose digital input readable by the CPU.
- General-purpose digital output writable by the CPU.
- Independent control of data direction for each port bit.
- Independent access for each port bit to Global Input and Global Output busses.
- Interrupt programmable to assert on rising edge, falling edge, or change from last pin state read.
- Output drive strength programmable in logic 0 and 1 states as strong, resistive (pull-up or pull-down), or high impedance.
- A slew rate controlled output mode is available.
- In high impedence, the digital input can be disabled to lower power consumption.

Port 1, Pin 0 is used in conjunction with device Test Mode and does not behave the same as other I/O ports immediately after reset. A device reset with Power On Reset (POR) will drive P1[0] high for 8 ms immediately after POR is released because there is a CPU hold-off time of approximately 64 ms before code execution begins. It will then drive P1[0] low for 8 ms. This can impact external circuits connected to Port 1, Pin 0.

In System Sleep State, GPIO Pins P2[4] and P2[6] should be held to a logic low or a false Low Voltage Detect interrupt may be triggered. The cause is in the System Sleep State, the internal Bandgap reference generator is turned off and the reference voltage is maintained on a capacitor.

The circumstances are that during sleep, the reference voltage on the capacitor is refreshed periodically at the sleep system duty cycle. Between refresh cycles, this voltage may leak slightly to either the positive supply or ground. If pins P2[4] or P2[6] are in a high state, the leakage to the positive supply is accelerated (especially at high temperature). Since the reference voltage is compared to the supply to detect a low voltage condition, this accelerated leakage to the positive supply voltage will cause that voltage to appear lower than it actually is, leading to the generation of a false Low Voltage Detect interrupt.

Port 0 and Port 2 have additional analog input and/or analog output capability. The specific routing and multiplexing of analog signals is shown in the following diagram:

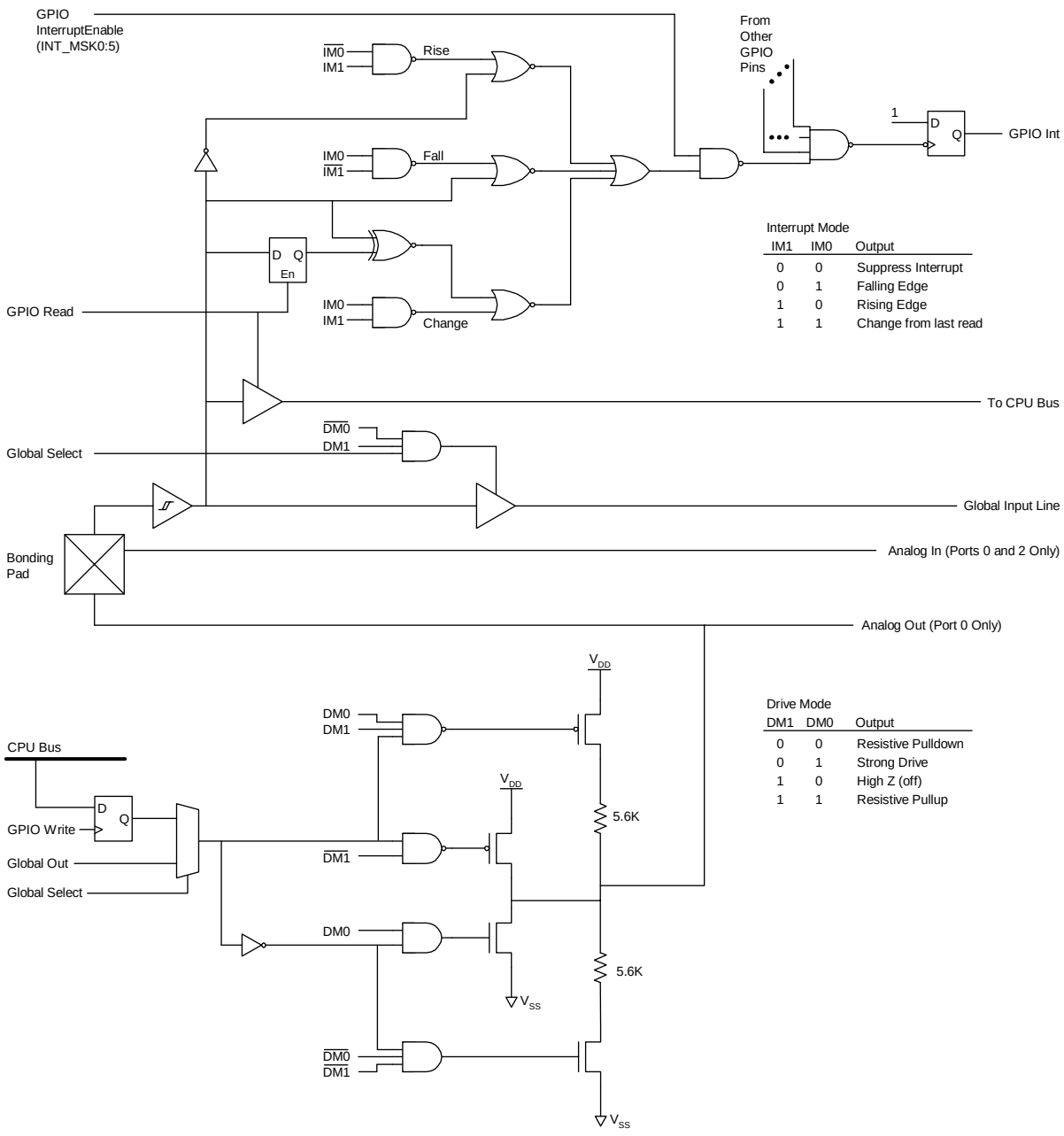

Figure 7: General Purpose I/O Pins

6.3.2 Port Drive Mode 1 Registers

Table 32: Port Drive Mode 1 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	DM1 [7]	DM1 [6]	DM1 [5]	DM1 [4]	DM1 [3]	DM1 [2]	DM1 [1]	DM1 [0]
Bit [7:0]: **DM1 [7:0]** See truth table for Port Drive Mode 0 Registers, above								

Port 0 Drive Mode 1 Register (PRT0DM1, Address = Bank 1, 01h)
Port 1 Drive Mode 1 Register (PRT1DM1, Address = Bank 1, 05h)
Port 2 Drive Mode 1 Register (PRT2DM1, Address = Bank 1, 09h)
Port 3 Drive Mode 1 Register (PRT3DM1, Address = Bank 1, 0Dh)
Port 4 Drive Mode 1 Register (PRT4DM1, Address = Bank 1, 11h)
Port 5 Drive Mode 1 Register (PRT5DM1, Address = Bank 1, 15h) **Note**: Port 5 is 4-bits wide

6.3.3 Port Interrupt Control 0 Registers

Table 33: Port Interrupt Control 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	IC0 [7]	IC0 [6]	IC0 [5]	IC0 [4]	IC0 [3]	IC0 [2]	IC0 [1]	IC0 [0]
Bit [7:0]: **IC0 [7:0]** The two Interrupt Control bits that control a particular port pin are treated as a pair and are decoded as follows: IC1 [x], IC0 [x] = 0 0 = Disabled (Default) IC1 [x], IC0 [x] = 0 1 = Falling Edge (-) IC1 [x], IC0 [x] = 1 0 = Rising Edge (+) IC1 [x], IC0 [x] = 1 1 = Change from Last Direct Read								

Port 0 Interrupt Control 0 Register (PRT0IC0, Address = Bank 1, 02h)
Port 1 Interrupt Control 0 Register (PRT1IC0, Address = Bank 1, 06h)
Port 2 Interrupt Control 0 Register (PRT2IC0, Address = Bank 1, 0Ah)
Port 3 Interrupt Control 0 Register (PRT3IC0, Address = Bank 1, 0Eh)
Port 4 Interrupt Control 0 Register (PRT4IC0, Address = Bank 1, 12h)
Port 5 Interrupt Control 0 Register (PRT5IC0, Address = Bank 1, 16h) **Note**: Port 5 is 4-bits wide

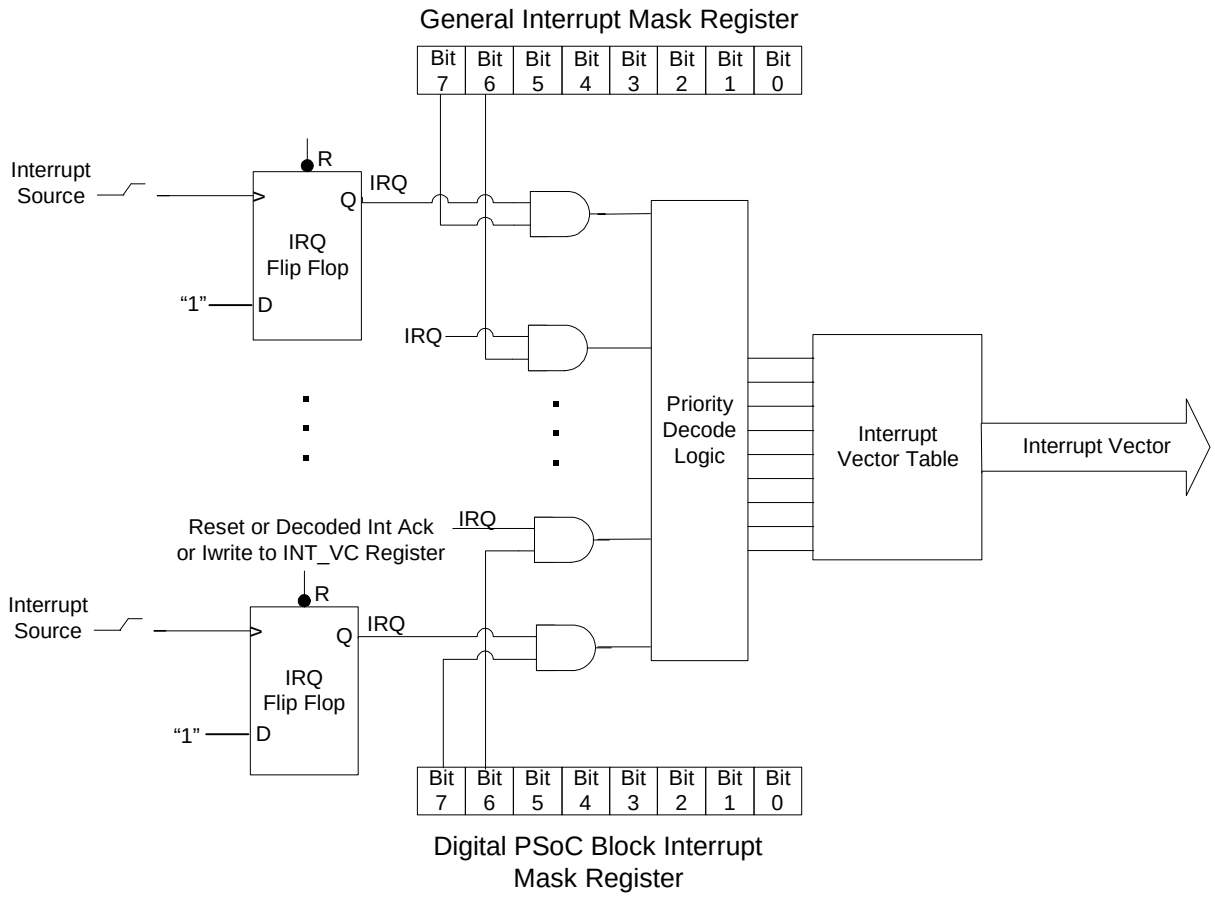

Figure 10: Interrupts Overview

9.3.6 Digital Communications Type A Block xx Control Register 0 When Used as SPI Transceiver

Table 58: Digital Communications Type A Block xx Control Register 0...

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/ Write	RW	R	R	R	R	RW	RW	RW
Bit Name	LSB First	Overrun	SPI Complete	TX Reg Empty	RX Reg Full	Clock Phase	Clock Polarity	Enable

Bit 7: **LSB First**
0 = MSB First
1 = LSB First

Bit 6: **Overrun**
0 = Indicates that no overrun has taken place
1 = Indicates the RX Data register was overwritten with a new byte before the previous one had been read
Reset when this register is read

Bit 5: **SPI Complete**
0 = Indicates the byte is in process of shifting out
1 = Indicates the byte has been shifted out (reset when register is read)
Optional interrupt source for both SPI Master and SPI Slave. Reset when this register is read

Bit 4: **TX Reg Empty**
0 = Indicates the TX Data register is not available to accept another byte
1 = Indicates the TX Data register is available to accept another byte
Default interrupt source for SPI Master. Reset when the TX Data Register (Data Register 1) is written.

Bit 3: **RX Reg Full**
0 = Indicates the RX Data register is empty
1 = Indicates a byte has been loaded into the RX Data register
Default interrupt source for SPI Slave. Reset when the RX Data Register (Data Register 2) is read

Bit 2: **Clock Phase**
0 = Data changes on leading edge and is latched on trailing edge
1 = Data is latched on leading edge and is changed on trailing edge

Bit 1: **Clock Polarity**
0 = Non-inverted (clock idle state is low)
1 = Inverted (clock idle state is high)

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

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)

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

3. Using CPU Interrupts

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

9.5.8 SPI Master - Serial Peripheral Interface (SPIM)

9.5.8.1 Summary

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

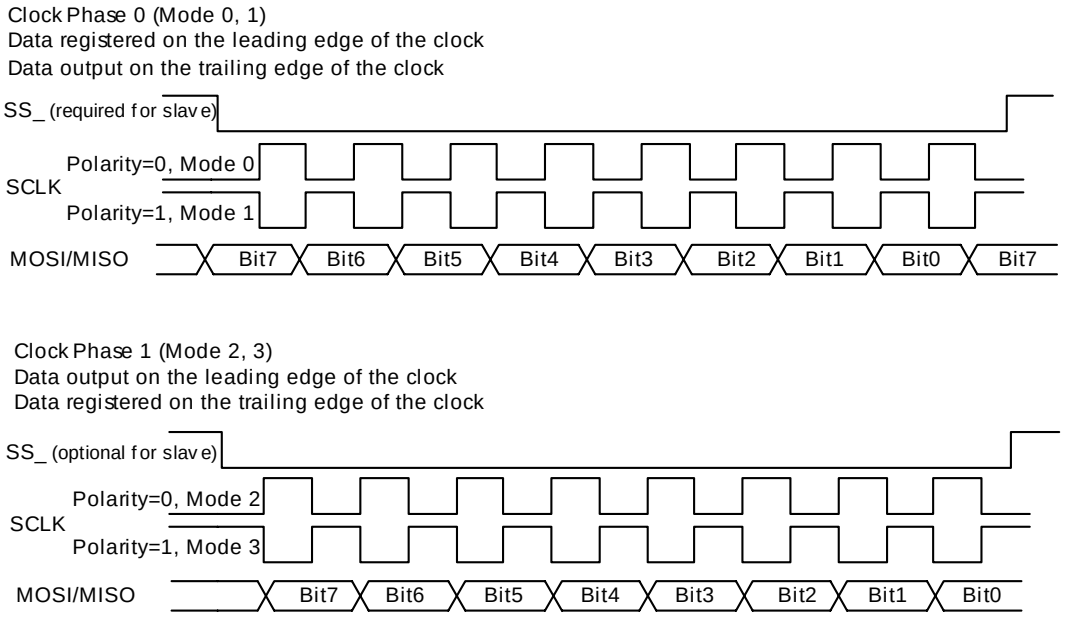

Figure 15: SPI Waveforms

9.5.8.2 Registers

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

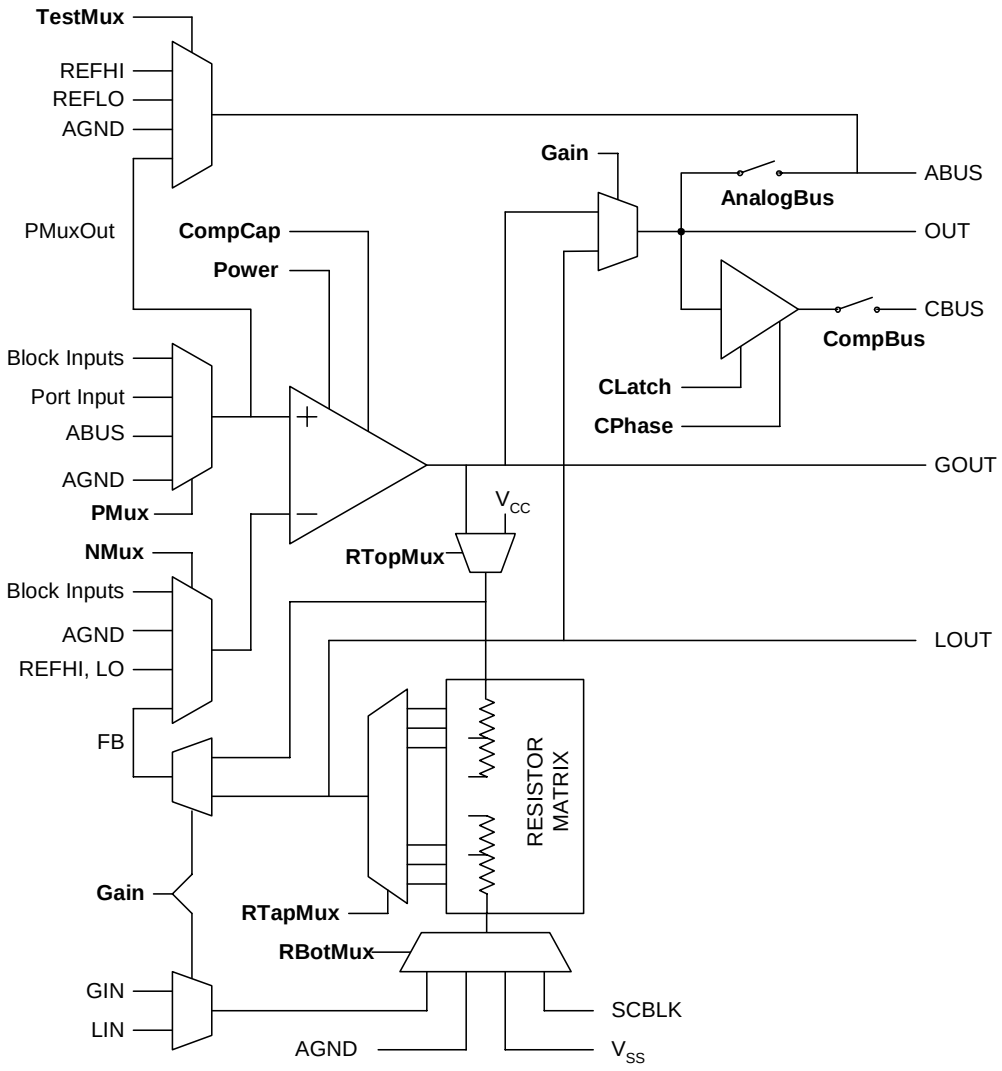

Figure 21: Analog Continuous Time PSoC Blocks

10.7.2 Registers

10.7.2.1 Analog Continuous Time Block xx Control 0 Register

The RTopMux and RBotMux bits control the connection of the two ends of the resistor string. The RTopMux bit controls the top end of the resistor string, which can either be connected to Vcc or to the op-amp output. The RBotMux bits control the connection of the bottom end of the resistor string.

The RTapMux bits control the center tap of the resistor string. Note that only relative weighting of units is given in the table.

The Gain and Loss columns correspond to the gain or loss obtained if the RTopMux and Gain bits are set so that the overall amplifier provides gain or loss.

The Gain bit controls whether the resistor string is connected around the op-amp as for gain (center tap to

10.7.2.2 Analog Continuous Time Block xx Control 1 Register

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

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

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

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

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

Table 67: Analog Continuous Time Block xx Control 1 Register

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

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

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

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

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

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

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

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

10.7.2.3 Analog Continuous Time Type A Block xx Control 2 Register

CPhase controls which internal clock phase the comparator data is latched on.

CLatch controls whether the latch is active or if it is always transparent.

CompCap controls whether the compensation capacitor is switched in or not in the op-amp. By not switching in the compensation capacitance, a much faster response can be obtained if the amplifier is being used as a comparator.

TestMux – selects block bypass mode for testing and characterization purposes.

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

Table 68: Analog Continuous Time Type A 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	CPhase	CLatch	CompCap	TestMux[2]	TestMux[1]	TestMux[0]	Power[1]	Power[0]

Bit 7: **CPhase**
0 = Comparator Control latch transparent on PHI1
1 = Comparator Control latch transparent on PHI2

Bit 6: **CLatch**
0 = Comparator Control latch is always transparent
1 = Comparator Control latch is active

Bit 5: **CompCap**
0 = Comparator Mode
1 = Op-amp Mode

Bit [4:2]: **TestMux [2:0]** Select block bypass mode for testing and characterization purposes

	ACA00	ACA01	ACA02	ACA03
1 0 0 = Positive Input to…	ABUS0	ABUS1	ABUS2	ABUS3
1 0 1 = AGND to…	ABUS0	ABUS1	ABUS2	ABUS3
1 1 0 = REFLO to…	ABUS0	ABUS1	ABUS2	ABUS3
1 1 1 = REFHI to…	ABUS0	ABUS1	ABUS2	ABUS3

0 x x = All Paths Off

Bit [1:0]: **Power [1:0]** Encoding for selecting 1 of 4 power levels
0 0 = Off
0 1 = Low (60 µA)
1 0 = Med (150 µA)
1 1 = High (500 µA)

Analog Continuous Time Block 00 Control 2 Register (ACA00CR2, Address = Bank 0/1, 73h)
Analog Continuous Time Block 01 Control 2 Register (ACA01CR2, Address = Bank 0/1, 77h)
Analog Continuous Time Block 02 Control 2 Register (ACA02CR2, Address = Bank 0/1, 7Bh)
Analog Continuous Time Block 03 Control 2 Register (ACA03CR2, Address = Bank 0/1, 7Fh)

10.8 Analog Switch Cap Type A PSoC Blocks

10.8.1 Introduction

The Analog Switch Cap Type A 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.

SC Integrator Block A supports Delta-Sigma, Successive Approximation and Incremental A/D Conversion, Capacitor DACs, and SC filters. It has three input arrays of binarily-weighted switched capacitors, allowing user programmability of the capacitor weights. This provides summing capability of two (CDAC) scaled inputs, and a non-switched capacitor input. Since the input of SC Block A has this additional switched capacitor, it is configured for the input stage of such a switched capacitor biquad filter. When followed by an SC Block B Integrator, this combination of blocks can be used to provide a full Switched Capacitor Biquad.

10.8.2.3 ACMux

The ACMux, as shown in Analog Switch Cap Type A Block xx Control 1 Register, controls the input muxing for both the A and C capacitor branches. The high order bit, ACMux[2], selects one of two inputs for the C branch. However, when the bit is high, it also overrides the two low order bits, forcing the A and C branches to the same source. The resulting condition is used to construct low pass biquad filters. See the individual AMux and CMux diagrams.

10.8.2.4 BMuxSCA/SCB

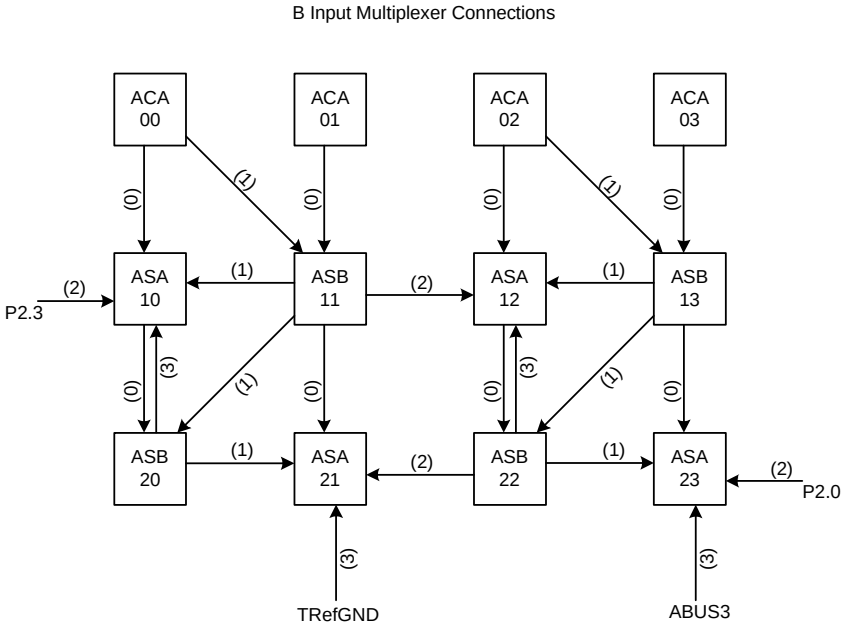

Figure 25: BMuxSCA/SCB Connections

10.8.3 Registers

10.8.3.1 Analog Switch Cap Type A Block xx Control 0 Register

FCap controls the size of the switched feedback capacitor in the integrator.

ClockPhase controls the internal clock phasing relative to the input clock phasing. ClockPhase affects the output of the analog column bus which is controlled by the AnalogBus bit in Control 2 Register (ASA10CR2, ASA12CR2, ASA21CR2, ASA23CR2).

ASign controls the switch phasing of the switches on the bottom plate of the ACap capacitor. The bottom plate samples the input or the reference.

The ACap bits set the value of the capacitor in the A path.

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

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/ Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	FCap	ClockPhase	ASign	ACap[4]	ACap[3]	ACap[2]	ACap[1]	ACap[0]

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

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

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

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

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

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

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

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

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

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

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

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.

12.0 Development Tools

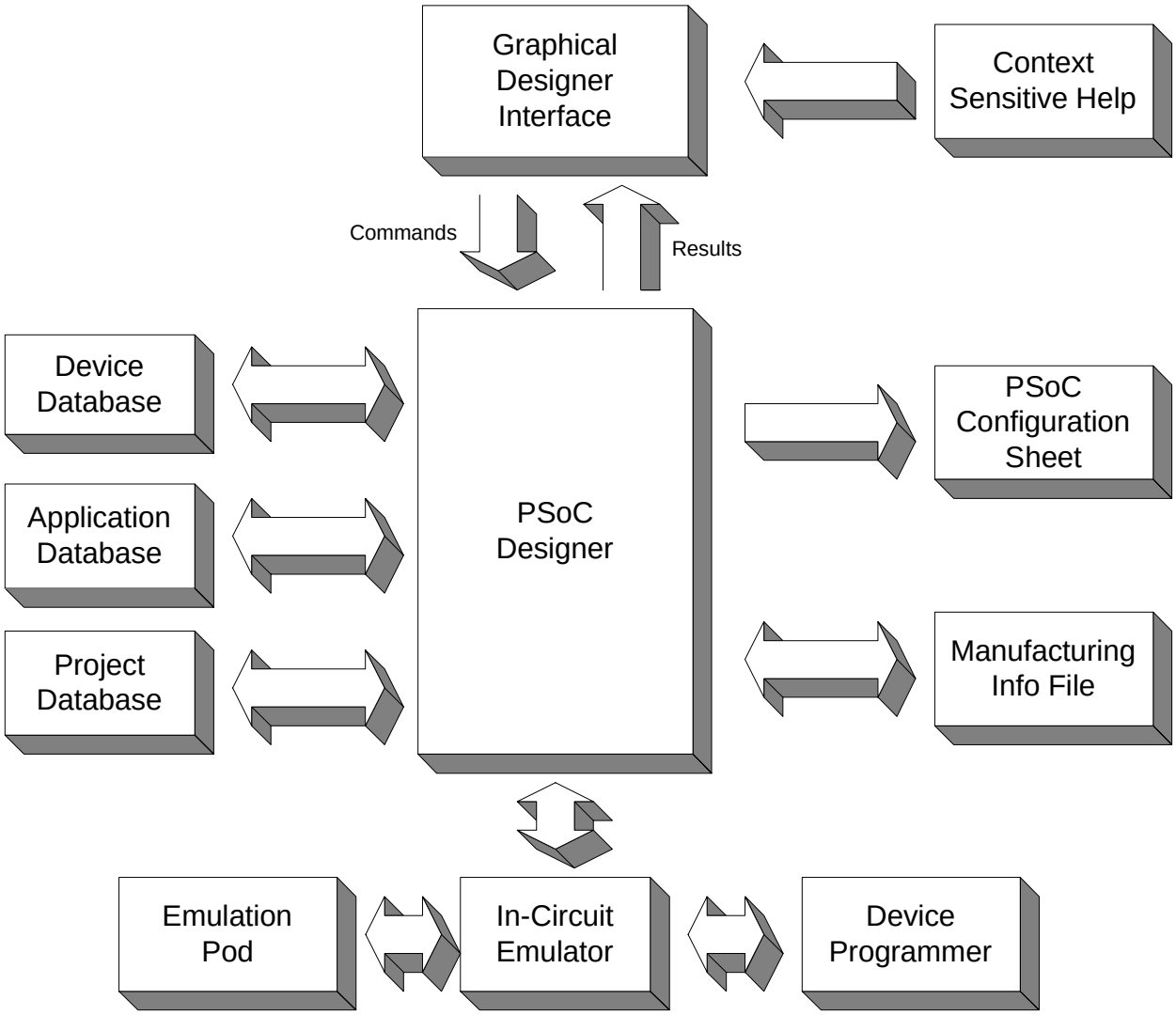

Figure 35: PSoC Designer Functional Flow

12.1 Overview

The Cypress MicroSystems PSoC Designer is a Microsoft® Windows-based, integrated development environment for the Programmable System-on-Chip (PSoC) devices. The PSoC Designer runs on Windows 98, Windows NT 4.0, Windows 2000, Windows Millennium (Me), or Windows XP.

PSoC Designer helps the customer to select an operating configuration for the microcontroller, write application code that uses the microcontroller, and debug the application. This system provides design database management by project, an integrated debugger with In-Circuit Emulator, in-system programming support, and the CYASM macro assembler for the CPUs.

PSoC Designer also supports a high-level C language compiler developed specifically for the devices in the family.

12.2 Integrated Development Environment Subsystems

12.2.1 Online Help System

The online help system displays online, context-sensitive help for the user. Designed for procedural and quick reference, each functional subsystem has its own context-sensitive help. This system also provides tutorials and links to FAQs and an Online Support Forum to aid the designer in getting started.

12.2.2 Device Editor

PSoC Designer has several main functions. The Device Editor subsystem lets the user select different onboard analog and digital component configurations for the PSoC blocks. PSoC Designer sets up power-on initialization tables for selected PSoC block configurations and creates source code for an application framework. The framework contains software to operate the selected components and, if the project uses more than one operating configuration, contains routines to switch between different sets of PSoC block configurations at runtime. PSoC Designer can print out a configuration sheet for given project configuration for use during application programming in conjunction with the Device Data Sheet. Once the framework is generated, the user can add application-specific code to flesh out the framework. It's also possible to change the selected components and regenerate the framework.

12.2.3 Assembler

The included CYASM macro assembler supports the M8C microcontroller instruction set and generates a load file ready for device programming or system debugging using the ICE hardware.

12.2.4 C Language Software Development

A C language compiler supports Cypress MicroSystems' PSoC family devices. Even if you have never worked in the C language before, the product quickly allows you to create complete C programs for the PSoC family devices.

The embedded, optimizing C compiler provides all the features of C tailored to the PSoC architecture. It includes a built-in macro assembler allowing assembly code to be merged seamlessly with C code. The link libraries automatically use absolute addressing or can be compiled in relative mode, and linked with other software modules to get absolute addressing.

The compiler comes complete with embedded libraries providing port and bus operations, standard keypad and display support, and extended math functionality.

12.2.5 Debugger

The PSoC Designer Debugger subsystem provides hardware in-circuit emulation, allowing the designer to test the program in a physical system while providing an internal view of the PSoC device. Debugger commands allow the designer to read and write program and data memory, read and write I/O registers, read and write CPU registers, set and clear breakpoints, and provide program run, halt, and step control. The debugger also allows the designer to create a trace buffer of registers and memory locations of interest.

12.3 Hardware Tools

12.3.1 In-Circuit Emulator

A low cost, high functionality ICE is available for development support. This hardware has the capability to program single devices.