PSoC® CY8C24533, CY8C23533, CY8C23433

Technical Reference Manual (TRM)

Document # 001-20559 Rev. *D

January 19, 2017

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone (USA): 800.858.1810
Phone (Intnl.): 408.943.2600
http://www.cypress.com

Copyrights

Contents Overview

Contents

Section A: Overview

The PSoC® family consists of programmable system-on-chips with on-chip controller devices. As described in this technical reference manual (TRM), a PSoC device includes configurable blocks of analog circuits and ***digital logic***, as well as programmable interconnect. This architecture allows the user to create customized peripheral configurations, to match the requirements of each individual application. Additionally, a fast CPU, Flash program memory, SRAM data memory, and configurable input/output (IO) are included in a range of pinouts.

This document is a technical reference manual for the PSoC device: CY8C24533, CY8C23533, CY8C23433CY8C24633. For the most up-to-date Ordering, Pinout, Packaging, or Electrical Specification information, refer to the PSoC data sheet. For the most current technical reference manual information, refer to the addendum. To obtain the newest product documentation, go to the Cypress web site at http://www.cypress.com/psoc. This section encompasses the following chapter:

- Pin Information on page 25

Document Organization

This manual is organized into sections and chapters, according to ***PSoC*** functionality. Each section begins with documentation interpretation, a top-level architectural explanation, PSoC device distinctions (if relevant), and a register summary (if applicable). Most chapters within the sections have an introduction, an architectural/application description, PSoC device distinctions (if relevant), register definitions, and timing diagrams. The sections are as follows:

- **Overview** – Presents the PSoC top-level architecture, PSoC device characteristics and distinctions, how to get started with helpful information, and document history and conventions. The PSoC device ***pinouts*** are detailed in the Pin Information chapter on page 25.
- **PSoC Core** – Describes the heart of the PSoC device in various chapters, beginning with an architectural overview and a summary list of registers pertaining to the PSoC core. See “PSoC Core” on page 31.
- **Register Reference** – Lists all PSoC device registers in Register Mapping Tables, on page 43, and presents bit-level detail of each PSoC register in its own Register Details chapter on page 47. Where applicable, detailed register descriptions are also located in each chapter.
- **Digital System** – Describes the configurable PSoC digital system in various chapters, beginning with an architectural overview and a summary list of registers pertaining to the digital system. See the “Digital System” on page 161.
- **Analog System** – Describes the configurable PSoC analog system in various chapters, beginning with an architectural overview and a summary list of registers pertaining to the analog system. See the “Analog System” on page 215.
- **System Resources** – Presents additional PSoC system resources, depending on the PSoC device, beginning with an overview and a summary list of registers pertaining to system resources. See “System Resources” on page 271.
- **Glossary** – Defines the specialized terminology used in this manual. Glossary terms are presented in ***bold, italic font*** throughout this manual. See the “Glossary” on page 321.
- **Index** – Lists the location of key topics and elements that constitute and empower the PSoC device. See the “Index” on page 401.

Top-Level Architecture

The PSoC block diagram on the next page illustrates the top-level architecture of the PSoC device. Each major grouping in the diagram is covered in this manual in its own section: PSoC Core, Digital System, Analog System, and the System Resources. Banding these four main areas together is the communication network of the system ***bus***.

PSoC Core

The PSoC Core is a powerful engine that supports a rich instruction set. It encompasses the ***SRAM*** for data storage, an ***interrupt*** controller for easy program execution to new addresses, sleep and watchdog timers, and multiple ***clock*** sources that include the phase locked loop (PLL), IMO (internal main oscillator), ILO (internal low speed oscillator), and ECO (32.768 kHz external crystal oscillator) for precision, programmable clocking. The clocks, together with programmable clock dividers (as a System Resource), provide the flexibility to integrate almost any timing requirement into the PSoC device.

The CPU core, called the M8C, is a powerful processor with speeds up to 24 MHz. The M8C is a four MIPS 8-***bit*** Harvard architecture microprocessor. Within the CPU core are the ***SROM*** and ***Flash*** memory components that provide flexible programming.

PSoC GPIOs provide connection to the CPU, digital and analog resources of the device. Each pin’s drive mode may be selected from eight options, allowing great flexibility in external interfacing. Every pin also has the capability to generate a system interrupt on high level, low level, and change from last read.

Digital System

The Digital System is composed of digital rows in a block array, and the Global, Array, and Row Digital Interconnects (GDI, ADI, and RDI, respectively).The digital system block is composed of 4 digital PSoC blocks. Each block is an 8-bit resource that can be used alone or combined with other blocks to form 8-, 16-, 24-, and 32-bit peripherals, which are called user modules.

The digital blocks can be connected to any GPIO through a series of global buses that can route any signal to any pin. The buses also allow for signal multiplexing and for performing logic operations. This configurability frees your designs from the constraints of a fixed peripheral controller.

Analog System

The Analog System is composed of analog columns in a block array, analog references, analog ***input*** muxing, and analog drivers. The analog system block is composed of 6 configurable blocks, each comprised of an opamp circuit allowing the creation of complex analog signal flows.

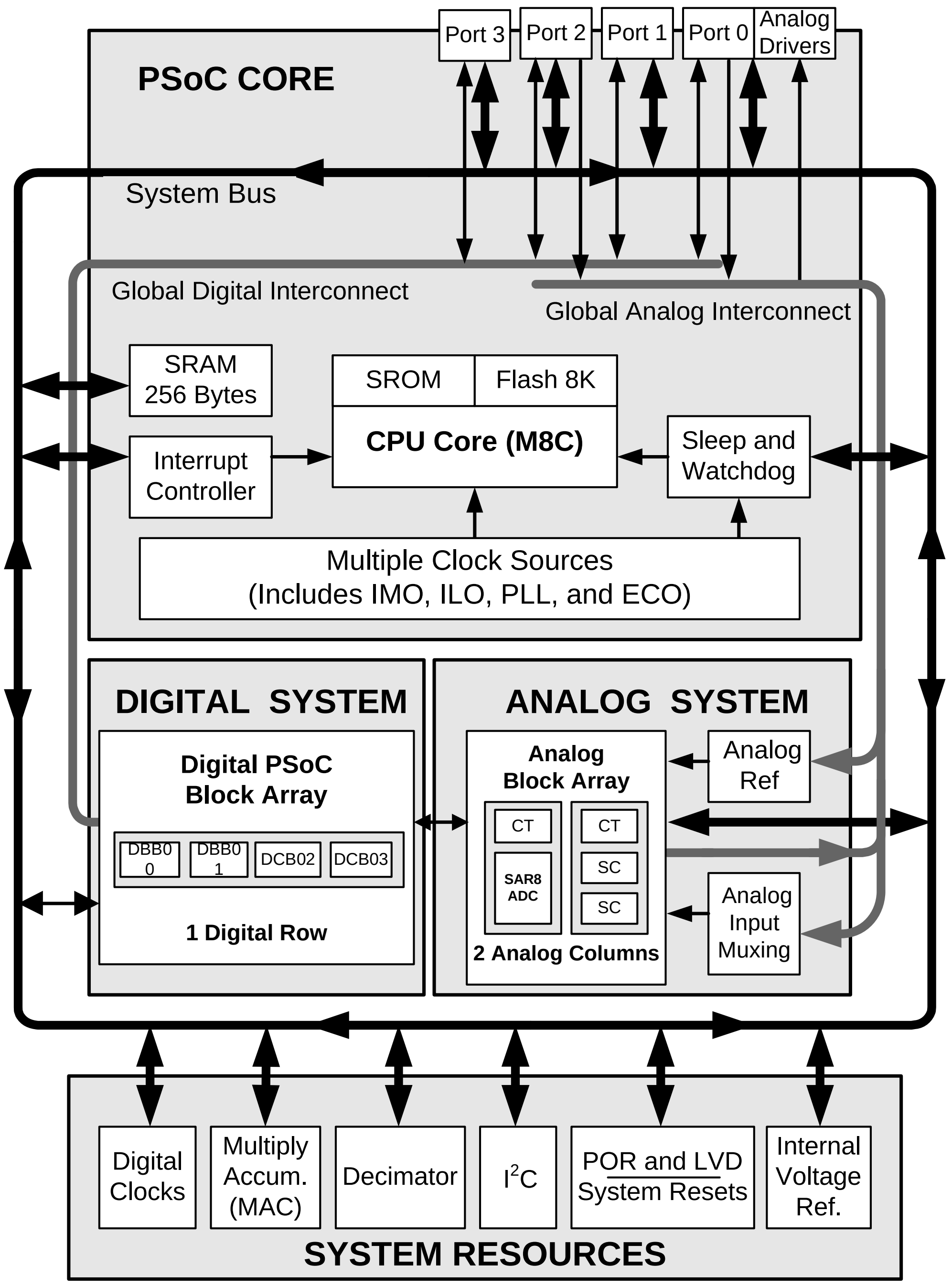
Analog blocks are arranged in a column of three, which includes one CT (Continuous Time) and two SC (Switched Capacitor) blocks. The Analog Column 0 contains the SAR8 ADC block rather than the standard SC blocks.

System Resources

The System Resources provide additional PSoC capability. These system resources include:

- Digital clocks to increase the flexibility of the PSoC device.
- One multiply accumulate (MAC) provides a fast 8-bit multiplier with 32-bit accumulate to assist in both general math as well as digital filters.
- The decimator provides a custom hardware filter for digital ***signal*** processing applications, including the creation of Delta Sigma ADCs.
- ***I2C*** functionality for implementing either I2C slave or master.
- Low Voltage Detection (LVD) interrupts can signal the application of falling voltage levels, while the advanced POR (Power On Reset) circuit eliminates the need for a system supervisor.
- An internal voltage reference that provides an absolute value of 1.3 V to a variety of PSoC subsystems.
- Various system resets supported by the M8C.

PSoC Top-Level Block Diagram



PSoC Device Characteristics

The PSoC ***digital*** system has 1 digital row and the ***analog*** system has 2 analog columns, as described in the following table.

PSoC Device Characteristics

PSoC Part Number	Digital IO (max)	Digital Rows	Digital Blocks	Analog Inputs	Analog Outputs	Analog Columns	Analog Blocks	Amount of SRAM	Amount of Flash
CY8C24423A	24	1	4	12	2	2	6	256 Bytes	4 KB
CY8C24533	26	1	4	12	2	2	4[a]	256 Bytes	8 KB
CY8C23533	26	1	4	12	2	2	4[a]	256 Bytes	8 KB
CY8C23433	26	1	4	12	2	2	4[a]	256 Bytes	8 KB
CY8C24633	25	1	4	12	2	2	4[a]	256 Bytes	8 KB

a. 2 CT, 2 SC.

The following table lists the resources available for CY8C24633, CY8C24533, CY8C23533, CY8C23433-specific PSoC device groups. The check mark or appropriate information denotes that a system resource is available for the PSoC device. Blank fields denote that the system resource is not available. These resources are detailed in the section titled “System Resources” on page 271.

Availability of System Resources for PSoC Devices

PSoC Part Number	Digital Clocks	I2C	Internal Voltage Ref	POR and LVD	System Resets	Decimator	Multiply Accumulate	SAR8 ADC	XRES Pin
CY8C24423A	✓	✓	✓	✓	✓	T1	1		
CY8C24533	✓	✓	✓	✓	✓	T1	1	✓	
CY8C23533	✓	✓	✓	✓	✓	T1	1	✓	✓
CY8C23433	✓	✓	✓	✓	✓	T1	1	✓	
CY8C24633	✓	✓	✓	✓	✓	T1	1	✓	

PSoC Device Distinctions

The PSoC device distinctions are listed in the table below and in each chapter section where it is appropriate. The PSoC device distinctions are significant exceptions or differences between PSoC groups and devices.

PSoC Device Distinctions

Device Distinctions	Devices Affected	Described in Chapter
Low Power Oscillator Capability The slow IMO (SLIMO) bit is available to enable SYSCLK operation at 6 MHz and 12 MHz, instead of only 24 MHz. The SLIMO bit is located in the CPU_SCR1 register on page 121.	CY8C24x23A CY8C24633CY8C24533, CY8C23533, CY8C23433	Internal Main Oscillator (IMO) chapter on page 15.
POR and LVD Trip Levels The lowest POR level is set for 2.4V operation; the next lowest is set for 3.0V operation (instead of 3.0V or 4.5V operation).	CY8C24x23A CY8C24533, CY8C23533, CY8C23433CY8C24633	POR and LVD chapter on page 319 and PSoC device data sheets.
Register Distinction CPU_SCR1 register on page 121 bit 4 (Slow IMO mode) is reserved.	CY8C24533, CY8C23533, CY8C23433CY8C24633	Internal Main Oscillator (IMO) chapter on page 15.
Register Distinction CPU_SCR1 register on page 121 bits 3 and 2 (ECO EXW and ECO EX, respectively) cannot be used.	CY8C24533, CY8C23533, CY8C23433CY8C24633	External Crystal Oscillator (ECO) chapter on page 21.
Register Distinction DEC_CR1 register on page 111 bit 7 (ECNT) is only available in devices with a type 1 decimator.	CY8C24x23A CY8C24533, CY8C23533, CY8C23433CY8C24633	Analog Interface chapter on page 219 and Decimator chapter on page 291.
Register Distinction OSC_GO_EN register on page 148 bit 7 is reserved.	CY8C24533, CY8C23533, CY8C23433CY8C24633	Digital Clocks chapter on page 275.

Getting Started

The quickest path to understanding PSoC is by reading the PSoC device's data sheet and using the *PSoC Designer Integrated Development Environment (IDE)*. This manual is useful for understanding the details of the PSoC integrated circuit. **Important Note:** For the most up-to-date Ordering, Packaging, or Electrical Specification information, refer to the individual PSoC device's data sheet or go to http://www.cypress.com/psoc.

Support

Free support for PSoC products is available online at http://www.cypress.com. Resources include Training Seminars, Discussion Forums, Application Notes, PSoC Consultants, TightLink Technical Support Email/Knowledge Base, and Application Support Technicians. Technical Support can be reached at http://www.cypress.com/support or by phone at:1-425-787-4814.

Product Upgrades

Cypress provides scheduled upgrades and version enhancements for PSoC Designer free of charge. You can order the upgrades from your distributor on CD-ROM or download them directly from http://www.cypress.com.

Development Kits

Development Kits are available from the following distributors: Digi-Key, Avnet, Arrow, and Future. The Cypress Online Store contains development kits, ***C*** compilers, and all accessories for PSoC development. Go to the Cypress Online Store web site at http://www.onfulfillment.com/cypressstore.

Document History

This section serves as a chronicle of the *PSoC CY8C24533, CY8C23533, CY8C23433 Technical Reference Manual*

PSoC Technical Reference Manual History

Version/ Release Date	Originator	Description of Change
** August, 2007	HMT	First release of the *PSoC CY8C24533 Technical Reference Manual*. This release encompasses the following PSoC device: CY8C24533.
*A February 2010	HMT	Add CY8C23533, CY8C23433 and update pinout diagrams.
*B July 2013	RJVB	Removed reference to the IMODIS bit.
*C December 2013	MSON	No content update; sunset review
*D January 2017	RJVB	Added information for "no glitch protection in the device for an external clock".

Documentation Conventions

There are only four distinguishing font types used in this manual, besides those found in the headings.

- The first is the use of *italics* when referencing a document title or file name.
- The second is the use of ***bold italics*** when referencing a term described in the Glossary of this manual.
- The third is the use of Times New Roman font, distinguishing equation examples.
- The fourth is the use of `Courier New` font, distinguishing code examples.

Register Conventions

The following table lists the register conventions that are specific to this manual. A more detailed set of register conventions is located in the Register Details chapter on page 47.

Register Conventions

Convention	Example	Description
‘x’ in a register name	ACBxxCR1	Multiple instances/address ranges of the same register
R	R : 00	Read register or bit(s)
W	W : 00	Write register or bit(s)
L	RL : 00	Logical register or bit(s)
C	RC : 00	Clearable register or bit(s)
00	RW : 00	Reset value is 0x00 or 00h
XX	RW : XX	Register is not reset
0,	0,04h	Register is in bank 0
1,	1,23h	Register is in bank 1
x,	x,F7h	Register exists in register bank 0 and register bank 1
Empty, grayed-out table cell	▨	Reserved bit or group of bits, unless otherwise stated

Numeric Naming

Hexidecimal numbers are represented with all letters in uppercase with an appended lowercase ‘h’ (for example, ‘14h’ or ‘3Ah’) and ***hexidecimal*** numbers may also be represented by a ‘0x’ prefix, the ***C*** coding convention. Binary numbers have an appended lowercase ‘b’ (for example, 01010100b’ or ‘01000011b’). Numbers not indicated by an ‘h’ or ‘b’ are ***decimal***.

Units of Measure

The following table lists the units of measure used in this manual.

Units of Measure

Symbol	Unit of Measure
dB	decibels
Hz	hertz
k	kilo, 1000
K	2^{10}, 1024
KB	1024 bytes
Kbit	1024 bits
kHz	kilohertz (32.000)
MHz	megahertz
μA	microampere
μF	microfarad
μs	microsecond
μV	microvolts
mA	milli-ampere
ms	milli-second
mV	milli-volts
ns	nanosecond
pF	picofarad
ppm	parts per million
V	volts

Acronyms

The following table lists the acronyms that are used in this manual.

Acronyms

Acronym	Description
ABUS	analog output bus
AC	alternating current
ADC	analog-to-digital converter
API	Application Programming Interface
BC	broadcast clock
BR	bit rate
BRA	bus request acknowledge
BRQ	bus request
CBUS	comparator bus
CI	carry in
CMP	compare
CO	carry out
CPU	central processing unit
CRC	cyclic redundancy check
CT	continuous time
DAC	digital-to-analog converter
DC	direct current
DI	digital or data input
DMA	direct memory access
DO	digital or data output
ECO	external crystal oscillator
FB	feedback
GIE	global interrupt enable
GPIO	general purpose IO
ICE	in-circuit emulator
IDE	integrated development environment
ILO	internal low speed oscillator
IMO	internal main oscillator
IO	input/output
IOR	IO read
IOW	IO write
IPOR	imprecise power on reset
IRQ	interrupt request
ISR	interrupt service routine
ISSP	in system serial programming
IVR	interrupt vector read
LFSR	linear feedback shift register
LRb	last received bit
LRB	last received byte
LSb	least significant bit
LSB	least significant byte
LUT	look-up table
MISO	master-in-slave-out
MOSI	master-out-slave-in
MSb	most significant bit
MSB	most significant byte
PC	program counter
PCH	program counter high
PCL	program counter low
PD	power down
PMA	PSoC® memory arbiter
POR	power on reset
PPOR	precision power on reset
PRS	pseudo random sequence
PSoC®	Programmable System-on-Chip™
PSSDC	power system sleep duty cycle
PWM	pulse width modulator
RAM	random access memory
RETI	return from interrupt
RI	row input
RO	row output
ROM	read only memory
RW	read/write
SAR	successive approximation register
SC	switched capacitor
SIE	serial interface engine
SE0	single-ended zero
SOF	start of frame
SP	stack pointer
SPI	serial peripheral interconnect
SPIM	serial peripheral interconnect master
SPIS	serial peripheral interconnect slave
SRAM	static random access memory
SROM	supervisory read only memory
SSADC	single slope ADC
SSC	supervisory system call
TC	terminal count
USB	universal serial bus
WDT	watchdog timer
WDR	watchdog reset
XRES	external reset

1. Pin Information

This chapter lists, describes, and illustrates CY8C24533, CY8C23533, CY8C23433CY8C24633 device pins and pinout configurations. For up-to-date Ordering, Pinout, and Packaging information, refer to the individual PSoC device's data sheet at http://www.cypress.com/psoc.

1.1 Pinouts

The PSoC CY8C24533, CY8C23533, CY8C23433CY8C24633 are available in 28-pin SSOP and 32-pin QFN and 56-pin SSOP OCDpackages. Refer to the following information for details. Every ***port*** pin (labeled with a "P"), except for ***Vss*** and ***Vdd***, and XRES in the following tables and illustrations, is capable of Digital IO.

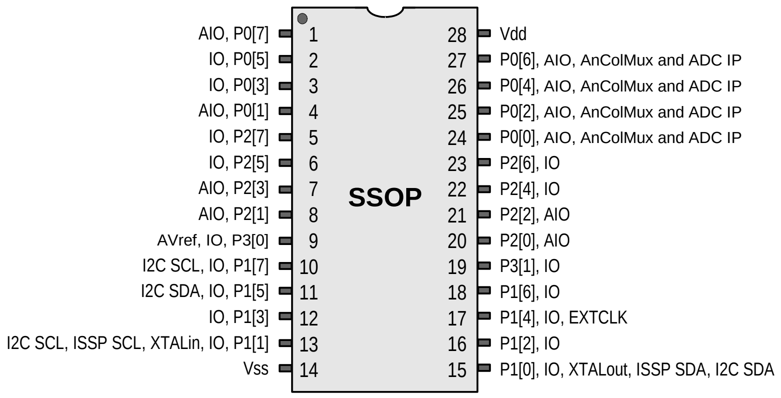
1.1.1 28-Pin Part Pinout

The 28-pin part is for the CY8C24533 CY8C24633 PSoC device.

Table 1-1. 28-Pin Part Pinout (SSOP)

Pin No.	Digital	Analog	Pin Name	Description
1	IO	I	P0[7]	Analog Col Mux IP and ADC IP
2	IO	IO	P0[5]	Analog Col Mux IP and Column O/P and ADC IP
3	IO	IO	P0[3]	Analog Col Mux IP and Column O/P and ADC IP
4	IO	I	P0[1]	Analog Col Mux IP and ADC IP
5	IO		P2[7]	GPIO
6	IO		P2[5]	GPIO
7	IO	I	P2[3]	Direct switched capacitor input
8	IO	I	P2[1]	Direct switched capacitor input
9	IO	AVref	P3[0]*	GPIO/ADC Vref (optional)
10	IO		P1[7]	I2C SCL
11	IO		P1[5]	I2C SDA
12	IO		P1[3]	GPIO
13	IO		P1[1]**	GPIO, Xtal input, I2C SCL, ISSP SCL
14	Power		Vss	Ground pin
15	IO		P1[0]**	GPIO, Xtal output, I2C SDA, ISSP SDA
16	IO		P1[2]	GPIO
17	IO		P1[4]	GPIO, external clock IP
18	IO		P1[6]	GPIO
19	IO		P3[1]***	GPIO
19			XRES	Active high pin reset with internal pull down
20	IO	I	P2[0]	Direct switched capacitor input
21	IO	I	P2[2]	Direct switched capacitor input
22	IO		P2[4]	GPIO
23	IO		P2[6]	GPIO
24	IO	I	P0[0]	Analog Col Mux IP and ADC IP
25	IO	I	P0[2]	Analog Col Mux IP and ADC IP
26	IO	I	P0[4]	Analog Col Mux IP and ADC IP
27	IO	I	P0[6]	Analog Col Mux IP and ADC IP
28	Power		Vdd	Supply voltage

CY8C24533 CY8C24633 PSoC Device



LEGEND: A = Analog, I = Input, and O = Output.
* Even though P3[0] is an odd port, it resides on the left side of the pinout.
** ISSP pin, which is not High Z at POR.
*** Even though P3[1] is an even port, it resides on the right side of the pinout.

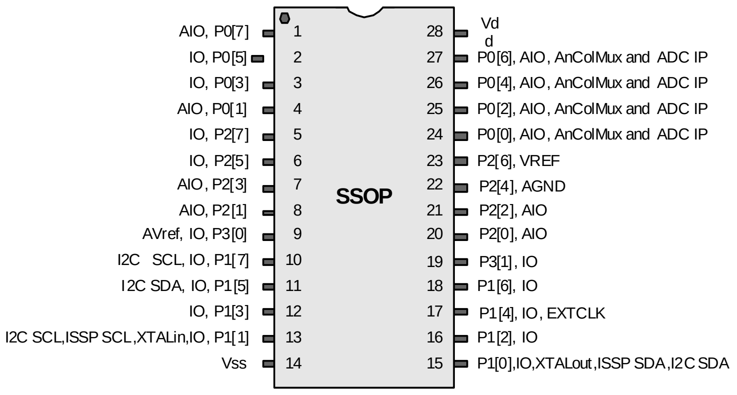
The 28-pin part is for the CY8C23433 PSoC device.

Table 1-2. 28-Pin Part Pinout (SSOP)

Pin No.	Digital	Analog	Pin Name	Description
1	IO	I	P0[7]	Analog Column Mux IP and ADC IP
2	IO	IO	P0[5]	Analog Column Mux IP and Column O/P and ADC IP
3	IO	IO	P0[3]	Analog Column Mux IP and Column O/P and ADC IP
4	IO	I	P0[1]	Analog Column Mux IP and ADC IP
5	IO		P2[7]	GPIO
6	IO		P2[5]	GPIO
7	IO	I	P2[3]	Direct Switched Capacitor Input
8	IO	I	P2[1]	Direct Switched Capacitor Input
9	IO	AVref	P3[0]*	GPIO/ADC Vref (optional)
10	IO		P1[7]	I2C SCL
11	IO		P1[5]	I2C SDA
12	IO		P1[3]	GPIO
13	IO		P1[1]**	GPIO, Xtal Input, I2C SCL, ISSP SCL
14	Power		Vss	Ground Pin
15	IO		P1[0]**	GPIO, Xtal Output, I2C SDA, ISSP SDA
16	IO		P1[2]	GPIO
17	IO		P1[4]	GPIO, External Clock IP
18	IO		P1[6]	GPIO
19	IO		P3[1]***	GPIO
20	IO	I	P2[0]	Direct Switched Capacitor Input
21	IO	I	P2[2]	Direct Switched Capacitor Input
22	IO		P2[4]	External Analog Ground (AGnd)
23	IO		P2[6]	Analog Voltage Reference (VRef)
24	IO	I	P0[0]	Analog Column Mux IP and ADC IP
25	IO	I	P0[2]	Analog Column Mux IP and ADC IP
26	IO	I	P0[4]	Analog Column Mux IP and ADC IP
27	IO	I	P0[6]	Analog Column Mux IP and ADC IP
28	Power		Vdd	Supply Voltage

LEGEND: A = Analog, I = Input, and O = Output.
* Even though P3[0] is an odd port, it resides on the left side of the pinout.
** ISSP pin, which is not High Z at POR.
*** Even though P3[1] is an even port, it resides on the right side of the pinout.

CY8C23433 28-Pin PSoC Device



1.1.2 32-Pin Part Pinout

The 32-pin part is for the CY8C23533 PSoC device.

Table 1-3. 32-Pin Part Pinout (QFN)

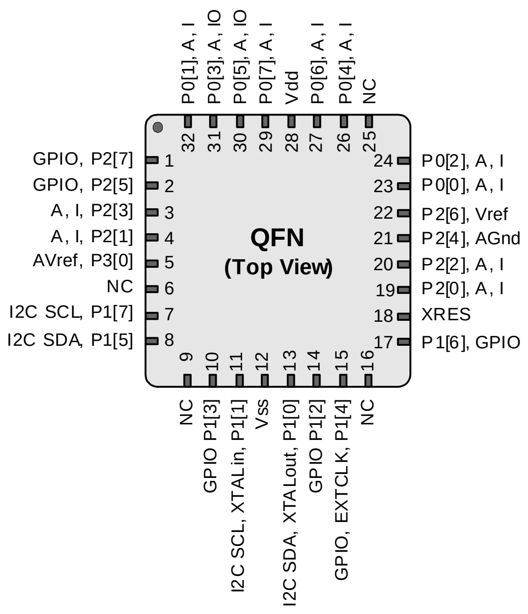
Pin No.	Type		Pin Name	Description
	Digital	Analog		
1	IO		P2[7]	GPIO
2	IO		P2[5]	GPIO
3	IO	I	P2[3]	Direct Switched Capacitor Block Input
4	IO	I	P2[1]	Direct Switched Capacitor Block Input
5	IO	AVref	P3[0]*	GPIO/ADC Vref (optional)
6			NC	No Connection
7	IO		P1[7]	I2C Serial Clock (SCL)
8	IO		P1[5]	I2C Serial Data (SDA)
9			NC	No Connection
10	IO		P1[3]	GPIO
11	IO		P1[1]**	GPIO, Crystal Input (XTALin), I2C Serial Clock (SCL), ISSP-SCLK
12	Power		Vss	Ground Connection
13	IO		P1[0]**	GPIO, Crystal Output (XTALout), I2C Serial Data (SDA), ISSP-SDATA
14	IO		P1[2]	GPIO
15	IO		P1[4]	GPIO, External Clock IP
16			NC	No Connection
17	IO		P1[6]	GPIO
18	Input		XRES	Active High External Reset with Internal Pull Down
19	IO	I	P2[0]	Direct Switched Capacitor Block Input
20	IO	I	P2[2]	Direct Switched Capacitor Block Input
21	IO		P2[4]	External Analog Ground (AGnd)
22	IO		P2[6]	External Voltage Reference (VRef)
23	IO	I	P0[0]	Analog Column Mux Input and ADC Input
24	IO	I	P0[2]	Analog Column Mux Input and ADC Input
25			NC	No Connection
26	IO	I	P0[4]	Analog Column Mux Input and ADC Input
27	IO	I	P0[6]	Analog Column Mux Input and ADC Input
28	Power		Vdd	Supply Voltage
29	IO	I	P0[7]	Analog Column Mux Input and ADC Input
30	IO	IO	P0[5]	Analog Column Mux Input, Column Output and ADC Input
31	IO	IO	P0[3]	Analog Column Mux Input, Column Output and ADC Input
32	IO	I	P0[1]	Analog Column Mux Input.and ADC Input

LEGEND: A = Analog, I = Input, and O = Output.

* Even though P3[0] is an odd port, it resides on the left side of the pinout.

** ISSP pin, which is not High Z at POR.

CY8C23533 32-Pin PSoC Device



1.1.3 56-Pin Part Pinout

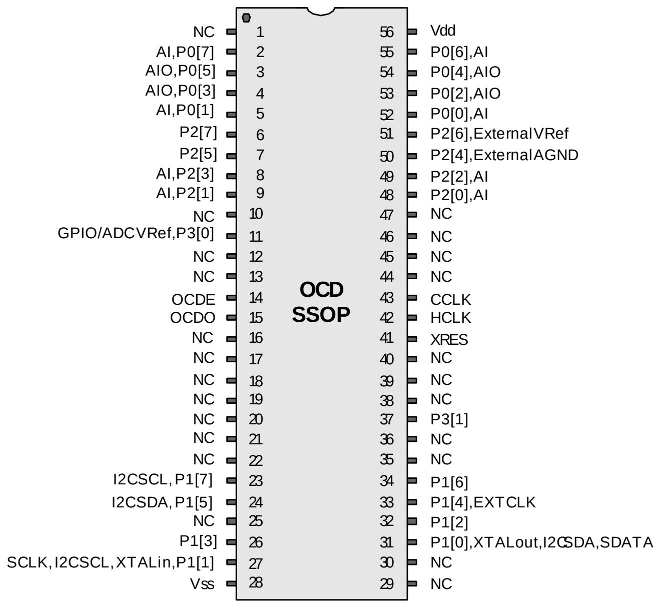
The 56-pin OCD (On-Chip Debug) part is for the CY8C24633 (CY8C24033) PSoC device.

Note OCD parts are only used for in-circuit debugging. OCD parts are NOT available for production.

Table 1-4. 56-Pin OCD Part Pinout (SSOP)

Pin No.	Name	Description
1	NC	No internal connection
2	P0[7]	Analog column mux input: AI
3	P0[5]	Analog column mux input and column output: AIO
4	P0[3]	Analog column mux input and column output: AIO
5	P0[1]	Analog column mux input: AI
6	P2[7]	
7	P2[5]	
8	P2[3]	Direct switched capacitor block input: AI
9	P2[1]	Direct switched capacitor block input: AI
10	NC	No internal connection
11	P3[0]	GPIO/ADC Vref (optional)
12	NC	No internal connection
13	NC	No internal connection
14	OCDE	OCD even data IO
15	OCDO	OCD odd data output
16	NC	No internal connection
17	NC	No internal connection
18	NC	No internal connection
19	NC	No internal connection
20	NC	No internal connection
21	NC	No internal connection
22	NC	No internal connection
23	P1[7]	I2C Serial Clock (SCL)
24	P1[5]	I2C Serial Data (SDA)
25	NC	No internal connection
26	P1[3]	
27	P1[1]*	Crystal (XTALin), I2C Serial Clock (SCL)
28	Vss	Ground connection
29	NC	No internal connection
30	NC	No internal connection
31	P1[0]*	Crystal (XTALout), I2C Serial Data (SDA)
32	P1[2]	
33	P1[4]	Optional External Clock Input (EXTCLK)
34	P1[6]	
35	NC	No internal connection
36	NC	No internal connection
37	P3[1]	GPIO
38	NC	No internal connection
39	NC	No internal connection
40	NC	No internal connection
41	XRES	Active high pin reset with internal pull down
42	HCLK	OCD high speed clock output
43	CCLK	OCD CPU clock output
44	NC	No internal connection
45	NC	No internal connection
46	NC	No internal connection
47	NC	No internal connection
48	P2[0]	Direct switched capacitor block input: AI
49	P2[2]	Direct switched capacitor block input: AI
50	P2[4]	External Analog Ground (AGND)
51	P2[6]	External Voltage Reference (VRef)
52	P0[0]	Analog column mux input: AI
53	P0[2]	Analog column mux input and column output: AIO
54	P0[4]	Analog column mux input and column output: AIO
55	P0[6]	Analog column mux input: AI
56	Vdd	Supply voltage

CY8C24033 OCD PSoC Device



NOT FOR PRODUCTION

LEGEND A = Analog, I = Input, O = Output.
* ISSP pin, which is not High Z at POR.

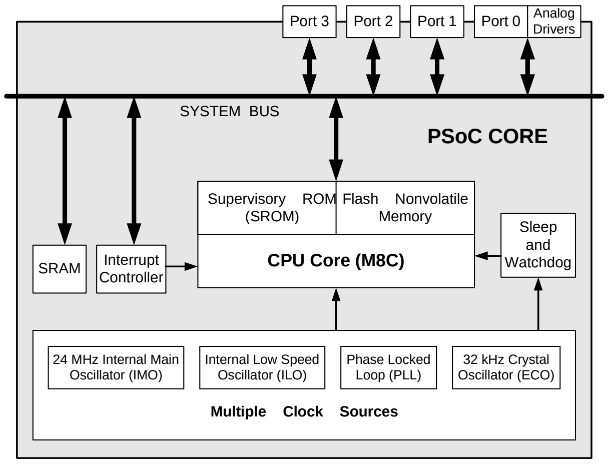
Section B: PSoC Core

The PSoC Core section discusses the core components of the PSoC devices: CY8C24533, CY8C23533, CY8C23433CY8C24533, and the registers associated with those components. This section encompasses the following chapters:

Top-Level Core Architecture

The figure below displays the top-level architecture of the PSoC's core. Each component of the figure is discussed at length in this section.

PSoC Core Block Diagram



Interpreting Core Documentation

The core section covers the heart of the PSoC device which includes the M8C ***microcontroller***; SROM, interrupt controller, GPIO, analog output drivers, and ***SRAM*** paging; multiple clock sources such as IMO, ILO, ECO, and PLL; and sleep and watchdog functionality.

The ***analog output*** drivers are described in this section and not the Analog System section because they are part of the PSoC core input and ***output*** signals.

Core Register Summary

The table below lists all the PSoC registers for the CPU core in ***address*** order within their system resource configuration. The bits that are grayed out are reserved bits. If these bits are written, they should always be written with a value of ‘0’. For the core registers, the first ‘x’ in some ***register*** addresses represents either bank 0 or bank 1. These registers are listed throughout this manual in bank 0, even though they are also available in bank 1.

The CY8C24533, CY8C23533, CY8C23433CY8C24633 PSoC devices have 1 digital row and 2 analog columns.

Summary Table of the Core Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
M8C REGISTER (page 44)										
x,F7h	CPU_F	PgMode[1:0]			XIO		Carry	Zero	GIE	RL : 02
SUPERVISORY ROM (SROM) REGISTERS (page 51)										
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW	ECO EX		IRAMDIS	# : 00
1,FAh	FLS_PR1							Bank[1:0]		RW:00
RAM PAGING (SRAM) REGISTERS (page 58)										
x,6Ch	TMP_DR0	Data[7:0]								RW : 00
x,6Dh	TMP_DR1	Data[7:0]								RW : 00
x,6Eh	TMP_DR2	Data[7:0]								RW : 00
x,6Fh	TMP_DR3	Data[7:0]								RW : 00
x,F7h	CPU_F	PgMode[1:0]			XIO		Carry	Zero	GIE	RL : 02
INTERRUPT CONTROLLER REGISTERS (page 64)										
0,DAh	INT_CLR0	VC3	Sleep	GPIO	SAR8 ADC		Analog 1	Analog 0	V Monitor	RW : 00
0,DBh	INT_CLR1					DCB03	DCB02	DBB01	DBB00	RW : 00
0,DDh	INT_CLR3								I2C	RW : 00
0,DEh	INT_MSK3	ENSWINT							I2C	RW : 00
0,E0h	INT_MSK0	VC3	Sleep	GPIO	SAR8 ADC		Analog 1	Analog 0	V Monitor	RW : 00
0,E1h	INT_MSK1					DCB03	DCB02	DBB01	DBB00	RW : 00
0,E2h	INT_VC	Pending Interrupt[7:0]								RC : 00
x,F7h	CPU_F	PgMode[1:0]			XIO		Carry	Zero	GIE	RL : 02
GENERAL PURPOSE IO (GPIO) REGISTERS (page 8)										
0,00h	PRT0DR	Data[7:0]								RW : 00
0,01h	PRT0IE	Interrupt Enables[7:0]								RW : 00
0,02h	PRT0GS	Global Select[7:0]								RW : 00
0,03h	PRT0DM2	Drive Mode 2[7:0]								RW : FF
1,00h	PRT0DM0	Drive Mode 0[7:0]								RW : 00

Summary Table of the Core Registers *(continued)*

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,01h	PRT0DM1	Drive Mode 1[7:0]								RW : FF
1,02h	PRT0IC0	Interrupt Control 0[7:0]								RW : 00
1,03h	PRT0IC1	Interrupt Control 1[7:0]								RW : 00
0,04h	PRT1DR	Data[7:0]								RW : 00
0,05h	PRT1IE	Interrupt Enables[7:0]								RW : 00
0,06h	PRT1GS	Global Select[7:0]								RW : 00
0,07h	PRT1DM2	Drive Mode 2[7:0]								RW : FF
1,04h	PRT1DM0	Drive Mode 0[7:0]								RW : 00
1,05h	PRT1DM1	Drive Mode 1[7:0]								RW : FF
1,06h	PRT1IC0	Interrupt Control 0[7:0]								RW : 00
1,07h	PRT1IC1	Interrupt Control 1[7:0]								RW : 00
0,08h	PRT2DR	Data[7:0]								RW : 00
0,09h	PRT2IE	Interrupt Enables[7:0]								RW : 00
0,0Ah	PRT2GS	Global Select[7:0]								RW : 00
0,0Bh	PRT2DM2	Drive Mode 2[7:0]								RW : FF
1,08h	PRT2DM0	Drive Mode 0[7:0]								RW : 00
1,09h	PRT2DM1	Drive Mode 1[7:0]								RW : FF
1,0Ah	PRT2IC0	Interrupt Control 0[7:0]								RW : 00
1,0Bh	PRT2IC1	Interrupt Control 1[7:0]								RW : 00
0,0Ch	PRT3DR	Data[7:0]								RW : 00
0,0Dh	PRT3IE	Interrupt Enables[7:0]								RW : 00
0,0Eh	PRT3GS	Global Select[7:0]								RW : 00
0,0Fh	PRT3DM2	Drive Mode 2[7:0]								RW : FF
1,0Ch	PRT3DM0	Drive Mode 0[7:0]								RW : 00
1,0Dh	PRT3DM1	Drive Mode 1[7:0]								RW : FF
1,0Eh	PRT3IC0	Interrupt Control 0[7:0]								RW : 00
1,0Fh	PRT3IC1	Interrupt Control 1[7:0]								RW : 00
ANALOG OUTPUT DRIVER REGISTER (page 14)										
1,62h	ABF_CR0	ACol1Mux		ABUF1EN		ABUF0EN		Bypass	PWR	RW : 00
INTERNAL MAIN OSCILLATOR (IMO) REGISTERS (page 16)										
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW	ECO EX		IRAMDIS	# : 00
1,E2h	OSC_CR2	PLLGAIN					EXTCLKEN	RSVD	SYSCLKX2DIS	RW : 00
1,E8h	IMO_TR	Trim[7:0]								W : 00

Summary Table of the Core Registers *(continued)*

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
INTERNAL LOW SPEED OSCILLATOR (ILO) REGISTER (page 19)										
1,E9h	ILO_TR			Bias Trim[1:0]		Freq Trim[3:0]				W : 00
EXTERNAL CRYSTAL OSCILLATOR (ECO) REGISTERS (page 23)										
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW	ECO EX		IRAMDIS	# : 00
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00
1,EBh	ECO_TR	PSSDC[1:0]								W : 00
PHASE-LOCKED LOOP (PLL) REGISTERS (page 27)										
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00
1,E2h	OSC_CR2	PLLGAIN					EXTCLKEN	RSVD	SYSCLKX2DIS	RW : 00
SLEEP AND WATCHDOG REGISTERS (page 33)										
0,E0h	INT_MSK0	VC3	Sleep	GPIO			Analog 1	Analog 0	V Monitor	RW : 00
0,E3h	RES_WDT	WDSL_Clear[7:0]								W : 00
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW	ECO EX		IRAMDIS	# : 00
x,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	# : XX
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00
1,E9h	ILO_TR			Bias Trim[1:0]		Freq Trim[3:0]				W : 00
1,EBh	ECO_TR	PSSDC[1:0]								W : 00

LEGEND

L	The and f, expr; or f, expr; and xor f, expr instructions can be used to modify this register.
#	Access is bit specific. Refer to the Register Details chapter on page 47 for additional information.
X	The value for power on reset is unknown.
x	An “x” before the comma in the address field indicates that this register can be accessed or written to no matter what bank is used.
C	Clearable register or bit(s).
R	Read register or bit(s).
W	Write register or bit(s).

2. CPU Core (M8C)

This chapter explains the CPU Core, called M8C, and its associated register. It covers the internal M8C registers, address spaces, ***instruction*** formats, and addressing modes. For additional information concerning the M8C instruction set, refer to the *PSoC Designer Assembly Language User Guide* available at the Cypress web site (http://www.cypress.com/psoc). For a complete table of the CPU Core registers, refer to the "Summary Table of the Core Registers" on page 32. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

2.1 Overview

The ***M8C*** is a four MIPS 8-bit Harvard architecture microprocessor. Selectable processor clock speeds from 93.7 kHz to 24 MHz allow the M8C to be tuned to a particular application's performance and power requirements. The M8C supports a rich instruction set which allows for efficient low level language support.

2.2 Internal Registers

The M8C has five internal registers that are used in program execution. The following is a list of these registers.

- Accumulator (A)
- Index (X)
- Program Counter (PC)
- Stack Pointer (SP)
- Flags (F)

All of the internal M8C registers are eight bits in width, except for the PC, which is 16 bits wide. Upon ***reset***, A, X, PC, and SP are reset to 00h. The Flag register (F) is reset to 02h, indicating that the Z ***flag*** is ***set***.

With each ***stack*** operation, the SP is automatically incremented or decremented so that it always points to the next stack ***byte*** in RAM. If the last byte in the stack is at address FFh, the ***stack pointer*** wraps to RAM address 00h. It is the ***firmware*** developer's responsibility to ensure that the stack does not overlap with user-defined variables in RAM.

With the exception of the F register, the M8C internal registers are not accessible via an explicit register address. The internal M8C registers are accessed using the following instructions:

- `MOV A, expr`
- `MOV X, expr`
- `SWAP A, SP`
- `OR F, expr`
- `JMP LABEL`

The F register can be read by using address F7h in either register bank

2.3 Address Spaces

The M8C has three address spaces: ***ROM***, ***RAM***, and registers. The ROM address space includes the supervisory ROM (SROM) and the Flash. The ROM address space is accessed via its own address and ***data bus***.

The ROM address space is composed of the Supervisory ROM and the on-chip Flash program store. Flash is organized into 64-byte blocks. The user need not be concerned with program store page boundaries, as the M8C automatically increments the 16-bit PC on every instruction making the block boundaries invisible to user code. Instructions occurring on a 256-byte Flash page boundary (with the exception of jmp instructions) incur an extra M8C clock cycle, as the upper byte of the PC is incremented.

The register address space is used to configure the PSoC microcontroller's programmable blocks. It consists of two banks of 256 bytes each. To switch between banks, the XIO bit in the Flag register is set or cleared (set for Bank1, cleared for Bank0). The common convention is to leave the bank set to Bank0 (XIO cleared), switch to Bank1 as needed (set XIO), then switch back to Bank0.

2.4 Instruction Set Summary

The instruction set is summarized in both Table 2-1 and Table 2-2 (in numeric and ***mnemonic*** order, respectively), and serves as a quick reference. If more information is needed, the Instruction Set Summary tables are described in detail in the *PSoC Designer Assembly Language User Guide* (refer to http://www.cypress.com/psoc).

Table 2-1. Instruction Set Summary Sorted Numerically by Opcode

Opcode Hex	Cycles	Bytes	Instruction Format	Flags
00	15	1	SSC	
01	4	2	ADD A, expr	C, Z
02	6	2	ADD A, [expr]	C, Z
03	7	2	ADD A, [X+expr]	C, Z
04	7	2	ADD [expr], A	C, Z
05	8	2	ADD [X+expr], A	C, Z
06	9	3	ADD [expr], expr	C, Z
07	10	3	ADD [X+expr], expr	C, Z
08	4	1	PUSH A	
09	4	2	ADC A, expr	C, Z
0A	6	2	ADC A, [expr]	C, Z
0B	7	2	ADC A, [X+expr]	C, Z
0C	7	2	ADC [expr], A	C, Z
0D	8	2	ADC [X+expr], A	C, Z
0E	9	3	ADC [expr], expr	C, Z
0F	10	3	ADC [X+expr], expr	C, Z
10	4	1	PUSH X	
11	4	2	SUB A, expr	C, Z
12	6	2	SUB A, [expr]	C, Z
13	7	2	SUB A, [X+expr]	C, Z
14	7	2	SUB [expr], A	C, Z
15	8	2	SUB [X+expr], A	C, Z
16	9	3	SUB [expr], expr	C, Z
17	10	3	SUB [X+expr], expr	C, Z
18	5	1	POP A	Z
19	4	2	SBB A, expr	C, Z
1A	6	2	SBB A, [expr]	C, Z
1B	7	2	SBB A, [X+expr]	C, Z
1C	7	2	SBB [expr], A	C, Z
1D	8	2	SBB [X+expr], A	C, Z
1E	9	3	SBB [expr], expr	C, Z
1F	10	3	SBB [X+expr], expr	C, Z
20	5	1	POP X	
21	4	2	AND A, expr	Z
22	6	2	AND A, [expr]	Z
23	7	2	AND A, [X+expr]	Z
24	7	2	AND [expr], A	Z
25	8	2	AND [X+expr], A	Z
26	9	3	AND [expr], expr	Z
27	10	3	AND [X+expr], expr	Z
28	11	1	ROMX	Z
29	4	2	OR A, expr	Z
2A	6	2	OR A, [expr]	Z
2B	7	2	OR A, [X+expr]	Z
2C	7	2	OR [expr], A	Z
2D	8	2	OR [X+expr], A	Z
2E	9	3	OR [expr], expr	Z
2F	10	3	OR [X+expr], expr	Z
30	9	1	HALT	
31	4	2	XOR A, expr	Z
32	6	2	XOR A, [expr]	Z
33	7	2	XOR A, [X+expr]	Z
34	7	2	XOR [expr], A	Z
35	8	2	XOR [X+expr], A	Z
36	9	3	XOR [expr], expr	Z
37	10	3	XOR [X+expr], expr	Z
38	5	2	ADD SP, expr	
39	5	2	CMP A, expr	if (A=B) Z=1 if (A<B) C=1
3A	7	2	CMP A, [expr]	
3B	8	2	CMP A, [X+expr]	
3C	8	3	CMP [expr], expr	
3D	9	3	CMP [X+expr], expr	
3E	10	2	MVI A, [[expr]++]	Z
3F	10	2	MVI [[expr]++], A	
40	4	1	NOP	
41	9	3	AND reg[expr], expr	Z
42	10	3	AND reg[X+expr], expr	Z
43	9	3	OR reg[expr], expr	Z
44	10	3	OR reg[X+expr], expr	Z
45	9	3	XOR reg[expr], expr	Z
46	10	3	XOR reg[X+expr], expr	Z
47	8	3	TST [expr], expr	Z
48	9	3	TST [X+expr], expr	Z
49	9	3	TST reg[expr], expr	Z
4A	10	3	TST reg[X+expr], expr	Z
4B	5	1	SWAP A, X	Z
4C	7	2	SWAP A, [expr]	Z
4D	7	2	SWAP X, [expr]	
4E	5	1	SWAP A, SP	Z
4F	4	1	MOV X, SP	
50	4	2	MOV A, expr	Z
51	5	2	MOV A, [expr]	Z
52	6	2	MOV A, [X+expr]	Z
53	5	2	MOV [expr], A	
54	6	2	MOV [X+expr], A	
55	8	3	MOV [expr], expr	
56	9	3	MOV [X+expr], expr	
57	4	2	MOV X, expr	
58	6	2	MOV X, [expr]	
59	7	2	MOV X, [X+expr]	
5A	5	2	MOV [expr], X	
5B	4	1	MOV A, X	Z
5C	4	1	MOV X, A	
5D	6	2	MOV A, reg[expr]	Z
5E	7	2	MOV A, reg[X+expr]	Z
5F	10	3	MOV [expr], [expr]	
60	5	2	MOV reg[expr], A	
61	6	2	MOV reg[X+expr], A	
62	8	3	MOV reg[expr], expr	
63	9	3	MOV reg[X+expr], expr	
64	4	1	ASL A	C, Z
65	7	2	ASL [expr]	C, Z
66	8	2	ASL [X+expr]	C, Z
67	4	1	ASR A	C, Z
68	7	2	ASR [expr]	C, Z
69	8	2	ASR [X+expr]	C, Z
6A	4	1	RLC A	C, Z
6B	7	2	RLC [expr]	C, Z
6C	8	2	RLC [X+expr]	C, Z
6D	4	1	RRC A	C, Z
6E	7	2	RRC [expr]	C, Z
6F	8	2	RRC [X+expr]	C, Z
70	4	2	AND F, expr	C, Z
71	4	2	OR F, expr	C, Z
72	4	2	XOR F, expr	C, Z
73	4	1	CPL A	Z
74	4	1	INC A	C, Z
75	4	1	INC X	C, Z
76	7	2	INC [expr]	C, Z
77	8	2	INC [X+expr]	C, Z
78	4	1	DEC A	C, Z
79	4	1	DEC X	C, Z
7A	7	2	DEC [expr]	C, Z
7B	8	2	DEC [X+expr]	C, Z
7C	13	3	LCALL	
7D	7	3	LJMP	
7E	10	1	RETI	C, Z
7F	8	1	RET	
8x	5	2	JMP	
9x	11	2	CALL	
Ax	5	2	JZ	
Bx	5	2	JNZ	
Cx	5	2	JC	
Dx	5	2	JNC	
Ex	7	2	JACC	
Fx	13	2	INDEX	Z

Note 1 Interrupt acknowledge to Interrupt Vector table = 13 cycles.

Note 2 The number of cycles required by an instruction is increased by one for instructions that span 256 byte page boundaries in the Flash memory space.

Table 2-2. Instruction Set Summary Sorted Alphabetically by Mnemonic

Opcode Hex	Cycles	Bytes	Instruction Format	Flags	Opcode Hex	Cycles	Bytes	Instruction Format	Flags	Opcode Hex	Cycles	Bytes	Instruction Format	Flags
09	4	2	ADC A, expr	C, Z	76	7	2	INC [expr]	C, Z	20	5	1	POP X	
0A	6	2	ADC A, [expr]	C, Z	77	8	2	INC [X+expr]	C, Z	18	5	1	POP A	Z
0B	7	2	ADC A, [X+expr]	C, Z	Fx	13	2	INDEX	Z	10	4	1	PUSH X	
0C	7	2	ADC [expr], A	C, Z	Ex	7	2	JACC		08	4	1	PUSH A	
0D	8	2	ADC [X+expr], A	C, Z	Cx	5	2	JC		7E	10	1	RETI	C, Z
0E	9	3	ADC [expr], expr	C, Z	8x	5	2	JMP		7F	8	1	RET	
0F	10	3	ADC [X+expr], expr	C, Z	Dx	5	2	JNC		6A	4	1	RLC A	C, Z
01	4	2	ADD A, expr	C, Z	Bx	5	2	JNZ		6B	7	2	RLC [expr]	C, Z
02	6	2	ADD A, [expr]	C, Z	Ax	5	2	JZ		6C	8	2	RLC [X+expr]	C, Z
03	7	2	ADD A, [X+expr]	C, Z	7C	13	3	LCALL		28	11	1	ROMX	Z
04	7	2	ADD [expr], A	C, Z	7D	7	3	LJMP		6D	4	1	RRC A	C, Z
05	8	2	ADD [X+expr], A	C, Z	4F	4	1	MOV X, SP		6E	7	2	RRC [expr]	C, Z
06	9	3	ADD [expr], expr	C, Z	50	4	2	MOV A, expr	Z	6F	8	2	RRC [X+expr]	C, Z
07	10	3	ADD [X+expr], expr	C, Z	51	5	2	MOV A, [expr]	Z	19	4	2	SBB A, expr	C, Z
38	5	2	ADD SP, expr		52	6	2	MOV A, [X+expr]	Z	1A	6	2	SBB A, [expr]	C, Z
21	4	2	AND A, expr	Z	53	5	2	MOV [expr], A		1B	7	2	SBB A, [X+expr]	C, Z
22	6	2	AND A, [expr]	Z	54	6	2	MOV [X+expr], A		1C	7	2	SBB [expr], A	C, Z
23	7	2	AND A, [X+expr]	Z	55	8	3	MOV [expr], expr		1D	8	2	SBB [X+expr], A	C, Z
24	7	2	AND [expr], A	Z	56	9	3	MOV [X+expr], expr		1E	9	3	SBB [expr], expr	C, Z
25	8	2	AND [X+expr], A	Z	57	4	2	MOV X, expr		1F	10	3	SBB [X+expr], expr	C, Z
26	9	3	AND [expr], expr	Z	58	6	2	MOV X, [expr]		00	15	1	SSC	
27	10	3	AND [X+expr], expr	Z	59	7	2	MOV X, [X+expr]		11	4	2	SUB A, expr	C, Z
70	4	2	AND F, expr	C, Z	5A	5	2	MOV [expr], X		12	6	2	SUB A, [expr]	C, Z
41	9	3	AND reg[expr], expr	Z	5B	4	1	MOV A, X	Z	13	7	2	SUB A, [X+expr]	C, Z
42	10	3	AND reg[X+expr], expr	Z	5C	4	1	MOV X, A		14	7	2	SUB [expr], A	C, Z
64	4	1	ASL A	C, Z	5D	6	2	MOV A, reg[expr]	Z	15	8	2	SUB [X+expr], A	C, Z
65	7	2	ASL [expr]	C, Z	5E	7	2	MOV A, reg[X+expr]	Z	16	9	3	SUB [expr], expr	C, Z
66	8	2	ASL [X+expr]	C, Z	5F	10	3	MOV [expr], [expr]		17	10	3	SUB [X+expr], expr	C, Z
67	4	1	ASR A	C, Z	60	5	2	MOV reg[expr], A		4B	5	1	SWAP A, X	Z
68	7	2	ASR [expr]	C, Z	61	6	2	MOV reg[X+expr], A		4C	7	2	SWAP A, [expr]	Z
69	8	2	ASR [X+expr]	C, Z	62	8	3	MOV reg[expr], expr		4D	7	2	SWAP X, [expr]	
9x	11	2	CALL		63	9	3	MOV reg[X+expr], expr		4E	5	1	SWAP A, SP	Z
39	5	2	CMP A, expr	if (A=B) Z=1 if (A<B) C=1	3E	10	2	MVI A, [[expr]++]	Z	47	8	3	TST [expr], expr	Z
3A	7	2	CMP A, [expr]		3F	10	2	MVI [[expr]++], A		48	9	3	TST [X+expr], expr	Z
3B	8	2	CMP A, [X+expr]		40	4	1	NOP		49	9	3	TST reg[expr], expr	Z
3C	8	3	CMP [expr], expr		29	4	2	OR A, expr	Z	4A	10	3	TST reg[X+expr], expr	Z
3D	9	3	CMP [X+expr], expr		2A	6	2	OR A, [expr]	Z	72	4	2	XOR F, expr	C, Z
73	4	1	CPL A	Z	2B	7	2	OR A, [X+expr]	Z	31	4	2	XOR A, expr	Z
78	4	1	DEC A	C, Z	2C	7	2	OR [expr], A	Z	32	6	2	XOR A, [expr]	Z
79	4	1	DEC X	C, Z	2D	8	2	OR [X+expr], A	Z	33	7	2	XOR A, [X+expr]	Z
7A	7	2	DEC [expr]	C, Z	2E	9	3	OR [expr], expr	Z	34	7	2	XOR [expr], A	Z
7B	8	2	DEC [X+expr]	C, Z	2F	10	3	OR [X+expr], expr	Z	35	8	2	XOR [X+expr], A	Z
30	9	1	HALT		43	9	3	OR reg[expr], expr	Z	36	9	3	XOR [expr], expr	Z
74	4	1	INC A	C, Z	44	10	3	OR reg[X+expr], expr	Z	37	10	3	XOR [X+expr], expr	Z
75	4	1	INC X	C, Z	71	4	2	OR F, expr	C, Z	45	9	3	XOR reg[expr], expr	Z
										46	10	3	XOR reg[X+expr], expr	Z

Note 1 Interrupt acknowledge to Interrupt Vector table = 13 cycles.

Note 2 The number of cycles required by an instruction is increased by one for instructions that span 256 byte page boundaries in the Flash memory space.

2.5 Instruction Formats

The M8C has a total of seven instruction formats which use instruction lengths of one, two, and three bytes. All instruction bytes are fetched from the program memory (Flash), using an address and data bus that are independent from the address and data buses used for register and RAM access.

While examples of instructions are given in this section, refer to the *PSoC Designer Assembly Language User Guide* for detailed information on individual instructions.

2.5.1 One-Byte Instructions

Many instructions, such as some of the `MOV` instructions, have single-byte forms because they do not use an address or data as an operand. As shown in Table 2-3, one-byte instructions use an 8-bit opcode. The set of one-byte instructions can be divided into four categories, according to where their results are stored.

Table 2-3. One-Byte Instruction Format

Byte 0
8-Bit Opcode

The first category of one-byte instructions are those that do not update any registers or RAM. Only the one-byte `NOP` and `SSC` instructions fit this category. While the ***program counter*** is incremented as these instructions execute, they do not cause any other internal M8C registers to be updated, nor do these instructions directly affect the register space or the RAM address space. The `SSC` instruction causes SROM code to run, which modifies RAM and the M8C internal registers.

The second category has only the two `PUSH` instructions in it. The `PUSH` instructions are unique, because they are the only one-byte instructions that cause a RAM address to be modified. These instructions automatically increment the SP.

The third category has only the `HALT` instruction in it. The `HALT` instruction is unique, because it is the only one-byte instruction that causes a user register to be modified. The `HALT` instruction modifies user register space address FFh (CPU_SCR register).

The final category for one-byte instructions are those that cause updates of the internal M8C registers. This category holds the largest number of instructions: `ASL`, `ASR`, `CPL`, `DEC`, `INC`, `MOV`, `POP`, `RET`, `RETI`, `RLC`, `ROMX`, `RRC`, SWAP. These instructions can cause the A, X, and SP registers or SRAM to update.

2.5.2 Two-Byte Instructions

The majority of M8C instructions are two bytes in length. While these instructions can be divided into categories identical to the one-byte instructions, this would not provide a useful distinction between the three two-byte instruction formats that the M8C uses.

Table 2-4. Two-Byte Instruction Formats

Byte 0	Byte 1
4-Bit Opcode	12-Bit Relative Address
8-Bit Opcode	8-Bit Data
8-Bit Opcode	8-Bit Address

The first two-byte instruction format, shown in the first row of Table 2-4, is used by short jumps and calls: `CALL`, `JMP`, `JACC`, `INDEX`, `JC`, `JNC`, `JNZ`, `JZ`. This instruction format uses only four bits for the instruction opcode, leaving 12 bits to store the relative destination address in a two's-complement form. These instructions can change program execution to an address relative to the current address by -2048 or +2047.

The second two-byte instruction format, shown in the second row of Table 2-4, is used by instructions that employ the Source Immediate addressing ***mode*** (see "Source Immediate" on page 39). The destination for these instructions is an internal M8C register, while the source is a constant value. An example of this type of instruction would be `ADD A, 7`.

The third two-byte instruction format, shown in the third row of Table 2-4, is used by a wide range of instructions and addressing modes. The following is a list of the addressing modes that use this third two-byte instruction format:

- Source Direct (`ADD A, [7]`)
- Source Indexed (`ADD A, [X+7]`)
- Destination Direct (`ADD [7], A`)
- Destination Indexed (`ADD [X+7], A`)
- Source Indirect Post Increment (`MVI A, [7]`)
- Destination Indirect Post Increment (`MVI [7], A`)

For more information on addressing modes see "Addressing Modes" on page 39.

2.5.3 Three-Byte Instructions

The three-byte instruction formats are the second most prevalent instruction formats. These instructions need three bytes because they either move data between two addresses in the user-accessible address space (registers and RAM) or they hold 16-bit absolute addresses as the destination of a long jump or long call.

Table 2-5. Three-Byte Instruction Formats

Byte 0	Byte 1	Byte 2
8-Bit Opcode	16-Bit Address (MSB, LSB)	
8-Bit Opcode	8-Bit Address	8-Bit Data
8-Bit Opcode	8-Bit Address	8-Bit Address

The first instruction format, shown in the first row of Table 2-5, is used by the `LJMP` and `LCALL` instructions. These instructions change program execution unconditionally to an absolute address. The instructions use an 8-bit opcode, leaving room for a 16-bit destination address.

The second three-byte instruction format, shown in the second row of Table 2-5, is used by the following two addressing modes:

- Destination Direct Source Immediate (`ADD [7], 5`)
- Destination Indexed Source Immediate (`ADD [X+7], 5`)

The third three-byte instruction format, shown in the third row of Table 2-5, is for the Destination Direct Source Direct addressing mode, which is used by only one instruction. This instruction format uses an 8-bit opcode followed by two 8-bit addresses. The first address is the destination address in RAM, while the second address is the source address in RAM. The following is an example of this instruction:

```
MOV [7], [5]
```

2.6 Addressing Modes

The M8C has ten addressing modes. These modes are detailed and located on the following pages:

- "Source Immediate" on page 39.
- "Source Direct" on page 40.
- "Source Indexed" on page 40.
- "Destination Direct" on page 41.
- "Destination Indexed" on page 41.
- "Destination Direct Source Immediate" on page 41.
- "Destination Indexed Source Immediate" on page 42.
- "Destination Direct Source Direct" on page 42.
- "Source Indirect Post Increment" on page 43.
- "Destination Indirect Post Increment" on page 43.

2.6.1 Source Immediate

For these instructions, the source value is stored in operand 1 of the instruction. The result of these instructions is placed in either the M8C A, F, or X register as indicated by the instruction's opcode. All instructions using the Source Immediate addressing mode are two bytes in length.

Table 2-6. Source Immediate

Opcode	Operand 1
Instruction	Immediate Value

Source Immediate Examples:

Source Code	Machine Code	Comments
`ADD A, 7`	`01 07`	The immediate value 7 is added to the Accumulator. The result is placed in the Accumulator.
`MOV X, 8`	`57 08`	The immediate value 8 is moved to the X register.
`AND F, 9`	`70 09`	The immediate value 9 is logically AND'ed with the F register and the result is placed in the F register.

2.6.2 Source Direct

For these instructions, the source address is stored in operand 1 of the instruction. During instruction execution, the address is used to retrieve the source value from RAM or register address space. The result of these instructions is placed in either the M8C A or X register as indicated by the instruction's opcode. All instructions using the Source Direct addressing mode are two bytes in length.

Table 2-7. Source Direct

Opcode	Operand 1
Instruction	Source Address

Source Direct Examples:

Source Code	Machine Code	Comments
`ADD A, [7]`	`02 07`	The value in memory at address 7 is added to the Accumulator and the result is placed in the Accumulator.
`MOV A, REG[8]`	`5D 08`	The value in the register space at address 8 is moved to the Accumulator.

2.6.3 Source Indexed

For these instructions, the source offset from the X register is stored in operand 1 of the instruction. During instruction execution, the current X register value is added to the signed offset, to determine the address of the source value in RAM or register address space. The result of these instructions is placed in either the M8C A or X register as indicated by the instruction's opcode. All instructions using the Source Indexed addressing mode are two bytes in length.

Table 2-8. Source Indexed

Opcode	Operand 1
Instruction	Source Index

Source Indexed Examples:

Source Code	Machine Code	Comments
`ADD A, [X+7]`	`03 07`	The value in memory at address X+7 is added to the Accumulator. The result is placed in the Accumulator.
`MOV X, [X+8]`	`59 08`	The value in RAM at address X+8 is moved to the X register.

2.6.4 Destination Direct

For these instructions, the destination address is stored in the machine code of the instruction. The source for the operation is either the M8C A or X register as indicated by the instruction's opcode. All instructions using the Destination Direct addressing mode are two bytes in length.

Table 2-9. Destination Direct

Opcode	Operand 1
Instruction	Destination Address

Destination Direct Examples:

Source Code	Machine Code	Comments
`ADD [7], A`	`04 07`	The value in the Accumulator is added to memory at address 7. The result is placed in memory at address 7. The Accumulator is unchanged.
`MOV REG[8], A`	`60 08`	The Accumulator value is moved to register space at address 8. The Accumulator is unchanged.

2.6.5 Destination Indexed

For these instructions, the destination offset from the X register is stored in the machine code for the instruction. The source for the operation is either the M8C A register or an immediate value as indicated by the instruction's opcode. All instructions using the Destination Indexed addressing mode are two bytes in length.

Table 2-10. Destination Indexed

Opcode	Operand 1
Instruction	Destination Index

Destination Indexed Example:

Source Code	Machine Code	Comments
`ADD [X+7], A`	`05 07`	The value in memory at address X+7 is added to the Accumulator. The result is placed in memory at address X+7. The Accumulator is unchanged.

2.6.6 Destination Direct Source Immediate

For these instructions, the destination address is stored in operand 1 of the instruction. The source value is stored in operand 2 of the instruction. All instructions using the Destination Direct Source Immediate addressing mode are three bytes in length.

Table 2-11. Destination Direct Source Immediate

Opcode	Operand 1	Operand 2
Instruction	Destination Address	Immediate Value

Destination Direct Source Immediate Examples:

Source Code	Machine Code	Comments
`ADD [7], 5`	`06 07 05`	The value in memory at address 7 is added to the immediate value 5. The result is placed in memory at address 7.
`MOV REG[8], 6`	`62 08 06`	The immediate value 6 is moved to register space at address 8.

2.6.7 Destination Indexed Source Immediate

For these instructions, the destination offset from the X register is stored in operand 1 of the instruction. The source value is stored in operand 2 of the instruction. All instructions using the Destination Indexed Source Immediate addressing mode are three bytes in length.

Table 2-12. Destination Indexed Source Immediate

Opcode	Operand 1	Operand 2
Instruction	Destination Index	Immediate Value

Destination Indexed Source Immediate Examples:

Source Code	Machine Code	Comments
`ADD [X+7], 5`	`07 07 05`	The value in memory at address X+7 is added to the immediate value 5. The result is placed in memory at address X+7.
`MOV REG[X+8], 6`	`63 08 06`	The immediate value 6 is moved to the register space at address X+8.

2.6.8 Destination Direct Source Direct

Only one instruction uses this addressing mode. The destination address is stored in operand 1 of the instruction. The source address is stored in operand 2 of the instruction. The instruction using the Destination Direct Source Direct addressing mode is three bytes in length.

Table 2-13. Destination Direct Source Direct

Opcode	Operand 1	Operand 2
Instruction	Destination Address	Source Address

Destination Direct Source Direct Example:

Source Code	Machine Code	Comments
`MOV [7], [8]`	`5F 07 08`	The value in memory at address 8 is moved to memory at address 7.

2.6.9 Source Indirect Post Increment

Only one instruction uses this addressing mode. The source address stored in operand 1 is actually the address of a pointer. During instruction execution, the pointer's current value is read to determine the address in RAM where the source value is found. The pointer's value is incremented after the source value is read. For PSoC microcontrollers with more than 256 bytes of RAM, the Data Page Read (MVR_PP) register is used to determine which RAM page to use with the source address. Therefore, values from pages other than the current page can be retrieved without changing the Current Page Pointer (CUR_PP). The pointer is always read from the current RAM page. For information on the MVR_PP and CUR_PP registers, see the Register Details chapter on page 47. The instruction using the Source Indirect Post Increment addressing mode is two bytes in length.

Table 2-14. Source Indirect Post Increment

Opcode	Operand 1
Instruction	Source Address Pointer

Source Indirect Post Increment Example:

Source Code	**Machine Code**	**Comments**
`MVI     A, [8]`	`3E 08`	The value in memory at address 8 (the indirect address) points to a memory location in RAM. The value at the memory location, pointed to by the indirect address, is moved to the Accumulator. The indirect address, at address 8 in memory, is then incremented.

2.6.10 Destination Indirect Post Increment

Only one instruction uses this addressing mode. The destination address stored in operand 1 is actually the address of a pointer. During instruction execution, the pointer's current value is read to determine the destination address in RAM where the Accumulator's value is stored. The pointer's value is incremented after the value is written to the destination address. For PSoC microcontrollers with more than 256 bytes of RAM, the Data Page Write (MVW_PP) register is used to determine which RAM page to use with the destination address. Therefore, values can be stored in pages other than the current page without changing the Current Page Pointer (CUR_PP). The pointer is always read from the current RAM page. For information on the MVR_PP and CUR_PP registers, see the Register Details chapter on page 47. The instruction using the Destination Indirect Post Increment addressing mode is two bytes in length.

Table 2-15. Destination Indirect Post Increment

Opcode	Operand 1
Instruction	Destination Address Pointer

Destination Indirect Post Increment Example:

Source Code	**Machine Code**	**Comments**
`MVI     [8], A`	`3F 08`	The value in memory at address 8 (the indirect address) points to a memory location in RAM. The Accumulator value is moved to the memory location pointed to by the indirect address. The indirect address, at address 8 in memory, is then incremented.

2.7 Register Definitions

The following register is associated with the CPU Core (M8C). The register description has an associated register table showing the bit structure. The bits that are grayed out in the table are reserved bits and are not detailed in the register description that follows. Reserved bits should always be written with a value of ‘0’.

2.7.1 CPU_F Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,F7h	CPU_F	PgMode[1:0]			XIO		Carry	Zero	GIE	RL : 02

LEGEND
L The AND F, expr; OR F, expr; and XOR F, expr flag instructions can be used to modify this register.
x An “x” before the comma in the address field indicates that this register can be read or written to no matter what bank is used.

The M8C Flag Register (CPU_F) provides read access to the M8C flags.

Bits 7 and 6: PgMode[1:0]. PgMode determines how the CUR_PP, STK_PP, and IDX_PP registers are used in forming effective RAM addresses for Direct Address mode and Indexed Address mode operands. PgMode also determines whether the stack page is determined by the STK_PP or IDX_PP register.

Bit 4: XIO. The IO Bank Select bit, also known as the register bank select bit, is used to select the register bank that is active for a register read or write. This bit allows the PSoC device to have 512 8-bit registers and therefore, can be thought of as the ninth address bit for registers. The address space accessed when the XIO bit is set to ‘0’ is called the ***user space***, while the address space accessed when the XIO bit is set to ‘1’ is called the ***configuration space***.

Bit 2: Carry. The Carry flag bit is set or cleared in response to the result of several instructions. It can also be manipulated by the flag-logic opcodes (for example, OR F, 4). See the *PSoC Designer Assembly Guide User Manual* for more details.

Bit 1: Zero. The Zero flag bit is set or cleared in response to the result of several instructions. It can also be manipulated by the flag-logic opcodes (for example, OR F, 2). See the *PSoC Designer Assembly Guide User Manual* for more details.

Bit 0: GIE. The state of the Global Interrupt Enable bit determines whether interrupts (by way of the interrupt request (IRQ)) are recognized by the M8C. This bit is set or cleared by the user, using the flag-logic instructions (for example, OR F, 1). GIE is also cleared automatically when an interrupt is processed, after the flag byte has been stored on the stack, preventing nested interrupts. If desired, the bit can be set in an ***interrupt service routine (ISR)***.

For GIE=1, the M8C samples the IRQ input for each instruction. For GIE=0, the M8C ignores the IRQ.

For additional information, refer to the CPU_F register on page 120.

3. Supervisory ROM (SROM)

This chapter discusses the Supervisory ROM (SROM) functions and its associated registers. For a complete table of the SROM registers, refer to the “Summary Table of the Core Registers” on page 32. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

3.1 Architectural Description

The SROM holds code that is used to boot the PSoC device, calibrate circuitry, and perform Flash operations. The functions provided by the SROM are called from code stored in the Flash or by device programmers.

The SROM is used to boot the part and provide ***interface*** functions to the Flash banks. (Table 3-1 lists the SROM functions.) The SROM functions are accessed by executing the Supervisory System Call instruction (SSC) which has an opcode of 00h. Before executing the SSC, the M8C’s ***accumulator*** needs to load with the desired SROM function code from Table 3-1.

Attempting to access undefined functions causes a halt. The SROM functions execute code with calls; therefore, the functions require stack space. With the exception of Reset, all of the SROM functions have a ***parameter block*** in SRAM that you must configure before executing the SSC.

Table 3-2 lists all possible parameter block variables. The meaning of each ***parameter***, with regards to a specific SROM function, is described later in this chapter. Because the SSC instruction clears the CPU_F PgMode bits, all parameter block variable addresses are in SRAM Page 0. The CPU_F value is automatically restored at the end of the SROM function.

Note For PSoC devices with more than 256 bytes of SRAM (that is, more than 1 page of SRAM, see the table titled “PSoC Device SRAM Availability, on page 55”), the MVR_PP and the MVW_PP pointers are not disabled by clearing the CPU_F PgMode bits. Therefore, the POINTER parameter is interpreted as an address in the page indicated by the MVI page pointers, when the supervisory operation is called. This allows the data ***buffer*** used in the supervisory operation to be located in any SRAM page. (See the RAM Paging chapter on page 55 for more details regarding the MVR_PP and MVW_PP pointers.)

Table 3-1. List of SROM Functions

Function Code	Function Name	Stack Space Needed	Page
00h	SWBootReset	0	46
01h	ReadBlock	7	47
02h	WriteBlock	10	48
03h	EraseBlock	9	48
06h	TableRead	3	49
07h	CheckSum	3	50
08h	Calibrate0	4	50
09h	Calibrate1	3	50

Note ProtectBlock (described on page 49) and EraseAll (described on page 49) SROM functions are not listed in the table above because they depend upon external programming.

Two important variables that are used for all functions are KEY1 and KEY2. These variables are used to help discriminate between valid SSCs and inadvertent SSCs. KEY1 must always have a value of 3Ah, while KEY2 must have the same value as the stack pointer (SP) when the SROM function begins execution. This would be the SP value when the SSC opcode is executed, plus three. For all SROM functions except SWBootReset, if either of the keys do not match the expected values, the M8C halts. The SWBootReset function does not check the key values. It only checks to see if the accumulator’s value is 0x00. The following code example puts the correct value in KEY1 and KEY2. The code is preceded by a HALT, to force the program to jump directly into the setup code and not accidentally run into it.

```
        halt
SSCOP:  mov [KEY1], 3ah
        mov X, SP
        mov A, X
        add A, 3
        mov [KEY2], A
```

Table 3-2. SROM Function Variables

Variable Name	SRAM Address
KEY1 / RETURN CODE	0,F8h
KEY2	0,F9h
BLOCKID	0,FAh
POINTER	0,FBh
CLOCK	0,FCh
Reserved	0,FDh
DELAY	0,FEh
Reserved	0,FFh

3.1.1 Additional SROM Feature

The SROM has the following additional feature.

Return Codes: These aid in the determination of success or failure of a particular function. The return code is stored in KEY1's position in the parameter block. The CheckSum and TableRead functions do not have return codes because KEY1's position in the parameter block is used to return other data.

Table 3-3. SROM Return Code Descriptions

Return Code Value	Description
00h	Success
01h	Function not allowed due to level of protection on the block.
02h	Software reset without hardware reset.
03h	Fatal error, SROM halted.

Note Read, write, and erase operations may fail if the target block is read or write protected. Block protection levels are set during device programming and cannot be modified from code in the PSoC device.

3.1.2 SROM Function Descriptions

3.1.2.1 SWBootReset Function

The SROM function SWBootReset is responsible for transitioning the device from a reset state to running ***user*** code. See "System Resets" on page 313 for more information on what events cause the SWBootReset function to execute.

The SWBootReset function is executed whenever the SROM is entered with an M8C accumulator value of 00h; the SRAM parameter block is not used as an input to the function. This happens, by design, after a ***hardware*** reset, because the M8C's accumulator is reset to 00h or when user code executes the SSC instruction with an accumulator value of 00h.

If the ***checksum*** of the calibration data is valid, the SWBootReset function ends by setting the internal M8C registers (CPU_SP, CPU_PC, CPU_X, CPU_F, CPU_A) to 00h writing 00h to most SRAM addresses in SRAM Page 0 and then begins to execute user code at address 0000h. (See Table 3-4 and the following paragraphs for more information on which SRAM addresses are modified.) If the checksum is not valid, an internal reset is executed and the boot process starts over. If this condition occurs, the internal reset status bit (IRESS) is set in the CPU_SCR1 register.

Table 3-4 documents the value of all the SRAM addresses in Page 0 after a successful SWBootReset. A cell in the table with "xx" indicates that the SRAM address is not modified by the SWBootReset function. A hex value in a cell indicates that the address should always have the indicated value after a successful SWBootReset. A cell with a "??" in it indicates that the value, after a SWBootReset, is determined by the value of IRAMDIS bit in the CPU_SCR1 register. If IRAMDIS is not set, these addresses are initialized to 00h. If IRAMDIS is set, these addresses are not modified by a SWBootReset after a watchdog reset.

The IRAMDIS bit allows variables to be preserved even if a watchdog reset (WDR) occurs. The IRAMDIS bit is reset by all system resets except watchdog reset. Therefore, this bit is only useful for watchdog resets and not general resets.

Table 3-4. SRAM Map Post SWBootReset (00h)

Address	0	1	2	3	4	5	6	7
	8	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0x0_	0x00	0x00	0x00	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x1_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x2_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x3_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x4_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x5_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x6_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x7_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x8_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x9_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0xA_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0xB_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0xC_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0xD_	??	??	??	??	??	??	??	??
	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0xE_	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0xF_	0x00	0x00	0x00	0x00	0x00	0x00	??	??
	0x00 0x02	xx	0x00	0x00	0xn	xx	0x00	0x00

Address F8h is the return code byte for all SROM functions (except Checksum and TableRead); for this function, the only acceptable values are 00h and 02h. Address FCh is the fail count variable. After POR (Power on Reset) or WDR, or XRES (External Reset), the variable is initialized to 00h by the SROM. Each time the checksum fails, the fail count is incremented. Therefore, if it takes two passes through SWBootReset to get a good checksum, the fail count would be 01h.

3.1.2.2 *ReadBlock Function*

The ReadBlock function is used to read 64 contiguous bytes from Flash: a ***block***. The number of blocks in a device is the total number of bytes divided by 64. Refer to Table 3-5 to determine the amount of space in your PSoC device.

Table 3-5. Flash Memory Organization

PSoC Device	Amount of Flash	Amount of SRAM	Number of Blocks per Bank	Number of Banks
CY8C24633	8 KB	256 Bytes	128	2
CY8C23433	8 KB	256 Bytes	128	2
CY8C23533	8 KB	256 Bytes	128	2
CY8C24533	8 KB	256 Bytes	128	2
CY8C24x23A	4 KB	256 Bytes	64	1

The first thing the ReadBlock function does is check the protection bits to determine if the desired BLOCKID is readable. If read protection is turned on, the ReadBlock function exits setting the accumulator and KEY2 back to 00h. KEY1 has a value of 01h, indicating a read failure.

If read protection is not enabled, the function reads 64 bytes from the Flash using a ROMX instruction and stores the results in SRAM using an MVI instruction. The 64 bytes are stored in SRAM, beginning at the address indicated by the value of the POINTER parameter. When the ReadBlock completes successfully, the accumulator, KEY1, and KEY2 all have a value of 00h.

If the PSoC device has more than one bank of Flash, the bank value in the FLS_PR1 register must be set prior to executing the SSC instruction. Refer to Table 3-5.

Note MVI [expr], A is used to store the Flash block contents in SRAM; thus, the MVW_PP register can be set to indicate which SRAM pages receive the data.

Table 3-6. ReadBlock Parameters (01h)

Name	Address	Type	Description
MVW_PP	0,D5h	Register	MVI write page pointer register.
KEY1	0,F8h	RAM	3Ah.
KEY2	0,F9h	RAM	Stack Pointer value+3, when SSC is executed.
BLOCKID	0,FAh	RAM	Flash block number.
POINTER	0,FBh	RAM	Addresses in SRAM where returned data should be stored.
FLS_PR1	1,FAh	Register	Flash bank number.

3.1.2.3 WriteBlock Function

The WriteBlock function is used to store data in the Flash. Data is moved 64 bytes at a time from SRAM to Flash using this function. Before a write can be performed, either an EraseAll or an EraseBlock must be completed successfully.

The first thing the WriteBlock function does is check the protection bits and determine if the desired BLOCKID is writeable. If write protection is turned on, the WriteBlock function exits, setting the accumulator and KEY2 back to 00h. KEY1 has a value of 01h, indicating a write failure. Write protection is set when the PSoC device is programmed externally and cannot be changed through the SSC function.

The BLOCKID of the ***Flash block***, where the data is stored, must be determined and stored at SRAM address FAh. For valid BLOCKID values, refer to Table 3-5.

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

The SRAM address, of the first of the 64 bytes to be stored in Flash, must be indicated using the POINTER variable in the parameter block (SRAM address FBh).

Finally, the CLOCK and DELAY value must be set correctly. The CLOCK value determines the length of the write ***pulse*** that is used to store the data in the Flash. The CLOCK and DELAY values are dependent on the CPU speed and must be set correctly. Refer to “Clocking” on page 53 for additional information.

If the PSoC device you are using has more than one bank of Flash, the bank value in the FLS_PR1 register must be set prior to executing the SSC instruction. Refer to Table 3-5.

Table 3-7. WriteBlock Parameters (02h)

Name	Address	Type	Description
MVR_PP	0,D4h	Register	MVI read page pointer register.
KEY1	0,F8h	RAM	3Ah.
KEY2	0,F9h	RAM	Stack Pointer value+3, when SSC is executed.
BLOCKID	0,FAh	RAM	Flash block number.
POINTER	0,FBh	RAM	First of 64 addresses in SRAM, where the data to be stored in Flash is located prior to calling WriteBlock.
CLOCK	0,FCh	RAM	Clock divider used to set the write pulse width.
DELAY	0,FEh	RAM	For a CPU speed of 12 MHz, set to 56h.
FLS_PR1	1,FAh	Register	Flash bank number.

3.1.2.4 EraseBlock Function

The EraseBlock function is used to erase a block of 64 contiguous bytes in Flash.

The first thing the EraseBlock function does is check the protection bits and determine if the desired BLOCKID is writeable. If write protection is turned on, the EraseBlock function exits, setting the accumulator and KEY2 back to 00h. KEY1 has a value of 01h, indicating a write failure.

To set up the parameter block for the EraseBlock function, correct key values must be stored in KEY1 and KEY2. The block number to be erased must be stored in the BLOCKID variable, and the CLOCK and DELAY values must be set based on the current CPU speed. For more information on setting the CLOCK and DELAY values, see “Clocking” on page 53.

If the PSoC device you are using has more than one bank of Flash, the bank value in the FLS_PR1 register must be set prior to executing the SSC instruction. Refer to Table 3-5.

Table 3-8. EraseBlock Parameters (03h)

Name	Address	Type	Description
KEY1	0,F8h	RAM	3Ah
KEY2	0,F9h	RAM	Stack Pointer value+3, when SSC is executed.
BLOCKID	0,FAh	RAM	Flash block number.
CLOCK	0,FCh	RAM	Clock divider used to set the erase pulse width.
DELAY	0,FEh	RAM	For a CPU speed of 12 MHz, set to 56h.
FLS_PR1	1,FAh	Register	Flash bank number.

3.1.2.5 ProtectBlock Function

The PSoC devices offer Flash protection on a block-by-block basis. Table 3-9 lists the protection modes available. In the table, ER and EW are used to indicate the ability to perform external reads and writes (that is, by an external programmer). For internal writes, IW is used. Internal reading is always permitted by way of the ROMX instruction. The ability to read by way of the SROM ReadBlock function is indicated by SR.

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

Table 3-9. ProtectBlock Modes

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

3.1.2.6 TableRead Function

The TableRead function gives the user access to part-specific data stored in the Flash during manufacturing. The Flash for these tables is separate from the program Flash and is not directly accessible.

One of the uses of the SROM TableRead function is to retrieve the values needed to optimize Flash programming for temperature. More information about how to use these values may be found in the section titled "Clocking" on page 53.

Table 3-10. TableRead Parameters (06h)

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

3.1.2.7 EraseAll Function

The EraseAll function performs a series of steps that destroys the user data in the Flash banks and resets the protection block in each Flash bank to all zeros (the unprotected state). This function may only be executed by an external programmer. If EraseAll is executed from code, the M8C halts without touching the Flash or protections.

Table 3-11. Flash Tables with Assigned Values in Flash Bank 0

	F8h	F9h	FAh	FBh	FCh	FDh	FEh	FFh
Table 0	Silicon ID							
Table 1	Voltage Reference Trim for 3.3V reg[1,EA]	IMO Trim for 3.3V reg[1,E8]	Room Temperature Calibration for 3.3V	Hot Temperature Calibration for 3.3V	Voltage Reference Trim for 5V reg[1,EA]	IMO Trim for 5V reg[1,E8]	Room Temperature Calibration for 5V	Hot Temperature Calibration for 5V
Table 2	Voltage Reference Trim for 2.7V reg[1,EA]	IMO Slow Trim 12 MHz Vdd = 2.7V	Room Temperature Calibration for 2.7V *	Hot Temperature Calibration for 2.7V *	IMO Slow Trim 6 MHz Vdd = 3.3V	IMO Slow Trim 6 MHz Vdd = 2.7V	IMO Slow Trim 6 MHz Vdd = 5.0V	
Table 3	M (cold)	B (cold)	Mult (cold)	M (hot)	B (hot)	Mult (hot)	00h	01h

* CY8C24x94 and CY7C64215 Table 2: FAh = IMO Trim 2 for 3.3V, FBh = IMO Trim 2 for 5V.

3.1.2.8 Checksum Function

The Checksum function calculates a 16-bit checksum over a user specifiable number of blocks, within a single ***Flash bank*** starting at block zero. The BLOCKID parameter is used to pass in the number of blocks to checksum. A BLOCKID value of '1' calculates the checksum of only block 0, a BLOCKID of '2' calculates the checksum of block 0 and block 1, and so on. A BLOCKID value of '0' calculates the checksum of the entire flash bank. Note that if the BLOCKID is greater than the number of blocks that the device has in a flash bank, the function calculates the checksum for the entire flash bank and repeats the checksum process again from block 0 in that flash bank. For example, for the CY8C24533 device, if the BLOCKID is equal to 150, the function calculates checksum for block 0 to block 127 and again for block 0 to block 21.

The 16-bit checksum is returned in KEY1 and KEY2. The parameter KEY1 holds the lower 8 bits of the checksum and the parameter KEY2 holds the upper 8 bits of the checksum. For devices with multiple Flash banks, the checksum function must be called once for each Flash bank. The SROM Checksum function operates on the Flash bank indicated by the Bank bit in the FLS_PR1 register.

Table 3-12. Checksum Parameters (07h)

Name	Address	Type	Description
KEY1	0,F8h	RAM	3Ah.
KEY2	0,F9h	RAM	Stack Pointer value+3, when SSC is executed.
BLOCKID	0,FAh	RAM	Number of Flash blocks to calculate checksum on.
FLS_PR1	1,FAh	Register	Flash bank number.

3.1.2.9 Calibrate0 Function

The Calibrate0 function transfers the calibration values stored in a special area of the Flash to their appropriate registers. This function may be executed at any time to set all calibration values back to their 5V values. However, it should not be necessary to call this function. This function is simply documented for completeness. 3.3V calibration values are accessed by way of the TableRead function, which is described in the section titled “TableRead Function” on page 49.

Table 3-13. Calibrate0 Parameters (08h)

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

3.1.2.10 Calibrate1 Function

While the Calibrate1 function is a completely separate function from Calibrate0, they perform the same function, which is to transfer the calibration values stored in a special area of the Flash to their appropriate registers. What is unique about Calibrate1 is that it calculates a checksum of the calibration data and, if that checksum is determined to be invalid, Calibrate1 causes a ***hardware reset*** by generating an internal reset. If this occurs, it is indicated by setting the Internal Reset Status bit (IRESS) in the CPU_SCR1 register.

The Calibrate1 function uses SRAM to calculate a checksum of the calibration data. The POINTER value is used to indicate the address of a 30-byte buffer used by this function. When the function completes, the 30 bytes is set to 00h.

An MVI A, [expr] and an MVI [expr], A instruction are used to move data between SRAM and Flash. Therefore, the MVI write pointer (MVW_PP) and the MVI read pointer (MVR_PP) must be specified to the same SRAM page to control the page of RAM used for the operations.

Calibrate1 was created as a sub-function of SWBootReset and the Calibrate1 function code was added to provide ***direct access***. For more information on how Calibrate1 works, see the SWBootReset section.

This function may be executed at any time to set all calibration values back to their 5V values. However, it should not be necessary to call this function. This function is simply documented for completeness. This function has no argument to select between 5V and 3.3V calibration values; therefore, it always defaults to 5V values. 3.3V calibration values are accessed by way of the TableRead function, which is described in the section titled “TableRead Function” on page 49.

Table 3-14. Calibrate1 Parameters (09h)

Name	Address	Type	Description
KEY1	0,F8h	RAM	3Ah
KEY2	0,F9h	RAM	Stack Pointer value+3, when SSC is executed.
POINTER	0,FBh	RAM	First of 30 SRAM addresses used by this function.
MVR_PP	0,D4h	Register	MVI write page pointer.
MVW_PP	0,D5h	Register	MVI read page pointer.

3.2 Register Definitions

The following registers are associated with the Supervisory ROM (SROM) and are listed in address order. The register descriptions have an associated register table showing the bit structure for that register. The bits in the tables that are grayed out are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of ‘0’. For a complete table of SROM registers, refer to the “Summary Table of the Core Registers” on page 32.

3.2.1 CPU_SCR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW *	ECO EX *		IRAMDIS	# : 00

LEGEND

x An “x” before the comma in the address field indicates that this register can be read or written to no matter what bank is used.

# Access is bit specific. Refer to the Register Details chapter on page 47 for additional information.

* Bits 3 and 2 (ECO EXW and ECO EX, respectively) cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

The System Status and Control Register 1 (CPU_SCR1) is used to convey the status and control of events related to internal resets and watchdog reset.

Bit 7: IRESS. The Internal Reset Status bit is a read only bit that can be used to determine if the booting process occurred more than once.

When this bit is set, it indicates that the SROM SWBootReset code was executed more than once. If this bit is not set, the SWBootReset was executed only once. In either case, the SWBootReset code does not allow execution from code stored in Flash until the M8C Core is in a safe operating mode with respect to supply voltage and Flash operation. There is no need for concern when this bit is set. It is provided for systems that may be sensitive to boot time, so that they can determine if the normal one-pass boot time was exceeded.

Bit 4: SLIMO. When set, the Slow IMO bit allows the active power dissipation of the PSoC device to be reduced by slowing down the IMO from 24 MHz to 6 MHz. The IMO trim value must also be changed when SLIMO is set. When not in external clocking mode, the IMO is the source for SYSCLK; therefore, when the speed of the IMO changes, so does SYSCLK.

Bit 3: ECO EXW. The ECO Exists Written bit is used as a status bit to indicate that the ECO EX bit has been previously written to. It is read only. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

Bit 2: ECO EX. The ECO Exists bit serves as a flag to the hardware, to indicate that an external crystal ***oscillator*** exists in the system. Just after boot, it may be written only once to a value of ‘1’ (crystal exists) or ‘0’ (crystal does not exist). If the bit is ‘0’, a switch-over to the ECO is locked out by hardware. If the bit is ‘1’, hardware allows the firmware to freely switch between the ECO and ILO. It should be written as early as possible after a ***Power On Reset (POR)*** or ***External Reset (XRES)*** event, where it is assumed that program execution integrity is high. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

Bit 0: IRAMDIS. The Initialize RAM Disable bit is a control bit that is readable and writeable. The ***default value*** for this bit is ‘0’, which indicates that the maximum amount of SRAM should be initialized on watchdog reset to a value of 00h. When the bit is ‘1’, the minimum amount of SRAM is initialized after a watchdog reset. For more information on this bit, see the “SROM Function Descriptions” on page 46.

For additional information, refer to the CPU_SCR1 register on page 121.

3.2.2 FLS_PR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,FAh	FLS_PR1							Bank[1:0]		RW : 00

The Flash Program Register 1 (FLS_PR1) is used to specify which Flash bank should be used for SROM operations.

Note This register has no effect on products with one Flash bank. Refer to the table titled "Flash Memory Organization" on page 47 to determine the number of Flash banks in PSoC devices.

Bits 1 and 0: Bank[1:0]. The Bank bits in this register indicate which Flash bank the SROM Flash functions should operate on. The default value for the Bank bit is zero. Flash bank 0 holds up to the first 8K of user code, as well as the cal table. The optional Flash banks 1, 2, and 3 hold additional user code.

For additional information, refer to the FLS_PR1 register on page 160.

3.3 Clocking

Successful programming and erase operations, on the Flash, require that the CLOCK and DELAY parameters be set correctly. To determine the proper value for the DELAY parameter only, the CPU speed must be considered. However, three factors should be used to determine the proper value for CLOCK: operating temperature, CPU speed, and characteristics of the individual device. Equations and additional information on calculating the DELAY and CLOCK values follow.

3.3.1 DELAY Parameter

To determine the proper value for the DELAY parameter, the CPU speed during the Flash operation must be considered. Equation 1 displays the equation for calculating DELAY based on a CPU speed value. In this equation the units for CPU are hertz (Hz).

$$DELAY = \frac{100 \times 10^{-6} \cdot CPU - 80}{13},$$
$$3MHz \le CPU \le 12MHz$$

Equation 1

Equation 2 shows the calculation of the DELAY value for a CPU speed of 12 MHz. The numerical result of this calculation should be rounded to the nearest whole number. In the case of a 12 MHz CPU speed, the correct value for DELAY is 86 (0x56).

$$DELAY = \frac{100 \times 10^{-6} \cdot 12 \times 10^{6} - 80}{13}$$

Equation 2

3.3.2 CLOCK Parameter

The CLOCK parameter must be calculated using different equations for erase and write operations. The erase value for CLOCK must be calculated first. In Equation 3, the erase CLOCK value is indicated by a subscript E after the word CLOCK and the write CLOCK value is indicated by a subscript W after the word CLOCK.

Before either CLOCK value can be calculated, the values for M, B, and Mult must be determined. These are device specific values that are stored in the Flash table 3 and are accessed by way of the TableRead SROM function (see the “TableRead Function” on page 49). If the operating temperature is at or below 0°C, the cold values should be used. For operating temperatures at or above 0°C, the hot values should be used. See Table 3-11 for more information.

Equations for calculating the correct value of CLOCK for write operations are first introduced with the assumption that the CPU speed is 12 MHz.

The equation for calculating the CLOCK value for an erase Flash operation is shown in Equation 3. In this equation the T has units of °C.

$$CLOCK_E = B - \frac{2M \cdot T}{256}$$

Equation 3

Using the correct values for B, M, and T, in the equation above, is required to achieve the endurance specifications of the Flash. However, for device programmers, where this calculation may be difficult to perform, the equation can be simplified by setting T to 0°C and using the hot value for B and M. This simplification is acceptable only if the total number of erase write cycles are kept to less than 10 and the operation is performed near room temperature. When T is set to 0, Equation 3 simplifies to the following.

$$CLOCK_E = B$$

Equation 4

Once a value for the erase CLOCK value has been determined, the write CLOCK value can be calculated. The equation to calculate the CLOCK value for a write is as follows.

$$CLOCK_W = \frac{CLOCK_E \cdot Mult}{64}$$

Equation 5

In the equation above, the correct value for Mult must be determined, based on temperature, in the same way that the B and M values were determined for Equation 3.

4. RAM Paging

This chapter explains the PSoC device’s use of RAM Paging and its associated registers. For a complete table of the RAM Paging registers, refer to the “Summary Table of the Core Registers” on page 32. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

4.1 Architectural Description

The M8C is an 8-bit CPU with an 8-bit address bus. The 8-bit memory address bus allows the M8C to access up to 256 bytes of SRAM, to increase the amount of available SRAM and preserve the M8C ***assembly*** language.

Table 4-1. PSoC Device SRAM Availability

PSoC Device	Amount of SRAM	Number of Pages
CY8C24x23A	256 Bytes	1 Page
CY8C24533	256 Bytes	1 Page
CY8C23533	256 Bytes	1 Page
CY8C23433	256 Bytes	1 Page
CY8C24633	256 Bytes	1 Page

The memory paging architecture consists of five areas:

- Stack Operations
- Interrupts
- MVI Instructions
- Current Page Pointer
- Indexed Memory Page Pointer

The first three of these areas have no dependency on the CPU_F register's PgMode bits and are covered in the next subsections after Basic Paging. The function of the last two depend on the CPU_F PgMode bits and are covered last.

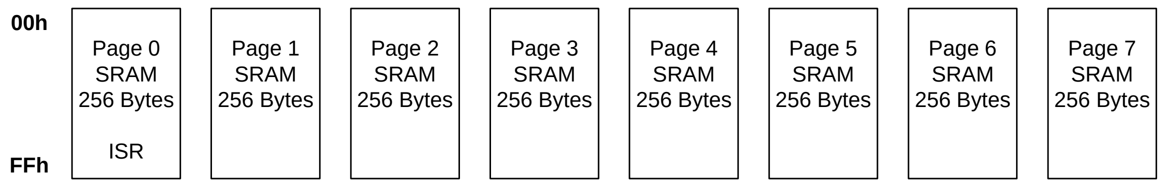
4.1.1 Basic Paging

The M8C is an 8-bit CPU with an 8-bit memory address bus. The memory address bus allows the M8C to access up to 256 bytes of SRAM. To increase the amount of SRAM, the M8C accesses memory page bits. The memory page bits are located in the CUR_PP register and allow for selection of one of eight SRAM pages. In addition to setting the page bits, Page mode must be enabled by setting the CPU_F[7] bit. If Page mode is not enabled, the page bits are ignored and all non-stack memory access is directed to Page 0.

Once Page mode is enabled and the page bits are set, all instructions that operate on memory access the SRAM page indicated by the page bits. The exceptions to this are the instructions that operate on the stack and the MVI instructions: PUSH, POP, LCALL, RETI, RET, CALL, and MVI.

See the description of Stack Operations and MVI Instructions below for a more detailed discussion.

Figure 4-1. Data Memory Organization



4.1.2 Stack Operations

As mentioned previously, the paging architecture's reset state puts the PSoC in a mode that is identical to that of a 256 byte PSoC device. Therefore, upon rest, all memory accesses are to Page 0. The SRAM page that stack operations uses is determined by the value of the three least significant bits of the Stack Page Pointer register (STK_PP). Stack operations have no dependency on the PgMode bits in the CPU_F register. Stack operations are those that use the Stack Pointer (SP) to calculate their affected address. Refer to the *PSoC Designer Assembly Language User Guide* for more information on all M8C instructions.

Stack memory accesses must be treated as a special case. If they are not, the stack could be fragmented across several pages. To prevent the stack from becoming fragmented, all instructions that operate on the stack automatically use the page indicated by the STK_PP register. Therefore, if a CALL is encountered in the program, the PSoC device automatically pushes the program counter onto the stack page indicated by STK_PP. Once the program counter is pushed, the SRAM paging mode automatically switches back to the precall mode. All other stack operations, such as RET and POP, follow the same rule as CALL. The stack is confined to a single SRAM page and the SP wraps from 00h to FFh and FFh to 00h. The user code must ensure that the stack is not damaged due to stack wrapping.

Because the value of the STK_PP register can be changed at any time, it is theoretically possible to manage the stack in such a way as to allow it to grow beyond one SRAM page or manage multiple stacks. However, the only supported use of the STK_PP register is when its value is set prior to the first stack operation and not changed again.

4.1.3 Interrupts

Interrupts, in a multi-page SRAM PSoC device, operate the same as interrupts in a 256 byte PSoC device. However, because the CPU_F register is automatically set to 0x00 on an interrupt and because of the non-linear nature of interrupts in a system, other parts of the PSoC memory paging architecture can be affected.

Interrupts are an abrupt change in program flow. If no special action is taken on interrupts by the PSoC device, the ***interrupt service routine (ISR)*** could be thrown into any SRAM page. To prevent this problem, the special addressing modes for all memory accesses, except for stack and MVI, are disabled when an ISR is entered. The special addressing modes are disabled when the CUP_F register is cleared. At the end of the ISR, the previous SRAM addressing mode is restored when the CPU_F register value is restored by the RETI instruction.

Therefore, all interrupt service ***routine*** code starts execution in SRAM Page 0. If it is necessary for the ISR to change to another SRAM page, it can be accomplished by changing the values of the CPU_F[7:6] bits to enable the special SRAM addressing modes. However, any change made to the CUR_PP, IDX_PP, or STK_PP registers persist after the ISR returns. Therefore, the ISR should save the current value of any paging register it modifies and restore its value before the ISR returns.

4.1.4 MVI Instructions

MVI instructions use data page pointers of their own (MVR_PP and MVW_PP). This allows a data buffer to be located away from other program variables, but accessible without changing the Current Page Pointer register (CUR_PP).

An MVI instruction performs three memory operations. Both forms of the MVI instruction access an address in SRAM that holds the data pointer (a memory read 1st access), incrementing that value and then storing it back in SRAM (a memory write 2nd access). This pointer value must reside in the current page, just as all other non-stack and non-indexed operations on memory must. However, the third memory operation uses the MVx_PP register. This third memory access can be either a read or a write, depending on which MVI instruction is used. The MVR_PP pointer is used for the MVI instruction that moves data into the accumulator. The MVW_PP pointer is used for the MVI instruction that moves data from the accumulator into SRAM. The MVI pointers are always enabled, regardless of the state of the Flag register page bits (CPU_F register).

4.1.5 Current Page Pointer

The Current Page Pointer is used to determine which SRAM page should be used for all memory accesses. Normal memory accesses are those not covered by other pointers including all non-stack, non-MVI, and non-indexed memory access instructions. The normal memory access instructions have the SRAM page they operate on determined by the value of the CUR_PP register. By default, the CUR_PP register has no affect on the SRAM page used for normal memory access, because all normal memory access is forced to SRAM Page 0.

The upper bit of the PgMode bits in the CPU_F register determines whether or not the CUR_PP register affects normal memory access. When the upper bit of the PgMode bits is set to ’0’, all normal memory access is forced to SRAM Page 0. This mode is automatically enabled when an ISR is entered. This is because, before the ISR is entered, the M8C pushes the current value of the CPU_F register onto the stack and then clears the CPU_F register. Therefore, by default, any normal memory access in an ISR is guaranteed to occur in SRAM Page 0.

When the RETI instruction is executed, to end the ISR, the previous value of the CPU_F register is restored, restoring the previous page mode. Note that this ISR behavior is the default and that the PgMode bits in the CPU_F register can be changed while in an ISR. If the PgMode bits are changed while in an ISR, the pre-ISR value is still restored by the RETI; but if the CUR_PP register is changed in the ISR, the ISR is also required to restore the value before executing the RETI instruction.

When the upper bit of the PgMode bits is set to ’1’, all normal memory access is forced to the SRAM page indicated by the value of the CUR_PP register. Table 4-2 gives a summary of the PgMode bit values and the corresponding Memory Paging mode.

4.1.6 Index Memory Page Pointer

The source indexed and destination indexed addressing modes to SRAM are treated as a unique addressing mode in a PSoC device, with more than one page of SRAM. An example of an indexed addressing mode is the MOV A, [X+expr] instruction. Note that register access also has indexed addressing; however, those instructions are not affected by the SRAM paging architecture.

Important Note If you are not using assembly to program a PSoC device, be aware that the ***compiler*** writer may restrict the use of some memory paging modes. Review the conventions in your compiler’s user guide for more information on restrictions or conventions associated with memory paging modes.

Indexed SRAM accesses operate in one of three modes:

- Index memory access modes are forced to SRAM Page 0.
- Index memory access modes are directed to the SRAM page indicated by the value in the STK_PP register.
- Index memory access is forced to the SRAM page indicated by the value in the IDX_PP register.

The mode is determined by the value of the PgMode bits in the CPU_F register. However, the final SRAM page that is used also requires setting either the Stack Page Pointer (STK_PP) register or the Index Page Pointer (IDX_PP) register. The table below shows the three indexed memory access modes. The third column of the table is provided for reference only.

Table 4-2. CPU_F PgMode Bit Modes

CPU_F PgMode BIts	Current SRAM Page	Indexed SRAM Page	Typical Use
00b	0	0	ISR*
01b	0	STK_PP	ISR with variables on stack
10b	CUR_PP	IDX_PP	
11b	CUR_PP	STK_PP	

* Mode used by SROM functions initiated by the SSC instruction.

After reset, the PgMode bits are set to 00b. In this mode, index memory accesses are forced to SRAM Page 0, just as they would be in a PSoC device with only 256 bytes of SRAM. This mode is also automatically enabled when an interrupt occurs in a PSoC device and is therefore considered the default ISR mode. This is because before the ISR is entered, the M8C pushes the current value of the CPU_F register on to the stack and then clears the CPU_F register. Therefore, by default, any indexed memory access in an ISR is guaranteed to occur in SRAM Page 0. When the RETI instruction is executed to end the ISR, the previous value of the CPU_F register is restored and the previous page mode is then also restored. Note that this ISR behavior is the default and that the PgMode bits in the CPU_F register may be changed while in an ISR. If the PgMode bits are changed while in an ISR, the pre-ISR value is still restored by the RETI; but if the STK_PP or IDX_PP registers are changed in the ISR, the ISR is also required to restore the values before executing the RETI instruction.

The most likely PgMode bit change, while in an ISR, is from the default value of 00b to 01b. In the 01b mode, indexed memory access is directed to the SRAM page indicated by the value of the STK_PP register. By using the PgMode, the value of the STK_PP register is not required to be modified. The STK_PP register is the register that determines which SRAM page the stack is located on. The 01b paging mode is intended to provide easy access to the stack, while in an ISR, by setting CPU_X register (just X in instruction format) equal to the value of SP using the MOV X, SP instruction.

The two previous paragraphs covered two of the three indexed memory access modes: STK_PP and forced to SRAM Page 0. Note, as shown in Table 4-2, that the STK_PP mode for indexed memory access is available under two PgMode settings. The 01b mode is intended for ISR use and the 11b mode is intended for non-ISR use. The third indexed memory access mode requires the PgMode bits to be set to 10b. In this mode indexed memory access is forced to the SRAM page indicated by the value of the IDX_PP register.

4.2 Register Definitions

The following registers are associated with RAM Paging and are listed in address order. The register descriptions have an associated register table showing the bit structure for that register. The bits in the tables that are grayed out are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of ‘0’. For a complete table of RAM Paging registers, refer to the “Summary Table of the Core Registers” on page 32.

4.2.1 TMP_DRx Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,6xh	TMP_DRx	Data[7:0]								RW : 00

LEGEND

x An ‘x’ before the comma in the address field indicates that this register can be read or written to no matter what bank is used. An “x” after the comma in the address field indicates that there are multiple instances of the register.

The Temporary Data Registers (TMP_DR0, TMP_DR1, TMP_DR2, and TMP_DR3) are used to enhance the performance in multiple SRAM page PSoC devices.

These registers have no pre-defined function (for example, the compiler and hardware do not use these registers) and exist for the user to use as desired.

Bits 7 to 0: Data[7:0]. Due to the paged SRAM architecture of PSoC devices with more than 256 bytes of SRAM, a value in SRAM may not always be accessible without first changing the current page.

The TMP_DRx registers are readable and writable registers that are provided to improve the performance of multiple SRAM page PSoC devices, by supplying some register space for data that is always accessible.

For expanded listing of TMP_DRx registers, refer to “Summary Table of the Core Registers” on page 32. For additional information, refer to TMP_DRx register on page 72.

4.2.2 CPU_F Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,F7h	CPU_F	PgMode[1:0]		▨	XIO	▨	Carry	Zero	GIE	RL : 02

LEGEND

L	The AND F, expr; OR F, expr; and XOR F, expr flag instructions can be used to modify this register.
x	An ‘x’ before the comma in the address field indicates that this register can be read or written to no matter what bank is used.

The M8C Flag Register (CPU_F) provides read access to the M8C flags.

Bits 7 and 6: PgMode[1:0]. PgMode determines how the CUR_PP and IDX_PP registers form effective RAM addresses for Direct Address mode and Indexed Address mode operands.

Bit 4: XIO. The IO Bank Select bit, also know as the register bank select bit, is used to select the register bank that is active for a register read or write. This bit allows the PSoC device to have 512 8-bit registers and, therefore, can be thought of as the ninth address bit for registers. The address space accessed when the XIO bit is set to ‘0’ is called the ***user space***, while the address space accessed when the XIO bit is set to ‘1’ is called the ***configuration space***.

Bit 2: Carry. The Carry Flag bit is set or cleared in response to the result of several instructions. It can also be manipulated by the flag-logic opcodes (for example, OR F, 4). See the *PSoC Designer Assembly Guide User Manual* for more details.

Bit 1: Zero. The Zero Flag bit is set or cleared in response to the result of several instructions. It can also be manipulated by the flag-logic opcodes (for example, OR F, 2). See the *PSoC Designer Assembly Guide User Manual* for more details.

Bit 0: GIE. The state of the Global Interrupt Enable bit determines whether interrupts (by way of the IRQ) are recognized by the M8C. This bit is set or cleared by the user, using the flag-logic instructions (for example, OR F, 1). GIE is also cleared automatically by the M8C upon entering the ISR, after the flag byte has been stored on the stack, preventing nested interrupts. Note that the bit can be set in an ISR if desired. For GIE=1, the M8C samples the IRQ input for each instruction. For GIE=0, the M8C ignores the IRQ.

For additional information, refer to the CPU_F register on page 120.

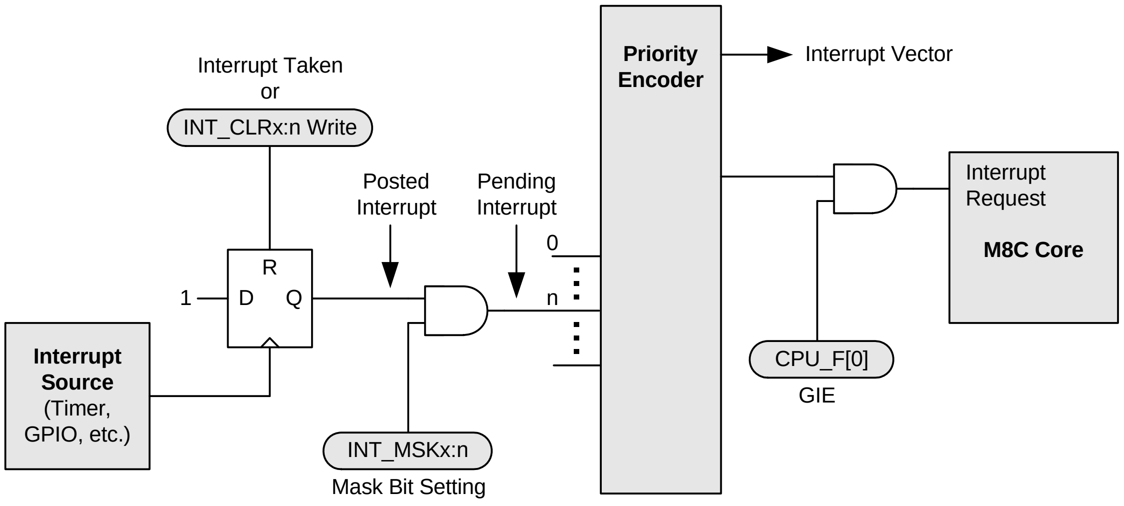
5. Interrupt Controller

This chapter presents the Interrupt Controller and its associated registers. The interrupt controller provides a mechanism for a hardware resource in PSoC devices, to change program execution to a new address without regard to the current task being performed by the code being executed. For a complete table of the Interrupt Controller registers, refer to the “Summary Table of the Core Registers” on page 32. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

5.1 Architectural Description

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

Figure 5-1. Interrupt Controller Block Diagram



The sequence of events that occur during interrupt processing is as follows:

1. An interrupt becomes active, either because (a) the interrupt condition occurs (for example, a timer expires), (b) a previously posted interrupt is enabled through an update of an interrupt ***mask*** register, or (c) an interrupt is pending and GIE is set from ‘0’ to ‘1’ in the CPU Flag register.
2. The current executing instruction finishes.
3. The internal interrupt routine executes, taking 13 cycles. During this time, the following actions occur:
   - The PCH, PCL, and Flag register (CPU_F) are pushed onto the stack (in that order).
   - The CPU_F register is then cleared. Since this clears the GIE bit to 0, additional interrupts are temporarily disabled.
   - The PCH (PC[15:8]) is cleared to zero.
   - The interrupt vector is read from the interrupt controller and its value is placed into PCL (PC[7:0]). This sets the program counter to point to the appropriate address in the interrupt table (for example, 001Ch for the GPIO interrupt).
4. Program execution vectors to the interrupt table. Typically, a LJMP instruction in the interrupt table sends execution to the user's interrupt service routine (ISR) for this interrupt. (See “Instruction Set Summary” on page 36.)

5. The ISR executes. Note that interrupts are disabled since GIE = 0. In the ISR, interrupts can be re-enabled if desired, by setting GIE = 1 (take care to avoid stack overflow in this case).
6. The ISR ends with a RETI instruction. This pops the Flag register, PCL, and PCH from the stack, restoring those registers. The restored Flag register re-enables interrupts, since GIE = 1 again.
7. Execution resumes at the next instruction, after the one that occurred before the interrupt. However, if there are more pending interrupts, the subsequent interrupts are processed before the next normal program instruction.

Interrupt Latency. The time between the assertion of an enabled interrupt and the start of its ISR can be calculated using the following equation:

$$\begin{aligned}&\textit{Latency} = \\ &\textit{Time for current instruction to finish} + \\ &\textit{Time for M8C to change program counter to interrupt address} + \\ &\textit{Time for LJMP instruction in interrupt table to execute.}\end{aligned}$$ **Equation 1**

For example, if the 5-cycle JMP instruction executes when an interrupt becomes active, the total number of CPU clock cycles before the ISR begins is as follows:

$$\begin{aligned}&\textit{(1 to 5 cycles for JMP to finish)} + \\ &\textit{(13 cycles for interrupt routine)} + \\ &\textit{(7 cycles for LJMP) = 21 to 25 cycles.}\end{aligned}$$ **Equation 2**

In the example above, at 24 MHz, 25 clock cycles take 1.042 μs.

Interrupt Priority. The priorities of the interrupts only come into consideration if more than one interrupt is pending during the same instruction cycle. In this case, the priority encoder (see Figure 5-1) generates an interrupt vector for the highest priority interrupt that is pending.

5.1.1 Posted versus Pending Interrupts

An interrupt is posted when its interrupt conditions occur. This results in the flip-flop in Figure 5-1 clocking in a ‘1’. The interrupt remains posted until the interrupt is taken or until it is cleared by writing to the appropriate INT_CLRx register.

A posted interrupt is not pending unless it is enabled by setting its interrupt mask bit (in the appropriate INT_MSKx register). All pending interrupts are processed by the priority encoder to determine the highest priority interrupt to be taken by the M8C if the Global Interrupt Enable bit is set in the CPU_F register.

Disabling an interrupt by clearing its interrupt mask bit (in the INT_MSKx register) does not clear a posted interrupt, nor does it prevent an interrupt from being posted. It simply prevents a posted interrupt from becoming pending.

It is especially important to understand the functionality of clearing posted interrupts, if the configuration of the PSoC device is changed by the application.

For example, if a digital PSoC block is configured as a counter and has posted an interrupt but is later reconfigured to a serial communications receiver, the posted interrupt from the counter remains. Therefore, if the digital PSoC block's INT_MSKx bit is set after configuring the block as a serial communications receiver, a pending interrupt is generated immediately. To prevent the carryover of posted interrupts from one configuration to the next, the INT_CLRx registers should be used to clear posted interrupts prior to enabling the digital PSoC block.

5.2 Application Description

The interrupt controller and its associated registers allow the user's code to respond to an interrupt from almost every functional block in the PSoC devices. Interrupts for all the digital blocks and each of the analog columns are available, as well as interrupts for supply voltage, sleep, variable clocks, and a general GPIO (pin) interrupt.

The registers associated with the interrupt controller allow interrupts to be disabled either globally or individually. The registers also provide a mechanism by which a user can ***clear*** all pending and posted interrupts, or clear individual posted or pending interrupts. A ***software*** mechanism is provided to set individual interrupts. Setting an interrupt by way of software is very useful during code development, when one may not have the complete hardware system necessary to generate a real interrupt.

The following table lists the interrupts for all PSoC devices (highlighting specifically the CY8C24533, CY8C23533, CY8C23433CY8C24633) and the priorities that are available in each PSoC device.

Table 5-1. PSoC Device Interrupt Table

Interrupt Priority	Interrupt Address	CY8C24633CY8 C24533, CY8C23533, CY8C23433	Interrupt Name
0 (Highest)	0000h	✓	Reset
1	0004h	✓	Supply Voltage Monitor
2	0008h	✓	Analog Column 0
3	000Ch	✓	Analog Column 1
5	0014h	✓	SAR8 ADC
6	0018h	✓	VC3
7	001Ch	✓	GPIO
8	0020h	✓	PSoC Block DBB00
9	0024h	✓	PSoC Block DBB01
10	0028h	✓	PSoC Block DCB02
11	002Ch	✓	PSoC Block DCB03
24	0060h	✓	I2C
25 (Lowest)	0064h	✓	Sleep Timer

5.3 Register Definitions

The following registers are associated with the Interrupt Controller and are listed in address order. The register descriptions have an associated register table showing the bit structure for that register. The bits in the tables that are grayed out are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of '0'. For a complete table of Interrupt Controller registers, refer to the "Summary Table of the Core Registers" on page 32.

Depending on the PSoC device you have, only certain bits are accessible to be read or written, such as the INT_CLR0 and INT_MSK0 registers that are analog column and digital row dependent. The analog column dependent registers have the column number listed to the right of the Address column. The digital row dependent registers are set up the same way, only with the term "Row" in the Address column.

5.3.1 INT_CLRx Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,DAh	INT_CLR0	VC3	Sleep	GPIO	SAR8 ADC		Analog 1	Analog 0	V Monitor	RW : 0
0,DBh	INT_CLR1					DCB03	DCB02	DBB01	DBB00	RW : 0
0,DDh	INT_CLR3								I2C	RW : 0

The Interrupt Clear Registers (INT_CLRx) are used to enable the individual interrupt sources' ability to clear posted interrupts.

There are three interrupt clear registers (INT_CLR0, INT_CLR1, and INT_CLR3) which may be referred to in general as INT_CLRx.The INT_CLRx registers are similar to the INT_MSKx registers in that they hold a bit for each interrupt source. Functionally the INT_CLRx registers are similar to the INT_VC register, although their operation is completely independent. When an INT_CLRx register is read, any bits that are set indicate an interrupt has been posted for that hardware resource. Therefore, reading these registers gives the user the ability to determine all posted interrupts.

The Enable Software Interrupt (ENSWINT) bit in INT_MSK3[7] determines the way an individual bit value written to an INT_CLR0 register is interpreted. When ENSWINT is cleared (the default state), writing 1's to an INT_CLRx register has no effect. However, writing 0's to an INT_CLRx register, when ENSWINT is cleared, causes the corresponding interrupt to clear. If the ENSWINT bit is set, any 0's written to the INT_CLRx registers are ignored. However, 1's written to an INT_CLRx register, while ENSWINT is set, causes an interrupt to post for the corresponding interrupt.

Note When using the INT_CLRx register to post an interrupt, the hardware interrupt source, such as a digital clock, must not have its interrupt output high. Therefore, it may be difficult to use software interrupts with interrupt sources that do not have enables such as VC3.

Software interrupts can aid in debugging interrupt service routines by eliminating the need to create system level interactions that are sometimes necessary to create a hardware-only interrupt.

5.3.1.1 INT_CLR0 Register

Bit 7: VC3. This bit allows posted VC3 interrupts to be read, cleared, or set.

Bit 6: Sleep. This bit allows posted sleep interrupts to be read, cleared, or set.

Bit 5: GPIO. This bit allows posted GPIO interrupts to be read, cleared, or set.

Bit 4: SAR8 ADC. This bit allows posted SAR8 ADC interrupts to be read, cleared, or set.

Bit 3: Analog 2. This bit allows posted analog column 2 interrupts to be read, cleared, or set.

Bit 2: Analog 1. This bit allows posted analog column 1 interrupts to be read, cleared, or set.

Bit 1: Analog 0. This bit allows posted analog column 0 interrupts to be read, cleared, or set.

Bit 0: V Monitor. This bit allows posted V monitor interrupts to be read, cleared, or set.

For additional information, refer to the INT_CLR0 register on page 99.

5.3.1.2 INT_CLR1 Register

Bit 3: DCB03. This bit allows posted DCB03 interrupts to be read, cleared, or set for row 0 block 3.

Bit 2: DCB02. This bit allows posted DCB02 interrupts to be read, cleared, or set for row 0 block 2.

Bit 1: DBB01. This bit allows posted DBB01 interrupts to be read, cleared, or set for row 0 block 1.

Bit 0: DBB00. This bit allows posted DBB00 interrupts to be read, cleared, or set for row 0 block 0.

For additional information, refer to the INT_CLR1 register on page 101.

5.3.1.3 INT_CLR3 Register

Bit 0: I2C. This bit allows posted I2C interrupts to be read, cleared, or set.

For additional information, refer to the INT_CLR3 register on page 102.

5.3.2 INT_MSKx Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,DEh	INT_MSK3	ENSWINT							I2C	RW : 00
0,E0h	INT_MSK0	VC3	Sleep	GPIO	SAR8 ADC		Analog 1	Analog 0	V Monitor	RW : 00
0,E1h	INT_MSK1					DCB03	DCB02	DBB01	DBB00	RW : 00

The Interrupt Mask Registers (INT_MSKx) are used to enable the individual interrupt sources' ability to create pending interrupts.

There are three interrupt ***mask*** registers (INT_MSK0, INT_MSK1, and INT_MSK3) which may be referred to in general as INT_MSKx. If cleared, each bit in an INT_MSKx register prevents a posted interrupt from becoming a pending interrupt (input to the priority encoder). However, an interrupt can still post even if its mask bit is zero. All INT_MSKx bits are independent of all other INT_MSKx bits.

If an INT_MSKx bit is set, the interrupt source associated with that mask bit may generate an interrupt that becomes a pending interrupt. For example, if INT_MSK0[5] is set and at least one GPIO pin is configured to generate an interrupt, the interrupt controller allows a GPIO interrupt request to post and become a pending interrupt for the M8C to respond to. If a higher priority interrupt is generated before the M8C responds to the GPIO interrupt, the higher priority interrupt is responded to and not the GPIO interrupt.

Each interrupt source may require configuration at a block level. Refer to the other chapters in this manual for information on how to configure an individual interrupt source.

5.3.2.1 INT_MSK3 Register

Bit 7: ENSWINT. This bit is a special non-mask bit that controls the behavior of the INT_CLRx registers. See the INT_CLRx register in this section for more information.

Bit 0: I2C. This bit allows posted I2C interrupts to be read, masked, or set.

For additional information, refer to the INT_MSK3 register on page 103.

5.3.2.2 INT_MSK0 Register

Bit 7: VC3. This bit allows posted VC3 interrupts to be read, masked, or set.

Bit 6: Sleep. This bit allows posted sleep interrupts to be read, masked, or set.

Bit 5: GPIO. This bit allows posted GPIO interrupts to be read, masked, or set.

Bit 4: SAR8 ADC. This bit allows posted SAR8 ADC interrupts to be read, masked, or set.

Bit 3: Analog 2. This bit allows posted analog column 2 interrupts to be read, masked, or set.

Bit 2: Analog 1. This bit allows posted analog column 1 interrupts to be read, masked, or set.

Bit 1: Analog 0. This bit allows posted analog column 0 interrupts to be read, masked, or set.

Bit 0: V Monitor. This bit allows posted V monitor interrupts to be read, masked, or set.

For additional information, refer to the INT_MSK0 register on page 104.

5.3.2.3 INT_MSK1 Register

Bit 3: DCB03. This bit allows posted DCB03 interrupts to be read, masked, or set for row 0 block 3.

Bit 2: DCB02. This bit allows posted DCB02 interrupts to be read, masked, or set for row 0 block 2.

Bit 1: DBB01. This bit allows posted DBB01 interrupts to be read, masked, or set for row 0 block 1.

Bit 0: DBB00. This bit allows posted DBB00 interrupts to be read, masked, or set for row 0 block 0.

For additional information, refer to the INT_MSK1 register on page 105.

5.3.3 INT_VC Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E2h	INT_VC	Pending Interrupt[7:0]								RC : 00

LEGEND
C Clearable register or bits.

The Interrupt Vector Clear Register (INT_VC) returns the next pending interrupt and clears all pending interrupts when written.

Bits 7 to 0: Pending Interrupt[7:0]. When the register is read, the ***least significant byte (LSB)***, of the highest priority pending interrupt, is returned. For example, if the GPIO and I2C interrupts are pending and the INT_VC register was read, the value 1Ch is be read. However, if no interrupts are pending, the value 00h is returned. This is the reset vector in the interrupt table; however, reading 00h from the INT_VC register should not be considered an indication that a system reset is pending. Rather, reading 00h from the INT_VC register simply indicates that there are no pending interrupts.

The highest priority interrupt, indicated by the value returned by a read of the INT_VC register, is removed from the list of pending interrupts when the M8C services an interrupt.

Reading the INT_VC register has limited usefulness. If interrupts are enabled, a read to the INT_VC register does not determine that an interrupt was pending before the interrupt was actually taken. However, while in an interrupt, a user may wish to read the INT_VC register to see what the next interrupt is. When the INT_VC register is written, with any value, all pending and posted interrupts are cleared by asserting the clear line for each interrupt.

For additional information, refer to the INT_VC register on page 106.

5.3.4 CPU_F Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,F7h	CPU_F	PgMode[1:0]			XIO		Carry	Zero	GIE	RL : 02

LEGEND

L The AND F, expr; OR F, expr; and XOR F, expr flag instructions can be used to modify this register.

x An “x” before the comma in the address field indicates that this register can be read or written to no matter what bank is used.

The M8C Flag Register (CPU_F) provides read access to the M8C flags. Note that only the GIE (Global Interrupt Enable) bit is related to the interrupt controller.

Bits 7 to 1. The CPU_F register holds bits that are used by different resources. For information on the other bits in this register, refer to the CPU Core (M8C) chapter on page 35.

Bit 0: GIE. The state of the Global Interrupt Enable bit determines whether interrupts (by way of the IRQ) are recognized by the M8C. This bit is set or cleared by the user, using the flag-logic instructions (for example, OR F, 1).

GIE is also cleared automatically by the M8C upon entering the interrupt service routine (ISR), after the flag byte has been stored on the stack, preventing nested interrupts. Note that the bit can be set in an ISR if desired.

For GIE=1, the M8C samples the IRQ input for each instruction. For GIE=0, the M8C ignores the IRQ.

For additional information, refer to the CPU_F register on page 120.

6. General Purpose IO (GPIO)

This chapter discusses the General Purpose IO (GPIO) and its associated registers, which is the circuit responsible for interfacing to the IO pins of a PSoC device. The GPIO blocks provide the interface between the M8C core and the outside world. They offer a large number of configurations to support several types of ***input/output (IO)*** operations for both digital and analog systems. For a complete table of the GPIO registers, refer to the “Summary Table of the Core Registers” on page 32. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

6.1 Architectural Description

The GPIO contains input buffers, output drivers, register bit storage, and configuration logic for connecting the PSoC device to the outside world.

IO ports are arranged with (up to) 8 bits per port. Each full port contains eight identical GPIO blocks, with connections to identify a unique address and register bit number for each block. Each GPIO block can be used for the following types of IO:

- Digital IO (digital input and output controlled by software)
- Global IO (digital PSoC block input and output)
- Analog IO (analog PSoC block input and output)

Each IO pin also has several drive modes, as well as interrupt capabilities. While all GPIO pins are identical and provide digital IO, some pins may not connect internally to analog functions.

The main block diagram for the GPIO block is shown in Figure 6-1. Note that some pins do not have all of the functionality shown, depending on internal connections.

6.1.1 Digital IO

One of the basic operations of the GPIO ports is to allow the M8C to send information out of the PSoC device and get information into the M8C from outside the PSoC device. This is accomplished by way of the port data register (PRTxDR). Writes from the M8C to the PRTxDR register store the data state, one bit per GPIO. In the standard non-bypass mode, the pin drivers drive the pin in response to this data bit, with a drive strength determined by the Drive mode setting (see Figure 6-1). The actual voltage on the pin depends on the Drive mode and the external ***load***.

The M8C can read the value of a port by reading the PRTxDR register address. When the M8C reads the PRTxDR register address, the current value of the pin voltage is translated into a logic value and returned to the M8C.

Note that the pin voltage can represent a different logic value than the last value written to the PRTxDR register. This is an important distinction to remember in situations such as the use of a read modify write to a PRTxDR register. Examples of read modify write instructions include ***AND***, ***OR***, and ***XOR***.

The following is an example of how a read modify write, to a PRTxDR register, can have an unexpected and even indeterminate result in certain systems. Consider a scenario where all bits of Port 1 on the PSoC device are in the Strong 1 Resistive 0 Drive mode; so that in some cases, the system the PSoC is in may pull up one of the bits.

```
mov   reg[PRT1DR], 0x00
or    reg[PRT1DR], 0x80
```

In the first line of code above, writing a 0x00 to the port does not affect any bits that happen to be driven by the system the PSoC is in. However, in the second line of code, it does not guarantee that only bit 7 is the one set to a strong 1. Because the OR instruction first reads the port, any bits that are in the pull up state are read as a ‘1’. These ones are then written back to the port. When this happens, the pin goes into a strong 1 state; therefore, if the pull up condition ends in the system, the PSoC keeps the pin value at a logic 1.

6.1.2 Global IO

The GPIO ports are also used to interconnect signals to and from the digital PSoC blocks, as global inputs or outputs.

The global IO feature of each GPIO (port pin) is off by default. To access the feature, two parameters must be changed. To configure a GPIO as a global input, the port global select bit must be set for the desired GPIO using the PRTxGS register. This sets BYP = 1 in Figure 6-1 and disconnects the output of the PRTxDR register from the pin. Also, the Drive mode for the GPIO must be set to the digital High Z state. (Refer to the “PRTxDMx Registers” on page 10 for more information.) To configure a GPIO as a global output, the port global select bit must again be set. But in this case, the drive state must be set to any of the non-High Z states.

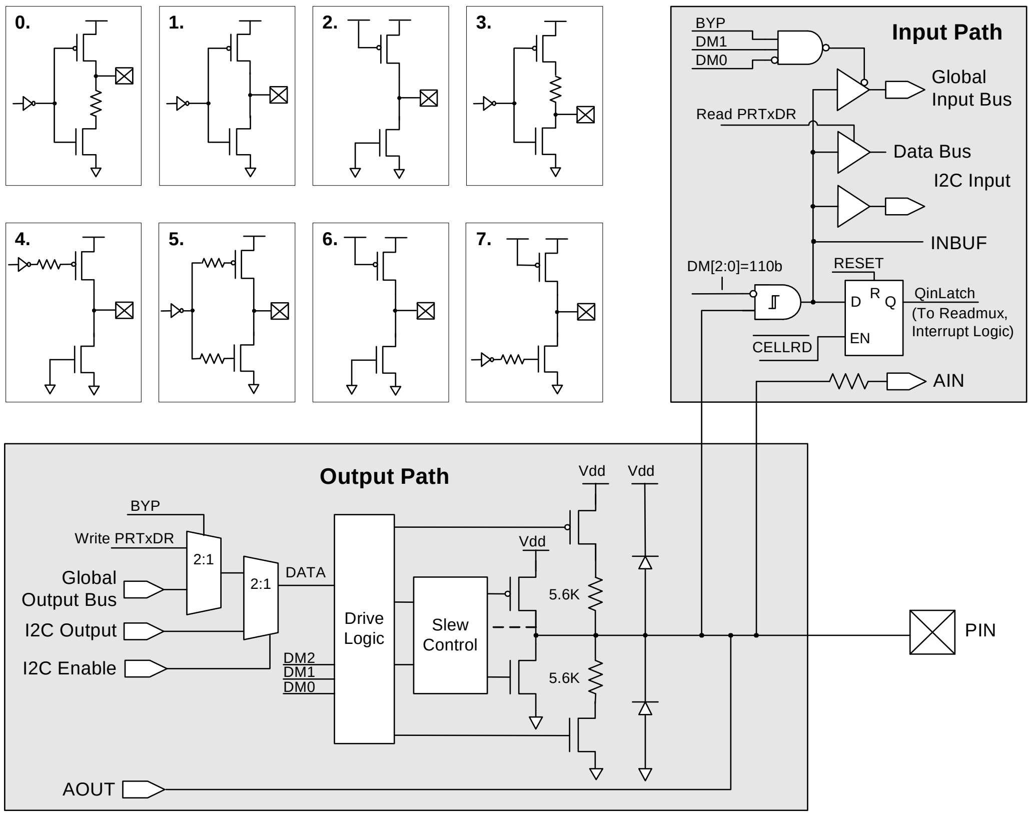
6.1.3 Analog Input

Analog signals can pass into the PSoC device core from PSoC device pins through the block’s AOUT pin. This provides a resistive ***path*** (~300 ohms) directly through the GPIO block. For analog modes, the GPIO block is typically configured into a high ***impedance*** analog drive mode (High Z). The mode turns off the Schmitt trigger on the input path, which may reduce power consumption and decrease internal switching noise when using a particular IO as an analog input. Refer to the Electrical Specifications chapter in the device data sheet.

Figure 6-1. GPIO Block Diagram

Drive Modes

DM2	DM1	DM0	Drive Mode	Diagram Number	Data = 0	Data = 1
0	0	0	Resistive Pull Down	0	Resistive	Strong
0	0	1	Strong Drive	1	Strong	Strong
0	1	0	High Impedance	2	High Z	High Z
0	1	1	Resistive Pull Up	3	Strong	Resistive
1	0	0	Open Drain, Drives High	4	High Z	Strong (Slow)
1	0	1	Slow Strong Drive	5	Strong (Slow)	Strong (Slow)
1	1	0	High Impedance Analog	6	High Z	High Z
1	1	1	Open Drain, Drives Low	7	Strong (Slow)	High Z



6.1.4 GPIO Block Interrupts

Each GPIO block can be individually configured for interrupt capability. Blocks are configured by pin interrupt enables and also by selection of the interrupt state. Blocks can be set to interrupt when the pin is high, low, or when it changes from the last time it was read. The block provides an open-drain interrupt output (INTO) that is connected to other GPIO blocks in a wire-OR fashion.

All pin interrupts that are wire-OR'ed together are tied to the same system GPIO interrupt. Therefore, if interrupts are enabled on multiple pins, the user's interrupt service routine must provide a mechanism to determine which pin was the source of the interrupt.

Using a GPIO interrupt requires the following steps:

1. Set the Interrupt mode in the GPIO pin block.
2. Enable the bit interrupt in the GPIO block.
3. Set the mask bit for the (global) GPIO interrupt.
4. Assert the overall Global Interrupt Enable.

The first two steps, bit interrupt enable and Interrupt mode, are set at the GPIO block level (that is, at each port pin), by way of the block's configuration registers.

The last two steps are common to all interrupts and are described in the Interrupt Controller chapter on page 61.

At the GPIO block level, asserting the INTO line depends only on the bit interrupt enable and the state of the pin relative to the chosen Interrupt mode. At the PSoC device level, due to their wire-OR nature, the GPIO interrupts are neither true edge-sensitive interrupts nor true level-sensitive interrupts. They are considered edge-sensitive for asserting, but level-sensitive for release of the wire-OR interrupt line.

If no GPIO interrupts are asserting, a GPIO interrupt occurs whenever a GPIO pin interrupt enable is set and the GPIO pin transitions, if not already transitioned, appropriately high or low, to match the Interrupt mode configuration. Once this happens, the INTO line pulls low to assert the GPIO interrupt. This assumes the other system-level enables are on, such as setting the global GPIO interrupt enable and the Global Interrupt Enable. Setting the pin interrupt enable may immediately assert INTO, if the Interrupt mode conditions are already being met at the pin.

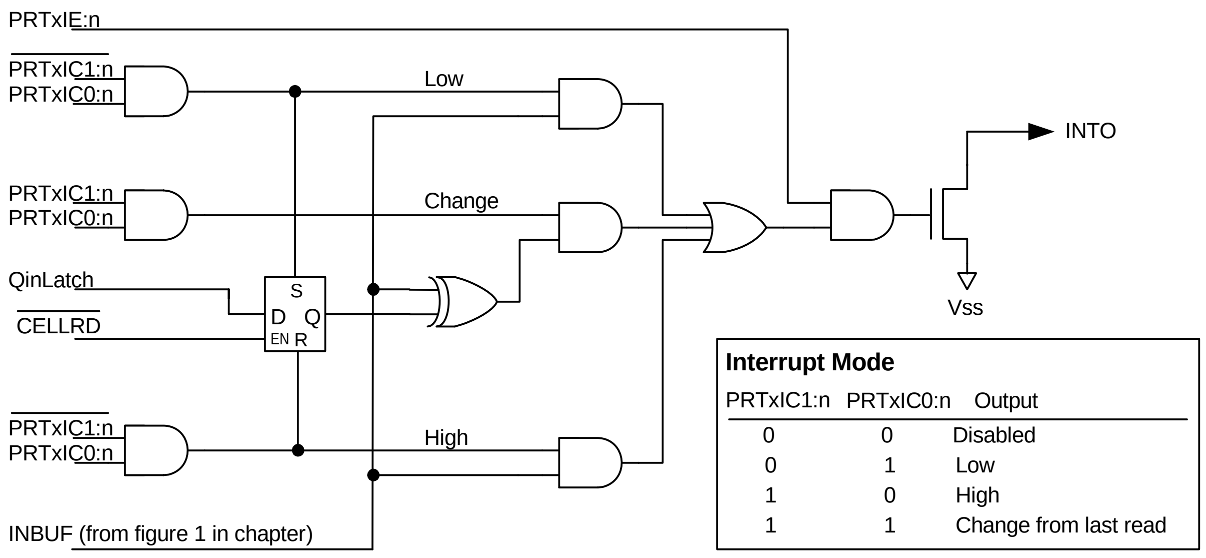
Once INTO pulls low, it continues to hold INTO low until one of these conditions change: (a) the pin interrupt enable is cleared; (b) the voltage at pin transitions to the opposite state; (c) in interrupt-on-change mode, the GPIO data register is read, thus setting the local interrupt level to the opposite state; or (d) the Interrupt mode is changed so that the current pin state does not create an interrupt. Once one of these conditions is met, the INTO releases. At this point, another GPIO pin (or this pin again) could assert its INTO pin, pulling the common line low to assert a new interrupt.

Note that the GPIO data register state is latched during read operation. Interrupt-on-change may not behave as expected if the input signal changes during the metastability time of the latch, that is, when the GPIO is being read.

Note the following behavior from this level-release feature. If one pin is asserting INTO and then a second pin asserts its INTO, when the first pin releases its INTO, the second pin is already driving INTO and thus no change is seen (that is, no new interrupt is asserted on the GPIO interrupt). Care must be taken, using polling or the states of the GPIO pin and Global Interrupt Enables, to catch all interrupts among a set of wire-OR GPIO blocks.

Figure 6-2 shows the interrupt logic portion of the block.

Figure 6-2. GPIO Interrupt Logic Diagram



Interrupt Mode

PRTxIC1:n	PRTxIC0:n	Output
0	0	Disabled
0	1	Low
1	0	High
1	1	Change from last read

6.2 Register Definitions

The following registers are associated with the General Purpose IO (GPIO) and are listed in address order. The register descriptions in this section have an associated register table showing the bit structure for that register. For a complete table of GPIO registers, refer to the "Summary Table of the Core Registers" on page 32.

For a selected GPIO block, the individual registers are addressed in the Summary Table of the Core Registers. In the register names, the 'x' is the port number, configured at the PSoC device level (x = 0 to 7, typically). All register values are readable, except for the PRTxDR register; reads of this register return the pin state instead of the register bit state.

6.2.1 PRTxDR Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	PRTxDR	Data[7:0]								RW : 00

LEGEND
xx An "x" after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the "Summary Table of the Core Registers" on page 32.

The Port Data Register (PRTxDR) allows for write or read access of the current logical equivalent of the voltage on the pin.

Bits 7 to 0: Data[7:0]. Writing the PRTxDR register bits sets the output drive state for the pin to high (for DIN=1) or low (DIN=0), unless a bypass mode is selected (either I2C Enable=1, or the global select register written high).

Reading the PRTxDR register returns the actual pin state, as seen by the input buffer. This may not be the same as the expected output state, if the load pulls the pin more strongly than the pin's configured output drive. See "Digital IO" on page 5 for a detailed discussion of digital IO.

For additional information, refer to the PRTxDR register on page 49.

6.2.2 PRTxIE Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	PRTxIE	Interrupt Enables[7:0]								RW : 00

LEGEND
xx An "x" after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the "Summary Table of the Core Registers" on page 32.

The Port Interrupt Enable Register (PRTxIE) is used to enable/disable the interrupt enable internal to the GPIO block.

Bits 7 to 0: Interrupt Enables[7:0]. A '1' enables the INTO output at the block and a '0' disables INTO so it is only High Z.

For additional information, refer to the PRTxIE register on page 50.

6.2.3 PRTxGS Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	PRTxGS	Global Select[7:0]								RW : 00

LEGEND

xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Summary Table of the Core Registers” on page 32.

The Port Global Select Register (PRTxGS) is used to select the block for connection to global inputs or outputs.

Bits 7 to 0:Global Select[7:0]. Writing this register high enables the global bypass (BYP = 1 in Figure 6-1). If the Drive mode is set to digital High Z (DM[2:0] = 010b), then the pin is selected for global input (PIN drives to the Global Input Bus). In non-High Z modes, the block is selected for global output (the Global Output Bus drives to PIN), bypassing the data register value (assuming I2C Enable = 0).

If the PRTxGS register is written to zero, the global in/out function is disabled for the pin and the pin reflects the value of PRT_DR.

For additional information, refer to PRTxGS register on page 51.

6.2.4 PRTxDMx Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	PRTxDM 2	Drive Mode 2[7:0]								RW : FF
1,xxh	PRTxDM 0	Drive Mode 0[7:0]								RW : 00
1,xxh	PRTxDM 1	Drive Mode 1[7:0]								RW : FF

LEGEND

xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Summary Table of the Core Registers” on page 32.

The Port Drive Mode Bit Registers (PRTxDMx) are used to specify the Drive mode for GPIO pins.

Bits 7 to 0: Drive Mode x[7:0]. In the PRTxDMx registers there are eight possible drive modes for each port pin. Three mode bits are required to select one of these modes, and these three bits are spread into three different registers (PRTxDM0, PRTxDM1, and PRTxDM2). The bit position of the effected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the three drive mode register bits that control the Drive mode for that pin (for example, bit[2] in PRT0DM0, bit[2] in PRT0DM1, and bit[2] in PRT0DM2). The three bits from the three registers are treated as a group. These are referred to as DM2, DM1, and DM0, or together as DM[2:0]. Drive modes are shown in Table 6-1.

For analog IO, the Drive mode should be set to one of the High Z modes, either 010b or 110b. The 110b mode has the advantage that the block’s digital input buffer is disabled, so no ***crowbar*** current flows even when the analog input is not close to either power rail. When digital inputs are needed on the same pin as analog inputs, the 010b Drive mode should be used. If the 110b Drive mode is used, the pin is always read as a zero by the CPU and the pin is not able to generate a useful interrupt. (It is not strictly required that a High Z mode be selected for analog operation.)

For global input modes, Drive mode must be set to 010b.

Table 6-1. Pin Drive Modes

Drive Modes			Pin State	Description
DM2	DM1	DM0		
0	0	0	Resistive pull down	Strong high, resistive low
0	0	1	Strong drive	Strong high, strong low
0	1	0	High impedance	High Z high and low, digital input enabled
0	1	1	Resistive pull up	Resistive high, strong low
1	0	0	Open drain high	Slow strong high, High Z low
1	0	1	Slow strong drive	Slow strong high, slow strong low
1	1	0	High impedance, analog (**reset state**)	High Z high and low, digital input disabled (for zero power) (**reset state**)
1	1	1	Open drain low	Slow strong low, High Z high

The GPIO provides a default Drive mode of high impedance, analog (High Z). This is achieved by forcing the reset state of all PRTxDM1 and PRTxDM2 registers to FFh.

The resistive drive modes place a ***resistance*** in series with the output, for low outputs (mode 000b) or high outputs (mode 011b). Strong Drive mode 001b gives the fastest edges at high DC drive strength. Mode 101b gives the same drive strength but with slower edges. The open-drain modes (100b and 111b) also use the slower edge rate drive. These modes enable open-drain functions such as I2C mode 111b (although the slow edge rate is not slow enough to meet the I2C fast mode specification).

For additional information, refer to the PRTxDM2 register on page 52, the PRTxDM0 register on page 123, and the PRTxDM1 register on page 124.

6.2.5 PRTxICx Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	PRTxIC0	Interrupt Control 0[7:0]								RW : 00
1,xxh	PRTxIC1	Interrupt Control 1[7:0]								RW : 00

LEGEND

xx An "x" after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the "Summary Table of the Core Registers" on page 32.

The Port Interrupt Control Registers (PRTxIC1 and PRTxIC0) are used to specify the Interrupt mode for GPIO pins.

Bits 7 to 0: Interrupt Control x[7:0]. In the PRTxICx registers, the Interrupt mode for the pin is determined by bits in these two registers. These are referred to as IC1 and IC0, or together as IC[1:0].

There are four possible interrupt modes for each port pin. Two mode bits are required to select one of these modes and these two bits are spread into two different registers (PRTxIC0 and PRTxIC1). The bit position of the effected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the interrupt control register bits that control the Interrupt mode for that pin (for example, bit[2] in PRT0IC0 and bit[2] in PRT0IC1). The two bits from the two registers are treated as a group.

The Interrupt mode must be set to one of the non-zero modes listed in Table 6-2, in order to get an interrupt from the pin.

The GPIO Interrupt mode "disabled" (00b) disables interrupts from the pin, even if the GPIO's bit interrupt enable is on (from the PRTxIE register).

Interrupt mode 01b means that the block asserts the interrupt line (INTO) when the pin voltage is low, providing the block's bit interrupt enable line is set (high).

Interrupt mode 10b means that the block asserts the interrupt line (INTO) when the pin voltage is high, providing the block's bit interrupt enable line is set (high).

Interrupt mode 11b means that the block asserts the interrupt line (INTO) when the pin voltage is the opposite of the last state read from the pin, providing the block's bit interrupt enable line is set high. This mode switches between low mode and high mode, depending on the last value read from the port during reads of the data register (PRTxDR). If the last value read from the GPIO was '0', the GPIO subsequently is in Interrupt High mode.

If the last value read from the GPIO was '1', the GPIO then is in Interrupt Low mode.

Table 6-2. GPIO Interrupt Modes

Interrupt Modes		Description
IC1	IC0	
0	0	Bit interrupt disabled, INTO de-asserted
0	1	Assert INTO when PIN = low
1	0	Assert INTO when PIN = high
1	1	Assert INTO when PIN = change from last read

Figure 6-3. GPIO Interrupt Mode 11b

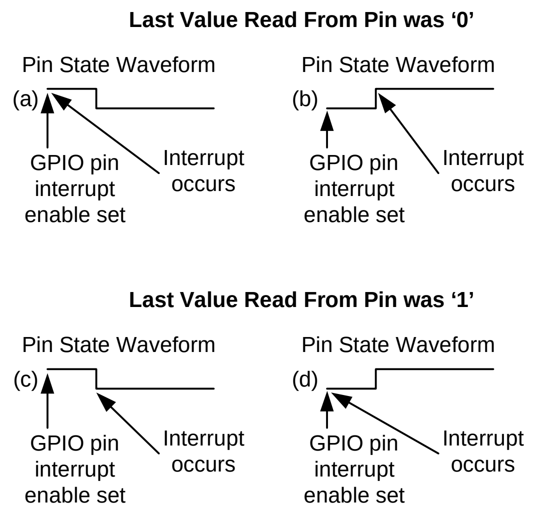

Last Value Read From Pin was '0'
Pin State Waveform
(a)
GPIO pin interrupt enable set
Interrupt occurs
Pin State Waveform
(b)
GPIO pin interrupt enable set
Interrupt occurs
Last Value Read From Pin was '1'
Pin State Waveform
(c)
GPIO pin interrupt enable set
Interrupt occurs
Pin State Waveform
(d)
GPIO pin interrupt enable set
Interrupt occurs

Figure 6-3 assumes that the GIE is set, GPIO interrupt mask is set, and that the GPIO Interrupt mode is set to 11b. The Change Interrupt mode is different from the other modes, in that it relies on the value of the GPIO's read latch to determine if the pin state has changed. Therefore, the port that contains the GPIO in question must be read during every interrupt service routine. If the port is not read, the Interrupt mode acts as if it is in high mode when the latch value is '0' and low mode when the latch value is '1'.

For additional information, refer to the PRTxIC0 register on page 125 and the PRTxIC1 register on page 126.

7. Analog Output Drivers

This chapter presents the Analog Output Drivers and their associated register. The analog output drivers provide a means for driving analog signals off the PSoC device. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47. For information on the analog system, refer to the “Analog System” on page 215.

7.1 Architectural Description

The CY8C24533, CY8C23533, CY8C23433CY8C24633 PSoC devices have up to two analog drivers used to output analog values on port pins.

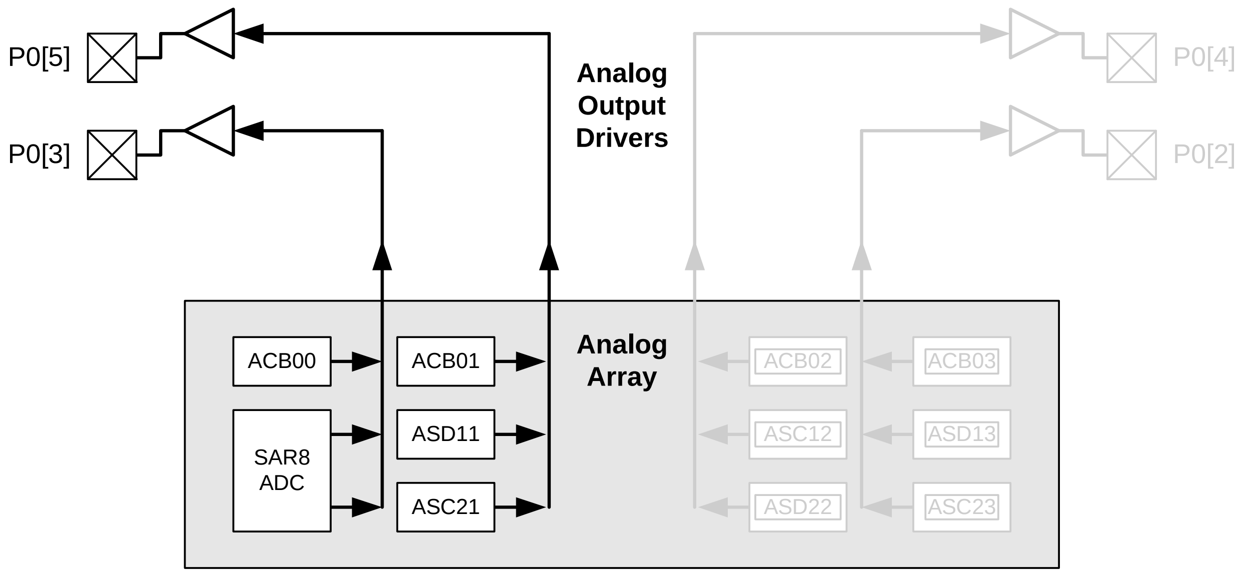
Table 7-1. PSoC Analog Output Drivers

Port Pin	CY8C24x23A	CY8C24533CY8C24633	CY8C23533	CY8C23433
P0[5]	✓	✓	✓	✓
P0[4]				
P0[3]	✓	✓	✓	✓
P0[2]				

Each of these drivers is a resource available to all the ***analog blocks*** in a particular analog column. Therefore, the number of analog output drivers matches the number of analog columns in a device. The user must select no more than one analog block per column to drive a signal on its analog output bus (ABUS), to serve as the input to the analog driver for that column. The output from the analog output driver for each column can be enabled and disabled using the Analog Output Driver register ABF_CR0. If the analog output driver is enabled, then it must have an analog block driving the ABUS for that column. Otherwise, the analog output driver can enter a high current consumption mode.

Figure 7-1 illustrates the drivers and their relationship within the analog array. For a detailed drawing of the analog output drivers in relation to the analog system, refer to the Analog Input Configuration chapter on page 241.

Figure 7-1. Analog Output Drivers for CY8C24533, CY8C23533, CY8C23433CY8C24633



7.2 Register Definitions

The following register is associated with the Analog Output Drivers. The register description has an associated register table showing the bit structure of the register. The bits that are grayed out in the table below are reserved bits and are not detailed in the register description that follows. Reserved bits should always be written with a value of '0'. Depending on the number of analog columns your PSoC device has (see the Cols. column in the register table below), some bits may be reserved (refer to the table titled "PSoC Device Characteristics" on page 20).

7.2.1 ABF_CR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,62h	ABF_CR0	ACol1Mux		ABUF1EN		ABUF0EN		Bypass	PWR	RW : 00

The Analog Output Buffer Control Register 0 (ABF_CR0) controls analog input muxes from Port 0 and the output buffer amplifiers that drive column outputs to device pins.

For more information on bit 7, see the Analog Input Configuration chapter on page 241.

Bit 7: ACol1MUX. A mux selects the output of column 0 input mux or column 1 input mux. When set, this bit sets the column 1 input to column 0 input mux output.

Bit 5: ABUF1EN. Enables the analog output buffer for Analog Column 1 (Pin P0[5]). A '0' disables the analog output buffer, a '1' enables.

Bit 3: ABUF0EN. Enables the analog output buffer for Analog Column 0 (Pin P0[3]). (1 Column: AGND). A '0' disables the analog output buffer, a '1' enables

Bit 1: Bypass. Bypass mode connects the analog output driver input directly to the output. When this bit is set, all analog output drivers are in bypass mode. This is a high impedance connection used primarily for measurement and calibration of internal references. Use of this feature is not recommended for customer designs.

Bit 0: PWR. This bit is used to set the power level of the analog output drivers. When this bit is set, all of the analog output drivers are in a High Power mode.

For additional information, refer to the ABF_CR0 register on page 135.

8. Internal Main Oscillator (IMO)

This chapter presents the Internal Main Oscillator (IMO) and its associated registers. The IMO produces clock signals of 24 MHz and 48 MHz. For a complete table of the IMO registers, refer to the “Summary Table of the Core Registers” on page 32. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

8.1 Architectural Description

The Internal Main Oscillator (IMO) outputs two clocks: a SYSCLK, which can be the internal 24 MHz clock or an external clock, and a SYSCLKX2 that is always twice the SYSCLK frequency. In the absence of a high-precision input source from the 32.768 kHz ***crystal oscillator***, the accuracy of the internal 24/48 MHz clocks is ±2.5% over temperature variation and two voltage ranges (3.3V ± 0.3V and 5.0V ± 0.25%). No external components are required to achieve this level of accuracy.

There is an option to phase lock this oscillator to the External Crystal Oscillator (ECO). The choice of crystal and its inherent accuracy determines the overall accuracy of the oscillator. The ECO must be stable prior to locking the frequency of the IMO to this reference source.

The frequency doubler circuit, which produces SYSCLKX2, can be disabled to save power. The lower frequency SYSCLK settings are available by setting the slow IMO (SLIMO) bit in the CPU_SCR1 register. With this bit set and the corresponding factory trim value applied to the IMO_TR register, SYSCLK can be lowered to 6 MHz. This offers lower device power consumption for systems that can operate with the reduced system clock. Slow IMO mode is discussed further in the “Application Description” on page 15.

8.2 Application Description

To save power, the IMO frequency can be reduced from 24 MHz to 6 MHz or 12 MHz using the SLIMO bit in the CPU_SCR1 register, in conjunction with the Trim values in the IMO_TR register. Note that the CY8C27x43, CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, CY8C22x13, CY7C603xx, and CYWUSB6953 devices do not have this functionality.

8.2.1 Trimming the IMO

An 8-bit register (IMO_TR) is used to trim the IMO. Bit 0 is the LSB and bit 7 is the MSB. The trim step size is approximately 80 kHz.

A factory trim setting is loaded into the IMO_TR register at boot time for 5V ± 0.25V operation, except for the CY7C603xx, which is 3.3V ± 0.25V. For operation in the voltage ranges of 3.3V ± 0.3V and 2.7V ± 0.3V, user code must modify the contents of this register with values stored in Flash bank 0 as shown in Table 3-11 on page 49. This is done with a Table Read command to the Supervisory ROM.

8.3 Register Definitions

The following registers are associated with the Internal Main Oscillator (IMO). The register descriptions have an associated register table showing the bit structure for that register. The bits in the tables that are grayed out are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of ‘0’. For a complete table showing all oscillator registers, refer to the “Summary Table of the Core Registers” on page 32.

8.3.1 CPU_SCR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW *	ECO EX *		IRAMDIS	# : 00

LEGEND

x An “x” before the comma in the address field indicates that this register can be read or written to no matter what bank is used.

# Access is bit specific. Refer to the Register Details chapter on page 47 for additional information.

* Bits 3 and 2 (ECO EXW and ECO EX, respectively) cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

The System Status and Control Register 1 (CPU_SCR1) is used to convey the status and control of events related to internal resets and watchdog reset.

Bit 7: IRESS. The Internal Reset Status bit is a read only bit that may be used to determine if the booting process occurred more than once.

When this bit is set, it indicates that the SROM SWBootReset code was executed more than once. If this bit is not set, the SWBootReset executed only once. In either case, the SWBootReset code does not allow execution from code stored in Flash until the M8C Core is in a safe operating mode with respect to supply voltage and Flash operation. There is no need for concern when this bit is set. It is provided for systems which may be sensitive to boot time, so that they can determine if the normal one-pass boot time is exceeded. For more information on the SWBootReest code see the Supervisory ROM (SROM) chapter on page 45.

Bit 4: SLIMO. When set, the Slow IMO bit allows the active power dissipation of the PSoC device to be reduced by slowing down the IMO from 24 MHz to 6 MHz. The IMO trim value must also be changed when SLIMO is set. When not in external clocking mode, the IMO is the source for SYSCLK; therefore, when the speed of the IMO changes, so does SYSCLK.

Bit 3: ECO EXW. The ECO Exists Written bit is used as a status bit to indicate that the ECO EX bit has been previously written to. It is read only. When this bit is a ‘1’, this indicates that the CPU_SCR1 register has been written to and is now locked. When this bit is a ‘0’, the register has not been written to since the last reset event. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

Bit 2: ECO EX. The ECO Exists bit serves as a flag to the hardware, to indicate that an external crystal ***oscillator*** exists in the system. Just after boot, it may be written *only once* to a value of ‘1’ (crystal exists) or ‘0’ (crystal does not exist). If the bit is ‘0’, a switch-over to the ECO is locked out by hardware. If the bit is ‘1’, hardware allows the firmware to freely switch between the ECO and ILO. It should be written as early as possible after a ***Power On Reset (POR)*** or ***External Reset (XRES)*** event, where it is assumed that program execution integrity is high. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

Bit 0: IRAMDIS. The Initialize RAM Disable bit is a control bit that is readable and writeable. The ***default value*** for this bit is ‘0’, which indicates that the maximum amount of SRAM should be initialized on watchdog reset to a value of 00h. When the bit is ‘1’, the minimum amount of SRAM is initialized after a watchdog reset. For more information on this bit, see the “SROM Function Descriptions” on page 46.

For additional information, refer to the CPU_SCR1 register on page 121.

8.3.2 OSC_CR2 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E2h	OSC_CR2	PLLGAIN					EXTCLKEN	RSVD	SYSCLKX2DIS	RW : 00

The Oscillator Control Register 2 (OSC_CR2) is used to configure various features of internal clock sources and clock nets.

Bit 7: PLLGAIN. This is the only bit in the OSC_CR2 register that directly influences the PLL. When set, this bit keeps the PLL in Low Gain mode. If this bit is held low, the lock time is less than 10 ms. If this bit is held high, the lock time is on the order of 50 ms. After lock is achieved, it is recommended that this bit be forced high to decrease the jitter on the output. If longer lock time is tolerable, the PLLGAIN bit can be held high all the time.

Bit 2: EXTCLKEN. When the EXTCLKEN bit is set, the external clock becomes the source for the internal clock tree, SYSCLK, which drives most PSoC device clocking functions. All external and internal signals, including the 32 kHz clock, whether derived from the Internal Low Speed Oscillator (ILO) or the crystal oscillator, are synchronized to this clock source. If an external clock is enabled, PLL mode should be off.

The external clock input is located on port P1[4]. When using this input, the pin Drive mode should be set to High Z (not High Z analog).

Bit 1: RSVD. Reserved bit - This bit should always be 0.

Bit 0: SYSCLKX2DIS. When SYSCLKX2DIS is set, the IMO's doubler is disabled. This results in a reduction of overall device power, on the order of 1 mA. It is advised that any application that does not require this doubled clock should have it turned off.

For additional information, refer to the OSC_CR2 register on page 153.

8.3.3 IMO_TR Register

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

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

The PSoC device specific value for 5V operation is loaded into the Internal Main Oscillator Trim register (IMO_TR) at boot time. The Internal Main Oscillator operates within specified tolerance over a voltage range of 4.75V to 5.25V, with no modification of this register. If the PSoC device operates at a lower voltage, user code must modify the contents of this register. For operation in the voltage range of 3.3V ± .3V, this is accomplished with a Table Read command to the Supervisory ROM, which supplies a trim value for operation in this range. For operation between these voltage ranges, user code can interpolate the best value using both available factory trim values.

It is strongly recommended that the user not alter the register value, unless Slow IMO mode is used.

Bits 7 to 0: Trim[7:0]. These bits are used to trim the Internal Main Oscillator. A larger value in this register increases the speed of the oscillator.

For additional information, refer to the IMO_TR register on page 156.

9. Internal Low Speed Oscillator

This chapter briefly explains the Internal Low Speed Oscillator (ILO) and its associated register. The Internal Low Speed Oscillator produces a 32 kHz clock. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

9.1 Architectural Description

The Internal Low Speed Oscillator (ILO) is an oscillator with a nominal frequency of 32 kHz. It is used to generate sleep wake-up interrupts and watchdog resets. This oscillator can also be used as a clocking source for the digital PSoC blocks.

The oscillator operates in three modes: normal power, low power, and off. The normal power mode consumes more current to produce a more accurate frequency. The low power mode is always used when the part is in a power down (sleep) state.

9.2 Register Definitions

The following register is associated with the Internal Low Speed Oscillator (ILO). The register description has an associated register table showing the bit structure. The bits in the table that are grayed out are reserved bits and are not detailed in the register description that follows. Note that reserved bits should always be written with a value of '0'.

9.2.1 ILO_TR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E9h	ILO_TR			Bias Trim[1:0]		Freq Trim[3:0]				W : 00

The Internal Low Speed Oscillator Trim Register (ILO_TR) sets the adjustment for the internal low speed oscillator.

The device-specific value, placed in the trim bits of this register at boot time, is based on factory testing. ***It is strongly recommended that the user not alter the values in the register***.

Bits 5 and 4: Bias Trim[1:0]. These two bits are used to set the bias current in the PTAT Current Source. Bit 5 gets inverted, so that a medium bias is selected when both bits are '0'. The ***bias current*** is set according to Table 9-1.

Table 9-1. Bias Current in PTAT

Bias Current	Bias Trim [1:0]
Medium Bias	00b
Maximum Bias	01b
Minimum Bias	10b
Reserved	11b

Bits 3 to 0: Freq Trim[3:0]. These four bits are used to trim the frequency. Bit 0 is the LSb and bit 3 is the MSb. Bit 3 gets inverted inside the register.

For additional information, refer to the ILO_TR register on page 157.

10. External Crystal Oscillator (ECO)

This chapter briefly explains the External Crystal Oscillator (ECO) and its associated registers. The 32.768 kHz external crystal oscillator circuit allows the user to replace the internal low speed oscillator with a more precise time source at low cost and low power. For a complete table of the External Crystal Oscillator registers, refer to the "Summary Table of the Core Registers" on page 32. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

10.1 Architectural Description

The External Crystal Oscillator (ECO) circuit uses an inexpensive watch crystal and two small value capacitors as external components, with all other components being on the PSoC device. The crystal oscillator may be configured to provide a reference to the Internal Main Oscillator (IMO) in PLL mode, for generating a 24 MHz system clock.

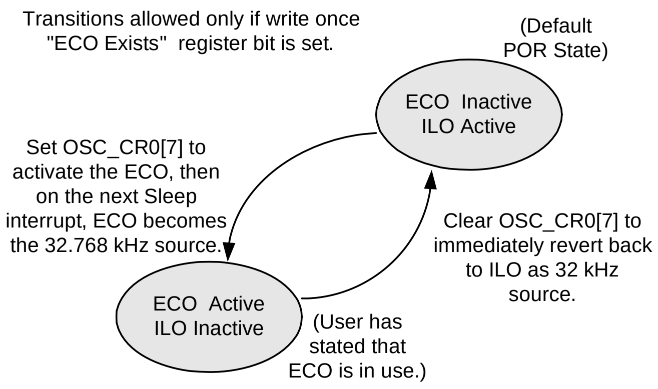
The XTALIn and XTALOut pins support connection of a 32.768 kHz watch crystal. To use the external crystal, bit 7 of the Oscillator Control 0 register (OSC_CR0) must be set (the default is off). The only external components needed are the crystal and the two capacitors that connect to Vdd. Note that transitions between the internal and external oscillator domains may produce glitches on the clock bus.

During the process of activating the ECO, there must be a hold-off period before using it as the 32.768 kHz source. This hold-off period is partially implemented in hardware using the sleep timer. Firmware must set up a sleep period of one second (maximum ECO ***settling time***), and then enable the ECO in the OSC_CR0 register. At the one second time-out (the sleep interrupt), the switch is made by hardware to the ECO. If the ECO is subsequently deactivated, the Internal Low Speed Oscillator (ILO) is again activated and the switch is made back to the ILO immediately.

The ECO Exists bit (ECO EX, bit 2 in the CPU_SCR1 register) is used to control whether the switch-over is allowed or locked. This is a write once bit. It is written early in code execution after a Power On Reset (POR) or External Reset (XRES) event. A '1' in this bit indicates to the hardware that a crystal exists in the system, and firmware is allowed to switch back and forth between ECO and ILO operation. If the bit is '0', switch-over to the ECO is locked out. The ECO Exists Written bit (ECO EXW, bit 3 in the CPU_SCR1 register) is read only and is set on the first write to this register. When this bit is '1', it indicates that the state of ECO EX is locked. This is illustrated in Figure 10-1.

Note Bits 3 and 2 (ECO EXW and ECO EX, respectively) in the CPU_SCR1 register cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

Figure 10-1. Transition Between ECO and ILO Operation



Transitions allowed only if write once "ECO Exists" register bit is set.
(Default POR State)
ECO Inactive ILO Active
Set OSC_CR0[7] to activate the ECO, then on the next Sleep interrupt, ECO becomes the 32.768 kHz source.
Clear OSC_CR0[7] to immediately revert back to ILO as 32 kHz source.
ECO Active ILO Inactive
(User has stated that ECO is in use.)

The firmware steps involved in switching between the Internal Low Speed Oscillator (ILO) to the 32.768 kHz External Crystal Oscillator (ECO) are as follows.

1. At reset, the PSoC device begins operation, using the ILO.
2. Set the ECO EX bit to allow crystal operation.
3. Select a sleep interval of one second, using bits[4:3] in the Oscillator Control 0 register (OSC_CR0), as the oscillator stabilization interval.
4. Enable the ECO by setting bit [7] in Oscillator Control 0 register (OSC_CR0) to '1'.

5. The ECO becomes the selected source at the end of the one-second interval on the edge created by the sleep interrupt logic. The one-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. Reset the sleep timer (if this does not interfere with any ongoing real-time clock operation), to guarantee the interval length. Note that the ILO continues to run until the oscillator is automatically switched over by the sleep timer interrupt.
6. It is strongly advised to wait the one-second stabilization period prior to engaging the PLL mode to lock the IMO frequency to the ECO frequency.

Note 1 The ILO switches back instantaneously by writing the 32 kHz Select Control bit to '0'.

Note 2 If the proper settings are selected in PSoC Designer, the above steps are automatically done in *boot.asm*.

Note 3 Transitions between oscillator domains may produce glitches on the 32 kHz clock bus. Functions that require accuracy on the 32 kHz clock should be enabled after the transition in oscillator domains.

10.1.1 ECO External Components

The external component connections and selections of the External Crystal Oscillator are illustrated in Figure 10-2.

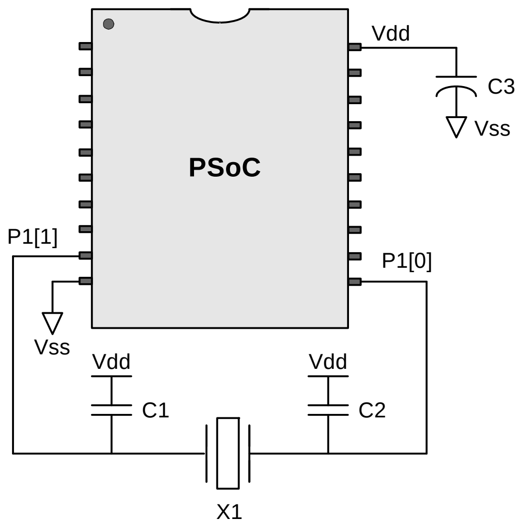
- Crystal – 32.768 kHz watch crystal such as Epson C-002RX.
- Capacitors – C1, C2 use NPO ceramic caps.

Use the equation below if you do not employ PLL mode.

$$C1 = C2 = 25\ pF - (Package\ Capacitance) - (Board\ Parasitic\ Capacitance)$$

An error of 1 pF in C1 and C2 gives about a 3 ppm error in frequency.

Figure 10-2. 20-Pin Example of the ECO External Connections



Refer to the device data sheet, in the packaging chapter, for typical package capacitances on crystal pins.

10.2 PSoC Device Distinctions

Bits 3 and 2 (ECO EXW and ECO EX, respectively) in the CPU_SCR1 register cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

10.3 Register Definitions

The following registers are associated with the External Crystal Oscillator and are listed in address order. Each register description has an associated register table showing the bit structure for that register. The bits that are grayed out in the tables below are reserved bits and are not detailed in the register descriptions. Note that reserved bits should always be written with a value of '0'. For a complete table of external crystal oscillator registers, refer to the "Summary Table of the Core Registers" on page 32.

10.3.1 CPU_SCR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW *	ECO EX *		IRAMDIS	# : 00

LEGEND
x An "x" before the comma in the address field indicates that this register can be read or written to no matter what bank is used.
# Access is bit specific. Refer to the Register Details chapter on page 47 for additional information.
* Bits 3 and 2 (ECO EXW and ECO EX, respectively) cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

The System Status and Control Register 1 (CPU_SCR1) is used to convey the status and control of events related to internal resets and watchdog reset.

Bit 7: IRESS. The Internal Reset Status bit is a read only bit that may be used to determine if the booting process occurred more than once.

When this bit is set, it indicates that the SROM SWBootReset code was executed more than once. If this bit is not set, the SWBootReset executed only once. In either case, the SWBootReset code does not allow execution from code stored in Flash until the M8C Core is in a safe operating mode with respect to supply voltage and Flash operation. There is no need for concern when this bit is set. It is provided for systems which may be sensitive to boot time, so that they can determine if the normal one-pass boot time is exceeded. For more information on the SWBootReest code see the Supervisory ROM (SROM) chapter on page 45.

Bit 4: SLIMO. When set, the Slow IMO bit allows the active power dissipation of the PSoC device to be reduced by slowing down the IMO from 24 MHz to 6 MHz. The IMO trim value must also be changed when SLIMO is set. When not in external clocking mode, the IMO is the source for SYSCLK; therefore, when the speed of the IMO changes, so does SYSCLK.

Bit 3: ECO EXW. The ECO Exists Written bit is used as a status bit to indicate that the ECO EX bit has been previously written to. It is read only. When this bit is a '1', this indicates that the CPU_SCR1 register has been written to and is now locked. When this bit is a '0', the register has not been written to since the last reset event. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

Bit 2: ECO EX. The ECO Exists bit serves as a flag to the hardware, to indicate that an external crystal ***oscillator*** exists in the system. Just after boot, it may be written *only once* to a value of '1' (crystal exists) or '0' (crystal does not exist). If the bit is '0', a switch-over to the ECO is locked out by hardware. If the bit is '1', hardware allows the firmware to freely switch between the ECO and ILO. It should be written as early as possible after a ***Power On Reset (POR)*** or ***External Reset (XRES)*** event, where it is assumed that program execution integrity is high. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

Bit 0: IRAMDIS. The Initialize RAM Disable bit is a control bit that is readable and writeable. The ***default value*** for this bit is '0', which indicates that the maximum amount of SRAM should be initialized on watchdog reset to a value of 00h. When the bit is '1', the minimum amount of SRAM is initialized after a watchdog reset. For more information on this bit, see the "SROM Function Descriptions" on page 46.

For additional information, refer to the CPU_SCR1 register on page 121.

10.3.2 OSC_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00

The Oscillator Control Register 0 (OSC_CR0) is used to configure various features of internal clock sources and clock nets.

Bit 7: 32k Select. By default, the 32 kHz clock source is the Internal Low Speed Oscillator (ILO). Optionally, the 32.768 kHz External Crystal Oscillator (ECO) may be selected.

Bit 6: PLL Mode. This is the only bit in the OSC_CR0 register that directly influences the Phase-Locked Loop (PLL). When set, this bit enables the PLL. The EXTCLKEN bit in the OSC_CR2 register should be set low during PLL operation. For information on the PLL, refer to the Phase-Locked Loop (PLL) chapter on page 27.

Bit 5: No Buzz. Normally, when the Sleep bit is set in the CPU_SCR register, all PSoC device systems are powered down, including the bandgap reference. However, to facilitate the detection of ***POR*** and ***LVD*** events at a rate higher than the sleep interval, the bandgap circuit is powered up periodically for about 60 µs at the Sleep System Duty Cycle (set in ECO_TR), which is independent of the sleep interval and typically higher. When the No Buzz bit is set, the Sleep System Duty Cycle value is overridden and the bandgap circuit is forced to be on during sleep. This results in a faster response to an LVD or POR event (continuous detection as opposed to periodic detection), at the expense of higher average sleep current.

Bits 4 and 3: Sleep[1:0]. The available sleep interval selections are shown in Table 10-1. Remember that when the ILO is the selected 32 kHz clock source, sleep intervals are approximate.

Table 10-1. Sleep Interval Selections

Sleep Interval OSC_CR[4:3]	Sleep Timer Clocks	Sleep Period (nominal)	Watchdog Period (nominal)
00b (default)	64	1.95 ms	6 ms
01b	512	15.6 ms	47 ms
10b	4,096	125 ms	375 ms
11b	32,768	1 sec	3 sec

Bits 2 to 0: CPU Speed[2:0]. The PSoC M8C may operate over a range of CPU clock speeds (see Table 10-2), allowing the M8C's performance and power requirements to be tailored to the application.

The reset value for the CPU Speed bits is zero; therefore, the default CPU speed is one-eighth of the clock source. The Internal Main Oscillator (IMO) is the default clock source for the CPU speed circuit; therefore, the default CPU speed is 3 MHz.

The CPU frequency is changed with a write to the OSC_CR0 register. There are eight frequencies generated from a power-of-two divide circuit, which are selected by a 3-bit code. At any given time, the CPU 8-to-1 clock mux is selecting one of the available frequencies, which is resynchronized to the 24 MHz master clock at the output.

Regardless of the CPU Speed bit's setting, if the actual CPU speed is greater than 12 MHz, the 24 MHz operating requirements apply. An example of this scenario is a device that is configured to use an external clock, which is supplying a frequency of 20 MHz. If the CPU speed register's value is 0b011, the CPU clock is 20 MHz. Therefore, the supply voltage requirements for the device are the same as if the part was operating at 24 MHz off of the IMO. The operating voltage requirements are not relaxed until the CPU speed is at 12 MHz or less.

Table 10-2. OSC_CR0[2:0] Bits: CPU Speed

Bits	Internal Main Oscillator	External Clock
000b	3 MHz	EXTCLK/ 8
001b	6 MHz	EXTCLK/ 4
010b	12 MHz	EXTCLK/ 2
011b	24 MHz	EXTCLK/ 1
100b	1.5 MHz	EXTCLK/ 16
101b	750 kHz	EXTCLK/ 32
110b	187.5 kHz	EXTCLK/ 128
111b	93.7 kHz	EXTCLK/ 256

For additional information, refer to the OSC_CR0 register on page 151.

10.3.3 ECO_TR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,EBh	ECO_TR	PSSDC[1:0]								W : 00

The External Crystal Oscillator Trim Register (ECO_TR) sets the adjustment for the 32.768 kHz External Crystal Oscillator.

The device-specific value placed in this register at boot time is based on factory testing. This register does not adjust the frequency of the external crystal oscillator.

It is strongly recommended that the user not alter the register value.

Bits 7 and 6: PSSDC[1:0]. These bits are used to set the sleep ***duty cycle***. These bits should not be altered.

For additional information, refer to the ECO_TR register on page 159.

11. Phase-Locked Loop (PLL)

This chapter presents the Phase-Locked Loop (PLL) and its associated registers. For a complete table of the PLL registers, refer to the “Summary Table of the Core Registers” on page 32. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

11.1 Architectural Description

A ***Phase-Locked Loop (PLL)*** function generates the system clock with crystal accuracy. It is designed to provide a 23.986 MHz oscillator, when utilized with an external 32.768 kHz crystal.

Although the PLL tracks crystal accuracy, it requires time to lock onto the reference frequency when first starting. The length of time depends on the PLLGAIN controlled by bit 7 of the OSC_CR2 register. If this bit is held low, the lock time is less than 10 ms. If this bit is held high, the lock time is on the order of 50 ms. After lock is achieved, it is recommended that this bit be forced high to decrease the ***jitter*** on the output. If longer lock time is tolerable, the PLLGAIN bit can be held high all the time.

After the 32.768 kHz External Crystal Oscillator (ECO) has been selected and enabled, the following procedure should be followed to enable the PLL and allow for proper frequency lock.

- Select a CPU frequency of 3 MHz or less.
- Enable the PLL.
- Wait between 10 and 50 ms, depending on bit 7 of the OSC_CR2 register.
- Set the CPU to a faster frequency, if desired. To do this, write the CPU Speed[2:0] bits in the OSC_CR0 register. The CPU frequency immediately changes when these bits are set.

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

11.2 Register Definitions

The following registers are associated with the Phase-Locked Loop (PLL) and are listed in address order. Each register description has an associated register table showing the bit structure for that register. The bits that are grayed out in the tables below are reserved bits and are not detailed in the register descriptions. Note that reserved bits should always be written with a value of ‘0’. For a complete table of the PLL registers, refer to the “Summary Table of the Core Registers” on page 32.

11.2.1 OSC_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00

The Oscillator Control Register 0 (OSC_CR0) is used to configure various features of internal clock sources and clock nets.

Bit 7: 32k Select. By default, the 32 kHz clock source is the Internal Low Speed Oscillator (ILO). Optionally, the 32.768 kHz External Crystal Oscillator (ECO) may be selected.

Bit 6: PLL Mode. This is the only bit in the OSC_CR0 register that directly influences the Phase-Locked Loop (PLL). When set, this bit enables the PLL. The EXTCLKEN bit in the OSC_CR2 register should be set low during PLL operation.

Bit 5: No Buzz. Normally, when the Sleep bit is set in the CPU_SCR register, all PSoC device systems are powered down, including the bandgap reference. However, to facilitate the detection of ***POR*** and ***LVD*** events at a rate higher than the sleep interval, the bandgap circuit is powered up periodically for about 60 μs at the Sleep System Duty Cycle (set in ECO_TR), which is independent of the sleep interval and typically higher. When the No Buzz bit is set, the Sleep System Duty Cycle value is overridden and the bandgap circuit is forced to be on during sleep. This results in a faster response to an LVD or POR event (continuous detection as opposed to periodic detection), at the expense of slightly higher average sleep current.

Bits 4 and 3: Sleep[1:0]. The available sleep interval selections are shown in Table 11-1. It must be remembered that when the ILO is the selected 32 kHz clock source, sleep intervals are approximate.

Table 11-1. Sleep Interval Selections

Sleep Interval OSC_CR[4:3]	Sleep Timer Clocks	Sleep Period (nominal)	Watchdog Period (nominal)
00b (default)	64	1.95 ms	6 ms
01b	512	15.6 ms	47 ms
10b	4096	125 ms	375 ms
11b	32,768	1 sec	3 sec

Bits 2 to 0: CPU Speed[2:0]. The PSoC M8C may operate over a range of CPU clock speeds (see Table 11-2), allowing the M8C's performance and power requirements to be tailored to the application.

The reset value for the CPU Speed bits is zero; therefore, the default CPU speed is one-eighth of the clock source. The Internal Main Oscillator (IMO) is the default clock source for the CPU speed circuit; therefore, the default CPU speed is 3 MHz.

The CPU frequency is changed with a write to the OSC_CR0 register. There are eight frequencies generated from a power-of-two divide circuit, which are selected by a 3-bit code. At any given time, the CPU 8-to-1 clock mux is selecting one of the available frequencies, which is resynchronized to the 24 MHz master clock at the output.

Regardless of the CPU Speed bit's setting, if the actual CPU speed is greater than 12 MHz, the 24 MHz operating requirements apply. An example of this scenario is a device that is configured to use an external clock, which is supplying a frequency of 20 MHz. If the CPU speed register's value is 0b011, the CPU clock is 20 MHz. Therefore, the supply voltage requirements for the device are the same as if the part was operating at 24 MHz off of the IMO. The operating voltage requirements are not relaxed until the CPU speed is at 12 MHz or less.

Some devices support the slow IMO option, as discussed in the IMO chapter in the "Architectural Description" on page 15. This offers an option to lower both system and CPU clock speed in order to save power.

An automatic protection mechanism is available for systems that need to run at peak CPU clock speed but cannot guarantee a high enough supply voltage for that clock speed. See the LVDTBEN bit in the "VLT_CR Register" on page 320 for more information.

Table 11-2. OSC_CR0[2:0] Bits: CPU Speed

Bits	6 MHz Internal Main Oscillator	24 MHz Internal Main Oscillator	External Clock
000b	750 kHz	3 MHz	EXTCLK/ 8
001b	1.5 MHz	6 MHz	EXTCLK/ 4
010b	3 MHz	12 MHz	EXTCLK/ 2
011b	6 MHz	24 MHz	EXTCLK/ 1
100b	375 kHz	1.5 MHz	EXTCLK/ 16
101b	187.5 kHz	750 kHz	EXTCLK/ 32
110b	93.7 kHz	187.5 kHz	EXTCLK/ 128
111b	46.9 kHz	93.7 kHz	EXTCLK/ 256

For additional information, refer to the OSC_CR0 register on page 151.

11.2.2 OSC_CR2 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E2h	OSC_CR2	PLLGAIN					EXTCLKEN	RSVD	SYSCLKX2DIS	RW : 00

The Oscillator Control Register 2 (OSC_CR2) is used to configure various features of internal clock sources and clock nets.

Bit 7: PLLGAIN. This is the only bit in the OSC_CR2 register that directly influences the PLL. When set, this bit keeps the PLL in Low Gain mode.

If this bit is held low, the lock time is less than 10 ms. If this bit is held high, the lock time is on the order of 50 ms. After lock is achieved, it is recommended that this bit be forced high to decrease the jitter on the output. If longer lock time is tolerable, the PLLGAIN bit can be held high all the time.

Bit 2: EXTCLKEN. When the EXTCLKEN bit is set, the external clock becomes the source for the internal clock tree, SYSCLK, which drives most PSoC device clocking functions. All external and internal signals, including the 32 kHz clock, whether derived from the Internal Low Speed Oscillator (ILO) or the crystal oscillator, are synchronized to this clock source. If an external clock is enabled, PLL mode should be off.

The external clock input is located on port P1[4]. When using this input, the pin Drive mode should be set to High Z (not High Z analog).

Bit 1: RSVD. Reserved bit - This bit should always be 0.

Bit 0: SYSCLKX2DIS. When SYSCLKX2DIS is set, the IMO’s doubler is disabled. This results in a reduction of overall device power, on the order of 1 mA. It is advised that any application that does not require this doubled clock should have it turned off. During emulation with the In-Circuit Emulator (ICE), the IMO’s doubler is always active regardless of the status of SYSCLKX2DIS.

For additional information, refer to the OSC_CR2 register on page 153.

12. Sleep and Watchdog

This chapter discusses the Sleep and Watchdog operations and their associated registers. For a complete table of the Sleep and Watchdog registers, refer to the "Summary Table of the Core Registers" on page 32. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

12.1 Architectural Description

Device components that are involved in Sleep and Watchdog operation are the selected 32 kHz clock (external crystal or internal), the sleep timer, the Sleep bit in the CPU_SCR0 register, the sleep circuit (to sequence going into and coming out of sleep), the bandgap refresh circuit (to periodically refresh the reference voltage during sleep), and the ***watchdog timer***.

The goal of sleep operation is to reduce average power consumption as much as possible. The system has a sleep state that can be initiated under firmware control. In this state, the CPU is stopped at an instruction boundary and the 24/48 MHz oscillator (IMO), the Flash memory module, and bandgap voltage reference are powered down. The only blocks that remain in operation are the 32 kHz oscillator (external crystal or internal), ***PSoC blocks*** clocked from the 32 kHz clock selection, and the supply voltage monitor circuit.

Analog PSoC blocks have individual power down settings that are controlled by firmware, independently of the sleep state. Continuous time analog blocks may remain in operation, since they do not require a clock source. Typically, switched capacitor analog blocks do not operate, since the internal sources of clocking for these blocks are stopped.

The system can only wake up from sleep as a result of an interrupt or reset event. The sleep timer can provide periodic interrupts to allow the system to wake up, poll peripherals, or do real-time functions, and then go to sleep again. The GPIO (pin) interrupt, supply monitor interrupt, analog column interrupts, and timers clocked externally or from the 32 kHz clock are examples of ***asynchronous*** interrupts that can also be used to wake the system up.

The Watchdog Timer (WDT) circuit is designed to assert a ***hardware reset*** to the device after a pre-programmed interval, unless it is periodically serviced in firmware. In the event that an unexpected execution path is taken through the code, this functionality serves to reboot the system. It also restarts the system from the CPU halt state.

Once the WDT is enabled, it is only disabled by a Power On Reset (POR) or an External Reset (XRES). A WDT reset leaves the WDT enabled. Therefore, if the WDT is used in an application, all code (including initialization code) must be written as though the WDT is enabled.

12.1.1 32 kHz Clock Selection

By default, the 32 kHz clock source is the Internal Low Speed Oscillator (ILO). Optionally, the 32.768 kHz External Crystal Oscillator (ECO) may be activated. This selection is made in bit 7 of the OSC_CR0 register. Selecting the ECO as the source for the 32 kHz clock allows the sleep timer and sleep interrupt to be used in real-time clock applications. Regardless of the clock source selected, the 32 kHz clock plays a key role in sleep functionality. It runs continuously and is used to sequence system wake up. It is also used to periodically refresh the bandgap voltage during sleep.

Refer to the External Crystal Oscillator (ECO) chapter on page 21, for details on activating an external crystal oscillator.

12.1.2 Sleep Timer

The sleep timer is a 15-bit up counter clocked by the currently selected 32 kHz clock source, either the ILO or ECO. This timer is always enabled. The exception to this is within an ***ICE*** (in-circuit ***emulator***) in ***debugger*** mode and when the Stop bit in the CPU_SCR0 is set; the sleep timer is disabled, so that the user does not get continual watchdog resets when a breakpoint is hit in the debugger environment.

If the associated sleep timer interrupt is enabled, a periodic interrupt to the CPU is generated based on the sleep interval selected from the OSC_CR0 register. The sleep timer functionality does not need to be directly associated with the sleep state. It can be used as a general purpose timer interrupt regardless of sleep state.

The reset state of the sleep timer is a count value of all zeros. There are two ways to reset the sleep timer. Any hardware reset (that is, POR, XRES, or Watchdog Reset (WDR), or Internal Reset (IRES) resets the sleep timer. There is also a method that allows the user to reset the sleep timer in firmware. A write of 38h to the RES_WDT register clears the sleep timer.

Note Any write to the RES_WDT register also clears the watchdog timer.

Clearing the sleep timer may be done at anytime to synchronize the sleep timer operation to CPU processing. A good example of this is after POR. The CPU hold-off, due to voltage ramp and others, may be significant. In addition, a significant amount of program initialization may be required. However, the sleep timer starts counting immediately after POR and is at an arbitrary count when user code begins execution. In this case, it may be desirable to clear the sleep timer before enabling the sleep interrupt initially, to ensure that the first sleep period is a full interval.

12.2 Application Description

The following are notes regarding sleep as it relates to firmware and application issues.

Note 1 If an interrupt is pending, enabled, and scheduled to be taken at the instruction boundary after the write to the sleep bit, the system does not go to sleep. The instruction still executes, but it is not able to set the SLEEP bit in the CPU_SCR0 register. Instead, the interrupt is taken and the effect of the sleep instruction is ignored.

Note 2 The Global Interrupt Enable (CPU_F register) does not need to be enabled to wake the system out of sleep state. Individual interrupt enables, as set in the interrupt mask registers, are sufficient. If the Global Interrupt Enable is not set, the CPU does not service the ISR associated with that interrupt. However, the system wakes up and continues executing instructions from the point at which it went to sleep. In this case, the user must manually clear the pending interrupt or subsequently enable the Global Interrupt Enable bit and let the CPU take the ISR. If a pending interrupt is not cleared, it is continuously asserted. Although the sleep bit may be written and the sleep sequence executed as soon as the device enters Sleep mode, the Sleep bit is cleared by the pending interrupt and Sleep mode is exited immediately.

Note 3 On wake up, the instruction immediately after the sleep instruction is executed before the interrupt service routine (if enabled). The instruction after the sleep instruction is pre-fetched, before the system actually goes to sleep. Therefore, when an interrupt occurs to wake the system up, the pre-fetched instruction is executed and then the interrupt service routine is executed. (If the Global Interrupt Enable is not set, instruction execution just continues where it left off before sleep.)

Note 4 If PLL mode is enabled, CPU frequency must be reduced to 3 MHz before going to sleep. Since the PLL overshoots as it attempts to re-lock after wake up, the CPU frequency must be relatively low. It is recommended to wait 10 ms after wake up, before normal CPU operating frequency may be restored.

Note 5 Analog power must be turned off by firmware before going to sleep, to achieve the smallest sleep current. The system sleep state does not control the analog array. There are individual power controls for each analog block and global power controls in the reference block. These power controls must be manipulated by firmware.

Note 6 If the Global Interrupt Enable bit is disabled, it can be safely enabled just before the instruction that writes the sleep bit. It is usually undesirable to get an interrupt on the instruction boundary, just before writing the Sleep bit. This means that on the return from interrupt, the sleep command executes, possibly bypassing any firmware preparations that are made in order to go to sleep. To prevent this, disable interrupts before preparations are made. After sleep preparations, enable global interrupts and write the Sleep bit with the two consecutive instructions as follows.

```
and f,~01h          // disable global interrupts
                    // (prepare for sleep, could
                    // be many instructions)
or f,01h            // enable global interrupts
mov reg[ffh],08h    // Set the sleep bit
```

Due to the timing of the Global Interrupt Enable instruction, it is not possible for an interrupt to occur immediately after that instruction. The earliest the interrupt can occur is after the next instruction (write to the Sleep bit) has been executed. Therefore, if an interrupt is pending, the sleep instruction is executed; but as described in Note 1, the sleep instruction is ignored. The first instruction executed after the ISR is the instruction after sleep.

12.3 Register Definitions

The following registers are associated with Sleep and Watchdog and are listed in address order. Each register description has an associated register table showing the bit structure for that register. The bits that are grayed out in the tables below are reserved bits and are not detailed in the register descriptions. Note that reserved bits should always be written with a value of '0'. For a complete table of the Sleep and Watchdog registers, refer to the "Summary Table of the Core Registers" on page 32.

12.3.1 INT_MSK0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E0h	INT_MSK0	VC3	Sleep	GPIO	SAR8 ADC		Analog 1	Analog 0	V Monitor	RW : 00

The Interrupt Mask Register 0 (INT_MSK0) is used to enable the individual sources' ability to create pending interrupts.

Only certain bits are accessible to be read or written in the analog column dependent INT_MSK0 register. In the table above, the analog column numbers are listed to the right in the Address column.

Bits 7 and 5 to 0. The INT_MSK0 register holds bits that are used by several different resources. For a full discussion of the INT_MSK0 register, see the Interrupt Controller chapter on page 61.

Bit 6: Sleep. This bit controls the sleep interrupt enable.

For additional information, refer to the INT_MSK0 register on page 104.

12.3.2 RES_WDT Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E3h	RES_WDT	WDSL_Clear[7:0]								W : 00

The Reset Watchdog Timer Register (RES_WDT) is used to clear the watchdog timer (a write of any value) and clear both the watchdog timer and the sleep timer (a write of 38h).

Bits 7 to 0: WDSL_Clear[7:0]. The Watchdog Timer (WDT) write-only register is designed to timeout at three rollover events of the sleep timer. Therefore, if only the WDT is cleared, the next Watchdog Reset (WDR) occurs anywhere from two to three times the current sleep interval setting. If the sleep timer is near the beginning of its count, the watchdog timeout is closer to three times.

However, if the sleep timer is very close to its ***terminal count***, the watchdog timeout is closer to two times. To ensure a full three times timeout, both the WDT and the sleep timer may be cleared. In applications that need a real-time clock, and thus cannot reset the sleep timer when clearing the WDT, the duty cycle at which the WDT must be cleared should be no greater than two times the sleep interval.

For additional information, refer to the RES_WDT register on page 107.

12.3.3 CPU_SCR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW *	ECO EX *		IRAMDIS	# : 00

LEGEND
x An "x" before the comma in the address field indicates that this register can be read or written to no matter what bank is used.
# Access is bit specific. Refer to the Register Details chapter on page 47 for additional information.
* Bits 3 and 2 (ECO EXW and ECO EX, respectively) cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

The System Status and Control Register 1 (CPU_SCR1) is used to convey the status and control of events related to internal resets and watchdog reset.

Bit 7: IRESS. The Internal Reset Status bit is a read only bit that may be used to determine if the booting process occurred more than once.

When this bit is set, it indicates that the SROM SWBootReset code executed more than once. If this bit is not set, the SWBootReset executed only once. In either case, the SWBootReset code does not allow execution from code stored in Flash until the M8C Core is in a safe operating mode with respect to supply voltage and Flash operation. There is no need for concern when this bit is set. It is provided for systems which may be sensitive to boot time, so that they can determine if the normal one-pass boot time is exceeded. For more information on the SWBootReest code see the Supervisory ROM (SROM) chapter on page 45.

Bit 4: SLIMO. When set, the Slow IMO bit allows the active power dissipation of the PSoC device to be reduced by slowing down the IMO from 24 MHz to 6 MHz. The IMO trim value must also be changed when SLIMO is set. When not in external clocking mode, the IMO is the source for SYSCLK; therefore, when the speed of the IMO changes, so does SYSCLK.

Bit 3: ECO EXW. The ECO Exists Written bit is used as a status bit to indicate that the ECO EX bit has been previously written to. It is read only. When this bit is a '1', this indicates that the CPU_SCR1 register has been written to and is now locked. When this bit is a '0', the register has not been written to since the last reset event. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

Bit 2: ECO EX. The ECO Exists bit serves as a flag to the hardware, to indicate that an external crystal ***oscillator*** exists in the system. Just after boot, it may be written *only once* to a value of '1' (crystal exists) or '0' (crystal does not exist). If the bit is '0', a switch-over to the ECO is locked out by hardware. If the bit is '1', hardware allows the firmware to freely switch between the ECO and ILO. It should be written as early as possible after a ***Power On Reset (POR)*** or ***External Reset (XRES)*** event, where it is assumed that program execution integrity is high. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

Bit 0: IRAMDIS. The Initialize RAM Disable bit is a control bit that is readable and writeable. The ***default value*** for this bit is '0', which indicates that the maximum amount of SRAM should be initialized on watchdog reset to a value of 00h. When the bit is '1', the minimum amount of SRAM is initialized after a watchdog reset. For more information on this bit, see the "SROM Function Descriptions" on page 46.

For additional information, refer to the CPU_SCR1 register on page 121.

12.3.4 CPU_SCR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	# : XX

LEGEND

X	The value for power on reset is unknown.
x	An “x” before the comma in the address field indicates that this register can be read or written to no matter what bank is used.
#	Access is bit specific. Refer to register detail for additional information.

The System Status and Control Register 0 (CPU_SCR0) is used to convey the status and control of events for various functions of the PSoC device.

Bit 7: GIES. The Global Interrupt Enable Status bit is a read only status bit and its use is discouraged. The GIES bit is a legacy bit which was used to provide the ability to read the GIE bit of the CPU_F register. However, the CPU_F register is now readable. When this bit is set, it indicates that the GIE bit in the CPU_F register is also set which, in turn, indicates that the microprocessor services interrupts.

Bit 5: WDRS. The WatchDog Reset Status bit may not be set. It is normally ‘0’ and automatically set whenever a watchdog reset occurs. The bit is readable and clearable by writing a zero to its bit position in the CPU_SCR0 register.

Bit 4: PORS. The Power On Reset Status (PORS) bit, which is the watchdog enable bit, is set automatically by a POR or External Reset (XRES). If the bit is cleared by user code, the watchdog timer is enabled. Once cleared, the only way to reset the PORS bit is to go through a POR or XRES. Thus, there is no way to disable the watchdog timer, other than to go through a POR or XRES.

Bit 3: Sleep. The Sleep bit is used to enter Low Power Sleep mode when set. To wake up the system, this register bit is cleared asynchronously by any enabled interrupt. There are two special features of this register bit that ensures proper sleep operation. First, the write to set the register bit is blocked, if an interrupt is about to be taken on that instruction boundary (immediately after the write). Second, there is a hardware interlock to ensure that, once set, the Sleep bit may not be cleared by an incoming interrupt until the sleep circuit has finished performing the sleep sequence and the system-wide power down signal has been asserted. This prevents the sleep circuit from being interrupted in the middle of the process of system power down, possibly leaving the system in an indeterminate state.

Bit 0: STOP. The STOP bit is readable and writeable. When set, the PSoC M8C stops executing code until a reset event occurs. This can be either a POR or WDR, or XRES. If an application wants to stop code execution until a reset, the preferred method is to use the HALT instruction rather than a register write to this bit.

For additional information, refer to the CPU_SCR0 register on page 122.

12.3.5 OSC_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00

The Oscillator Control Register 0 (OSC_CR0) is used to configure various features of internal clock sources and clock nets.

Bit 7: 32k Select. By default, the 32 kHz clock source is the Internal Low Speed Oscillator (ILO). Optionally, the 32.768 kHz External Crystal Oscillator (ECO) may be selected.

Bit 6: PLL Mode. This is the only bit in the OSC_CR0 register that directly influences the Phase-Locked Loop (PLL). When set, this bit enables the PLL. The EXTCLKEN bit in the OSC_CR2 register should be set low during PLL operation. For information on the PLL, refer to the Phase-Locked Loop (PLL) chapter on page 27.

Bit 5: No Buzz. Normally, when the Sleep bit is set in the CPU_SCR register, all PSoC device systems are powered down, including the bandgap reference. However, to facilitate the detection of ***POR*** and ***LVD*** events at a rate higher than the sleep interval, the bandgap circuit is powered up periodically for about 60 µs at the Sleep System Duty Cycle (set in ECO_TR), which is independent of the sleep interval and typically higher. When the No Buzz bit is set, the Sleep System Duty Cycle value is overridden and the bandgap circuit is forced to be on during sleep. This results in a faster response to an LVD or POR event (continuous detection as opposed to periodic detection), at the expense of slightly higher average sleep current.

Bits 4 and 3: Sleep[1:0]. The available sleep interval selections are shown in Table 12-1. The accuracy of the sleep intervals is dependent on the accuracy of the oscillator used.

Table 12-1. Sleep Interval Selections

Sleep Interval OSC_CR[4:3]	Sleep Timer Clocks	Sleep Period (nominal)	Watchdog Period (nominal)
00b (default)	64	1.95 ms	6 ms
01b	512	15.6 ms	47 ms
10b	4,096	125 ms	375 ms
11b	32,768	1 sec	3 sec

Bits 2 to 0: CPU Speed[2:0]. The PSoC M8C operates over a range of CPU clock speeds (see Table 12-2), allowing the M8C's performance and power requirements to be tailored to the application.

The reset value for the CPU Speed bits is zero; therefore, the default CPU speed is one-eighth of the clock source. The Internal Main Oscillator (IMO) is the default clock source for the CPU speed circuit; therefore, the default CPU speed is 3 MHz.

The CPU frequency is changed with a write to the OSC_CR0 register. There are eight frequencies generated from a power-of-two divide circuit, which are selected by a 3-bit code. At any given time, the CPU 8-to-1 clock mux is selecting one of the available frequencies, which is resynchronized to the 24 MHz master clock at the output.

Regardless of the CPU Speed bit's setting, if the actual CPU speed is greater than 12 MHz, the 24 MHz operating requirements apply. An example of this scenario is a device that is configured to use an external clock, which is supplying a frequency of 20 MHz. If the CPU speed register's value is 011b, the CPU clock is 20 MHz. Therefore, the supply voltage requirements for the device are the same as if the part was operating at 24 MHz off of the IMO. The operating voltage requirements are not relaxed until the CPU speed is at 12 MHz or less.

Table 12-2. OSC_CR0[2:0] Bits: CPU Speed

Bits	Internal Main Oscillator	External Clock
000b	3 MHz	EXTCLK/ 8
001b	6 MHz	EXTCLK/ 4
010b	12 MHz	EXTCLK/ 2
011b	24 MHz	EXTCLK/ 1
100b	1.5 MHz	EXTCLK/ 16
101b	750 kHz	EXTCLK/ 32
110b	187.5 kHz	EXTCLK/ 128
111b	93.7 kHz	EXTCLK/ 256

For additional information, refer to the OSC_CR0 register on page 151.

12.3.6 ILO_TR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E9h	ILO_TR			Bias Trim[1:0]		Freq Trim[3:0]				W : 00

The Internal Low Speed Oscillator Trim Register (ILO_TR) sets the adjustment for the internal low speed oscillator.

The device-specific value, placed in the trim bits of this register at boot time, is based on factory testing. ***It is strongly recommended that the user not alter the register value***.

Bits 5 and 4: Bias Trim[1:0]. These two bits are used to set the bias current in the PTAT Current Source. Bit 5 gets inverted, so that a medium bias is selected when both bits are '0'. The bias current is set according to Table 12-3.

Table 12-3. Bias Current in PTAT

Bias Current	Bias Trim [1:0]
Medium Bias	00b
Maximum Bias	01b
Minimum Bias	10b
Not needed *	11b

* About 15% higher than the minimum bias.

Bits 3 to 0: Freq Trim[3:0]. These four bits are used to trim the frequency. Bit 0 is the LSb and bit 3 is the MSb. Bit 3 gets inverted inside the register.

For additional information, refer to the ILO_TR register on page 157.

12.3.7 ECO_TR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,EBh	ECO_TR	PSSDC[1:0]								W : 00

The External Crystal Oscillator Trim Register (ECO_TR) sets the adjustment for the 32.768 kHz external crystal oscillator.

The value placed in this register is based on factory testing. This register does not adjust the frequency of the external crystal oscillator. ***It is strongly recommended that the user not alter the register value.***

Bits 7 and 6: PSSDC[1:0]. These bits are used to set the sleep duty cycle. These bits should not be altered.

For additional information, refer to the ECO_TR register on page 159.

12.4 Timing Diagrams

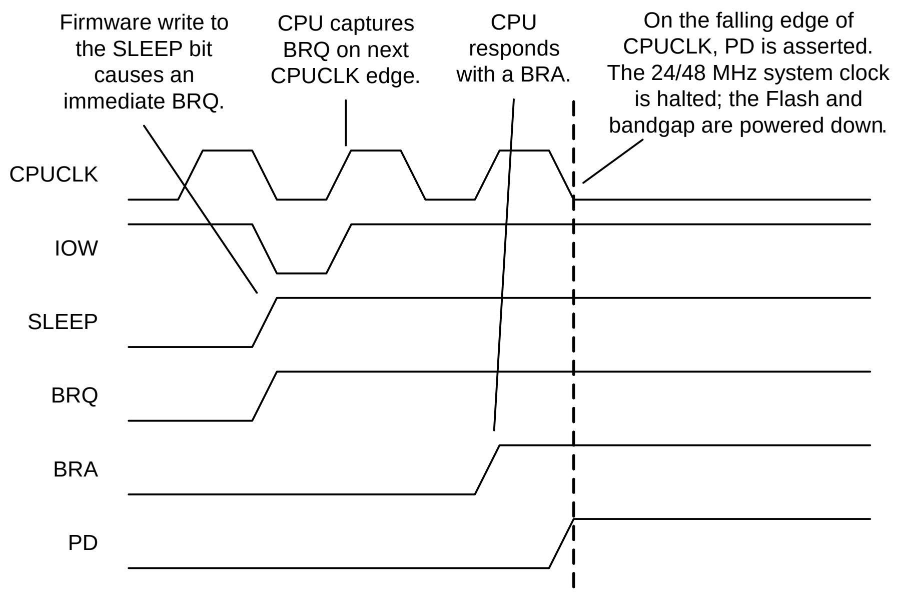
12.4.1 Sleep Sequence

The Sleep bit, in the CPU_SCR0 register, is an input into the sleep logic circuit. This circuit is designed to sequence the device into and out of the hardware sleep state. The hardware sequence to put the device to sleep is shown in Figure 12-1 and is defined as follows.

1. Firmware sets the Sleep bit in the CPU_SCR0 register. The Bus Request (BRQ) signal to the CPU is immediately asserted: This is a request by the system to halt CPU operation at an instruction boundary.
2. The CPU issues a Bus Request Acknowledge (BRA) on the following ***positive edge*** of the CPU clock.
3. The sleep logic waits for the following ***negative edge*** of the CPU clock and then asserts a system-wide Power Down (PD) signal. In Figure 12-1, the CPU is halted and the system-wide power down signal is asserted.

The system-wide PD signal controls three major circuit blocks: the Flash memory module, the Internal Main Oscillator (24/48 MHz oscillator that is also called the IMO), and the bandgap voltage reference. These circuits transition into a zero power state. The only operational circuits on the PSoC device are the ILO (or optional ECO), the bandgap refresh circuit, and the supply voltage monitor circuit. Note that the system sleep state does not apply to the analog array. Power down settings for individual analog blocks and references must be done in firmware, prior to executing the sleep instruction.

Figure 12-1. Sleep Sequence


Firmware write to the SLEEP bit causes an immediate BRQ.
CPU captures BRQ on next CPUCLK edge.
CPU responds with a BRA.
On the falling edge of CPUCLK, PD is asserted. The 24/48 MHz system clock is halted; the Flash and bandgap are powered down.
CPUCLK
IOW
SLEEP
BRQ
BRA
PD

12.4.2 Wake Up Sequence

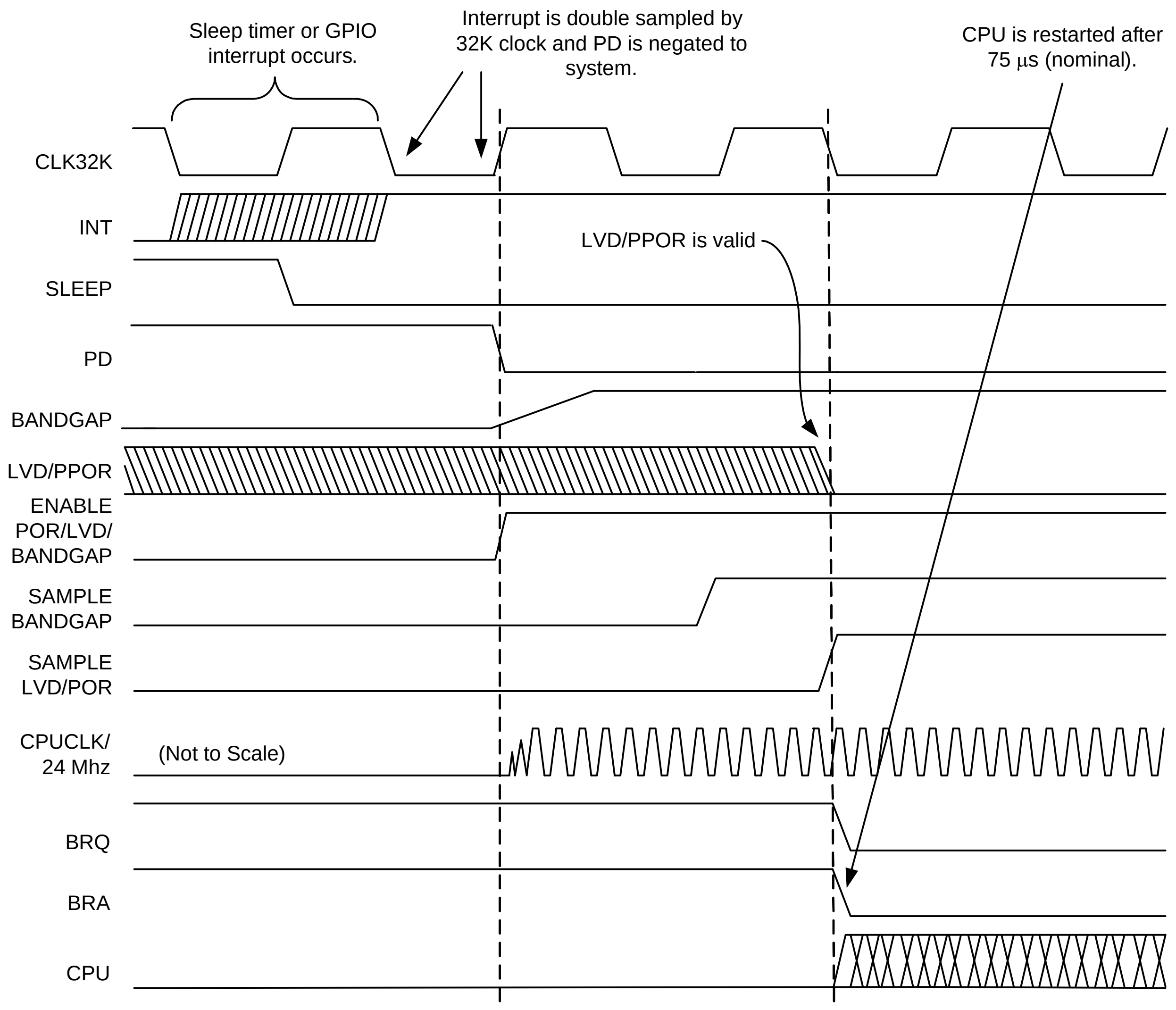
Once asleep, the only event that can wake the system up is an interrupt. The Global Interrupt Enable of the CPU flag register does not need to be set. Any unmasked interrupt wakes the system up. It is optional for the CPU to actually take the interrupt after the wake up sequence.

The wake up sequence is synchronized to the 32 kHz clock for purposes of sequencing a startup delay, to allow the Flash memory module enough time to power up before the CPU asserts the first read access. Another reason for the delay is to allow the IMO, bandgap, and LVD/POR circuits time to settle before actually being used in the system. As shown in Figure 12-2, the wake up sequence is as follows:

1. The wake up interrupt occurs and is synchronized by the negative edge of the 32 kHz clock.
2. At the following positive edge of the 32 kHz clock, the system-wide PD signal is negated. The Flash memory module, IMO, and bandgap any POR/LVD circuits are all powered up to a normal operating state.
3. At the next positive edge of the 32 kHz clock, the values of the bandgap are settled and sampled.
4. At the following negative edge of the 32 kHz clock (after about 15 µs, nominal) the values of the POR/LVD signals have settled and are sampled. The BRQ signal is negated by the sleep logic circuit. On the following CPU clock, BRA is negated by the CPU and instruction execution resumes.

The wake up times (interrupt to CPU operational) range from two to three 32 kHz cycles or 61 - 92 µs (nominal).

Figure 12-2. Wake Up Sequence

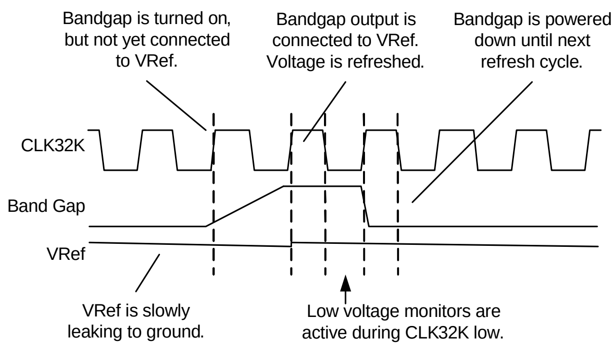

Sleep timer or GPIO interrupt occurs.
Interrupt is double sampled by 32K clock and PD is negated to system.
CPU is restarted after 75 µs (nominal).
CLK32K
INT
SLEEP
PD
BANDGAP
LVD/PPOR
LVD/PPOR is valid
ENABLE POR/LVD/ BANDGAP
SAMPLE BANDGAP
SAMPLE LVD/POR
CPUCLK/ 24 Mhz
(Not to Scale)
BRQ
BRA
CPU

12.4.3 Bandgap Refresh

During normal operation, the bandgap circuit provides a voltage reference (VRef) to the system, for use in the analog blocks, Flash, and ***low voltage detect (LVD)*** circuitry. Normally, the bandgap output is connected directly to the VRef signal. However, during sleep, the ***bandgap reference*** generator block and LVD circuits are completely powered down. The bandgap and LVD blocks are periodically re-enabled during sleep, in order to monitor for low voltage conditions. This is accomplished by turning on the bandgap periodically, allowing it time to start up for a full 32 kHz clock period, and connecting it to VRef to refresh the reference voltage for the following 32 kHz clock period as shown in Figure 12-3.

During the second 32 kHz clock period of the refresh cycle, the LVD circuit is allowed to settle during the ***high time*** of the 32 kHz clock. During the low period of the second 32 kHz clock, the LVD interrupt is allowed to occur.

Figure 12-3. Bandgap Refresh Operation


Bandgap is turned on, but not yet connected to VRef.
Bandgap output is connected to VRef. Voltage is refreshed.
Bandgap is powered down until next refresh cycle.
CLK32K
Band Gap
VRef
VRef is slowly leaking to ground.
Low voltage monitors are active during CLK32K low.

The rate at which the refresh occurs is related to the 32 kHz clock and controlled by the Power System Sleep Duty Cycle (PSSDC), bits [7:6] of the ECO_TR register). Table 12-4 enumerates the available selections. The default setting (256 sleep timer counts) is applicable for many applications, giving a typical average device current under 5 µA.

Table 12-4. Power System Sleep Duty Cycle Selections

PSSDC	Sleep Timer Counts	Period (nominal)
00b (default)	256	8 ms
01b	1024	31.2 ms
10b	64	2 ms
11b	16	500 µs

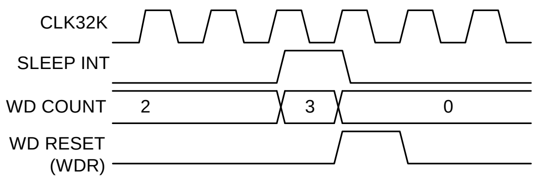
12.4.4 Watchdog Timer

On device boot up, the Watchdog Timer (WDT) is initially disabled. The PORS bit in the System Control Register controls the enabling of the WDT. On boot, the PORS bit is initially set to '1', indicating that either a POR or XRES event has occurred. The WDT is enabled by clearing the PORS bit. Once this bit is cleared and the watchdog timer is enabled, it cannot be subsequently disabled. (The PORS bit cannot be set to '1' in firmware; it can only be cleared.)

The only way to disable the Watchdog function, after it is enabled, is through a subsequent POR or XRES. Although the WDT is disabled during the first time through initialization code after a POR or XRES, all code should be written as if it is enabled (that is, the WDT should be cleared periodically). This is because, in the initialization code after a WDR event, the watchdog timer is enabled so all code must be aware of this.

The watchdog timer is three counts of the sleep timer interrupt output. The watchdog interval is three times the selected sleep timer interval. The available selections for the watchdog interval are shown in Table 12-1. When the sleep timer interrupt is asserted, the watchdog timer increments. When the counter reaches three, a terminal count is asserted. This terminal count is registered by the 32 kHz clock. Therefore, the WDR (Watchdog Reset) signal goes high after the following edge of the 32 kHz clock and held asserted for one cycle (30 µs nominal). The ***flip-flop*** that registers the WDT terminal count is not reset by the WDR signal when it is asserted, but is reset by all other resets. This timing is shown in Figure 12-4.

Figure 12-4. Watchdog Reset



Once enabled, the WDT must be periodically cleared in firmware. This is accomplished with a write to the RES_WDT register. This write is data independent, so any write clears the watchdog timer. (Note that a write of 38h also clears the sleep timer.) If for any reason the firmware fails to clear the WDT within the selected interval, the circuit asserts WDR to the device. WDR is equivalent in effect to any other reset. All internal registers are set to their reset state, see the table titled "Details of Functionality for Various Resets" on page 318. An important aspect to remember about WDT resets is that RAM initialization can be disabled (IRAMDIS in the CPU_SCR1 register). In this case, the SRAM contents are unaffected; so that when a WDR occurs, program variables are persistent through this reset.

In practical application, it is important to know that the watchdog timer interval can be anywhere between two and three times the sleep timer interval. The only way to guarantee that the WDT interval is a full three times that of the sleep interval is to clear the sleep timer (write 38h) when clearing the WDT register. However, this is not possible in applications that use the sleep timer as a real-time clock. In the case where firmware clears the WDT register without clearing the sleep timer, this can occur at any point in a given sleep timer interval. If it occurs just before the terminal count of a sleep timer interval, the resulting WDT interval is just over two times that of the sleep timer interval.

12.5 Power Consumption

Sleep mode power consumption consists of the items in the following tables.

In Table 12-5, the typical block currents shown do not represent maximums. These currents do not include any analog block currents that may be on during Sleep mode.

Table 12-5. Continuous Currents

	Current
IPOR	1 μA
ICLK32K (ILO/ECO)	1 μA

While the CLK32K can be turned off in Sleep mode, this mode is not useful since it makes it impossible to restart unless an Imprecise Power On Reset (IPOR) occurs. (The Sleep bit can not be cleared without CLK32K.) During the sleep mode buzz, the bandgap is on for two cycles and the LVD circuitry is on for one cycle. Time-averaged currents from periodic sleep mode 'buzz', with periodic count of N, are listed in Table 12-6.

Table 12-6. Time-Averaged Currents

	Current
IBG (Bandgap)	(2/N) * 60 μA
ILVD (LVD comparators)	(2/N) * 50 μA

Table 12-7 lists example currents for N=256 and N=1024. Device leakage currents add to the totals in the table.

Table 12-7. Example Currents

	N=256	N=1024
IPOR	1	1
CLK32K	1	1
IBG	0.46	0.12
ILVD	0.4	0.1
Total	2.9 μA	2.2 μA

Section C: Register Reference

The Register Reference section discusses the registers of the PSoC device. It lists all the registers in mapping tables, in address order. For easy reference, each register is linked to the page of a detailed description located in the next chapter. This section encompasses the following chapter:

Register General Conventions

The register conventions specific to this section and the Register Details chapter are listed in the following table.

Register Conventions

Convention	Description
Empty, grayed-out table cell	Illustrates a reserved bit or group of bits
'x' before the comma in an address	Indicates the register exists in register bank 1 and register bank 2
'x' in a register name	Indicates that there are multiple instances/address ranges of the same registe
R	Read register or bit(s)
W	Write register or bit(s)
L	Logical register or bit(s)
C	Clearable register or bit(s)
#	Access is bit specific

Register Naming Conventions

The register naming convention specific to this section for arrays of PSoC blocks and their registers is:

<Prefix>mn<Suffix>
where m=row index, n=column index

Therefore, ASD11CR3 is a register for an analog PSoC block in row 1 column 1.

Register Mapping Tables

The PSoC device has a total register address space of 512 bytes. The register space is also referred to as IO space and is broken into two parts. The XIO bit in the Flag register (CPU_F) determines which bank the user is currently in. When the XIO bit is set, the user is said to be in the "extended" address space or the "configuration" registers.

Register Map Bank 0 Table: User Space

Name	Addr (0,Hex)	Access	Page	Name	Addr (0,Hex)	Access	Page	Name	Addr (0,Hex)	Access	Page	Name	Addr (0,Hex)	Access	Page
PRT0DR	00	RW	49		40				80				C0		
PRT0IE	01	RW	50		41				81				C1		
PRT0GS	02	RW	51		42				82				C2		
PRT0DM2	03	RW	52		43				83				C3		
PRT1DR	04	RW	49		44			ASD11CR0	84	RW	79		C4		
PRT1IE	05	RW	50		45			ASD11CR1	85	RW	80		C5		
PRT1GS	06	RW	51		46			ASD11CR2	86	RW	81		C6		
PRT1DM2	07	RW	52		47			ASD11CR3	87	RW	82		C7		
PRT2DR	08	RW	49		48				88				C8		
PRT2IE	09	RW	50		49				89				C9		
PRT2GS	0A	RW	51		4A				8A				CA		
PRT2DM2	0B	RW	52		4B				8B				CB		
PRT3DR	0C	RW	49		4C				8C				CC		
PRT3IE	0D	RW	50		4D				8D				CD		
PRT3GS	0E	RW	51		4E				8E				CE		
PRT3DM2	0F	RW	52		4F				8F				CF		
	10				50				90				D0		
	11				51				91				D1		
	12				52				92				D2		
	13				53				93				D3		
	14				54			ASC21CR0	94	RW	83		D4		
	15				55			ASC21CR1	95	RW	84		D5		
	16				56			ASC21CR2	96	RW	85	I2C_CFG	D6	RW	94
	17				57			ASC21CR3	97	RW	86	I2C_SCR	D7	#	95
	18				58				98			I2C_DR	D8	RW	97
	19				59				99			I2C_MSCR	D9	#	98
	1A				5A				9A			INT_CLR0	DA	RW	99
	1B				5B				9B			INT_CLR1	DB	RW	101
	1C				5C				9C				DC		
	1D				5D				9D			INT_CLR3	DD	RW	102
	1E				5E				9E			INT_MSK3	DE	RW	103
	1F				5F				9F				DF		
DBB00DR0	20	#	53	AMX_IN	60	RW	64		A0			INT_MSK0	E0	RW	104
DBB00DR1	21	W	54		61				A1			INT_MSK1	E1	RW	105
DBB00DR2	22	RW	55		62				A2			INT_VC	E2	RC	106
DBB00CR0	23	#	56	ARF_CR	63	RW	65		A3			RES_WDT	E3	W	107
DBB01DR0	24	#	53	CMP_CR0	64	#	66		A4			DEC_DH	E4	RC	108
DBB01DR1	25	W	54	ASY_CR	65	#	67		A5			DEC_DL	E5	RC	109
DBB01DR2	26	RW	55	CMP_CR1	66	RW	68		A6			DEC_CR0	E6	RW	110
DBB01CR0	27	#	56	SARADC_DL	67	RW	69		A7			DEC_CR1	E7	RW	111
DCB02DR0	28	#	53						A8			MUL0_X	E8	W	112
DCB02DR1	29	W	54	SARADC_CR0	69	#	70		A9			MUL0_Y	E9	W	113
DCB02DR2	2A	RW	55	SARADC_CR1	6A	RW	71		AA			MUL0_DH	EA	R	114
DCB02CR0	2B	#	56		6B				AB			MUL0_DL	EB	R	115
DCB03DR0	2C	#	53	TMP_DR0	6C	RW	72		AC			ACC0_DR1	EC	RW	116
DCB03DR1	2D	W	54	TMP_DR1	6D	RW	72		AD			ACC0_DR0	ED	RW	117
DCB03DR2	2E	RW	55	TMP_DR2	6E	RW	72		AE			ACC0_DR3	EE	RW	118
DCB03CR0	2F	#	56	TMP_DR3	6F	RW	72		AF			ACC0_DR2	EF	RW	119
	30			ACB00CR3	70	RW	73	RDI0RI	B0	RW	87		F0		
	31			ACB00CR0	71	RW	74	RDI0SYN	B1	RW	88		F1		
	32			ACB00CR1	72	RW	76	RDI0IS	B2	RW	89		F2		
	33			ACB00CR2	73	RW	78	RDI0LT0	B3	RW	90		F3		
	34			ACB01CR3	74	RW	73	RDI0LT1	B4	RW	91		F4		
	35			ACB01CR0	75	RW	74	RDI0RO0	B5	RW	92		F5		
	36			ACB01CR1 *	76	RW	76	RDI0RO1	B6	RW	93		F6		
	37			ACB01CR2 *	77	RW	78		B7			CPU_F	F7	RL	120
	38				78				B8				F8		
	39				79				B9				F9		
	3A				7A				BA				FA		
	3B				7B				BB				FB		
	3C				7C				BC				FC		
	3D				7D				BD				FD		
	3E				7E				BE			CPU_SCR1	FE	#	121

Gray fields are reserved.   # Access is bit specific.

Name	Addr (0,Hex)	Access	Page	Name	Addr (0,Hex)	Access	Page	Name	Addr (0,Hex)	Access	Page	Name	Addr (0,Hex)	Access	Page
	3F				7F				BF			CPU_SCR0	FF	#	122

Gray fields are reserved. # Access is bit specific.

Register Map Bank 1 Table: Configuration Space

Name	Addr (1,Hex)	Access	Page	Name	Addr (1,Hex)	Access	Page	Name	Addr (1,Hex)	Access	Page	Name	Addr (1,Hex)	Access	Page
PRT0DM0	00	RW	123		40				80				C0		
PRT0DM1	01	RW	124		41				81				C1		
PRT0IC0	02	RW	125		42				82				C2		
PRT0IC1	03	RW	126		43				83				C3		
PRT1DM0	04	RW	123		44			ASD11CR0	84	RW	79		C4		
PRT1DM1	05	RW	124		45			ASD11CR1	85	RW	80		C5		
PRT1IC0	06	RW	125		46			ASD11CR2	86	RW	81		C6		
PRT1IC1	07	RW	126		47			ASD11CR3	87	RW	82		C7		
PRT2DM0	08	RW	123		48				88				C8		
PRT2DM1	09	RW	124		49				89				C9		
PRT2IC0	0A	RW	125		4A				8A				CA		
PRT2IC1	0B	RW	126		4B				8B				CB		
PRT3DM0	0C	RW	123		4C				8C				CC		
PRT3DM1	0D	RW	124		4D				8D				CD		
PRT3IC0	0E	RW	125		4E				8E				CE		
PRT3IC1	0F	RW	126		4F				8F				CF		
	10				50				90			GDI_O_IN	D0	RW	144
	11				51				91			GDI_E_IN	D1	RW	145
	12				52				92			GDI_O_OU	D2	RW	146
	13				53				93			GDI_E_OU	D3	RW	147
	14				54			ASC21CR0	94	RW	83		D4		
	15				55			ASC21CR1	95	RW	84		D5		
	16				56			ASC21CR2	96	RW	85		D6		
	17				57			ASC21CR3	97	RW	86		D7		
	18				58				98				D8		
	19				59				99				D9		
	1A				5A				9A				DA		
	1B				5B				9B				DB		
	1C				5C				9C				DC		
	1D				5D				9D			OSC_GO_EN	DD	RW	148
	1E				5E				9E			OSC_CR4	DE	RW	149
	1F				5F				9F			OSC_CR3	DF	RW	150
DBB00FN	20	RW	127	CLK_CR0	60	RW	133		A0			OSC_CR0	E0	RW	151
DBB00IN	21	RW	129	CLK_CR1	61	RW	134		A1			OSC_CR1	E1	RW	152
DBB00OU	22	RW	131	ABF_CR0	62	RW	135		A2			OSC_CR2	E2	RW	153
	23			AMD_CR0	63	RW	136		A3			VLT_CR	E3	RW	154
DBB01FN	24	RW	127		64				A4			VLT_CMP	E4	R	155
DBB01IN	25	RW	129		65				A5				E5		
DBB01OU	26	RW	131	AMD_CR1	66	RW	137		A6				E6		
	27			ALT_CR0	67	RW	138		A7				E7		
DCB02FN	28	RW	127		68			SARADC_TRS	A8	RW	139	IMO_TR	E8	W	156
DCB02IN	29	RW	129		69			SARADC_TRCL	A9	RW	140	ILO_TR	E9	W	157
DCB02OU	2A	RW	131		6A			SARADC_TRCH	AA	RW	141	BDG_TR	EA	RW	158
	2B				6B			SARADC_CR2	AB	#	142	ECO_TR	EB	W	159
DCB03FN	2C	RW	127	TMP_DR0	6C	RW	72	SARADC_LCR	AC	RW	143		EC		
DCB03IN	2D	RW	129	TMP_DR1	6D	RW	72						ED		
DCB03OU	2E	RW	131	TMP_DR2	6E	RW	72		AE				EE		
	2F			TMP_DR3	6F	RW	72		AF				EF		
	30			ACB00CR3	70	RW	73	RDI0RI	B0	RW	87		F0		
	31			ACB00CR0	71	RW	74	RDI0SYN	B1	RW	88		F1		
	32			ACB00CR1	72	RW	76	RDI0IS	B2	RW	89		F2		
	33			ACB00CR2	73	RW	78	RDI0LT0	B3	RW	90		F3		
	34			ACB01CR3	74	RW	73	RDI0LT1	B4	RW	91		F4		
	35			ACB01CR0	75	RW	74	RDI0RO0	B5	RW	92		F5		
	36			ACB01CR1	76	RW	76	RDI0RO1	B6	RW	93		F6		
	37			ACB01CR2 *	77	RW	78		B7			CPU_F	F7	RL	120
	38				78				B8				F8		

Gray fields are reserved. # Access is bit specific.

Name	Addr (1,Hex)	Access	Page	Name	Addr (1,Hex)	Access	Page	Name	Addr (1,Hex)	Access	Page	Name	Addr (1,Hex)	Access	Page
	39				79				B9				F9		
	3A				7A				BA			FLS_PR1	FA	RW	160
	3B				7B				BB				FB		
	3C				7C				BC				FC		
	3D				7D				BD				FD		
	3E				7E				BE			CPU_SCR1	FE	#	121
	3F				7F				BF			CPU_SCR0	FF	#	122

Gray fields are reserved.   # Access is bit specific.

13. Register Details

This chapter is a reference for all the PSoC device registers in address order, for Bank 0 and Bank 1. The most detailed descriptions of the PSoC registers are in the Register Definitions section of each chapter. The registers that are in both banks are incorporated with the Bank 0 registers, designated with an 'x', rather than a '0' preceding the comma in the address. Bank 0 registers are listed first and begin on page 49. Bank 1 registers are listed second and begin on page 123. A condensed view of all the registers is shown in the "Register Map Bank 0 Table: User Space" on page 44 and the "Register Map Bank 1 Table: Configuration Space" on page 45.

13.1 Maneuvering Around the Registers

For ease-of-use, this chapter has been formatted so that there is one register per page, although some registers use two pages. On each page, from top to bottom, there are four sections:

1. Register name and address (from lowest to highest).
2. Register table showing the bit organization, with reserved bits grayed out.
3. Written description of register specifics or links to additional register information.
4. Detailed register bit descriptions.

Note that some registers are directly related to the digital and analog functions; therefore, these registers might have more than one register table (number 2 above). This is due to the fact that the PSoC devices have different digital row and analog column characteristics which use different bits in the same register. To find out the number of digital rows and analog columns your PSoC device has, refer to the table below.

PSoC Device Characteristics

PSoC Part Number	Digital IO (max)	Digital Rows	Digital Blocks	Analog Inputs	Analog Outputs	Analog Columns	Analog Blocks
CY8C24x23A	24	1	4	12	2	2	6
CY8C24533	26	1	4	12	2	2	4[a]
CY8C23533	26	1	4	12	2	2	4[a]
CY8C23433	26	1	4	12	2	2	4[a]
CY8C24633	25	1	4	12	2	2	4[b]

a. 2 CT, 2 SC, 1 SAR8 ADC.
b. 2 CT, 2 SC, 1 SAR8 ADC.

Use the register tables, in addition to the detailed register bit descriptions, to determine which bits are reserved for some smaller PSoC devices. Reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'.

Register Conventions

The following table lists the register conventions that are specific to this chapter.

Register Conventions

Convention	Example	Description
‘x’ in a register name	ACBxxCR1	Multiple instances/address ranges of the same register
R	R : 00	Read register or bit(s)
W	W : 00	Write register or bit(s)
L	RL : 00	Logical register or bit(s)
C	RC : 00	Clearable register or bit(s)
00	RW : 00	Reset value is 0x00 or 00h
XX	RW : XX	Register is not reset
0,	0,04h	Register is in bank 0
1,	1,23h	Register is in bank 1
x,	x,F7h	Register exists in register bank 0 and register bank 1
Empty, grayed-out table cell	▨	Reserved bit or group of bits, unless otherwise stated

13.1.1 Register Naming Conventions

There are a few register naming conventions used in this manual to abbreviate repetitious register information by using a lower case ‘x’ in the register name. The convention to interpret these register names is as follows.

- For all registers, an ‘x’ before the comma in the address field indicates that the register can be accessed or written to no matter what bank is used. For example, the M8C Flag register’s (CPU_F) address is ’x,F7h’ meaning it is located in bank 0 and bank 1 at F7h.
- For digital block registers, the first ‘x’ in some register names represents either “B” for basic or “C” for communication. For rows of digital PSoC blocks and their registers, the second ‘x’ set represents <Prefix>mn<Suffix>, where m=row index, n=column index. Therefore, DCB32CR0 (written DxBxxCR0) is a digital communication register for a digital PSoC block in row 3 column 2.
- For digital row registers, the ‘x’ in the digital register’s name represents the digital row index. For example, the RDIxIS register name encompasses four registers: one for each digital row index and unique address (RDI0IS, RDI2IS, and RDI3IS).
- For analog column registers, the naming convention for the switched capacitor and continuous time registers and their arrays of PSoC blocks is <Prefix>mn<Suffix>, where m=row index, n=column index. Therefore, ASC21CR2 (written ASCxxCR2) is a register for an analog PSoC block in row 2 column 1

13.2 Bank 0 Registers

The following registers are all in bank 0 and are listed in address order. An ‘x’ before the comma in the register’s address indicates that the register can be accessed independent of the XIO bit in the CPU_F register. Registers that are in both Bank 0 and Bank 1 are listed in address order in Bank 0. For example, the RDIxLT1 register has an address of x,B4h and is in both Bank 0 and Bank 1.

13.2.1 PRTxDR

Port Data Register

Individual Register Names and Addresses:

PRT0DR : 0,00h	PRT1DR : 0,04h	PRT2DR : 0,08h	PRT3DR : 0,0Ch

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

This register allows for write or read access of the current logical equivalent of the voltage on the pin.

For additional information, refer to the “Register Definitions” on page 8 in the GPIO chapter.

Bit	Name	Description
7:0	**Data[7:0]**	Write value to port or read value from port. Reads return the state of the pin, not the value in the PRTxDR register.

13.2.2 PRTxIE

Port Interrupt Enable Register

Individual Register Names and Addresses:

PRT0IE : 0,01h	PRT1IE : 0,05h	PRT2IE : 0,09h	PRT3IE : 0,0Dh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Interrupt Enables[7:0]							

This register is used to enable or disable the interrupt enable internal to the GPIO block.

For additional information, refer to the "Register Definitions" on page 8 in the GPIO chapter.

Bit	Name	Description
7:0	**Interrupt Enables[7:0]**	A bit set in this register enables the corresponding port pin interrupt. 0 Port pin interrupt disabled for the corresponding pin. 1 Port pin interrupt enabled for the corresponding pin.

13.2.3 PRTxGS

Port Global Select Register

Individual Register Names and Addresses:

PRT0GS : 0,02h	PRT1GS : 0,06h	PRT2GS : 0,0Ah	PRT3GS : 0,0Eh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Global Select[7:0]							

This register is used to select the block for connection to global inputs or outputs.

For additional information, refer to the “Register Definitions” on page 8 in the GPIO chapter.

Bit	Name	Description
7:0	**Global Select[7:0]**	A bit set in this register connects the corresponding port pin to an internal global bus. This connection is used to input or output digital signals to or from the digital blocks. 0 Global function disabled. The pin value is determined by the PRTxDR bit value and port configuration registers. 1 Global function enabled. Direction depends on mode bits for the pin (registers PRTxDM0, PRTxDM1, and PRTxDM2).

13.2.4 PRTxDM2

Port Drive Mode Bit 2 Register

Individual Register Names and Addresses:

PRT0DM2 : 0,03h	PRT1DM2 : 0,07h	PRT2DM2 : 0,0Bh	PRT3DM2 : 0,0Fh

	7	6	5	4	3	2	1	0
Access : POR	RW : FF							
Bit Name	Drive Mode 2[7:0]							

This register is one of three registers whose combined value determines the unique Drive mode of each bit in a GPIO port.

In this register, there are eight possible drive modes for each port pin. Three mode bits are required to select one of these modes, and these three bits are spread into three different registers (the PRTxDM0 register on page 123, the PRTxDM1 register on page 124, and the PRTxDM2 register). The bit position of the affected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the three drive mode register bits that control the Drive mode for that pin (for example: PRT0DM0[2], PRT0DM1[2], and PRT0DM2[2]). The three bits from the three registers are treated as a group. These are referred to as DM2, DM1, and DM0, or together as DM[2:0].

All Drive mode bits are shown in the sub-table below ([**2**10] refers to the combination (in order) of bits in a given bit position); however, this register only controls the ***most significant bit (MSb)*** of the Drive mode.

For additional information, refer to the “Register Definitions” on page 8 in the GPIO chapter.

Bit	Name	Description
7:0	**Drive Mode 2[7:0]**	Bit 2 of the Drive mode, for each pin of an 8-bit GPIO port.

[210]	**Pin Output High**	**Pin Output Low**	**Notes**
000b	Strong	Resistive	
001b	Strong	Strong	
010b	High Z	High Z	Digital input enabled.
011b	Resistive	Strong	
100b	Slow + strong	High Z	
101b	Slow + strong	Slow + strong	
110b	High Z	High Z	Reset state. Digital input disabled for zero power.
111b	High Z	Slow + strong	I2C Compatible mode.

Note A bold digit, in the table above, signifies that the digit is used in this register.

13.2.5 DxBxxDR0

Digital Basic/Communication Type B Block Data Register 0

Individual Register Names and Addresses:

DBB00DR0 : 0,20h DBB01DR0 : 0,24h DCB02DR0 : 0,28h DCB03DR0 : 0,2Ch

	7	6	5	4	3	2	1	0
Access : POR	R : 00							
Bit Name	Data[7:0]							

This register is the data register for a digital block.

The use of this register is dependent on which function is selected for its block. This selection is made in the FN[2:0] bits of the DxBxxFN register on page 127. (For the Timer, Counter, Dead Band, and CRCPRS functions, a read of the DxBxxDR0 register returns 00h and transfers DxBxxDR0 to DxBxxDR2.)

The naming convention for the digital basic/communication and control registers is as follows. The first ‘x’ in the digital register’s name represents either “B” for basic or “C” for communication. For rows of digital PSoC blocks and their registers, the second ‘x’ set represents <Prefix>mn<Suffix>, where m=row index, n=column index. Therefore, DBB21DR0 is a digital basic register for a digital PSoC block in row 2 column 1. For additional information, refer to the “Register Definitions” on page 187 in the Digital Blocks chapter.

Bit	Name	Description
7:0	**Data[7:0]**	Data for selected function.

Block Function	**Register Function**	**DCB Only**
Timer	Count Value	No
Counter	Count Value	No
Dead Band	Count Value	No
CRCPRS	LFSR *	No
SPIM	Shifter	Yes
SPIS	Shifter	Yes
TXUART	Shifter	Yes
RXUART	Shifter	Yes

* ***Linear Feedback Shift Register (LFSR)***

13.2.6 DxBxxDR1

Digital Basic/Communication Type B Block Data Register 1

Individual Register Names and Addresses:

DBB00DR1 : 0,21h	DBB01DR1 : 0,25h	DCB02DR1 : 0,29h	DCB03DR1 : 0,2Dh

	7	6	5	4	3	2	1	0
Access : POR	W : 00							
Bit Name	Data[7:0]							

This register is the data register for a digital block.

The use of this register is dependent on which function is selected for its block. This selection is made in the FN[2:0] bits of the DxBxxFN register on page 127. Refer to the DxBxxDR0 register on page 53 for naming convention and digital row availability information. For additional information, refer to the "Register Definitions" on page 187 in the Digital Blocks chapter.

Bit	Name	Description
7:0	**Data[7:0]**	Data for selected function.

Block Function	**Register Function**	**DCB Only**
Timer	Period	No
Counter	Period	No
Dead Band	Period	No
CRCPRS	Polynomial	No
SPIM	TX Buffer	Yes
SPIS	TX Buffer	Yes
TXUART	TX Buffer	Yes
RXUART	Not applicable	Yes

13.2.7 DxBxxDR2

Digital Basic/Communication Type B Block Data Register 2

Individual Register Names and Addresses:

DBB00DR2 : 0,22h	DBB01DR2 : 0,26h	DCB02DR2 : 0,2Ah	DCB03DR2 : 0,2Eh

	7	6	5	4	3	2	1	0
Access : POR	RW* : 00							
Bit Name	Data[7:0]							

This register is the data register for a digital block.

The use of this register is dependent on which function is selected for its block. This selection is made in the FN[2:0] bits of the DxBxxFN register on page 127. Refer to the DxBxxDR0 register on page 53 for naming convention and digital row availability information. For additional information, refer to the “Register Definitions” on page 187 in the Digital Blocks chapter.

* If the block is configured as SPIM, SPIS, or RXUART, this register is read only.

Bit	Name	Description
7:0	**Data[7:0]**	Data for selected function.

Block Function	**Register Function**	**DCB Only**
Timer	Capture/Compare	No
Counter	Compare	No
Dead Band	Buffer	No
CRCPRS	Seed/Residue	No
SPIM	RX Buffer	Yes
SPIS	RX Buffer	Yes
TXUART	Not applicable	Yes
RXUART	RX Buffer	Yes

13.2.8 DxBxxCR0 (Timer Control)

Digital Basic/Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DBB00CR0 : 0,23h   DBB01CR0 : 0,27h   DCB02CR0 : 0,2Bh   DCB03CR0 : 0,2Fh

	7	6	5	4	3	2	1	0
Access : POR						RW : 0	RW : 0	RW : 0
Bit Name						TC Pulse Width	Capture Int	Enable

This register is the Control register for a timer, if the DxBxxFN register is configured as a ‘000’.

Refer to the DxBxxDR0 register on page 53 for naming convention and digital row availability information. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 187 in the Digital Blocks chapter.

Bit	Name	Description
2	**TC Pulse Width**	Primary output: 0 Terminal Count pulse width is one-half a block clock. Supports a period value of 00h. 1 Terminal Count pulse width is one full block clock.
1	**Capture Int**	0 Interrupt is selected with Mode bit 0 in the Function (DxBxxFN) register. 1 Block interrupt is caused by a hardware ***capture*** event (overrides Mode bit 0 selection).
0	**Enable**	0 Timer is not enabled. 1 Timer is enabled.

13.2.9 DxBxxCR0 (Counter Control)

Digital Basic/Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DBB00CR0: 0,23h　DBB01CR0: 0,27h　DCB02CR0: 0,2Bh　DCB03CR0: 0,2Fh

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

This register is the Control register for a counter, if the DxBxxFN register is configured as a ‘001’.

Refer to the DxBxxDR0 register on page 53 for naming convention and digital row availability information. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 187 in the Digital Blocks chapter.

Bit	Name	Description	
0	**Enable**	0	Counter is not enabled.
		1	Counter is enabled.

13.2.10 DxBxxCR0 (Dead Band Control)

Digital Basic/Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DBB00CR0: 0,23h	DBB01CR0: 0,27h	DCB02CR0: 0,2Bh	DCB03CR0: 0,2Fh

	7	6	5	4	3	2	1	0
Access : POR						RW : 0	RW : 0	RW : 0
Bit Name						Bit Bang Clock	Bit Bang Mode	Enable

This register is the Control register for a dead band, if the DxBxxFN register is configured as a ‘100’.

Refer to the DxBxxDR0 register on page 53 for naming convention and digital row availability information. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 187 in the Digital Blocks chapter.

Bit	Name	Description
2	**Bit Bang Clock**	When Bit Bang mode is enabled, the output of this register bit is substituted for the PWM reference. This register may be toggled by user firmware to generate PHI1 and PH2 output clocks with the programmed dead time.
1	**Bit Bang Mode**	0 Dead Band Generator uses the previous block primary output as the input reference. 1 Dead Band Generator uses the Bit Bang Clock register as the input reference.
0	**Enable**	0 Dead Band Generator is not enabled. 1 Dead Band Generator is enabled.

13.2.11 DxBxxCR0 (CRCPRS Control)

Digital Basic/Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DBB00CR0: 0,23h	DBB01CR0: 0,27h	DCB02CR0: 0,2Bh	DCB03CR0: 0,2Fh

	7	6	5	4	3	2	1	0
Access : POR							RW : 0	RW : 0
Bit Name							Pass Mode	Enable

This register is the Control register for a CRCPRS, if the DxBxxFN register is configured as a '010'.

Refer to the DxBxxDR0 register on page 53 for naming convention and digital row availability information. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 187 in the Digital Blocks chapter.

Bit	Name	Description
1	**Pass Mode**	If selected, the DATA input selection is driven directly to the primary output and the block interrupt output. The CLK input selection is driven directly to the auxiliary output. 0 Normal CRC/PRS outputs. 1 Outputs are overridden.
0	**Enable**	0 CRC/PRS is not enabled. 1 CRC/PRS is enabled.

13.2.12 DCBxxCR0 (SPIM Control)

Digital Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DCB02CR0: 0,2Bh	DCB03CR0: 0,2Fh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	R : 0	R : 0	R : 1	R : 0	RW : 0	RW : 0	RW : 0
Bit Name	LSb First	Overrun	SPI Complete	TX Reg Empty	RX Reg Full	Clock Phase	Clock Polarity	Enable

This register is the Control register for an SPIM, if the DxBxxFN register is configured as a ‘110’.

The LSb First, Clock Phase, and Clock Polarity bits are configuration bits and should never be changed once the block is enabled. They can be set at the same time that the block is enabled. Refer to the DxBxxDR0 register on page 53 for naming convention and digital row availability information. For additional information, refer to the “Register Definitions” on page 187 in the Digital Blocks chapter.

Bit	Name	Description	
7	**LSb First**	This bit should not be changed during an SPI transfer.	
		0	Data is shifted out MSb first.
		1	Data is shifted out LSb first.
6	**Overrun**	0	No overrun has occurred.
		1	Overrun has occurred. Indicates that a new byte is received and loaded into the RX Buffer before the previous one is read. It is cleared on a read of this (CR0) register.
5	**SPI Complete**	0	Indicates that a byte may still be in the process of shifting out, or no transmission is active.
		1	Indicates that a byte is shifted out and all associated clocks are generated. It is cleared on a read of this (CR0) register. Optional interrupt.
4	**TX Reg Empty**	Reset state and the state when the block is disabled is ‘1’.	
		0	Indicates that a byte is currently buffered in the TX register.
		1	Indicates that a byte is written to the TX register and cleared on write of the TX Buffer (DR1) register. This is the default interrupt. This status is initially asserted on block enable; however, the TX Reg Empty interrupt occurs only after the first data byte is written and transferred into the shifter.
3	**RX Reg Full**	0	RX register is empty.
		1	A byte is received and loaded into the RX register. It is cleared on a read of the RX Buffer (DR2) register.
2	**Clock Phase**	0	Data is latched on the leading clock edge. Data changes on the trailing edge (Modes 0, 1).
		1	Data changes on the leading clock edge. Data is latched on the trailing edge (Modes 2, 3).
1	**Clock Polarity**	0	Non-inverted, clock idles low (Modes 0, 2).
		1	Inverted, clock idles high (Modes 1, 3).
0	**Enable**	0	SPI Master is not enabled.
		1	SPI Master is enabled.

13.2.13 DCBxxCR0 (SPIS Control)

Digital Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DCB02CR0: 0,2Bh DCB03CR0: 0,2Fh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	R : 0	R : 0	R : 1	R : 0	RW : 0	RW : 0	RW : 0
Bit Name	LSb First	Overrun	SPI Complete	TX Reg Empty	RX Reg Full	Clock Phase	Clock Polarity	Enable

This register is the Control register for an SPIS, if the DxBxxFN register is configured as a ‘110’.

The LSb First, Clock Phase, and Clock Polarity bits are configuration bits and should never be changed once the block is enabled. They can be set at the same time that the block is enabled. Refer to the DxBxxDR0 register on page 53 for naming convention and digital row availability information. For additional information, refer to the “Register Definitions” on page 187 in the Digital Blocks chapter.

Bit	Name	Description
7	**LSb First**	This bit should not be changed during an SPI transfer. 0 Data is shifted out MSb first. 1 Data is shifted out LSb first.
6	**Overrun**	0 No overrun has occurred. 1 Overrun has occurred. Indicates that a new byte is received and loaded into the RX Buffer before the previous one is read. It is cleared on a read of this (CR0) register.
5	**SPI Complete**	0 Indicates that a byte may still be in the process of shifting out, or no transmission is active. 1 Indicates that a byte is shifted out and all associated clocks are generated. It is cleared on a read of this (CR0) register. Optional interrupt.
4	**TX Reg Empty**	Reset state and the state when the block is disabled is ‘1’. 0 Indicates that a byte is currently buffered in the TX register. 1 Indicates that a byte is written to the TX register and cleared on write of the TX Buffer (DR1) register. This is the default interrupt. This status is initially asserted on block enable; however, the TX Reg Empty interrupt occurs only after the first data byte is written and transferred into the shifter.
3	**RX Reg Full**	0 RX register is empty. 1 A byte is received and loaded into the RX register. It is cleared on a read of the RX Buffer (DR2) register.
2	**Clock Phase**	0 Data is latched on the leading clock edge. Data changes on the trailing edge (Modes 0, 1). 1 Data changes on the leading clock edge. Data is latched on the trailing edge (Modes 2, 3).
1	**Clock Polarity**	0 Non-inverted, clock idles low (Modes 0, 2). 1 Inverted, clock idles high (Modes 1, 3).
0	**Enable**	0 SPI Slave is not enabled. 1 SPI Slave is enabled.

13.2.14 DCBxxCR0 (UART Transmitter Control)

Digital Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DCB02CR0: 0,2Bh DCB03CR0: 0,2Fh

	7	6	5	4	3	2	1	0
Access : POR	▨	▨	R : 0	R : 1	▨	RW : 0	RW : 0	RW : 0
Bit Name	▨	▨	TX Complete	TX Reg Empty	▨	Parity Type	Parity Enable	Enable

This register is the Control register for a UART transmitter, if the DxBxxFN register is configured as a ‘101’.

Refer to the DxBxxDR0 register on page 53 for naming convention and digital row availability information. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 187 in the Digital Blocks chapter. For the Receive mode definition, refer to section 13.2.15 on page 63.

Bit	Name	Description	
5	**TX Complete**	0	Indicates that a byte may still be in the process of shifting out.
		1	Indicates that a byte is shifted out and all associated framing bits are generated. Optional interrupt. Cleared on a read of this (CR0) register.
4	**TX Reg Empty**	Reset state and the state when the block is disabled is ‘1’.	
		0	Indicates that a byte is currently buffered in the TX register.
		1	Indicates that a byte is written to the TX register and cleared on write of the TX Buffer register. This is the default interrupt. TX Reg Empty interrupt occurs only after the first data byte is written and transferred into the shifter.
2	**Parity Type**	0	Even parity.
		1	Odd parity.
1	**Parity Enable**	0	Parity is not enabled.
		1	Parity is enabled, frame includes parity bit.
0	**Enable**	0	Serial Transmitter is not enabled.
		1	Serial Transmitter is enabled.

13.2.15 DCBxxCR0 (UART Receiver Control)

Digital Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DCB02CR0: 0,2Bh DCB03CR0: 0,2Fh

	7	6	5	4	3	2	1	0
Access : POR	R : 0	R : 0	R : 0	R : 0	R : 0	RW : 0	RW : 0	RW : 0
Bit Name	Parity Error	Overrun	Framing Error	RX Active	RX Reg Full	Parity Type	Parity Enable	Enable

This register is the Control register for a UART receiver, if the DxBxxFN register is configured as a ‘101’.

Refer to the DxBxxDR0 register on page 53 for naming convention and digital row availability information. For additional information, refer to the “Register Definitions” on page 187 in the Digital Blocks chapter. For the Transmit mode definition, refer to section 13.2.14 on page 62.

Bit	Name	Description	
7	**Parity Error**	0	Indicates that no parity error has occurred.
		1	Valid when RX Reg Full is set, indicating that a parity error has occurred in the received byte and cleared on a read of this (CR0) register.
6	**Overrun**	0	Indicates that no overrun has occurred.
		1	Valid when RX Reg Full is set, indicating that the byte in the RX Buffer register has not been read before the next byte is loaded. It is cleared on a read of this (CR0) register.
5	**Framing Error**	0	Indicates no framing error has occurred.
		1	Valid when RX Reg Full is set, indicating that a framing error has occurred (a logic 0 was sampled at the STOP bit, instead of the expected logic 1). It is cleared on a read of this (CR0) register.
4	**RX Active**	0	Indicates that no reception is in progress.
		1	Indicates that a reception is in progress. It is set by the detection of a START bit and cleared at the ***sampling*** of the STOP bit.
3	**RX Reg Full**	0	Indicates that the RX Buffer register is empty.
		1	Indicates that a byte is received and transferred to the RX Buffer (DR2) register. This bit is cleared when the RX Buffer register (DR2) is read by the CPU. Interrupt source.
2	**Parity Type**	0	Even parity.
		1	Odd parity.
1	**Parity Enable**	0	Parity is not enabled.
		1	Parity is enabled, frame includes parity bit.
0	**Enable**	0	Serial Receiver is not enabled.
		1	Serial Receiver is enabled.

13.2.16 AMX_IN

Analog Input Select Register

Individual Register Names and Addresses:

AMX_IN: 0,60h

2 COLUMN	7	6	5	4	3	2	1	0
Access : POR					RW : 0		RW : 0	
Bit Name					ACI1[1:0]		ACI0[1:0]	

This register controls the analog muxes that feed signals in from port pins into the analog column.

Use the register tables above, in addition to the detailed register bit descriptions below, to determine which bits are reserved for some smaller PSoC devices. Note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 243 in the Analog Input Configuration chapter.

Bits	Name	Description
3:2	**ACI1[1:0]**	Selects the Analog Column Mux 1. For 1 column, these are even inputs. 00b ACM1 P0[0] 01b ACM1 P0[2] 10b ACM1 P0[4] 11b ACM1 P0[6] **Note** ACol1Mux (ABF_CR0, Address 1,62h) 0 AC1 = ACM1 1 AC1 = ACM0
1:0	**ACI0[1:0]**	Selects the Analog Column Mux 0. For 1 column, these are odd inputs. 00b ACM0 P0[1] 01b ACM0 P0[3] 10b ACM0 P0[5] 11b ACM0 P0[7]

13.2.17 ARF_CR

Analog Reference Control Register

Individual Register Names and Addresses:

ARF_CR: 0,63h

	7	6	5	4	3	2	1	0
Access : POR		RW : 0	RW : 0			RW : 0		
Bit Name		HBE	REF[2:0]			PWR[2:0]		

This register is used to configure various features of the configurable analog references.

In the table above, note that the reserved bit is a gray table cell and is not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 246 in the Analog Reference chapter.

Bits	Name	Description
6	**HBE**	Bias level control for opamps. 0 Low bias mode for analog array. 1 High bias mode for analog array.
5:3	**REF[2:0]**	Analog Array Reference Control (values with respect to Vss). These three bits select the sources for analog ***ground*** (AGND), the high reference (RefHi), and the low reference (RefLo). The following table applies to 2 column PSoC devices:

	AGND	**RefHi**	**RefLo**
000b	Vdd/2	Vdd/2 + Bandgap	Vdd/2 - Bandgap
001b	P2[4]	P2[4] + P2[6]	P2[4] - P2[6]
010b	Vdd/2	Vdd/2 + Vdd/2	Vdd/2 - Vdd/2
011b	2 *x* Bandgap	2 *x* Bandgap + Bandgap	2 *x* Bandgap - Bandgap
100b	2 *x* Bandgap	2 *x* Bandgap + P2[6]	2 *x* Bandgap - P2[6]
101b	P2[4]	P2[4] + Bandgap	P2[4] - Bandgap
110b	Bandgap	Bandgap + Bandgap	Bandgap - Bandgap
111b	1.6 *x* Bandgap	1.6 *x* Bandgap + 1.6 *x* Bandgap	1.6 *x* Bandgap - 1.6 *x* Bandgap

Bits	Name	Description
2:0	**PWR[2:0]**	Analog Array Power Control:

	Reference	**CT Block**	**SC Blocks**
000b	Off	Off	Off
001b	Low	On	Off
010b	Medium	On	Off
011b	High	On	Off
100b	Off	Off	Off
101b	Low	On	On
110b	Medium	On	On
111b	High	On	On

13.2.18 CMP_CR0

Analog Comparator Bus 0 Register

Individual Register Names and Addresses:

CMP_CR0: 0,64h

	7	6	5	4	3	2	1	0
Access : POR			R : 0				RW : 0	
Bit Name			COMP[1:0]				AINT[1:0]	

This register is used to poll the analog column comparator bits and select column interrupts.

Use the register tables above, in addition to the detailed register bit descriptions below, to determine which bits are reserved for some smaller PSoC devices. Note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 227 in the Analog Interface chapter.

Bits	Name	Description
5	**COMP[1]**	Comparator bus state for column 1. This bit is updated on the rising edge of PHI2, unless the comparator latch disable bits are set (refer to the CLDISx bits in the CMP_CR1 register). If the comparator latch disable bits are set, then this bit is transparent to the comparator bus in the analog array.
4	**COMP[0]**	Comparator bus state for column 0. This bit is updated on the rising edge of PHI2, unless the comparator latch disable bits are set (refer to the CLDISx bits in the CMP_CR1 register). If the comparator latch disable bits are set, then this bit is transparent to the comparator bus in the analog array.
1	**AINT[1]**	Controls the selection of the analog comparator interrupt for column 1. 0 The comparator data bit from the column is the input to the interrupt controller. 1 The falling edge of PHI2 for the column is the input to the interrupt controller.
0	**AINT[0]**	Controls the selection of the analog comparator interrupt for column 0. 0 The comparator data bit from the column is the input to the interrupt controller. 1 The falling edge of PHI2 for the column is the input to the interrupt controller.

13.2.19 ASY_CR

Analog Synchronization Control Register

Individual Register Names and Addresses:

ASY_CR: 0,65h

	7	6	5	4	3	2	1	0
Access : POR		W : 0			RW : 0	RW : 0		RW : 0
Bit Name		SARCNT[2:0]			SARSIGN	SARCOL[1:0]		SYNCEN

This register is used to control SAR operation, except for the SYNCEN bit, which is associated with analog register write stalling.

Use the register table above, in addition to the detailed register bit descriptions below, to determine which bits are reserved for some smaller PSoC devices. Note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 227 in the Analog Interface chapter.

Bits	Name	Description
6:4	**SARCNT[2:0]**	Initial SAR count. This field is initialized to the number of SAR bits to process. **Note** Any write to the SARCNT bits, other than '0', results in a modification of the read back of any analog register in the analog array. These bits must always be zero, except for SAR processing.
3	**SARSIGN**	This bit adjusts the SAR comparator based on the type of block addressed. In a DAC configuration with more than one analog block (more than 6 bits), this bit is set to '0' when processing the most significant block. It is set to '1' when processing the least significant block., because the least significant block is an inverting input to the most significant block.
2:1	**SARCOL[1:0]**	The selected column corresponds with the position of the SAR comparator block. Note that the comparator and DAC can be in the same block. 00b Analog Column 0 is the source for SAR comparator. 01b Analog Column 1 is the source for SAR comparator. 10b 11b
0	**SYNCEN**	Set to '1', stalls the CPU until the rising edge of PHI1, if a write to a register within an analog SC block takes place. 0 CPU stalling disabled. 1 CPU stalling enabled.

13.2.20 CMP_CR1

Analog Comparator Bus 1 Register

Individual Register Names and Addresses:

CMP_CR1: 0,66h

	7	6	5	4	3	2	1	0
Access : POR			RW : 0	RW : 0				
Bit Name			CLDIS[1]	CLDIS[0]				

This register is used to override the analog column comparator synchronization.

By default, the analog comparator bus is synchronized by the column clock and driven to the digital comparator bus for use in the digital array and the interrupt controller. The CLDIS bits are used to bypass the synchronization. This bypass mode can be used in power down operation to wake the device out of sleep, as a result of an analog column interrupt. Most devices update the comparator bus on the rising edge of PHI2.

Use the register table above, in addition to the detailed register bit descriptions below, to determine which bits are reserved for some smaller PSoC devices. Note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 227 in the Analog Interface chapter.

Bits	Name	Description
5	**CLDIS[1]**	Controls the comparator output latch, column 1. 0 Comparator bus synchronization is enabled. 1 Comparator bus synchronization is disabled.
4	**CLDIS[0]**	Controls the comparator output latch, column 0. 0 Comparator bus synchronization is enabled. 1 Comparator bus synchronization is disabled.

13.2.21 SARADC_DL

SAR8 ADC Conversion Low Byte Results Register

Individual Register Names and Addresses:

SARADC_DL : 0,67h

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

This register is the raw conversion data register of the SAR8 ADC module.

For additional information, refer to the "Register Definitions" on page 266 in the SAR8 ADC Block chapter.

Bit	Name	Description
7:0	**Data[7:0]**	The read out data of this register is dependant on the right side scale bits' setting, which are defined in the SARADC_CR2 register. The scale size setting affects the final read out results, but not the ADC conversion results in this register.

13.2.22 SARADC_CR0

SAR8 ADC Control Register 0

Individual Register Names and Addresses:

SARADC_CR0 : 0,69h

	7	6	5	4	3	2	1	0
Access : POR		RW : 00				RW : 1	RC : 0	RC : 0
Bit Name		ADC Channel[3:0]				Data Ready	Start/Busy	ADCEN

This register is used to control normal ADC operation and show ADC status.

In the table above, note that the reserved bit is a grayed table cell and is not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 266 in the SAR8 ADC Block chapter.

Bit	Name	Description	
6:3	**ADC Channel[3:0]**	ADC channel is:	
		0000	P0[0]
		0001	P0[1]
		0010	P0[2]
		0011	P0[3]
		0100	P0[4]
		0101	P0[5]
		0110	P0[6]
		0111	P0[7]
		1000	CT1
		1001	CT2
2	**Data Ready**	0	New conversion data is not ready or the conversion data has already been read out. Writing ‘1’ to Start/Busy bit automatically clears bit to ‘0’.
		1	ADC has newest conversion data, which has never been read out.
1	**Start/Busy**	0	ADC has finished the operation. If firmware writes a '1' to this bit it means that firmware triggers the ADC to perform the sample and conversion from the next system clock cycle. If this bit is already '1' (in ADC conversion mode), the new write of '1' forces the ADC to cancel the ongoing conversion and restart one new conversion from the next system clock cycle. When the ADC is set to free running, this signal means start a new ADC conversion. The ADC then automatically starts the new conversion after it finishes the previous conversion.
		1	ADC is busy.
0	**ADCEN**	The ADC automatically enters low power mode right after it finishes the conversion even in Enable mode. If the ADC has been disabled, it does not perform any operation.	
		0	Disable ADC operation.
		1	Enable ADC operation.

13.2.23 SARADC_CR1

SAR8 ADC Control Register 1

Individual Register Names and Addresses:

SARADC_CR1 : 0,6Ah

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0				RW : 0	RW : 0	RW : 0
Bit Name	PWRSELADC	PWRSELR2R				Align Source[1:0]		Align Enable

This register is used to control normal ADC operation and show ADC status.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 266 in the SAR8 ADC Block chapter.

Bit	Name	Description	
7	**PWRSELADC**	0	ADC analog block obtains power supply from inside VPWR.
		1	ADC analog block obtains power supply from P3[0]. P3[0] must supply no less than 2 mA current.
6	**PWRSELR2R**	0	ADC R2R reference generation block obtains power supply from inside VPWR. P3[0] must be less than or equal to VPWR, and ADC R2R reference generation block power supply must be less than or equal to that of the ADC analog block.
		1	ADC R2R reference generation block obtains power supply from P3[0]. P3[0] must supply no less than 0.3 mA current.
2:1	**Align Source [1:0]**	00	Low and high channels are completely independent. Both can trigger ADC. Each is 8-bit accuracy.
		01	Only low channel can trigger ADC. It’s 8-bit accuracy.
		10	Only high channel can trigger ADC. It’s 8-bit accuracy.
		11	High and low channels combine together to form a 16-bit trigger source. It’s 16-bit accuracy.
0	**Align Enable**	0	Auto align function is disabled.
		1	Auto align function is enabled.

13.2.24 TMP_DRx

Temporary Data Register

Individual Register Names and Addresses:

TMP_DR0 : x,6Ch	TMP_DR1 : x,6Dh	TMP_DR2 : x,6Eh	TMP_DR3 : x,6Fh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

This register is used to enhance the performance in multiple SRAM page PSoC devices.

All bits in this register are reserved for PSoC devices with 256 bytes of SRAM. Refer to the table titled "PSoC Device SRAM Availability" on page 55. For additional information, refer to the "Register Definitions" on page 58 in the RAM Paging chapter.

Bit	Name	Description
7:0	**Data[7:0]**	General purpose register space.

13.2.25 ACBxxCR3

Analog Continuous Time Type B Block Control Register 3

Individual Register Names and Addresses:

ACB00CR3 : x,70h ACB01CR3 : x,74h

	7	6	5	4	3	2	1	0
Access : POR					RW : 0	RW : 0	RW : 0	RW : 0
Bit Name					LPCMPEN	CMOUT	INSAMP	EXGAIN

This register is one of four registers used to configure a type B continuous time PSoC block.

The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index, n=column index; therefore, ACB01CR3 is a register for an analog PSoC block in row 0 column 1. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 251 in the Continuous Time Block chapter.

Bits	Name	Description	
3	**LPCMPEN**	0	Low power comparator is disabled.
		1	Low power comparator is enabled.
2	**CMOUT**	0	No connection to column output.
		1	Connect Common mode to column output.
1	**INSAMP**	0	Normal mode.
		1	Connect amplifiers across column to form an Instrumentation Amp.
0	**EXGAIN**	0	Standard Gain mode.
		1	High Gain mode (see the ACBxxCR0 register on page 74).

13.2.26 ACBxxCR0

Analog Continuous Time Type B Block Control Register 0

Individual Register Names and Addresses:

ACB00CR0 : x,71h ACB01CR0 : x,75h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0				RW : 0	RW : 0	RW : 0	
Bit Name	RTapMux[3:0]				Gain	RTopMux	RBotMux[1:0]	

This register is one of four registers used to configure a type B continuous time PSoC block.

The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index, n=column index; therefore, ACB01CR0 is a register for an analog PSoC block in row 0 column 1. For additional information, refer to the "Register Definitions" on page 251 in the Continuous Time Block chapter.

Bits	Name	Description
7:4	**RTapMux[3:0]**	Encoding for selecting one of 18 resistor taps. The four bits of RTapMux[3:0] allow selection of 16 taps. The two additional ***tap*** selections are provided using ACBxxCR3 bit 0, EXGAIN. The EXGAIN bit only affects the RTapMux values 0h and 1h.

RTap	EXGAIN	Rf	Ri	Loss	Gain
0h	1	47	1	0.0208	48.000
1h	1	46	2	0.0417	24.000
0h	0	45	3	0.0625	16.000
1h	0	42	6	0.1250	8.000
2h	0	39	9	0.1875	5.333
3h	0	36	12	0.2500	4.000
4h	0	33	15	0.3125	3.200
5h	0	30	18	0.3750	2.667
6h	0	27	21	0.4375	2.286
7h	0	24	24	0.5000	2.000
8h	0	21	27	0.5625	1.778
9h	0	18	30	0.6250	1.600
Ah	0	15	33	0.6875	1.455
Bh	0	12	36	0.7500	1.333
Ch	0	9	39	0.8125	1.231
Dh	0	6	42	0.8750	1.143
Eh	0	3	45	0.9375	1.067
Fh	0	0	48	1.0000	1.000

Bits	Name	Description
3	**Gain**	Select gain or loss configuration for output tap. 0 Loss. 1 Gain.
2	**RTopMux**	Encoding for feedback resistor select. 0 Rtop to Vdd. 1 Rtop to opamp's output.

(continued on next page)

13.2.26 ACBxxCR0 *(continued)*

1:0 **RBotMux[1:0]**

Encoding for feedback resistor select. Bits [1:0] are overridden if bit 1 of the ACBxxCR3 register is set. In that case, the bottom of the resistor string is connected across columns. Note that available mux inputs vary by individual PSoC block. In the table below, only columns ACB00 and ACB01 are used by the 2 column analog PSoC blocks.

	ACB00	**ACB01**
00b	ACB01	ACB00
01b	AGND	AGND
10b	Vss	Vss
11b	Vss	ASD11

13.2.27 ACBxxCR1

Analog Continuous Time Type B Block Control Register 1

Individual Register Names and Addresses:

ACB00CR1 : x,72h ACB01CR1 : x,76h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0			RW : 0		
Bit Name	AnalogBus	CompBus	NMux[2:0]			PMux[2:0]		

This register is one of four registers used to configure a type B continuous time PSoC block.

The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index, n=column index; therefore, ACB01CR1 is a register for an analog PSoC block in row 0 column 1. For additional information, refer to the "Register Definitions" on page 251 in the Continuous Time Block chapter.

Bits	Name	Description
7	**AnalogBus**	Enable output to the analog bus. 0 Disable output to analog column bus. 1 Enable output to analog column bus.
6	**CompBus**	Enable output to the comparator bus. 0 Disable output to comparator bus. 1 Enable output to comparator bus.
5:3	**NMux[2:0]**	Encoding for negative input select. Note that available mux inputs vary by individual PSoC block. In the table below, only columns ACB00 and ACB01 are used by the 2 column analog PSoC blocks.

	ACB00	**ACB01**
000b	ACB01	ACB00
001b	AGND	AGND
010b	RefLo	RefLo
011b	RefHi	RefHi
100b	FB#	FB#
101b	Vss	ASD11
110b	ASD11	Vss
111b	Port Inputs	Port Inputs

# Feedback point from tap of the feedback resistor as defined by corresponding CR0 bits [7:4] and CR3 bit 0.

(continued on next page)

13.2.27 ACBxxCR1 *(continued)*

2:0	**PMux[2:0]**	Encoding for positive input select. Note that available mux inputs vary by individual PSoC block.

The following table is used by the 2 column analog PSoC blocks.

	ACB00	**ACB01**
000b	RefLo	Vss
001b	Port Inputs	Port Inputs
010b	ACB01	ACB00
011b	AGND	AGND
100b	Vss	ASD11
101b	ASD11	Vss
110b	ABUS0	ABUS1
111b	FB#	FB#

# Feedback point from tap of the feedback resistor as defined by corresponding CR0 bits [7:4] and CR3 bit 0.

13.2.28 ACBxxCR2

Analog Continuous Time Type B Block Control Register 2

Individual Register Names and Addresses:

ACB00CR2 : x,73h ACB01CR2 : x,77h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0		RW : 0	
Bit Name	CPhase	CLatch	CompCap	TMUXEN	TestMux[1:0]		PWR[1:0]	

This register is one of four registers used to configure a type B continuous time PSoC block.

The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index, n=column index; therefore, ACB01CR2 is a register for an analog PSoC block in row 0 column 1. For additional information, refer to the "Register Definitions" on page 251 in the Continuous Time Block chapter.

Bits	Name	Description
7	**CPhase**	0 Comparator Control latch is transparent on PHI1. 1 Comparator Control latch is transparent on PHI2.
6	**CLatch**	0 Comparator Control latch is always transparent. 1 Comparator Control latch is active.
5	**CompCap**	0 Comparator Mode. 1 Opamp Mode.
4	**TMUXEN**	Test Mux. 0 Disabled. 1 Enabled.
3:2	**TestMux[1:0]**	Select block bypass mode. Note that available mux inputs vary by individual PSoC block and TMUXEN must be set. In the table below, columns ACB00 and ACB01 are used by the 2 column PSoC blocks.
1:0	**PWR[1:0]**	Encoding for selecting one of four power levels. High Bias mode doubles the power at each of these settings. See bit 6 in the ARF_CR register on page 65. 00b Off. 01b Low. 10b Medium. 11b High.

TestMux[1:0]:

		ACB00	**ACB01**
00b	Positive Input to	ABUS0	ABUS1
01b	AGND to	ABUS0	ABUS1
10b	RefLo to	ABUS0	ABUS1
11b	RefHi to	ABUS0	ABUS1

13.2.29 ASDxxCR0

Analog Switch Cap Type D Block Control Register 0

Individual Register Names and Addresses:

ASD11CR0 : x,84h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 00				
Bit Name	FCap	ClockPhase	ASign	ACap[4:0]				

This register is one of four registers used to configure a type D switched capacitor PSoC block.

The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index, n=column index; therefore, ASD11CR0 is a register for an analog PSoC block in row 1 column 1. For additional information, refer to the "Register Definitions" on page 258 in the Switched Capacitor Block chapter.

Bits	Name	Description
7	**FCap**	F Capacitor value selection bit. 0 16 capacitor units. 1 32 capacitor units.
6	**ClockPhase**	The ClockPhase controls the clock phase of the comparator within the switched capacitor blocks, as well as the clock phase of the switches. 0 ***Switch phasing*** is Internal PHI1 = External PHI1. Comparator Capture Point Event is triggered by Falling PHI2 and Comparator Output Point Event is triggered by Rising PHI1. 1 Switch phasing is Internal PHI1 = External PHI2. Comparator Capture Point Event is triggered by Falling PHI1 and Comparator Output Point Event is triggered by Rising PHI2.
5	**ASign**	0 Input sampled on Internal PHI1. Reference Input sampled on Internal PHI2. Positive gain. 1 Input sampled on Internal PHI2. Reference Input sampled on Internal PHI1. Negative gain.
4:0	**ACap[4:0]**	Binary encoding for 32 possible capacitor sizes for capacitor ACap.

13.2.30 ASDxxCR1

Analog Switch Cap Type D Block Control Register 1

Individual Register Names and Addresses:

ASD11CR1 : x,85h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0			RW : 00				
Bit Name	AMux[2:0]			BCap[4:0]				

This register is one of four registers used to configure a type D switched capacitor PSoC block.

The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index, n=column index; therefore, ASD11CR1 is a register for an analog PSoC block in row 1 column 1. For additional information, refer to the “Register Definitions” on page 258 in the Switched Capacitor Block chapter.

Bits	Name	Description
7:5	**AMux[2:0]**	Encoding for selecting A and C inputs for C Type blocks and A inputs for D Type blocks. (Note that available mux inputs vary by individual PSoC block.) In the table below, only column ASD11 is used by the 2 column analog PSoC blocks.
		The following table is used by the 2 column analog PSoC blocks.
4:0	**BCap[4:0]**	Binary encoding for 32 possible capacitor sizes for capacitor BCap.

	ASD11
000b	ACB01
001b	P2[2]
010b	P2[3]
011b	ASC21
100b	RefHi
101b	ACB00
110b	Reserved
111b	Reserved

13.2.31 ASDxxCR2

Analog Switch Cap Type D Block Control Register 2

Individual Register Names and Addresses:

ASD11CR2 : x,86h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 00				
Bit Name	AnalogBus	CompBus	AutoZero	CCap[4:0]				

This register is one of four registers used to configure a type D switched capacitor PSoC block.

The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index, n=column index; therefore, ASD11CR2 is a register for an analog PSoC block in row 1 column 1. For additional information, refer to the "Register Definitions" on page 258 in the Switched Capacitor Block chapter.

Bits	Name	Description
7	**AnalogBus**	Enable output to the analog bus. Note that ClockPhase in ASDxxCR0 register, bit 6, also effects this bit: Sample + Hold mode is allowed only if ClockPhase = 0. 0 Disable output to analog column bus. 1 Enable output to analog column bus.
6	**CompBus**	Enable output to the comparator bus. 0 Disable output to comparator bus. 1 Enable output to comparator bus.
5	**AutoZero**	Bit for controlling the AutoZero switch. 0 Shorting switch is not active. Input cap branches shorted to opamp input. 1 Shorting switch is enabled during Internal PHI1. Input cap branches shorted to analog ground during Internal PHI1 and to opamp input during Internal PHI2.
4:0	**CCap[4:0]**	Binary encoding for 32 possible capacitor sizes for capacitor CCap.

13.2.32 ASDxxCR3

Analog Switch Cap Type D Block Control Register 3

Individual Register Names and Addresses:

ASD11CR3 : x,87h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0		RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	
Bit Name	ARefMux[1:0]		FSW1	FSW0	BSW	BMuxSD	PWR[1:0]	

This register is one of four registers used to configure a type D switched capacitor PSoC block.

The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index, n=column index; therefore, ASD11CR3 is a register for an analog PSoC block in row 1 column 1. For additional information, refer to the “Register Definitions” on page 258 in the Switched Capacitor Block chapter.

Bits	Name	Description
7:6	**ARefMux[1:0]**	Encoding for selecting reference input. 00b Analog ground is selected. 01b RefHi input selected. (This is usually the high reference.) 10b RefLo input selected. (This is usually the low reference.) 11b 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.)
5	**FSW1**	Bit for controlling gated switches. 0 Switch is disabled. 1 If the FSW1 bit is set to ‘1’, the state of the switch is determined by the AutoZero bit. If the AutoZero bit is ‘0’, the switch is enabled at all times. If the AutoZero bit is ‘1’, the switch is enabled only when the Internal PHI2 is high.
4	**FSW0**	Bits for controlling gated switches. 0 Switch is disabled. 1 Switch is enabled when PHI1 is high.
3	**BSW**	Enable switching in branch. 0 B branch is a continuous time path. 1 B branch is switched with Internal PHI2 sampling.
2	**BMuxSD**	Encoding for selecting B inputs. (Note that the available mux inputs vary by individual PSoC block.) In the table below, only columns ASD20 and ASD11 are used by the 2 column analog PSoC blocks. **ASD20** **ASD11** 0 ASD11 ACB00 1 ASC10 ACB01 The following table is used by the 1 column analog PSoC blocks. **ASD11** 0 Reserved 1 ACB01
1:0	**PWR[1:0]**	Encoding for selecting one of four power levels. 00b Off 10b Medium 01b Low 11b High

13.2.33 ASCxxCR0

Analog Switch Cap Type C Block Control Register 0

Individual Register Names and Addresses:

ASC21CR0 : x,94h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 00				
Bit Name	FCap	ClockPhase	ASign	ACap[4:0]				

This register is one of four registers used to configure a type C switched capacitor PSoC block.

The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index, n=column index; therefore, ASC21CR0 is a register for an analog PSoC block in row 2 column 1. For additional information, refer to the "Register Definitions" on page 258 in the Switched Capacitor Block chapter.

Bits	Name	Description
7	**FCap**	F Capacitor value selection bit. 0 16 capacitor units. 1 32 capacitor units.
6	**ClockPhase**	The ClockPhase controls the clock phase of the comparator within the switched capacitor blocks, as well as the clock phase of the switches. 0 ***Switch phasing*** is Internal PHI1 = External PHI1. Comparator Capture Point Event is triggered by Falling PHI2 and Comparator Output Point Event is triggered by Rising PHI1. 1 Switch phasing is Internal PHI1 = External PHI2. Comparator Capture Point Event is triggered by Falling PHI1 and Comparator Output Point Event is triggered by Rising PHI2.
5	**ASign**	0 Input sampled on Internal PHI1. Reference Input sampled on Internal PHI2. Positive gain. 1 Input sampled on Internal PHI2. Reference Input sampled on Internal PHI1. Negative gain.
4:0	**ACap[4:0]**	Binary encoding for 32 possible capacitor sizes for capacitor ACap.

13.2.34 ASCxxCR1

Analog Switch Cap Type C Block Control Register 1

Individual Register Names and Addresses:

ASC21CR1 : x,95h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0			RW : 00				
Bit Name	ACMux[2:0]			BCap[4:0]				

This register is one of four registers used to configure a type C switched capacitor PSoC block.

The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index, n=column index; therefore, ASC12CR1 is a register for an analog PSoC block in row 1 column 2. For additional information, refer to the "Register Definitions" on page 258 in the Switched Capacitor Block chapter.

Bits	Name	Description
7:5	**ACMux[2:0]**	Encoding to select A and C inputs. (Note that available mux inputs vary by individual PSoC block.) For 2 Column Analog PSoC Blocks:
4:0	**BCap[4:0]**	Binary encoding for 32 possible capacitor sizes of the capacitor BCap.

ASC21

	A Inputs	*C Inputs*
000b	ASD11	ASD11
001b	P2[1]	ASD11
010b	RefHi	ASD11
011b	Vtemp	ASD11
100b	P2[3]	ASD11
101b	P2[1]	ASD11
110b	ABUS1	ASD11
111b	P2[2]	ASD11

13.2.35 ASCxxCR2

Analog Switch Cap Type C Block Control Register 2

Individual Register Names and Addresses:

ASC21CR2 : x,96h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 00				
Bit Name	AnalogBus	CompBus	AutoZero	CCap[4:0]				

This register is one of four registers used to configure a type C switched capacitor PSoC block.

The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index, n=column index; therefore, ASC21CR2 is a register for an analog PSoC block in row 2 column 1. For additional information, refer to the “Register Definitions” on page 258 in the Switched Capacitor Block chapter.

Bits	Name	Description
7	**AnalogBus**	Enable output to the analog bus. Note that ClockPhase in the ASCxxCR0 register on page 83, bit 6, also affects this bit: Sample + Hold mode is allowed only if ClockPhase = 0. 0 Disable output to analog column bus. 1 Enable output to analog column bus.
6	**CompBus**	Enable output to the comparator bus. 0 Disable output to comparator bus. 1 Enable output to comparator bus.
5	**AutoZero**	Bit for controlling gated switches. 0 Shorting switch is not active. Input cap branches shorted to opamp input. 1 Shorting switch is enabled during Internal PHI1. Input cap branches shorted to analog ground during Internal PHI1 and to opamp input during Internal PHI2.
4:0	**CCap[4:0]**	Binary encoding for 32 possible capacitor sizes of the capacitor CCap.

13.2.36 ASCxxCR3

Analog Switch Cap Type C Block Control Register 3

Individual Register Names and Addresses:

ASC21CR3 : x,97h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0		RW : 0	RW : 0	RW : 0		RW : 0	
Bit Name	ARefMux[1:0]		FSW1	FSW0	BMuxSC[1:0]		PWR[1:0]	

This register is one of four registers used to configure a type C switched capacitor PSoC block.

The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index, n=column index; therefore, ASC21CR3 is a register for an analog PSoC block in row 2 column 1. For additional information, refer to the "Register Definitions" on page 258 in the Switched Capacitor Block chapter.

Bits	Name	Description
7:6	**ARefMux[1:0]**	Encoding for selecting reference input. 00b Analog ground is selected. 01b RefHi input selected. 10b RefLo input selected. 11b 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.)
5	**FSW1**	Bit for controlling the FSW1 switch. 0 Switch is disabled. 1 If the FSW1 bit is set to '1', the state of the switch is determined by the AutoZero bit. If the AutoZero bit is '0', the switch is enabled at all times. If the AutoZero bit is '1', the switch is enabled only when the Internal PHI2 is high.
4	**FSW0**	Bit for controlling the FSW0 switch. 0 Switch is disabled. 1 Switch is enabled when PHI1 is high.
3:2	**BMuxSC[1:0]**	Encoding for selecting B inputs. Note that the available mux inputs vary by individual PSoC block. For 2 Column Analog PSoC Blocks: **ASC21** 00b ASD11 01b P2[1] 10b P2[0] 11b TrefGND
1:0	**PWR[1:0]**	Encoding for selecting one of four power levels. 00b Off 01b Low 10b Medium 11b High

13.2.37 RDIxRI

Row Digital Interconnect Row Input Register

Individual Register Names and Addresses:

RDI0RI : x,B0h

	7	6	5	4	3	2	1	0
Access : POR							RW : 0	
Bit Name							RI0[1:0]	

This register is used to control the input mux that determines which global inputs drive the row inputs.

The ‘x’ in the digital register’s name represents the digital row index. For additional information, refer to the “Register Definitions” on page 173 in the Row Digital Interconnect chapter.

Bit	Name	Description
1:0	**RI0[1:0]**	Select source for row input 0. 00b GIE[0] 01b GIE[4] 10b GIO[0] 11b GIO[4]

13.2.38 RDIxSYN

Row Digital Interconnect Synchronization Register

Individual Register Names and Addresses:

RDI0SYN : x,B1h

	7	6	5	4	3	2	1	0
Access : POR								RW : 0
Bit Name								RI0SYN

This register is used to control the input synchronization.

The ‘x’ in the digital register’s name represents the digital row index. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 173 in the Row Digital Interconnect chapter.

Bit	Name	Description	
0	**RI0SYN**	0	Row input 0 is synchronized to the SYSCLK system clock.
		1	Row input 0 is passed without synchronization.

13.2.39 RDIxIS

Row Digital Interconnect Input Select Register

Individual Register Names and Addresses:

RDI0IS : x,B2h

	7	6	5	4	3	2	1	0
Access : POR			RW : 0		RW : 0	RW : 0	RW : 0	RW : 0
Bit Name			BCSEL[1:0]		IS3	IS2	IS1	IS0

This register is used to configure the inputs to the digital row LUTS and select a broadcast driver from another row if present.

The 'x' in the digital register's name represents the digital row index. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 173 in the Row Digital Interconnect chapter.

Bit	Name	Description	
5:4	**BCSEL[1:0]**	When the BCSEL value is equal to the row number, the ***tri-state*** buffer that drives the row broadcast ***net*** from the input select mux is disabled, so that one of the row's blocks may drive the local row broadcast net.	
		00b	Row 0 drives row broadcast net.
		01b	Row 1 drives row broadcast net. Reserved for 1 row PSoC devices.
		10b	Row 2 drives row broadcast net. Reserved for 1 and 2 row PSoC devices.
		11b	Row 3 drives row broadcast net. Reserved for 1, 2, and 3 row PSoC devices.
3	**IS3**	0	The 'A' input of LUT3 is RO[3].
		1	The 'A' input of LUT3 is RI[3].
2	**IS2**	0	The 'A' input of LUT2 is RO[2].
		1	The 'A' input of LUT2 is RI[2].
1	**IS1**	0	The 'A' input of LUT1 is RO[1].
		1	The 'A' input of LUT1 is RI[1].
0	**IS0**	0	The 'A' input of LUT0 is RO[0].
		1	The 'A' input of LUT0 is RI[0].

13.2.40 RDIxLT0

Row Digital Interconnect Logic Table Register 0

Individual Register Names and Addresses:

RDI0LT0 : x,B3h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0				RW : 0			
Bit Name	LUT1[3:0]				LUT0[3:0]			

This register is used to select the logic function of the digital row LUTS.

The ‘x’ in the digital register’s name represents the digital row index. For additional information, refer to the “Register Definitions” on page 173 in the Row Digital Interconnect chapter.

Bit	Name	Description
7:4	**LUT1[3:0]**	Select ***logic function*** for LUT1.

	Function
0h	FALSE
1h	A AND B
2h	A AND $\overline{B}$
3h	A
4h	$\overline{A}$ AND B
5h	B
6h	A XOR B
7h	A OR B
8h	A NOR B
9h	A XNOR B
Ah	$\overline{B}$
Bh	A OR $\overline{B}$
Ch	$\overline{A}$
Dh	$\overline{A}$ OR B
Eh	A NAND B
Fh	TRUE

Bit	Name	Description
3:0	**LUT0[3:0]**	Select logic function for LUT0.

	Function
0h	FALSE
1h	A AND B
2h	A AND $\overline{B}$
3h	A
4h	$\overline{A}$ AND B
5h	B
6h	A XOR B
7h	A OR B
8h	A NOR B
9h	A XNOR B
Ah	$\overline{B}$
Bh	A OR $\overline{B}$
Ch	$\overline{A}$
Dh	$\overline{A}$ OR B
Eh	A NAND B
Fh	TRUE

13.2.41 RDIxLT1

Row Digital Interconnect Logic Table Register 1

Individual Register Names and Addresses:

RDI0LT1 : x,B4h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0				RW : 0			
Bit Name	LUT3[3:0]				LUT2[3:0]			

This register is used to select the logic function of the digital row LUTS.

The ‘x’ in the digital register’s name represents the digital row index. For additional information, refer to the “Register Definitions” on page 173 in the Row Digital Interconnect chapter.

Bit	Name	Description
7:4	**LUT3[3:0]**	Select logic function for LUT3.

	Function
0h	FALSE
1h	A AND B
2h	A AND $\overline{B}$
3h	A
4h	$\overline{A}$ AND B
5h	B
6h	A XOR B
7h	A OR B
8h	A NOR B
9h	A XNOR B
Ah	$\overline{B}$
Bh	A OR $\overline{B}$
Ch	$\overline{A}$
Dh	$\overline{A}$ OR B
Eh	A NAND B
Fh	TRUE

Bit	Name	Description
3:0	**LUT2[3:0]**	Select logic function for LUT2.

	Function
0h	FALSE
1h	A AND B
2h	A AND $\overline{B}$
3h	A
4h	$\overline{A}$ AND B
5h	B
6h	A XOR B
7h	A OR B
8h	A NOR B
9h	A XNOR B
Ah	$\overline{B}$
Bh	A OR $\overline{B}$
Ch	$\overline{A}$
Dh	$\overline{A}$ OR B
Eh	A NAND B
Fh	TRUE

13.2.42 RDIxRO0

Row Digital Interconnect Row Output Register 0

Individual Register Names and Addresses:

RDI0RO0 : x,B5h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	GOO5EN	GOO1EN	GOE5EN	GOE1EN	GOO4EN	GOO0EN	GOE4EN	GOE0EN

This register is used to select the global nets that the row outputs drive.

The 'x' in the digital register's name represents the digital row index. For additional information, refer to the "Register Definitions" on page 173 in the Row Digital Interconnect chapter.

Bit	Name	Description	
7	**GOO5EN**	0 1	Disable Row's LUT1 output to global output. Enable Row's LUT1 output to GOO[5].
6	**GOO1EN**	0 1	Disable Row's LUT1 output to global output. Enable Row's LUT1 output to GOO[1].
5	**GOE5EN**	0 1	Disable Row's LUT1 output to global output. Enable Row's LUT1 output to GOE[5].
4	**GOE1EN**	0 1	Disable Row's LUT1 output to global output. Enable Row's LUT1 output to GOE[1].
3	**GOO4EN**	0 1	Disable Row's LUT0 output to global output. Enable Row's LUT0 output to GOO[4].
2	**GOO0EN**	0 1	Disable Row's LUT0 output to global output. Enable Row's LUT0 output to GOO[0].
1	**GOE4EN**	0 1	Disable Row's LUT0 output to global output. Enable Row's LUT0 output to GOE[4].
0	**GOE0EN**	0 1	Disable Row's LUT0 output to global output. Enable Row's LUT0 output to GOE[0].

13.2.43 RDIxRO1

Row Digital Interconnect Row Output Register 1

Individual Register Names and Addresses:

RDI0RO1 : x,B6h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	GOO7EN	GOO3EN	GOE7EN	GOE3EN	GOO6EN	GOO2EN	GOE6EN	GOE2EN

This register is used to select the global nets that the row outputs drive.

The ‘x’ in the digital register’s name represents the digital row index. For additional information, refer to the “Register Definitions” on page 173 in the Row Digital Interconnect chapter.

Bit	Name	Description	
7	**GOO7EN**	0 1	Disable Row’s LUT3 output to global output. Enable Row’s LUT3 output to GOO[7].
6	**GOO3EN**	0 1	Disable Row’s LUT3 output to global output. Enable Row’s LUT3 output to GOO[3].
5	**GOE7EN**	0 1	Disable Row’s LUT3 output to global output. Enable Row’s LUT3 output to GOE[7].
4	**GOE3EN**	0 1	Disable Row’s LUT3 output to global output. Enable Row’s LUT3 output to GOE[3].
3	**GOO6EN**	0 1	Disable Row’s LUT2 output to global output. Enable Row’s LUT2 output to GOO[6].
2	**GOO2EN**	0 1	Disable Row’s LUT2 output to global output. Enable Row’s LUT2 output to GOO[2].
1	**GOE6EN**	0 1	Disable Row’s LUT2 output to global output. Enable Row’s LUT2 output to GOE[6].
0	**GOE2EN**	0 1	Disable Row’s LUT2 output to global output. Enable Row’s LUT2 output to GOE[2].

13.2.44 I2C_CFG

I^{2}C Configuration Register

Individual Register Names and Addresses:

I2C_CFG: 0,D6h

	7	6	5	4	3	2	1	0
Access : POR		RW : 0	RW : 0	RW : 0	RW : 0		RW : 0	RW : 0
Bit Name		PSelect	Bus Error IE	Stop IE	Clock Rate[1:0]		Enable Master	Enable Slave

This register is used to set the basic operating modes, baud rate, and selection of interrupts.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 298 in the I2C chapter.

Bit	Name	Description
6	**PSelect**	I2C Pin Select. 0 P1[5] and P1[7] 1 P1[0] and P1[1] **Note** Read the I2C chapter for a discussion of the side effects of choosing the P1[0] and P1[1] pair of pins.
5	**Bus Error IE**	Bus Error Interrupt Enable. 0 Disabled. 1 Enabled. An interrupt is generated on the detection of a Bus Error.
4	**Stop IE**	Stop Interrupt Enable. 0 Disabled. 1 Enabled. An interrupt is generated on the detection of a Stop Condition.
3:2	**Clock Rate[1:0]**	00b 100K Standard Mode 01b 400K Fast Mode 10b 50K Standard Mode 11b Reserved
1	**Enable Master**	Writing a '0' to both the Enable Master and Enable Slave bits holds the I2C hardware in reset. 0 Disabled 1 Enabled
0	**Enable Slave**	Writing a '0' to both the Enable Master and Enable Slave bits holds the I2C hardware in reset. 0 Disabled 1 Enabled

13.2.45 I2C_SCR

I^{2}C Status and Control Register

Individual Register Names and Addresses:

I2C_SCR: 0,D7h

	7	6	5	4	3	2	1	0
Access : POR	RC : 0	RC : 0	RC : 0	RW : 0	RC : 0	RW : 0	RC : 0	RC : 0
Bit Name	Bus Error	Lost Arb	Stop Status	ACK	Address	Transmit	LRB	Byte Complete

This register is used by both master and slave to control the flow of data bytes and to keep track of the bus state during a transfer.

Bits in this register are held in reset until one of the enable bits in I2C_CFG is set. For additional information, refer to the “Register Definitions” on page 298 in the I2C chapter.

Bit	Name	Description
7	**Bus Error**	0 This status bit must be cleared by firmware by writing a ‘0’ to the bit position. It is never cleared by the hardware. 1 A misplaced Start or Stop condition is detected.
6	**Lost Arb**	0 This bit is set immediately on lost arbitration; however, it does not cause an interrupt. This status may be checked after the following Byte Complete interrupt. Any Start detect or a write to the Start or Restart generate bits (I2C_MSCR register), when operating in Master mode, also clears the bit. 1 Lost Arbitration.
5	**Stop Status**	0 This status bit must be cleared by firmware with write of ‘0’ to the bit position. It is never cleared by the hardware. 1 A Stop condition is detected.
4	**ACK**	Acknowledge Out. This bit is automatically cleared by hardware on a Byte Complete event. 0 NACK the last received byte. 1 ACK the last received byte
3	**Address**	0 This status bit must be cleared by firmware with write of ‘0’ to the bit position. 1 The received byte is a slave address.
2	**Transmit**	Transmit bit is set by firmware to define the direction of the byte transfer. Any Start detect or a write to the Start or Restart generate bits, when operating in Master mode, also clears the bit. 0 Receive mode 1 Transmit mode
1	**LRB**	Last Received Bit. The value of the 9th bit in a Transmit sequence, which is the Acknowledge bit from the receiver. Any Start detect or a write to the Start or Restart generate bits, when operating in Master mode, also clears the bit. 0 Last transmitted byte was ACK’ed by the receiver. 1 Last transmitted byte was NACK’ed by the receiver.

(continued on next page)

13.2.45 I2C_SCR *(continued)*

0	**Byte Complete**	Transmit/Receive Mode: 0 No completed transmit/receive since last cleared by firmware. Any Start detect or a write to the Start or Restart generate bits, when operating in Master mode, also clears the bit. Transmit Mode: 1 Eight bits of data have been transmitted and an ACK or NACK has been received. Receive Mode: 1 Eight bits of data have been received.

13.2.46 I2C_DR

I^{2}C Data Register

Individual Register Names and Addresses:

I2C_DR: 0,D8h

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

This register provides read/write access to the Shift register.

This register is read only for received data and write only for transmitted data. For additional information, refer to the "Register Definitions" on page 298 in the I2C chapter.

Bit	Name	Description
7:0	**Data[7:0]**	Read received data or write data to transmit.

13.2.47 I2C_MSCR

I^{2}C Master Status and Control Register

Individual Register Names and Addresses:

I2C_MSCR: 0,D9h

	7	6	5	4	3	2	1	0
Access : POR					R : 0	R : 0	RW : 0	RW : 0
Bit Name					Bus Busy	Master Mode	Restart Gen	Start Gen

This register implements I2C framing controls and provides Bus Busy status.

Bits in this register are held in reset until one of the enable bits in I2C_CFG is set. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 298 in the I2C chapter.

Bit	Name	Description
3	**Bus Busy**	This bit is set to the following. 0 When a Stop condition is detected (from any bus master). 1 When a Start condition is detected (from any bus master).
2	**Master Mode**	This bit is set/cleared by hardware when the device is operating as a master. 0 Stop condition detected, generated by this device. 1 Start condition detected, generated by this device.
1	**Restart Gen**	This bit is cleared by hardware when the Restart generation is complete. 0 Restart generation complete. 1 Generate a Restart condition.
0	**Start Gen**	This bit is cleared by hardware when the Start generation is complete. 0 Start generation complete. 1 Generate a Start condition and send a byte (address) to the I2C bus, if bus is not busy.

13.2.48 INT_CLR0

Interrupt Clear Register 0

Individual Register Names and Addresses:

INT_CLR0: 0,DAh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0		RW : 0	RW : 0	RW : 0
Bit Name	VC3	Sleep	GPIO	SAR8 ADC		Analog 1	Analog 0	V Monitor

This register is used to enable the individual interrupt sources' ability to clear posted interrupts.

When bits in this register are read, a '1' is returned for every bit position that has a corresponding posted interrupt. When bits in this register are written with a '0' and ENSWINT is not set, posted interrupts are cleared at the corresponding bit positions. If there was not a posted interrupt, there is no effect. When bits in this register are written with a '1' and ENSWINT is set, an interrupt is posted in the interrupt controller. Note that the ENSWINT bit is in the INT_MSK3 register on page 103.

Use the register table above, in addition to the detailed register bit descriptions below, to determine which bits are reserved for some smaller PSoC devices. Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 64 in the Interrupt Controller chapter.

Bit	Name	Description	
7	**VC3**	Read 0	No posted interrupt for Variable Clock 3.
		Read 1	Posted interrupt present for Variable Clock 3.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for Variable Clock 3.
6	**Sleep**	Read 0	No posted interrupt for sleep timer.
		Read 1	Posted interrupt present for sleep timer.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for sleep timer.
5	**GPIO**	Read 0	No posted interrupt for general purpose inputs and outputs (pins).
		Read 1	Posted interrupt present for GPIO (pins).
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for general purpose inputs and outputs (pins).
4	**SAR8 ADC**	Read 0	No posted interrupt for SAR8 ADC.
		Read 1	Posted interrupt present for SAR8 ADC.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for SAR8 ADC.

(continued on next page)

13.2.48 INT_CLR0 *(continued)*

2	**Analog 1**	Read 0	No posted interrupt for analog columns.
		Read 1	Posted interrupt present for analog columns
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for analog columns.
1	**Analog 0**	Read 0	No posted interrupt for analog columns.
		Read 1	Posted interrupt present for analog columns
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for analog columns.
0	**V Monitor**	Read 0	No posted interrupt for supply voltage monitor.
		Read 1	Posted interrupt present for supply voltage monitor.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for supply voltage monitor.

13.2.49 INT_CLR1

Interrupt Clear Register 1

Individual Register Names and Addresses:

INT_CLR1: 0,DBh

	7	6	5	4	3	2	1	0
Access : POR					RW : 0	RW : 0	RW : 0	RW : 0
Bit Name					DCB03	DCB02	DBB01	DBB00

This register is used to clear posted interrupts for digital blocks or generate interrupts. When bits in this register are read, a ‘1’ is returned for every bit position that has a corresponding posted interrupt. When bits in this register are written with a ‘0’ and ENSWINT is not set, posted interrupts are cleared at the corresponding bit positions. If there was not a posted interrupt, there is no effect. When bits in this register are written with a ‘1’ and ENSWINT is set, an interrupt is posted in the interrupt controller. Note that the ENSWINT bit is in the INT_MSK3 register on page 103.

Use the register table above, in addition to the detailed register bit descriptions below, to determine which bits are reserved for some smaller PSoC devices. Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 64 in the Interrupt Controller chapter.

Bit	Name	Description	
3	**DCB03**	Digital Communications Block type B, row 0, position 3.	
		Read 0	No posted interrupt.
		Read 1	Posted interrupt present.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt.
2	**DCB02**	Digital Communications Block type B, row 0, position 2.	
		Read 0	No posted interrupt.
		Read 1	Posted interrupt present.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt.
1	**DBB01**	Digital Basic Block type B, row 0, position 1.	
		Read 0	No posted interrupt.
		Read 1	Posted interrupt present.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt.
0	**DBB00**	Digital Basic Block type B, row 0, position 0.	
		Read 0	No posted interrupt.
		Read 1	Posted interrupt present.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt.

13.2.50 INT_CLR3

Interrupt Clear Register 3

Individual Register Names and Addresses:

INT_CLR3: 0,DDh

	7	6	5	4	3	2	1	0
Access : POR								RW : 0
Bit Name								I2C

This register is used to enable the I2C interrupt sources' ability to clear posted interrupts.

When bits in this register are read, a '1' is returned for every bit position that has a corresponding posted interrupt. When bits in this register are written with a '0' and ENSWINT is cleared, any posted interrupt is cleared. If there was not a posted interrupt, there is no effect. When bits in this register are written with a '1' and ENSWINT is set, an interrupt is posted in the interrupt controller. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 64 in the Interrupt Controller chapter.

Bit	Name	Description	
0	**I2C**	Read 0	No posted interrupt for I2C.
		Read 1	Posted interrupt present for I2C.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for I2C.

13.2.51 INT_MSK3

Interrupt Mask Register 3

Individual Register Names and Addresses:

INT_MSK3: 0,DEh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0							RW : 0
Bit Name	ENSWINT							I2C

This register is used to enable the I2C's ability to create pending interrupts and enable software interrupts.

When an interrupt is masked off, the mask bit is '0'. The interrupt still posts in the interrupt controller. Therefore, clearing the mask bit only prevents a posted interrupt from becoming a pending interrupt. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 64 in the Interrupt Controller chapter.

Bit	Name	Description	
7	**ENSWINT**	0	Disable software interrupts.
		1	Enable software interrupts.
0	**I2C**	0	Mask I2C interrupt.
		1	Unmask I2C interrupt.

13.2.52 INT_MSK0

Interrupt Mask Register 0

Individual Register Names and Addresses:

INT_MSK0: 0,E0h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0		RW : 0	RW : 0	RW : 0
Bit Name	VC3	Sleep	GPIO	SAR8 ADC		Analog 1	Analog 0	V Monitor

This register is used to enable the individual sources' ability to create pending interrupts.

When an interrupt is masked off, the mask bit is '0'. The interrupt still posts in the interrupt controller. Therefore, clearing the mask bit only prevents a posted interrupt from becoming a pending interrupt. Use the register table above, in addition to the detailed register bit descriptions below, to determine which bits are reserved for some smaller PSoC devices. Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 64 in the Interrupt Controller chapter.

Bit	Name	Description	
7	**VC3**	0	Mask VC3 interrupt.
		1	Unmask VC3 interrupt.
6	**Sleep**	0	Mask sleep interrupt.
		1	Unmask sleep interrupt.
5	**GPIO**	0	Mask GPIO interrupt.
		1	Unmask GPIO interrupt.
4	**SAR8 ADC**	0	Mask analog interrupt, SAR8 ADC.
		1	Unmask SAR8 ADC interrupt.
2	**Analog 1**	0	Mask analog interrupt, column 1.
		1	Unmask analog interrupt.
1	**Analog 0**	0	Mask analog interrupt, column 0.
		1	Unmask analog interrupt.
0	**V Monitor**	0	Mask voltage monitor interrupt.
		1	Unmask voltage monitor interrupt.

13.2.53 INT_MSK1

Interrupt Mask Register 1

Individual Register Names and Addresses:

INT_MSK1: 0,E1h

	7	6	5	4	3	2	1	0
Access : POR					RW : 0	RW : 0	RW : 0	RW : 0
Bit Name					DCB03	DCB02	DBB01	DBB00

This register is used to enable the individual sources' ability to create pending interrupts for digital blocks.

When an interrupt is masked off, the mask bit is '0'. The interrupt still posts in the interrupt controller. Therefore, clearing the mask bit only prevents a posted interrupt from becoming a pending interrupt. Use the register table above, in addition to the detailed register bit descriptions below, to determine which bits are reserved for some smaller PSoC devices. Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 64 in the Interrupt Controller chapter.

Bit	Name	Description	
3	**DCB03**	0	Mask Digital Communication Block, row 0, position 3 off.
		1	Unmask Digital Communication Block, row 0, position 3.
2	**DCB02**	0	Mask Digital Communication Block, row 0, position 2 off.
		1	Unmask Digital Communication Block, row 0, position 2.
1	**DBB01**	0	Mask Digital Basic Block, row 0, position 1 off.
		1	Unmask Digital Basic Block, row 0, position 1.
0	**DBB00**	0	Mask Digital Basic Block, row 0, position 0 off.
		1	Unmask Digital Basic Block, row 0, position 0.

13.2.54 INT_VC

Interrupt Vector Clear Register

Individual Register Names and Addresses:

INT_VC: 0,E2h

	7	6	5	4	3	2	1	0
Access : POR	RC : 00							
Bit Name	Pending Interrupt[7:0]							

This register returns the next pending interrupt and clears all pending interrupts when written.

For additional information, refer to the “Register Definitions” on page 64 in the Interrupt Controller chapter.

Bit	Name	Description	
7:0	**Pending Interrupt[7:0]**	Read	Returns vector for highest priority pending interrupt.
		Write	Clears all pending and posted interrupts.

13.2.55 RES_WDT

Reset Watchdog Timer Register

Individual Register Names and Addresses:

RES_WDT: 0,E3h

	7	6	5	4	3	2	1	0
Access : POR	W : 00							
Bit Name	WDSL_Clear[7:0]							

This register is used to clear the watchdog timer and clear both the watchdog timer and the sleep timer.

For additional information, refer to the "Register Definitions" on page 33 in the Sleep and Watchdog chapter.

Bit	Name	Description
7:0	**WDSL_Clear[7:0]**	Any write clears the watchdog timer. A write of 38h clears both the watchdog and sleep timers.

13.2.56 DEC_DH

Decimator Data High Register

Individual Register Names and Addresses:

DEC_DH: 0,E4h

	7	6	5	4	3	2	1	0
Access : POR	RC : XX							
Bit Name	Data High Byte[7:0]							

This register is a dual purpose register and is used to read the high byte of the decimator's output or clear the decimator.

When a hardware reset occurs, the internal state of the decimator is reset, but the output data registers (DEC_DH and DEC_DL) are not. For additional information, refer to the "Register Definitions" on page 293 in the Decimator chapter.

Bit	Name	Description	
7:0	**Data High Byte[7:0]**	Read	Returns the high byte of the decimator.
		Write	Clears the 16-bit accumulator values. Either the DEC_DH or DEC_DL register may be written to clear the accumulators (that is, it is not necessary to write both).

13.2.57 DEC_DL

Decimator Data Low Register

Individual Register Names and Addresses:

DEC_DL: 0,E5h

	7	6	5	4	3	2	1	0
Access : POR	RC : XX							
Bit Name	Data Low Byte[7:0]							

This register is a dual purpose register and is used to read the low byte of the decimator's output or clear the decimator.

When a hardware reset occurs, the internal state of the decimator is reset, but the output data registers (DEC_DH and DEC_DL) are not. For additional information, refer to the "Register Definitions" on page 293 in the Decimator chapter.

Bit	Name	Description	
7:0	**Data Low Byte[7:0]**	Read	Returns the low byte of the decimator.
		Write	Clears the 16-bit accumulator values. Either the DEC_DH or DEC_DL register may be written to clear the accumulators (that is, it is not necessary to write both).

13.2.58 DEC_CR0

Decimator Control Register 0

Individual Register Names and Addresses:

DEC_CR0: 0,E6h

	7	6	5	4	3	2	1	0
Access : POR			RW : 0		RW : 0	RW : 0		RW : 0
Bit Name			IGEN[1:0]		ICLKS0	DCOL[1:0]		DCLKS0

This register contains control bits to access hardware support for ADC operation.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 293 in the Decimator chapter.

Bits	Name	Description
5:4	**IGEN[1:0]**	Incremental/SSADC Gate Enable. Selects on a column basis which comparator outputs are gated with the SSADC selected PWM source. 1h Analog Column 0 2h Analog Column 1
3	**ICLKS0**	Incremental/SSADC Gate Source. This bit selects any one of the digital blocks in your device. The bit value for a digital block number that does not exist in a specific PSoC should be considered reserved. For example, a PSoC device with 2 rows may choose any block numbered 0x or 1x, but not a block numbered 2x or 3x. ICLKS0 0b Digital block 02 1b Digital block 12
2:1	**DCOL[1:0]**	Decimator Column Source. Selects the analog comparator column as a data source for the decimator. 00b Analog Column 0 01b Analog Column 1 10b Reserved 11b Reserved
0	**DCLKS0**	Decimator Latch Select. This bit selects any one of the digital blocks in your device. The bit value for a digital block number that does not exist in a specific PSoC should be considered reserved. For example, a PSoC device with 2 rows may choose any block numbered 0x or 1x, but not a block numbered 2x or 3x. **DCLKS0** 0b Digital block 02 1b Digital block 12

13.2.59 DEC_CR1

Decimator Control Register 1

Individual Register Names and Addresses:

DEC_CR1: 0,E7h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	ECNT	IDEC	ICLKS3	ICLKS2	ICLKS1	DCLKS3	DCLKS2	DCLKS1

This register is used to configure signals for ADC operation.

Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 293 in the Decimator chapter.

Bits	Name	Description
7	**ECNT**	0 Disable decimator as a counter for incremental ADC. Configure for delta sigma operation. 1 Enable decimator as a counter for incremental ADC operation.
6	**IDEC**	Invert the Digital Block Latch Control (selected by DCLKS3, DCLKS2, DCLKS1, and DCLKS0). 0 Non-inverted 1 Inverted
5:3	**ICLKSx**	Incremental/SSADC Gate Source. Along with ICLKS0 in DEC_CR0, selects any one of the digital blocks in your device. The bit value for a digital block number that does not exist in a specific PSoC should be considered reserved. For example, a PSoC device with 2 rows may choose any block numbered 0x or 1x, but not a block numbered 2x or 3x.
(5:3 *(cont.)*	**ICLKSx**	**ICLKS3, ICLKS2, ICLKS1** 000b Digital block 02 001b Digital block 12 010b Digital block 01 011b Digital block 11 100b Digital block 00 101b Digital block 10 110b Digital block 03 111b Digital block 13
2:0	**DCLKSx**	Decimator Latch Select. Along with DCLKS0 in DEC_CR0, selects any one of the digital blocks in your device. The bit value for a digital block number that does not exist in a specific PSoC should be considered reserved. For example, a PSoC device with 2 rows may choose any block numbered 0x or 1x, but not a block numbered 2x or 3x. **DCLKS3, DCLKS2, DCLKS1** 000b Digital block 02 001b Digital block 12 010b Digital block 01 011b Digital block 11 100b Digital block 00 101b Digital block 10 110b Digital block 03 111b Digital block 13

13.2.60 MULx_X

Multiply Input X Register

Individual Register Names and Addresses:

MUL0_X : 0,E8h

	7	6	5	4	3	2	1	0
Access : POR	W : XX							
Bit Name	Data[7:0]							

This register is one of two multiplicand registers for the signed 8-bit multiplier in the PSoC MAC.

For additional information, refer to the "Register Definitions" on page 287 in the Multiply Accumulate chapter.

Bit	Name	Description
7:0	**Data[7:0]**	X multiplicand for MAC 8-bit multiplier.

13.2.61 MULx_Y

Multiply Input Y Register

Individual Register Names and Addresses:

MUL0_Y : 0,E9h

	7	6	5	4	3	2	1	0
Access : POR	W : XX							
Bit Name	Data[7:0]							

This register is one of two multiplicand registers for the signed 8-bit multiplier in the PSoC MAC.

For additional information, refer to the "Register Definitions" on page 287 in the Multiply Accumulate chapter.

Bit	Name	Description
7:0	**Data[7:0]**	Y multiplicand for MAC 8-bit multiplier.

13.2.62 MULx_DH

Multiply Result High Byte Register

Individual Register Names and Addresses:

MUL0_DH : 0,EAh

	7	6	5	4	3	2	1	0
Access : POR	R : XX							
Bit Name	Data[7:0]							

This register holds the most significant byte of the 16-bit product.

For additional information, refer to the “Register Definitions” on page 287 in the Multiply Accumulate chapter.

Bit	Name	Description
7:0	**Data[7:0]**	High byte of MAC multiplier 16-bit product.

13.2.63 MULx_DL

Multiply Result Low Byte Register

Individual Register Names and Addresses:

MUL0_DL : 0,EBh

	7	6	5	4	3	2	1	0
Access : POR	R : XX							
Bit Name	Data[7:0]							

This register holds the least significant byte of the 16-bit product.

For additional information, refer to the “Register Definitions” on page 287 in the Multiply Accumulate chapter.

Bit	Name	Description
7:0	**Data[7:0]**	Low byte of MAC multiplier 16-bit product.

13.2.64 MACx_X/ACCx_DR1

Accumulator Data Register 1

Individual Register Names and Addresses:

MAC0_X/ACC0_DR1 : 0,ECh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

This is the multiply accumulate X register and the second byte of the accumulated value.

For additional information, refer to the “Register Definitions” on page 287 in the Multiply Accumulate chapter.

Bit	Name	Description	
7:0	**Data[7:0]**	Read	Returns the 2nd byte of the 32-bit accumulated value. The 2nd byte is next to the least significant byte for the accumulated value.
		Write	X multiplicand for the MAC 16-bit multiply and 32-bit accumulator.

13.2.65 MACx_Y/ACCx_DR0

Accumulator Data Register 0

Individual Register Names and Addresses:

MAC0_Y/ACC0_DR0 : 0,EDh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

This is the multiply accumulate Y register and the first byte of the accumulated value.

For additional information, refer to the “Register Definitions” on page 287 in the Multiply Accumulate chapter.

Bit	Name	Description	
7:0	**Data[7:0]**	Read	Returns the 1st byte of the 32-bit accumulated value. The 1st byte is the least significant byte for the accumulated value.
		Write	Y multiplicand for the MAC 16-bit multiply and 32-bit accumulate.

13.2.66 MACx_CL0/ACCx_DR3

Accumulator Data Register 3

Individual Register Names and Addresses:

MAC0_CL0/ACC0_DR3 : 0,EEh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

This is an accumulator clear register and the fourth byte of the accumulated value.

For additional information, refer to the "Register Definitions" on page 287 in the Multiply Accumulate chapter.

Bit	Name	Description	
7:0	**Data[7:0]**	Read	Returns the 4th byte of the 32-bit accumulated value. The 4th byte is the ***most significant byte (MSB)*** for the accumulated value.
		Write	Writing any value to this address clears all four bytes of the Accumulator.

13.2.67 MACx_CL1/ACCx_DR2

Accumulator Data Register 2

Individual Register Names and Addresses:

MAC0_CL1/ACC0_DR2 : 0,EFh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

This is an accumulator clear register and the third byte of the accumulated value.

For additional information, refer to the “Register Definitions” on page 287 in the Multiply Accumulate chapter.

Bit	Name	Description	
7:0	**Data[7:0]**	Read	Returns the 3rd byte of the 32-bit accumulated value. The 3rd byte is next to most significant byte for the accumulated value.
		Write	Writing any value to this address clears all four bytes of the Accumulator.

13.2.68 CPU_F

M8C Flag Register

Individual Register Names and Addresses:

CPU_F: x,F7h

	7	6	5	4	3	2	1	0
Access : POR	RL : 0		▨	RL : 0	▨	RL : 0	RL : 0	RL : 0
Bit Name	PgMode[1:0]		▨	XIO	▨	Carry	Zero	GIE

This register provides read access to the M8C flags.

The AND f, expr; OR f, expr; and XOR f, expr flag instructions can be used to modify this register. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 44 in the M8C chapter and the “Register Definitions” on page 64 in the Interrupt Controller chapter.

Bit	Name	Description	
7:6	**PgMode[1:0]**	00b	Direct Address mode and Indexed Address mode operands are referred to RAM Page 0, regardless of the values of CUR_PP and IDX_PP. Note that this condition prevails on entry to an ISR when the CPU_F register is cleared.
		01b	Direct Address mode instructions are referred to Page 0. Indexed Address mode instructions are referred to the RAM page specified by the stack page pointer, STK_PP.
		10b	Direct Address mode instructions are referred to the RAM page specified by the current page pointer, CUR_PP. Indexed Address mode instructions are referred to the RAM page specified by the index page pointer, IDX_PP.
		11b	Direct Address mode instructions are referred to the RAM page specified by the current page pointer, CUR_PP. Indexed Address mode instructions are referred to the RAM page specified by the stack page pointer, STK_PP.
4	**XIO**	0	Normal register address space.
		1	Extended register address space. Primarily used for configuration.
2	**Carry**	Set by the M8C CPU core to indicate whether there has been a carry in the previous logical/arithmetic operation.	
		0	No carry
		1	Carry
1	**Zero**	Set by the M8C CPU core to indicate whether there has been a zero result in the previous logical/arithmetic operation.	
		0	Not equal to zero
		1	Equal to zero
0	**GIE**	0	M8C does not process any interrupts.
		1	Interrupt processing enabled.

13.2.69 CPU_SCR1

System Status and Control Register 1

Individual Register Names and Addresses:

CPU_SCR1: x,FEh

	7	6	5	4	3	2	1	0
Access : POR	R : 0				R : 0	RW : 0		RW : 0
Bit Name	IRESS				ECO EXW	ECO EX		IRAMDIS

This register is used to convey the status and control of events related to internal resets and watchdog reset.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 51 in the SROM chapter or "Register Definitions" on page 23 of the External Crystal Oscillator (ECO) chapter.

Note

5. Bits 3 and 2 (ECO EXW and ECO EX, respectively) cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

Bit	Name	Description
7	**IRESS**	This bit is read only. 0 Boot phase only executed once. 1 Boot phase occurred multiple times.
3	**ECO EXW**	ECO Exists Written. 1 The ECO Exists Written bit has been written with a '1' or '0' and is now locked. 0 The ECO Exists Written bit has never been written in User mode.
2	**ECO EX**	ECO Exists (write once – see the explanation in "Register Definitions" on page 23). 1 ECO operation exists (set/reset OSC_CR[7] to enable/disable). 0 ECO operation does not exist. 32 kHz clock source is locked to operate from the ILO.
0	**IRAMDIS**	0 SRAM is initialized to 00h after POR, XRES, and WDR. 1 Address 03h - D7h of SRAM Page 0 are not modified by WDR.

13.2.70 CPU_SCR0

System Status and Control Register 0

Individual Register Names and Addresses:

CPU_SCR0: x,FFh

	7	6	5	4	3	2	1	0
Access : POR	R : 0		RC : 0	RC : 1	RW : 0			RW : 0
Bit Name	GIES		WDRS	PORS	Sleep			STOP

This register is used to convey the status and control of events for various functions of a PSoC device.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 33 in the Sleep and Watchdog chapter.

Bit	Name	Description
7	**GIES**	Global interrupt enable status. It is recommended that the user read the Global Interrupt Enable Flag bit from the CPU_F register on page 120. This bit is read only for GIES. Its use is discouraged, as the Flag register is now readable at address x,F7h (read only).
5	**WDRS**	Watchdog Reset Status. This bit may not be set by user code; however, it may be cleared by writing it with a ’0’. 0 No Watchdog Reset occurred. 1 Watchdog Reset occurred.
4	**PORS**	Power On Reset Status. This bit may not be set by user code; however, it may be cleared by writing it with a ’0’. 0 Power On Reset has not occurred and watchdog timer is enabled. 1 Set after external reset or Power On Reset.
3	**Sleep**	Set by the user to enable the CPU sleep state. CPU remains in Sleep mode until any interrupt is pending. 0 Normal operation. 1 Sleep.
0	**STOP**	0 M8C is free to execute code. 1 M8C is halted. Can only be cleared by POR or WDR.

13.3 Bank 1 Registers

The following registers are all in bank 1 and are listed in address order. Registers that are in both Bank 0 and Bank 1 are listed in address order in the section titled “Bank 0 Registers” on page 49.

13.3.1 PRTxDM0

Port Drive Mode Bit Register 0

Individual Register Names and Addresses:

PRT0DM0 : 1,00h	PRT1DM0 : 1,04h	PRT2DM0 : 1,08h	PRT3DM0 : 1,0Ch

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Drive Mode 0[7:0]							

This register is one of three registers whose combined value determines the unique Drive mode of each bit in a GPIO port.

In register PRTxDM0 there are eight possible drive modes for each port pin. Three mode bits are required to select one of these modes, and these three bits are spread into three different registers (PRTxDM0, “PRTxDM1” on page 124, and “PRTxDM2” on page 52). The bit position of the effected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the three Drive Mode register bits that control the Drive mode for that pin (for example, Bit[2] in PRT0DM0, bit[2] in PRT0DM1, and bit[2] in PRT0DM2). The three bits from the three registers are treated as a group. These are referred to as DM2, DM1, and DM0, or together as DM[2:0].

All Drive mode bits are shown in the sub-table below ([21**0**] refers to the combination (in order) of bits in a given bit position); however, this register only controls the ***least significant bit (LSb)*** of the Drive mode.

For additional information, refer to the “Register Definitions” on page 8 in the GPIO chapter.

Bit	Name	Description
7:0	**Drive Mode 0[7:0]**	Bit 0 of the Drive mode, for each of 8-port pins, for a GPIO port.

[210]	Pin Output High	Pin Output Low	Notes
00**0**b	Strong	Resistive	
00**1**b	Strong	Strong	
01**0**b	High Z	High Z	Digital input enabled.
01**1**b	Resistive	Strong	
10**0**b	Slow + strong	High Z	
10**1**b	Slow + strong	Slow + strong	
11**0**b	High Z	High Z	Reset state. Digital input disabled for zero power.
11**1**b	High Z	Slow + strong	I2C Compatible mode.

Note A bold digit, in the table above, signifies that the digit is used in this register.

13.3.2 PRTxDM1

Port Drive Mode Bit Register 1

Individual Register Names and Addresses:

PRT0DM1 : 1,01h	PRT1DM1 : 1,05h	PRT2DM1 : 1,09h	PRT3DM1 : 1,0Dh

	7	6	5	4	3	2	1	0
Access : POR	RW : FF							
Bit Name	Drive Mode 1[7:0]							

This register is one of three registers whose combined value determines the unique Drive mode of each bit in a GPIO port.

In register PRTxDM1 there are eight possible drive modes for each port pin. Three mode bits are required to select one of these modes, and these three bits are spread into three different registers ("PRTxDM0" on page 123, PRTxDM1, and "PRTxDM2" on page 52). The bit position of the effected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the three Drive Mode register bits that control the Drive mode for that pin (for example, Bit[2] in PRT0DM0, bit[2] in PRT0DM1, and bit[2] in PRT0DM2). The three bits from the three registers are treated as a group. These are referred to as DM2, DM1, and DM0, or together as DM[2:0].

All Drive mode bits are shown in the sub-table below ([2**1**0] refers to the combination (in order) of bits in a given bit position); however, this register only controls the middle bit of the Drive mode.

For additional information, refer to the "Register Definitions" on page 8 in the GPIO chapter.

Bit	Name	Description
7:0	**Drive Mode 1[7:0]**	Bit 1 of the Drive mode, for each of 8-port pins, for a GPIO port.

[2**1**0]	Pin Output High	Pin Output Low	Notes
0**0**0b	Strong	Resistive	
0**0**1b	Strong	Strong	
0**1**0b	High Z	High Z	Digital input enabled.
0**1**1b	Resistive	Strong	
1**0**0b	Slow + strong	High Z	
1**0**1b	Slow + strong	Slow + strong	
1**1**0b	High Z	High Z	Reset state. Digital input disabled for zero power.
1**1**1b	High Z	Slow + strong	I2C Compatible mode.

Note A bold digit, in the table above, signifies that the digit is used in this register.

13.3.3 PRTxIC0

Port Interrupt Control Register 0

Individual Register Names and Addresses:

PRT0IC0 : 1,02h	PRT1IC0 : 1,06h	PRT2IC0 : 1,0Ah	PRT3IC0 : 1,0Eh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Interrupt Control 0[7:0]							

This register is one of two registers whose combined value determine the unique Interrupt mode of each bit in a GPIO port.

In register PRTxIC0 there are four possible interrupt modes for each port pin. Two mode bits are required to select one of these modes and these two bits are spread into two different registers (PRTxIC0 and "PRTxIC1" on page 126). The bit position of the effected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the interrupt control register bits that control the Interrupt mode for that pin (for example, Bit[2] in PRT0IC0 and bit[2] in PRT0IC1). The two bits from the two registers are treated as a group. In the sub-table below, "[**0**]" refers to the combination (in order) of bits in a given position, one bit from PRTxIC1 and one bit from PRTxIC0.

For additional information, refer to the "Register Definitions" on page 8 in the GPIO chapter.

Bit	Name	Description	
7:0	**Interrupt Control 0[7:0]**	[1**0**]	**Interrupt Type**
		0**0**b	Disabled
		0**1**b	Low
		1**0**b	High
	‘	1**1**b	Change from last read

Note A bold digit, in the table above, signifies that the digit is used in this register.

13.3.4 PRTxIC1

Port Interrupt Control Register 1

Individual Register Names and Addresses:

PRT0IC1 : 1,03h	PRT1IC1 : 1,07h	PRT2IC1 : 1,0Bh	PRT3IC1 : 1,0Fh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Interrupt Control 1[7:0]							

This register is one of two registers whose combined value determine the unique Interrupt mode of each bit in a GPIO port.

In register PRTxIC1 there are four possible interrupt modes for each port pin. Two mode bits are required to select one of these modes and these two bits are spread into two different registers ("PRTxIC0" on page 125 and PRTxIC1). The bit position of the effected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the interrupt control register bits that control the Interrupt mode for that pin (for example, Bit[2] in PRT0IC0 and bit[2] in PRT0IC1). The two bits from the two registers are treated as a group. In the sub-table below, "[**1**]" refers to the combination (in order) of bits in a given position, one bit from PRTxIC1 and one bit from PRTxIC0.

For additional information, refer to the "Register Definitions" on page 8 in the GPIO chapter.

Bit	Name	Description
7:0	**Interrupt Control 1[7:0]**	[**1**0] **Interrupt Type** **0**0b Disabled **0**1b Low **1**0b High **1**1b Change from last read **Note** A bold digit, in the table above, signifies that the digit is used in this register.

13.3.5 DxBxxFN

Digital Basic/Communications Type B Block Function Register

Individual Register Names and Addresses:

DBB00FN : 1,20h DBB01FN : 1,24h DCB02FN : 1,28h DCB03FN : 1,2Ch

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0		RW : 0		
Bit Name	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]		

This register contains the primary Mode and Function bits that determine the function of the block.

Before changing any of the configuration registers (DxBxxFN, DxBxxIN, and DxBxxOU), disable the corresponding digital block by setting bit 0 in the CR0 or DxBxxCR0 register to ‘0’. The values in the DxBxxFN register should not be changed while the block is enabled. After all configuration changes are made, enable the block by setting bit 0 in the DxBxxCR0 register to ‘1’.

The naming convention for this register is as follows. The first ‘x’ in the digital register’s name represents either “B” for basic or “C” for communication. For rows of digital PSoC blocks and their registers, the second ‘x’ set represents <Prefix>mn<Suffix>, where m=row index, n=column index. Therefore, DCB12FN is a digital communication register for a digital PSoC block in row 1 column 2. Depending on the digital row characteristics of your PSoC device, some addresses may not be available. For additional information, refer to the “Register Definitions” on page 187 in the Digital Blocks chapter.

Bit	Name	Description	
7	**Data Invert**	0	Data input is non-inverted.
		1	Data input is inverted.
6	**BCEN**	Enable Primary Function Output to drive the broadcast net.	
		0	Disable
		1	Enable
5	**End Single**	0	Block is not the end of a chained function or the function is not chainable.
		1	Block is the end of a chained function or a standalone block in a chainable function.
4:3	**Mode[1:0]**	These bits are function dependent and are described by function as follows.	
	Timer or Counter:	Mode[0] signifies the interrupt type.	
		0	Interrupt on Terminal Count
		1	Interrupt on Compare True
		Mode[1] signifies the compare type.	
		0	Compare on Less Than or Equal
		1	Compare on Less Than
	CRCPRS:	Mode[1:0] are encoded as the Compare Type.	
		00b	Compare on Equal
		01b	Compare on Less Than or Equal
		10b	Reserved
		11b	Compare on Less Than

(continued on next page)

13.3.5 DxBxxFN *(continued)*

4:3 *(cont.)*	Dead Band:	Mode[1:0] are encoded as the Kill Type.	
		00b	Synchronous Restart KILL mode
		01b	Disable KILL mode
		10b	Asynchronous KILL mode
		11b	Reserved
	UART:	Mode[0] signifies the Direction.	
		0	Receiver
		1	Transmitter
		Mode[1] signifies the Interrupt Type.	
		0	Interrupt on TX Reg Empty
		1	Interrupt on TX Complete
	SPI:	Mode[0] signifies the Type.	
		0	Master
		1	Slave
		Mode[1] signifies the Interrupt Type.	
		0	Interrupt on TX Reg Empty
		1	Interrupt on SPI Complete
2:0	**Function[2:0]**	000b	Timer (chainable)
		001b	Counter (chainable)
		010b	CRCPRS (chainable)
		011b	Reserved
		100b	Dead Band
		101b	UART (DCBxx blocks only)
		110b	SPI (DCBxx blocks only)
		111b	Reserved

13.3.6 DxBxxIN

Digital Basic/Communications Type B Block Input Register

Individual Register Names and Addresses:

DBB00IN : 1,21h DBB01IN : 1,25h DCB02IN : 1,29h DCB03IN : 1,2Dh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0				RW : 0			
Bit Name	Data Input[3:0]				Clock Input[3:0]			

These registers are used to select the data and clock inputs.

Before changing any of the configuration registers (DxBxxFN, DxBxxIN, and DxBxxOU), disable the corresponding digital block by setting bit 0 in the CR0 or DxBxxCR0 register to '0'. The values in this register should not be changed while the block is enabled. After all configuration changes are made, enable the block by setting bit 0 in the CR0 register to '1'.

The naming convention for this register is as follows. The first 'x' in the digital register's name represents either "B" for basic or "C" for communication. For rows of digital PSoC blocks and their registers, the second 'x' set represents <Prefix>mn<Suffix>, where m=row index, n=column index. Therefore, DCB12IN is a digital communication register for a digital PSoC block in row 1 column 2. Depending on the digital row characteristics of your PSoC device, some addresses may not be available. For additional information, refer to the "Register Definitions" on page 187 in the Digital Blocks chapter.

Bit	Name	Description	
7:4	**Data Input[3:0]**	0h	Low (0)
		1h	High (1)
		2h	Row broadcast net
		3h	Chain function to previous block (low (0) in block DBB00IN)
		4h	Analog column comparator 0
		5h	Analog column comparator 1
		6h	Analog column comparator 2
		7h	Analog column comparator 3
		8h	Row output 0
		9h	Row output 1
		Ah	Row output 2
		Bh	Row output 3
		Ch	Row input 0
		Dh	Row input 1
		Eh	Row input 2
		Fh	Row input 3

(continued on next page)

13.3.6 DxBxxIN *(continued)*

3:0	**Clock Input[3:0]**	0h	Clock disabled (low)
		1h	VC3
		2h	Row broadcast net
		3h	Previous block primary output (low for DBB00)
		4h	SYSCLKX2
		5h	VC1
		6h	VC2
		7h	CLK32K
		8h	Row output 0
		9h	Row output 1
		Ah	Row output 2
		Bh	Row output 3
		Ch	Row input 0
		Dh	Row input 1
		Eh	Row input 2
		Fh	Row input 3

13.3.7 DxBxxOU

Digital Basic/Communications Type B Block Output Register

Individual Register Names and Addresses:

DBB00OU : 1,22h	DBB01OU : 1,26h	DCB02OU : 1,2Ah	DCB03OU : 1,2Eh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0		RW : 0	RW : 0		RW : 0	RW : 0	
Bit Name	AUXCLK		AUXEN	AUX IO Select[1:0]		OUTEN	Output Select[1:0]	

This register is used to control the connection of digital block outputs to the available row interconnect and control clock resynchronization.

Before changing any of the configuration registers (DxBxxFN, DxBxxIN, and DxBxxOU), disable the corresponding digital block by setting bit 0 in the CR0 or DxBxxCR0 register to ‘0’. The values in this register should not be changed while the block is enabled. After all configuration changes are made, enable the block by setting bit 0 in the DxBxxCR0 register to ‘1’.

The naming convention for this register is as follows. The first ‘x’ in the digital register’s name represents either “B” for basic or “C” for communication. For rows of digital PSoC blocks and their registers, the second ‘x’ set represents <Prefix>mn<Suffix>, where m=row index, n=column index. Therefore, DBB12OU is a digital basic register for a digital PSoC block in row 1 column 2. Depending on the digital row characteristics of your PSoC device, some addresses may not be available. For additional information, refer to the “Register Definitions” on page 187 in the Digital Blocks chapter.

Bit	Name	Description		
7:6	**AUXCLK**	00b	No sync	16-to-1 clock mux output
		01b	Synchronize	Output of 16-to-1 clock mux to SYSCLK
		10b	Synchronize	Output of 16-to-1 clock mux to SYSCLKX2
		11b	SYSCLK	Directly connect SYSCLK to block clock input
5	**AUXEN**	Auxiliary IO Enable (function dependent)		
		All Functions except SPI Slave: Enable Auxiliary Output Driver		
		0	Disabled	
		1	Enabled	
		SPI Slave: Input Source for SS_		
		0	Row Input [3:0], as selected by the AUX IO Select bits	
		1	Force SS_ Active	
4:3	**AUX IO Select[1:0]**	Auxiliary IO Select Function Output (function dependent)		
		All Functions except SPI Slave: Row Output Select		
		00b	Row Output 0	
		01b	Row Output 1	
		10b	Row Output 2	
		11b	Row Output 3	
		SPI Slave Source for SS_ Input if AUXEN =0		
		00b	Row Input 0	
		01b	Row Input 1	
		10b	Row Input 2	
		11b	Row Input 3	

(continued on next page)

13.3.7 DxBxxOU *(continued)*

4:3 *(cont.)*	**AUX IO Select[1:0]**	SPI Slave Source for SS_ Input if AUXEN =1	
		00b	Force SS_ Active
		01b	Reserved
		10b	Reserved
		11b	Reserved
2	**OUTEN**	Enable Primary Function Output Driver	
		0	Disabled
		1	Enabled
1:0	**Output Select[1:0]**	Row Output Select for Primary Function Output	
		00b	Row Output 0
		01b	Row Output 1
		10b	Row Output 2
		11b	Row Output 3

13.3.8 CLK_CR0

Analog Column Clock Control Register 0

Individual Register Names and Addresses:

CLK_CR0: 1,60h

	7	6	5	4	3	2	1	0
Access : POR					RW : 0		RW : 0	
Bit Name					AColumn1[1:0]		AColumn0[1:0]	

This register is used to select the clock source for an individual analog column.

Each column has two bits that select the column clock input source. The resulting column clock frequency is the selected input clock frequency divided by four. Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 227 in the Analog Interface chapter.

Bits	Name	Description
3:2	**AColumn1[1:0]**	Clock selection for column 1. 00b Variable Clock 1 (VC1) 01b Variable Clock 2 (VC2) 10b Analog Clock 0 (ACLK0) 11b Analog Clock 1 (ACLK1)
1:0	**AColumn0[1:0]**	Clock selection for column 0. 00b Variable Clock 1 (VC1) 01b Variable Clock 2 (VC2) 10b Analog Clock 0 (ACLK0) 11b Analog Clock 1 (ACLK1)

13.3.9 CLK_CR1

Analog Clock Source Control Register 1

Individual Register Names and Addresses:

CLK_CR1: 1,61h

	7	6	5	4	3	2	1	0
Access : POR		RW : 0	RW : 0			RW : 0		
Bit Name		SHDIS	ACLK1[2:0]			ACLK0[2:0]		

This register is used to select the clock source for an individual analog column.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 227 in the Analog Interface chapter.

Bits	Name	Description
6	**SHDIS**	Sample and hold disable. 0 Enabled 1 Disabled
5:3	**ACLK1[2:0]**	Select the clocking source for Analog Clock 1. 000b Digital Basic Block 00 001b Digital Basic Block 01 010b Digital Communication Block 02 011b Digital Communication Block 03 100b Digital Basic Block 10 101b Digital Basic Block 11 110b Digital Communication Block 12 111b Digital Communication Block 13
2:0	**ACLK0[2:0]**	Select the clocking source for Analog Clock 0. 000b Digital Basic Block 00 001b Digital Basic Block 01 010b Digital Communication Block 02 011b Digital Communication Block 03 100b Digital Basic Block 10 101b Digital Basic Block 11 110b Digital Communication Block 12 111b Digital Communication Block 13

13.3.10 ABF_CR0

Analog Output Buffer Control Register 0

Individual Register Names and Addresses:

ABF_CR0: 1,62h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0		RW : 0		RW : 0		RW : 0	RW : 0
Bit Name	ACol1Mux		ABUF1EN		ABUF0EN		Bypass	PWR

This register controls analog input muxes from Port 0.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 14 in the Analog Output Drivers chapter or the “Register Definitions” on page 243 in the Analog Input Configuration chapter.

Bits	Name	Description
7	**ACol1Mux**	0 Set column 1 input to column 1 input mux output (1 Column: selects among P0[6,4,2,0]). 1 Set column 1 input to column 0 input mux output (1 Column: selects among P0[7,5,3,1]).
5	**ABUF1EN**	Enables the analog output buffer for Analog Column 1 (Pin P0[5]). 0 Disable analog output buffer. 1 Enable analog output buffer.
3	**ABUF0EN**	Enables the analog output buffer for Analog Column 0 (Pin P0[3]) (1 Column: AGND). 0 Disable analog output buffer. 1 Enable analog output buffer.
1	**Bypass**	Connects the positive input of the amplifier(s) directly to the output(s). Amplifiers must be disabled when in Bypass mode. 0 Disable 1 Enable
0	**PWR**	Determines power level of all output buffers. 0 Low output power 1 High output power

13.3.11 AMD_CR0

Analog Modulation Control Register 0

Individual Register Names and Addresses:

AMD_CR0: 1,63h

	7	6	5	4	3	2	1	0
Access : POR						RW : 0		
Bit Name						AMOD0[2:0]		

This register is used to select the modulator bits used with each column.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 227 in the Analog Interface chapter.

Bits	Name	Description
2:0	**AMOD0[2:0]**	Analog modulation control signal selection for column 0. 000b Zero (off) 001b Global Output Bus, even bus bit 1 (GOE[1]) 010b Global Output Bus, even bus bit 0 (GOE[0]) 011b Row 0 Broadcast Bus 100b Analog Column Comparator 0 101b Analog Column Comparator 1 110b Analog Column Comparator 2 111b Analog Column Comparator 3

13.3.12 AMD_CR1

Analog Modulation Control Register 1

Individual Register Names and Addresses:

AMD_CR1: 1,66h

	7	6	5	4	3	2	1	0
Access : POR						RW : 0		
Bit Name						AMOD1[2:0]		

This register is used to select the modulator bits used with each column.

Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 227 in the Analog Interface chapter.

Bits	Name	Description
2:0	**AMOD1[2:0]**	Analog modulation control signal selection for column 1.
		000b Zero (off)
		001b Global Output Bus, even bus bit 1 (GOE[1])
		010b Global Output Bus, even bus bit 0 (GOE[0])
		011b Row 0 Broadcast Bus
		100b Analog Column Comparator 0
		101b Analog Column Comparator 1
		110b Analog Column Comparator 2
		111b Analog Column Comparator 3

13.3.13 ALT_CR0

Analog LUT Control Register 0

Individual Register Names and Addresses:

ALT_CR0: 1,67h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0				RW : 0			
Bit Name	LUT1[3:0]				LUT0[3:0]			

This register is used to select the logic function.

For additional information, refer to the “Register Definitions” on page 227 in the Analog Interface chapter.

Bits	Name	Description
7:4	**LUT1[3:0]**	Select 1 of 16 logic functions for output of comparator bus 1.

	Function
0h	FALSE
1h	A AND B
2h	A AND $\overline{B}$
3h	A
4h	$\overline{A}$ AND B
5h	B
6h	A XOR B
7h	A OR B
8h	A NOR B
9h	A XNOR B
Ah	$\overline{B}$
Bh	A OR $\overline{B}$
Ch	$\overline{A}$
Dh	$\overline{A}$ OR B
Eh	A NAND B
Fh	TRUE

Bits	Name	Description
3:0	**LUT0[3:0]**	Select 1 of 16 logic functions for output of comparator bus 0.

	Function
0h	FALSE
1h	A AND B
2h	A AND $\overline{B}$
3h	A
4h	$\overline{A}$ AND B
5h	B
6h	A XOR B
7h	A OR B
8h	A NOR B
9h	A XNOR B
Ah	$\overline{B}$
Bh	A OR $\overline{B}$
Ch	$\overline{A}$
Dh	$\overline{A}$ OR B
Eh	A NAND B
Fh	TRUE

13.3.14 SARADC_TRS

SAR8 ADC Auto Align/Trigger Source Register

Individual Register Names and Addresses:

SARADC_TRS : 1,A8h

	7	6	5	4	3	2	1	0
Access : POR	RW : 00		RW : 00		RW : 00		RW : 00	
Bit Name	DCB03_HL[1:0]		DCB02_HL[1:0]		DBB01_HL[1:0]		DBB00_HL[1:0]	

This register is where the ADC auto align/trigger source is set.

Select any digital blocks in digital ROW0 as the align/trigger source for ADC conversion. You can also select two adjacent blocks in digital ROW0 as the align/trigger source for ADC conversion. For example, if there is one PWM8 UM in block DBB0x and the system needs the ADC auto aligned with the PWM pulse, you can select the DBB0x as auto align/trigger source to the ADC. Through setting the SARADC_TRCL register, you can trigger the ADC start conversion at any point during one PWM cycle. You can also select two adjacent blocks as auto align/trigger source to align with a 16-bit PWM, timer, or counter. Be sure that the block you select is the same block you place the user module that aligns with the ADC.

For additional information, refer to the “Register Definitions” on page 266 in the SAR8 ADC Block chapter.

Bits	Name	Description	
7:6	**DCB03_HL[1:0]**	Bit 7	
		0	DCB03 DR0 is not driving the channel high.
		1	DCB03 DR0 is driving the channel high.
		Bit 6	
		0	DCB03 DR0 is not driving the channel low.
		1	DCB03 DR0 is driving the channel low.
5:4	**DCB02_HL[1:0]**	Bit 5	
		0	DCB02 DR0 is not driving the channel high.
		1	DCB02 DR0 is driving the channel high.
		Bit 4	
		0	DCB02 DR0 is not driving the channel low.
		1	DCB02 DR0 is driving the channel low.
3:2	**DBB01_HL[1:0]**	Bit 3	
		0	DBB01 DR0 is not driving the channel high.
		1	DBB01 DR0 is driving the channel high.
		Bit 2	
		0	DBB01 DR0 is not driving the channel low.
		1	DBB01 DR0 is driving the channel low.
1:0	**DBB00_HL[1:0]**	Bit 1	
		0	DBB00 DR0 is not driving the channel high.
		1	DBB00 DR0 is driving the channel high.
		Bit 0	
		0	DBB00 DR0 is not driving the channel low.
		1	DBB00 DR0 is driving the channel low.

Note The digital block with the smaller number has higher priority to output to the related channel. The circuitry guarantees only the highest priority bit to be set to 1 when the user enables multiple blocks to one channel. The user can see only one bit set for one channel if they read back the register data.

13.3.15 SARADC_TRCL

SAR8 ADC Low Channel Comparator Data Register

Individual Register Names and Addresses:

SARADC_TRCL : 1,A9h

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	CMP_L[7:0]							

This register is the ADC auto align/trigger comparator data register.

A trigger occurs when the low channel trigger is enabled, a selected digital block is enabled, and the selected digital block's DR0 data is equal to the data of this register.

For additional information, refer to the "Register Definitions" on page 266 in the SAR8 ADC Block chapter.

Bits	Name	Description
7:0	**CMP_L[7:0]**	The comparator data for low channel trigger.

13.3.16 SARADC_TRCH

SAR8 ADC High Channel Comparator Data Register

Individual Register Names and Addresses:

SARADC_TRCH : 1,AAh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	CMP_H[7:0]							

This register is the ADC auto align/trigger comparator data register.

A trigger occurs when the high channel trigger is enabled, a selected digital block is enabled, and the selected digital block's DR0 data is equal to the data of this register.

For additional information, refer to the "Register Definitions" on page 266 in the SAR8 ADC Block chapter.

Bits	Name	Description
7:0	**CMP_H[7:0]**	The comparator data for high channel trigger.

13.3.17 SARADC_CR2

SAR8 ADC Control Register 2

Individual Register Names and Addresses:

SARADC_CR2 : 1,ABh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0			RW : 02		
Bit Name	Test Enable	Free Run	Scale Size[2:0]			ADC Clock[2:0]		

The SAR8 ADC Control Register 2 (SARADC_CR2) is used to control ADC settings.

For additional information, refer to the "Register Definitions" on page 266 in the SAR8 ADC Block chapter.

Bits	Name	Description	
7	**Test Enable**	0	ADC in user mode.
		1	ADC in test mode.
6	**Free Run**	0	ADC in one-shot mode.
		1	ADC in free running mode.
5:3	**Scale Size[2:0]**	000	ADC raw results are directly read out.
		001	ADC raw results are divided by 2 when read out.
		010	ADC raw results are divided by 4 when read out.
		011	ADC raw results are divided by 8 when read out.
		100	ADC raw results are divided by 16 when read out.
		101	ADC raw results are divided by 32 when read out.
		010	ADC raw results are divided by 64 when read out.
		This feature does not affect the raw data in ADC raw results register. It only affects the data read out to the MCU.	
2:0	**ADC Clock[2:0]**	000	ADC clock is SYSCLK.
		001	ADC clock is SYSCLK/2
		010	ADC clock is SYSCLK/4.
		011	ADC clock is SYSCLK/8.
		100	ADC clock is SYSCLK/16.
		101	ADC clock is SYSCLK/32.
		010	ADC clock is SYSCLK/64.
		ADC sample rate is ADC clock divided by 8, and its conversion time is ADC clock period multiplied by 8. The maximum ADC clock speed should be no more than 6 MHz. The ADC automatically goes into test mode after conversion, so for one-shot mode, the low-speed ADC clock consumes more power than the high-speed ADC clock. In free running mode, the low-speed ADC clock consumes less power than the high-speed ADC clock.	

13.3.18 SARADC_LCR

SAR8 ADC Reference Voltage Generator Control Register

Individual Register Names and Addresses:

SARADC_LCR : 1,ACh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	DA_L[7:0]							

This register Is the ADC DA register used for reference voltage generation control. It is write only in Test mode.

For additional information, refer to the “Register Definitions” on page 266 in the SAR8 ADC Block chapter.

Bits	Name	Description
7:0	**DA_L [7:0]**	The low byte control for reference voltage.

13.3.19 GDI_O_IN

Global Digital Interconnect Odd Inputs Register

Individual Register Names and Addresses:

GDI_O_IN: 1,D0h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	GIONOUT7	GIONOUT6	GIONOUT5	GIONOUT4	GIONOUT3	GIONOUT2	GIONOUT1	GIONOUT0

This register is used to configure a global input to drive a global output.

For additional information, refer to the "Register Definitions" on page 167 in the Global Digital Interconnect chapter.

Bit	Name	Description	
7	**GIONOUT7**	0 1	GIO[7] does not drive GOO[7]. GIO[7] drives its value on to GOO[7].
6	**GIONOUT6**	0 1	GIO[6] does not drive GOO[6]. GIO[6] drives its value on to GOO[6].
5	**GIONOUT5**	0 1	GIO[5] does not drive GOO[5]. GIO[5] drives its value on to GOO[5].
4	**GIONOUT4**	0 1	GIO[4] does not drive GOO[4]. GIO[4] drives its value on to GOO[4].
3	**GIONOUT3**	0 1	GIO[3] does not drive GOO[3]. GIO[3] drives its value on to GOO[3].
2	**GIONOUT2**	0 1	GIO[2] does not drive GOO[2]. GIO[2] drives its value on to GOO[2].
1	**GIONOUT1**	0 1	GIO[1] does not drive GOO[1]. GIO[1] drives its value on to GOO[1].
0	**GIONOUT0**	0 1	GIO[0] does not drive GOO[0]. GIO[0] drives its value on to GOO[0].

13.3.20 GDI_E_IN

Global Digital Interconnect Even Inputs Register

Individual Register Names and Addresses:

GDI_E_IN: 1,D1h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	GIENOUT7	GIENOUT6	GIENOUT5	GIENOUT4	GIENOUT3	GIENOUT2	GIENOUT1	GIENOUT0

This register is used to configure a global input to drive a global output.

For additional information, refer to the "Register Definitions" on page 167 in the Global Digital Interconnect chapter.

Bit	Name	Description	
7	**GIENOUT7**	0 1	GIE[7] does not drive GOE[7]. GIE[7] drives its value on to GOE [7].
6	**GIENOUT6**	0 1	GIE[6] does not drive GOE[6]. GIE[6] drives its value on to GOE [6].
5	**GIENOUT5**	0 1	GIE[5] does not drive GOE[5]. GIE[5] drives its value on to GOE [5].
4	**GIENOUT4**	0 1	GIE[4] does not drive GOE[4]. GIE[4] drives its value on to GOE [4].
3	**GIENOUT3**	0 1	GIE[3] does not drive GOE[3]. GIE[3] drives its value on to GOE [3].
2	**GIENOUT2**	0 1	GIE[2] does not drive GOE[2]. GIE[2] drives its value on to GOE [2].
1	**GIENOUT1**	0 1	GIE[1] does not drive GOE[1]. GIE[1] drives its value on to GOE [1].
0	**GIENOUT0**	0 1	GIE[0] does not drive GOE[0]. GIE[0] drives its value on to GOE [0].

13.3.21 GDI_O_OU

Global Digital Interconnect Odd Outputs Register

Individual Register Names and Addresses:

GDI_O_OU: 1,D2h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	GOOUTIN7	GOOUTIN6	GOOUTIN5	GOOUTIN4	GOOUTIN3	GOOUTIN2	GOOUTIN1	GOOUTIN0

This register is used to configure a global output to drive a global input.

For additional information, refer to the “Register Definitions” on page 167 in the Global Digital Interconnect chapter.

Bit	Name	Description	
7	**GOOUTIN7**	0 1	GOO[7] does not drive GIO[7]. GOO[7] drives its value on to GIO[7].
6	**GOOUTIN6**	0 1	GOO[6] does not drive GIO[6]. GOO[6] drives its value on to GIO[6].
5	**GOOUTIN5**	0 1	GOO[5] does not drive GIO[5]. GOO[5] drives its value on to GIO[5].
4	**GOOUTIN4**	0 1	GOO[4] does not drive GIO[4]. GOO[4] drives its value on to GIO[4].
3	**GOOUTIN3**	0 1	GOO[3] does not drive GIO[3]. GOO[3] drives its value on to GIO[3].
2	**GOOUTIN2**	0 1	GOO[2] does not drive GIO[2]. GOO[2] drives its value on to GIO[2].
1	**GOOUTIN1**	0 1	GOO[1] does not drive GIO[1]. GOO[1] drives its value on to GIO[1].
0	**GOOUTIN0**	0 1	GOO[0] does not drive GIO[0]. GOO[0] drives its value on to GIO[0].

13.3.22 GDI_E_OU

Global Digital Interconnect Even Outputs Register

Individual Register Names and Addresses:

GDI_E_OU: 1,D3h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	GOEUTIN7	GOEUTIN6	GOEUTIN5	GOEUTIN4	GOEUTIN3	GOEUTIN2	GOEUTIN1	GOEUTIN0

This register is used to configure a global output to drive a global input.

For additional information, refer to the "Register Definitions" on page 167 in the Global Digital Interconnect chapter.

Bit	Name	Description	
7	**GOEUTIN7**	0 1	GOE[7] does not drive GIE[7]. GOE[7] drives its value on to GIE[7].
6	**GOEUTIN6**	0 1	GOE[6] does not drive GIE[6]. GOE[6] drives its value on to GIE[6].
5	**GOEUTIN5**	0 1	GOE[5] does not drive GIE[5]. GOE[5] drives its value on to GIE[5].
4	**GOEUTIN4**	0 1	GOE[4] does not drive GIE[4]. GOE[4] drives its value on to GIE[4].
3	**GOEUTIN3**	0 1	GOE[3] does not drive GIE[3]. GOE[3] drives its value on to GIE[3].
2	**GOEUTIN2**	0 1	GOE[2] does not drive GIE[2]. GOE[2] drives its value on to GIE[2].
1	**GOEUTIN1**	0 1	GOE[1] does not drive GIE[1]. GOE[1] drives its value on to GIE[1].
0	**GOEUTIN0**	0 1	GOE[0] does not drive GIE[0]. GOE[0] drives its value on to GIE[0].

13.3.23 OSC_GO_EN

Oscillator to Global Outputs Enable Register

Individual Register Names and Addresses:

OSC_GO_EN: 1,DDh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	SLPINT	VC3	VC2	VC1	SYSCLKX2	SYSCLK	CLK24M	CLK32K

This register is used to enable tri-state buffers that connect specific system clocks to specific global output even nets.

For additional information, refer to the “Register Definitions” on page 279 in the Digital Clocks chapter.

Bit	Name	Description	
7	**SLPINT**	0 1	The sleep interrupt is not driven onto a global net. The sleep interrupt is driven onto GOE[7].
6	**VC3**	0 1	The VC3 clock is not driven onto a global net The VC3 clock is driven onto GOE[6]
5	**VC2**	0 1	The VC2 clock is not driven onto a global net The VC2 clock is driven onto GOE[5]
4	**VC1**	0 1	The VC1 clock is not driven onto a global net The VC1 clock is driven onto GOE[4]
3	**SYSCLKX2**	0 1	The 2 times system clock is not driven onto a global net The 2 times system clock is driven onto GOE[3]
2	**SYSCLK**	0 1	The system clock is not driven onto a global net The system clock is driven onto GOE[2]
1	**CLK24M**	0 1	The 24 MHz clock is not driven onto a global net The 24 MHz system clock is driven onto GOE[1]
0	**CLK32K**	0 1	The 32 kHz clock is not driven onto a global net The 32 kHz system clock is driven onto GOE[0]

13.3.24 OSC_CR4

Oscillator Control Register 4

Individual Register Names and Addresses:

OSC_CR4: 1,DEh

	7	6	5	4	3	2	1	0
Access : POR							RW : 0	
Bit Name							VC3 Input Select[1:0]	

This register selects the input clock to variable clock 3 (VC3).

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 279 in the Digital Clocks chapter.

Bit	Name	Description
1:0	**VC3 Input Select[1:0]**	Selects the clocking source for the VC3 Clock Divider. 00b SYSCLK 01b VC1 10b VC2 11b SYSCLKX2

13.3.25 OSC_CR3

Oscillator Control Register 3

Individual Register Names and Addresses:

OSC_CR3: 1,DFh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	VC3 Divider[7:0]							

This register selects the divider value for variable clock 3 (VC3).

The output frequency of the VC3 Clock Divider is the input frequency divided by the value in this register, plus one. For example, if this register contains 07h, the clock frequency output from the VC3 Clock Divider is one eighth the input frequency. For additional information, refer to the "Register Definitions" on page 279 in the Digital Clocks chapter.

Bit	Name	Description
7:0	**VC3 Divider[7:0]**	Refer to the OSC_CR4 register. 00h Input Clock 01h Input Clock / 2 02h Input Clock / 3 03h Input Clock / 4 ... ... FCh Input Clock / 253 FDh Input Clock / 254 FEh Input Clock / 255 FFh Input Clock / 256

13.3.26 OSC_CR0

Oscillator Control Register 0

Individual Register Names and Addresses:

OSC_CR0: 1,E0h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0		RW : 0		
Bit Name	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]		

This register is used to configure various features of internal clock sources and clock nets.

Bit	Name	Description
7	**32k Select**	0 Internal low precision 32 kHz oscillator. 1 External crystal 32.768 kHz oscillator.
6	**PLL Mode**	0 Disabled. 1 Enabled. Internal main oscillator is frequency locked to External Crystal Oscillator.
5	**No Buzz**	0 BUZZ bandgap during power down. 1 Bandgap is always powered even during sleep.
4:3	**Sleep[1:0]**	Sleep Interval. 00b 1.95 ms (512 Hz) 01b 15.6 ms (64 Hz) 10b 125 ms (8 Hz) 11b 1 s (1 Hz)
2:0	**CPU Speed[2:0]**	These bits set the CPU clock speed, based on the system clock (SYSCLK). SYSCLK is 24 MHz by default, but it can optionally be set to 6 MHz on some PSoC devices (see the "Architectural Description" on page 15), or driven from an external clock.

	6 MHz IMO	**24 MHz IMO**	**External Clock**
000b	750 kHz	3 MHz	EXTCLK / 8
001b	1.5 MHz	6 MHz	EXTCLK / 4
010b	3 MHz	12 MHz	EXTCLK / 2
011b	6 MHz	24 MHz	EXTCLK / 1
100b	375 kHz	1.5 MHz	EXTCLK / 16
101b	187.5 kHz	750 kHz	EXTCLK / 32
110b	46.9 kHz	187.5 kHz	EXTCLK / 128
111b	23.4 kHz	93.7 kHz	EXTCLK / 256

13.3.27 OSC_CR1

Oscillator Control Register 1

Individual Register Names and Addresses:

OSC_CR1: 1,E1h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0				RW : 0			
Bit Name	VC1 Divider[3:0]				VC2 Divider[3:0]			

This register selects the divider value for variable clocks 1 and 2 (VC1 and VC2).

For additional information, refer to the "Register Definitions" on page 279 in the Digital Clocks chapter.

Bit	Name	Description		
7:4	**VC1 Divider[3:0]**		**Internal Main Oscillator**	**External Clock**
		0h	24 MHz	EXTCLK / 1
		1h	12 MHz	EXTCLK / 2
		2h	8 MHz	EXTCLK / 3
		3h	6 MHz	EXTCLK / 4
		4h	4.8 MHz	EXTCLK / 5
		5h	4 MHz	EXTCLK / 6
		6h	3.43 MHz	EXTCLK / 7
		7h	3 MHz	EXTCLK / 8
		8h	2.67 MHz	EXTCLK / 9
		9h	2.40 MHz	EXTCLK / 10
		Ah	2.18 MHz	EXTCLK / 11
		Bh	2.00 MHz	EXTCLK / 12
		Ch	1.85 MHz	EXTCLK / 13
		Dh	1.71 MHz	EXTCLK / 14
		Eh	1.6 MHz	EXTCLK / 15
		Fh	1.5 MHz	EXTCLK / 16
3:0	**VC2 Divider[3:0]**		**Internal Main Oscillator**	**External Clock**
		0h	(24 / (OSC_CR1[7:4]+1)) / 1	(EXTCLK / (OSC_CR1[7:4]+1)) / 1
		1h	(24 / (OSC_CR1[7:4]+1)) / 2	(EXTCLK / (OSC_CR1[7:4]+1)) / 2
		2h	(24 / (OSC_CR1[7:4]+1)) / 3	(EXTCLK / (OSC_CR1[7:4]+1)) / 3
		3h	(24 / (OSC_CR1[7:4]+1)) / 4	(EXTCLK / (OSC_CR1[7:4]+1)) / 4
		4h	(24 / (OSC_CR1[7:4]+1)) / 5	(EXTCLK / (OSC_CR1[7:4]+1)) / 5
		5h	(24 / (OSC_CR1[7:4]+1)) / 6	(EXTCLK / (OSC_CR1[7:4]+1)) / 6
		6h	(24 / (OSC_CR1[7:4]+1)) / 7	(EXTCLK / (OSC_CR1[7:4]+1)) / 7
		7h	(24 / (OSC_CR1[7:4]+1)) / 8	(EXTCLK / (OSC_CR1[7:4]+1)) / 8
		8h	(24 / (OSC_CR1[7:4]+1)) / 9	(EXTCLK / (OSC_CR1[7:4]+1)) / 9
		9h	(24 / (OSC_CR1[7:4]+1)) / 10	(EXTCLK / (OSC_CR1[7:4]+1)) / 10
		Ah	(24 / (OSC_CR1[7:4]+1)) / 11	(EXTCLK / (OSC_CR1[7:4]+1)) / 11
		Bh	(24 / (OSC_CR1[7:4]+1)) / 12	(EXTCLK / (OSC_CR1[7:4]+1)) / 12
		Ch	(24 / (OSC_CR1[7:4]+1)) / 13	(EXTCLK / (OSC_CR1[7:4]+1)) / 13
		Dh	(24 / (OSC_CR1[7:4]+1)) / 14	(EXTCLK / (OSC_CR1[7:4]+1)) / 14
		Eh	(24 / (OSC_CR1[7:4]+1)) / 15	(EXTCLK / (OSC_CR1[7:4]+1)) / 15
		Fh	(24 / (OSC_CR1[7:4]+1)) / 16	(EXTCLK / (OSC_CR1[7:4]+1)) / 16

13.3.28 OSC_CR2

Oscillator Control Register 2

Individual Register Names and Addresses:

OSC_CR2: 1,E2h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0					RW : 0	RW : 0	RW : 0
Bit Name	PLLGAIN					EXTCLKEN	RSVD	SYSCLKX2DIS

This register is used to configure various features of internal clock sources and clock nets.

In OCD mode (OCDM=1), bits [1:0] have no effect. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 279 in the Digital Clocks chapter.

Bit	Name	Description
7	**PLLGAIN**	Phase-locked loop gain. 0 Recommended value, normal gain. 1 Reduced gain to make PLL more tolerant to noisy or jittery crystal input.
2	**EXTCLKEN**	External clock mode enable. 0 Disabled. Operate from internal main oscillator. 1 Enabled. Operate from clock supplied at port P1[4].
1	**RSVD**	Reserved bit - This bit should always be 0.
0	**SYSCLKX2DIS**	48 MHz clock source disable. 0 Enabled. If enabled, system clock net is forced on. 1 Disabled for power reduction.

13.3.29 VLT_CR

Voltage Monitor Control Register

Individual Register Names and Addresses:

VLT_CR: 1,E3h

	7	6	5	4	3	2	1	0
Access : POR			RW : 0		RW : 0	RW : 0		
Bit Name			PORLEV[1:0]		LVDTBEN	VM[2:0]		

This register is used to set the trip points for POR, LVD, and the supply pump.

Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 320 in the POR and LVD chapter.

Bit	Name	Description
5:4	**PORLEV[1:0]**	Sets the POR level per the DC electrical specifications in the PSoC device data sheet. 00b POR level for 2.4 V or 3V operation (refer to the PSoC device data sheet) 01b POR level for 3.0V or 4.5V operation (refer to the PSoC device data sheet) 10b POR level for 4.75V operation 11b Reserved
3	**LVDTBEN**	Enables reset of CPU speed register by LVD comparator output. 0 Disables CPU speed throttle-back. 1 Enables CPU speed throttle-back.
2:0	**VM[2:0]**	Sets the LVD and pump levels per the DC electrical specifications in the PSoC device data sheet, for those PSoC devices with this feature. 000b Lowest voltage setting 001b 010b . 011b . 100b . 101b 110b 111b Highest voltage setting

13.3.30 VLT_CMP

Voltage Monitor Comparators Register

Individual Register Names and Addresses:

VLT_CMP: 1,E4h

	7	6	5	4	3	2	1	0
Access : POR						R : 0	R : 0	R : 0
Bit Name						PUMP	LVD	PPOR

This register is used to read the state of internal supply voltage monitors.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 320 in the POR and LVD chapter.

Bit	Name	Description
2	**PUMP**	Read state of pump comparator. 0 Vdd is above trip point. 1 Vdd is below trip point.
1	**LVD**	Reads state of LVD comparator. 0 Vdd is above LVD trip point. 1 Vdd is below LVD trip point.
0	**PPOR**	Reads state of Precision POR comparator (only useful with PPOR reset disabled, with PORLEV[1:0] in VLT_CR register set to 11b). 0 Vdd is above PPOR trip voltage. 1 Vdd is below PPOR trip voltage.

13.3.31 IMO_TR

Internal Main Oscillator Trim Register

Individual Register Names and Addresses:

IMO_TR: 1,E8h

	7	6	5	4	3	2	1	0
Access : POR	W : 00							
Bit Name	Trim[7:0]							

This register is used to manually center the oscillator's output to a target frequency.

It is strongly recommended that the user not alter this register's values. The value in this register should not be changed. For additional information, refer to the "Register Definitions" on page 16 in the Internal Main Oscillator chapter.

Bit	Name	Description
7:0	**Trim[7:0]**	The value of this register is used to trim the Internal Main Oscillator. Its value is set to the best value for the device during boot. ***The value of these bits should not be changed***. 00h Lowest frequency setting 01h ... ... 7Fh 80h Design center setting 81h ... ... FEh FFh Highest frequency setting

13.3.32 ILO_TR

Internal Low Speed Oscillator Trim Register

Individual Register Names and Addresses:

ILO_TR: 1,E9h

	7	6	5	4	3	2	1	0
Access : POR			W : 0		W : 0			
Bit Name			Bias Trim[1:0]		Freq Trim[3:0]			

This register sets the adjustment for the Internal Low Speed Oscillator (ILO).

It is strongly recommended that the user not alter this register's values. The trim bits are set to factory specifications and should not be changed. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 19 in the Internal Low Speed Oscillator chapter.

Bit	Name	Description
5:4	**Bias Trim[1:0]**	The value of this register is used to trim the Internal Low Speed Oscillator. Its value is set to the device specific, best value during boot. ***The value of these bits should not be changed.*** 00b Medium bias 01b Maximum bias (recommended) 10b Minimum bias 11b Intermediate Bias * * About 15% higher than the minimum bias.
3:0	**Freq Trim[3:0]**	The value of this register is used to trim the Internal Low Speed Oscillator. Its value is set to the device specific, best value during boot. ***The value of these bits should not be changed.***

13.3.33 BDG_TR

Bandgap Trim Register

Individual Register Names and Addresses:

BDG_TR: 1,EAh

	7	6	5	4	3	2	1	0
Access : POR		RW : 0	RW : 1		RW : 8			
Bit Name		AGNDBYP	TC[1:0]		V[3:0]			

This register is used to adjust the bandgap and add an RC filter to AGND.

Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 312 in the Internal Voltage Reference chapter.

Bit	Name	Description
6	**AGNDBYP**	If set, an external bypass capacitor on AGND may be connected to Port 2[4]. 0 Disable 1 Enable
5:4	**TC[1:0]**	The value of these bits is used to trim the temperature coefficient. Their value is set to the best value for the device during boot. ***The value of these bits should not be changed.***
3:0	**V[3:0]**	The value of these bits is used to trim the bandgap reference. Their value is set to the best value for the device during boot. ***The value of these bits should not be changed.***

13.3.34 ECO_TR

External Crystal Oscillator Trim Register

Individual Register Names and Addresses:

ECO_TR: 1,EBh

	7	6	5	4	3	2	1	0
Access : POR	W : 0							
Bit Name	PSSDC[1:0]							

This register sets the adjustment for the 32.768 kHz External Crystal Oscillator.

The value in this register should not be changed. The value is used to trim the 32.768 kHz external crystal oscillator and is set to the device specific, best value during boot. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 23 in the External Crystal Oscillator (ECO) chapter.

Bit	Name	Description
7:6	**PSSDC[1:0]**	Sleep duty cycle. Controls the ratios (in numbers of 32.768 kHz clock periods) of "on" time versus "off" time for PORLVD, Bandgap reference, and pspump. ***These bits should not be changed.*** 00b 1 / 128 01b 1 / 512 10b 1 / 32 11b 1 / 8

13.3.35 FLS_PR1

Flash Program Register 1

Individual Register Names and Addresses:

FLS_PR1: 1,FAh

	7	6	5	4	3	2	1	0
Access : POR							RW : 0	
Bit Name							Bank[1:0]	

This register is used to specify which Flash bank should be used for SROM operations.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the Supervisory ROM (SROM) chapter on page 45.

Bit	Name	Description
1:0	**Bank[1:0]**	Selects the active Flash bank for supervisory operations. No affect in User mode. 00b Flash Bank 0 01b Flash Bank 1 10b Flash Bank 2 11b Flash Bank 3

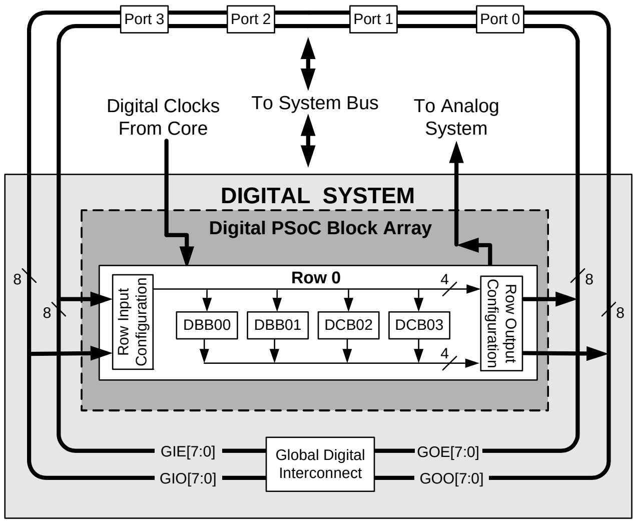
Section D: Digital System

The configurable Digital System section discusses the digital components of the PSoC device and the registers associated with those components. This section encompasses the following chapters:

Top-Level Digital Architecture

The figure below displays the top-level architecture of the PSoC's digital system. Each component of the figure is discussed at length in this section.

Digital System Block Diagram



Interpreting the Digital Documentation

Information in this section covers the CY8C24533, CY8C23533, CY8C23433CY8C24633 PSoC devices. The following table lists the resources available for the CY8C24533, CY8C23533, CY8C23433CY8C24633 (and related) PSoC devices. While reading the digital system section, keep in mind the number of digital rows that are in the CY8C24533, CY8C23533, CY8C23433CY8C24633 is 1.

PSoC Device Characteristics

PSoC Part Number	Digital IO (max)	Digital Rows	Digital Blocks	Analog Inputs	Analog Outputs	Analog Columns	Analog Blocks
CY8C24x23A	24	1	4	12	2	2	6
CY8C24533	26	1	4	12	2	2	4[a]
CY8C23533	26	1	4	12	2	2	4[a]
CY8C23433	26	1	4	12	2	2	4[a]
CY8C24633	25	1	4	12	2	2	4[b]

a. 2 CT, 2 SC, 1 SAR8 ADC.
b. 2 CT, 2 SC, 1 SAR8 ADC.

Digital Register Summary

The table below lists all the PSoC registers for the digital system in address order (Add. column) within their system resource configuration. The bits that are grayed out are reserved bits. If these bits are written, they should always be written with a value of ‘0’. The naming conventions for the digital row registers and the digital block registers are detailed in their respective table title rows.

Note that the CY8C24533, CY8C23533, CY8C23433CY8C24633 are 1 row devices.

Summary Table of the Digital Registers

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
GLOBAL DIGITAL INTERCONNECT (GDI) REGISTERS (page 167)										
1,D0h	GDI_O_IN	GIONOUT7	GIONOUT6	GIONOUT5	GIONOUT4	GIONOUT3	GIONOUT2	GIONOUT1	GIONOUT0	RW : 00
1,D1h	GDI_E_IN	GIENOUT7	GIENOUT6	GIENOUT5	GIENOUT4	GIENOUT3	GIENOUT2	GIENOUT1	GIENOUT0	RW : 00
1,D2h	GDI_O_OU	GOOUTIN7	GOOUTIN6	GOOUTIN5	GOOUTIN4	GOOUTIN3	GOOUTIN2	GOOUTIN1	GOOUTIN0	RW : 00
1,D3h	GDI_E_OU	GOEUTIN7	GOEUTIN6	GOEUTIN5	GOEUTIN4	GOEUTIN3	GOEUTIN2	GOEUTIN1	GOEUTIN0	RW : 00
DIGITAL ROW REGISTERS (page 173)										
x,B0h	RDI0RI							RI0[1:0]		RW : 00
x,B1h	RDI0SYN								RI0SYN	RW : 00
x,B2h	RDI0IS			BCSEL[1:0]		IS3	IS2	IS1	IS0	RW : 00
x,B3h	RDI0LT0	LUT1[3:0]				LUT0[3:0]				RW : 00
x,B4h	RDI0LT1	LUT3[3:0]				LUT2[3:0]				RW : 00
x,B5h	RDI0RO0	GOO5EN	GOO1EN	GOE5EN	GOE1EN	GOO4EN	GOO0EN	GOE4EN	GOE0EN	RW : 00
x,B6h	RDI0RO1	GOO7EN	GOO3EN	GOE7EN	GOE3EN	GOO6EN	GOO2EN	GOE6EN	GOE2EN	RW : 00
DIGITAL BLOCK REGISTERS (page 187)										
Digital Block Data and Control Registers (page 187)										
0,20h	DBB00DR0	Data[7:0]								# : 00
0,21h	DBB00DR1	Data[7:0]								W : 00
0,22h	DBB00DR2	Data[7:0]								# : 00
0,23h	DBB00CR0	Function control/status bits for selected function[6:0]							Enable	# : 00
1,20h	DBB00FN	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]			RW : 00
1,21h	DBB00IN	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,22h	DBB00OU	AUXCLK		AUXEN	AUX IO Select[1:0]		OUTEN	Output Select[1:0]		RW : 00
0,24h	DBB01DR0	Data[7:0]								# : 00
0,25h	DBB01DR1	Data[7:0]								W : 00
0,26h	DBB01DR2	Data[7:0]								# : 00
0,27h	DBB01CR0	Function control/status bits for selected function[6:0]							Enable	# : 00
1,24h	DBB01FN	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]			RW : 00
1,25h	DBB01IN	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,26h	DBB01OU	AUXCLK		AUXEN	AUX IO Select[1:0]		OUTEN	Output Select[1:0]		RW : 00
0,28h	DCB02DR0	Data[7:0]								# : 00
0,29h	DCB02DR1	Data[7:0]								W : 00
0,2Ah	DCB02DR2	Data[7:0]								# : 00
0,2Bh	DCB02CR0	Function control/status bits for selected function[6:0]							Enable	# : 00
1,28h	DCB02FN	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]			RW : 00
1,29h	DCB02IN	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,2Ah	DCB02OU	AUXCLK		AUXEN	AUX IO Select[1:0]		OUTEN	Output Select[1:0]		RW : 00

Summary Table of the Digital Registers *(continued)*

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,2Ch	DCB03DR0	Data[7:0]								# : 00
0,2Dh	DCB03DR1	Data[7:0]								W : 00
0,2Eh	DCB03DR2	Data[7:0]								# : 00
0,2Fh	DCB03CR0	Function control/status bits for selected function[6:0]							Enable	# : 00
1,2Ch	DCB03FN	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]			RW : 00
1,2Dh	DCB03IN	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,2Eh	DCB03OU	AUXCLK		AUXEN	AUX IO Select[1:0]		OUTEN	Output Select[1:0]		RW : 00
Digital Block Interrupt Mask Registers (page 187)										
0,E1h	INT_MSK1					DCB03	DCB02	DBB01	DBB00	RW : 00

LEGEND

x	An ‘x’ before the comma in the address field indicates that this register can be read or written to no matter what bank is used. R: Read register or bit(s).
#	Access is bit specific. Refer to the Register Details chapter on page 47 for additional information.
R	Read register or bit(s).
W	Write register or bit(s).

14. Global Digital Interconnect (GDI)

This chapter discusses the Global Digital Interconnect (GDI) and its associated registers. The PSoC devices, CY8C24533, CY8C23533, CY8C23433CY8C24633, have the exact same global digital interconnect options, varying only in the number of 8-bit ports connected to the globals, as all PSoC CY8C2xxxx devices (except for the CY8C25122 and CY8C26xxx devices). For a complete table of the GDI registers, refer to the "Summary Table of the Digital Registers" on page 162. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

14.1 Architectural Description

Global Digital Interconnect (GDI) consists of four 8-bit buses (refer to the figures that follow). Two of the buses are input buses, which allow signals to pass from the device pins to the core of the PSoC device. These buses are called Global Input Odd (GIO[7:0]) and Global Input Even (GIE[7:0]). The other two buses are output buses that allow signals to pass from the core of the PSoC device to the device pins. They are called Global Output Odd (GOO[7:0]) and Global Output Even (GOE[7:0]). The word "odd" or "even" in the bus name indicates which device ports the bus connects to. Buses with odd in their name connect to all odd numbered ports. Buses with even in their name connect to all even numbered ports.

There are two ends to the global digital interconnect core signals and port pins. An end may be configured as a source or a destination. For example, a GPIO pin may be configured to drive a global input or receive a global output and drive it to the package pin. Globals cannot "loop through" a GPIO. Currently, there are two types of core signals connected to the global buses. The digital blocks, which may be a source or a destination for a global ***net***, and system clocks, which may only drive global nets.

Many of the digital clocks may also be driven on to the global bus to allow the clocks to route directly to IO pins. This is shown in the global interconnect block diagrams on the following pages. For more information on this feature, see the Digital Clocks chapter on page 275.

Each global input and global output has a ***keeper*** on it. The keeper sets the value of the global to '1' on system reset and holds the last driven value of the global should it become un-driven.

The primary goal, of the architectural block diagrams that follow, is to communicate the relationship between global buses (GOE, GOO, GIE, GIO) and pins. Note that any global input may be connected to its corresponding global output, using the tri-state buffers located in the corners of the figures. Also, global outputs may be shorted to global inputs using these tri-state buffers. The rectangle in the center of the figure represents the array of digital PSoC blocks.

14.1.1 28-Pin Global Interconnect

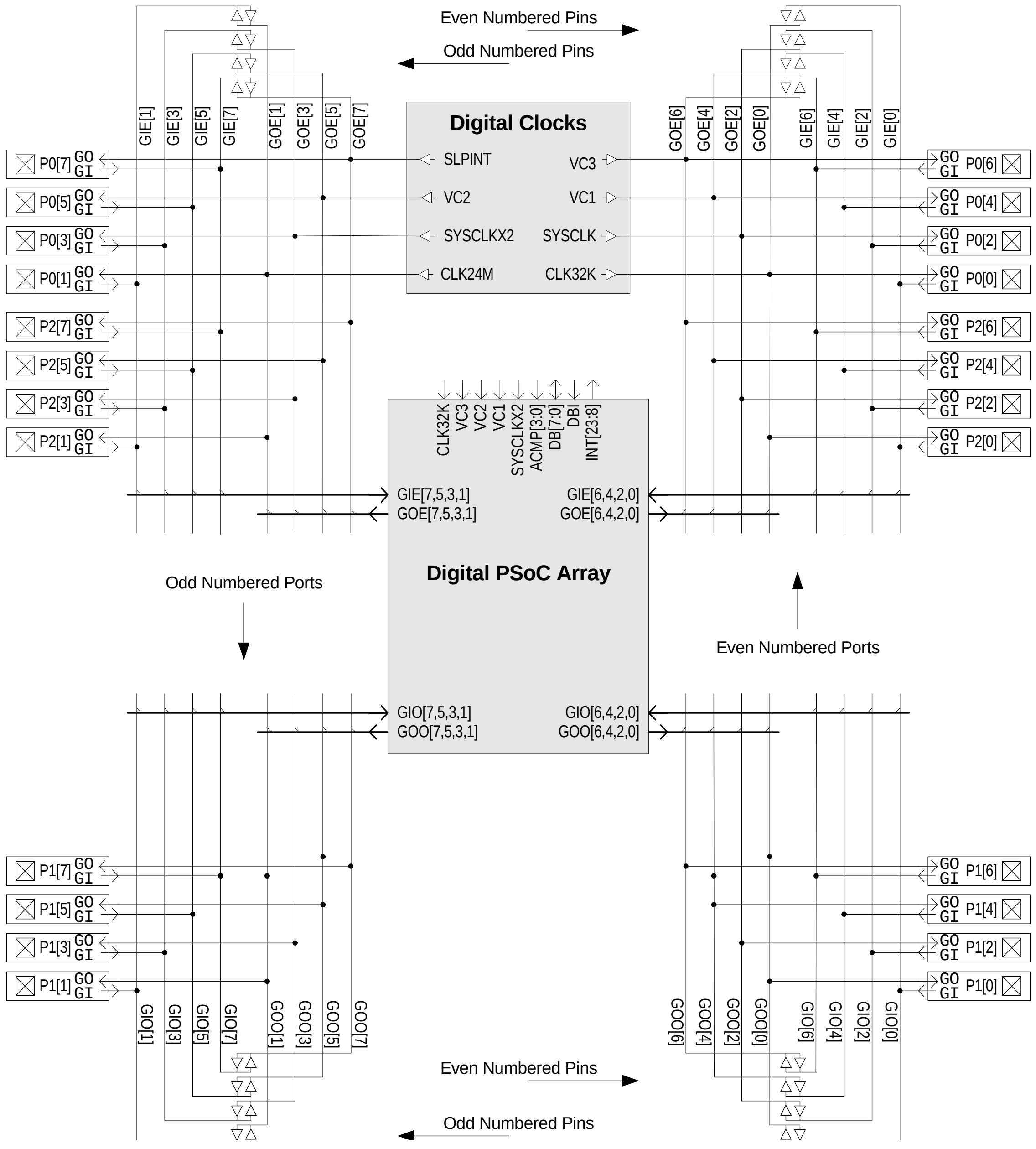
For the 28-pin PSoC device, there are three 8-bit ports. Therefore, there are two ports connected to the even global buses and one port connected to the odd global buses. Table 14-1 lists the mapping between global buses and ports.

Table 14-1. 28- (up to 32-) Pin Global Bus to Port Mapping

Global Bus	Ports
GIO[7:0], GOO[7:0]	P1
GIE[7:0], GOE[7:0]	P0, P2

Because up to two ports are connected to a single global bus, there is a one-to-many mapping between individual nets in a global bus and port pins. For example, if GIE[1] is used to bring an input signal into a digital PSoC block, either pin P0[1] or P2[1] may be used. The same is true for the outputs. For example, if GOE[3] is used to carry a signal from a digital PSoC block to a port pin, either or both of the following pins may be used: P0[3] or P2[3]. Only Port 1 pins connect to the GIO/GOO globals in the 28-pin PSoC device.

Figure 14-1. Global Interconnect Block Diagram for a 28- (up to 32-) Pin Package


Even Numbered Pins
Odd Numbered Pins
Digital Clocks
SLPINT
VC3
VC2
VC1
SYSCLKX2
SYSCLK
CLK24M
CLK32K
GIE[1] GIE[3] GIE[5] GIE[7] GOE[1] GOE[3] GOE[5] GOE[7]
GOE[6] GOE[4] GOE[2] GOE[0] GIE[6] GIE[4] GIE[2] GIE[0]
P0[7] P0[5] P0[3] P0[1] P2[7] P2[5] P2[3] P2[1]
P0[6] P0[4] P0[2] P0[0] P2[6] P2[4] P2[2] P2[0]
GO GI
CLK32K VC3 VC2 VC1 SYSCLKX2 ACMP[3:0] DB[7:0] DBI INT[23:8]
GIE[7,5,3,1]
GOE[7,5,3,1]
GIE[6,4,2,0]
GOE[6,4,2,0]
Digital PSoC Array
Odd Numbered Ports
Even Numbered Ports
GIO[7,5,3,1]
GOO[7,5,3,1]
GIO[6,4,2,0]
GOO[6,4,2,0]
P1[7] P1[5] P1[3] P1[1]
P1[6] P1[4] P1[2] P1[0]
GIO[1] GIO[3] GIO[5] GIO[7] GOO[1] GOO[3] GOO[5] GOO[7]
GOO[6] GOO[4] GOO[2] GOO[0] GIO[6] GIO[4] GIO[2] GIO[0]
Even Numbered Pins
Odd Numbered Pins

14.2 Register Definitions

The following registers are associated with the Global Digital Interconnect and are listed in address order. Each register description has an associated register table showing the bit structure for that register. For a complete table of GDI registers, refer to the "Summary Table of the Digital Registers" on page 162.

14.2.1 GDI_x_IN Registers

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,D0h	GDI_O_IN	GIONOUT7	GIONOUT6	GIONOUT5	GIONOUT4	GIONOUT3	GIONOUT2	GIONOUT1	GIONOUT0	RW : 00
1,D1h	GDI_E_IN	GIENOUT7	GIENOUT6	GIENOUT5	GIENOUT4	GIENOUT3	GIENOUT2	GIENOUT1	GIENOUT0	RW : 00

The Global Digital Interconnect Odd and Even Input Registers (GDI_x_IN) are used to configure a global input to drive a global output.

The PSoC device has a configurable Global Digital Interconnect (GDI). Note that the GDI_x_IN and GDI_x_OU registers should never have the same bits connected. This results in multiple drivers of one bus.

Bits 7 to 0: GIxNOUTx. Using the configuration bits in the GDI_x_IN registers, a global input net may be configured to drive its corresponding global output net. For example,

$$GIE[7] \rightarrow GOE[7]$$

The configurability of the GDI does not allow odd and even nets or nets with different indexes to be connected. The following are examples of connections that are not possible in the PSoC devices.

$$GOE[7] \nrightarrow GIO[7]$$
$$GOE[0] \nrightarrow GIE[7]$$

There are a total of 16 bits that control the ability of global inputs to drive global outputs. These bits are in the GDI_x_IN registers. Table 14-2 enumerates the meaning of each bit position in either of the GDI_O_IN or GDI_E_IN registers.

Table 14-2. GDI_x_IN Register

GDI_x_IN[0]	0: No connection between GIx[0] to GOx[0] 1: Allow GIx[0] to drive GOx[0]
GDI_x_IN[1]	0: No connection between GIx[1] to GOx[1] 1: Allow GIx[1] to drive GOx[1]
GDI_x_IN[2]	0: No connection between GIx[2] to GOx[2] 1: Allow GIx[2] to drive GOx[2]
GDI_x_IN[3]	0: No connection between GIx[3] to GOx[3] 1: Allow GIx[3] to drive GOx[3]
GDI_x_IN[4]	0: No connection between GIx[4] to GOx[4] 1: Allow GIx[4] to drive GOx[4]
GDI_x_IN[5]	0: No connection between GIx[5] to GOx[5] 1: Allow GIx[5] to drive GOx[5]
GDI_x_IN[6]	0: No connection between GIx[6] to GOx[6] 1: Allow GIx[6] to drive GOx[6]
GDI_x_IN[7]	0: No connection between GIx[7] to GOx[7] 1: Allow GIx[7] to drive GOx[7]

For additional information, refer to the GDI_O_IN register on page 144 and the GDI_E_IN register on page 145.

14.2.2 GDI_x_OU Registers

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,D2h	GDI_O_OU	GOOUTIN7	GOOUTIN6	GOOUTIN5	GOOUTIN4	GOOUTIN3	GOOUTIN2	GOOUTIN1	GOOUTIN0	RW : 00
1,D3h	GDI_E_OU	GOEUTIN7	GOEUTIN6	GOEUTIN5	GOEUTIN4	GOEUTIN3	GOEUTIN2	GOEUTIN1	GOEUTIN0	RW : 00

The Global Digital Interconnect Odd and Even Output Registers (GDI_x_OU) are used to configure a global output to drive a global input.

The PSoC device has a configurable Global Digital Interconnect (GDI). Note that the GDI_x_IN and GDI_x_OU registers should never have the same bits connected. This results in multiple drivers of one bus.

Bits 7 to 0: GOxUTINx. Using the configuration bits in the GDI_x_OU registers, a global output net may be configured to drive its corresponding global input. For example,

$$GOE[7] \rightarrow GIE[7]$$

The configurability of the GDI does not allow odd and even nets or nets with different indexes to be connected. The following are examples of connections that are not possible in the PSoC devices.

$$GOE[7] \nrightarrow GIO[7]$$
$$GOE[0] \nrightarrow GIE[7]$$

There are a total of 16 bits that control the ability of global outputs to drive global inputs. These bits are in the GDI_x_OU registers. Table 14-3 enumerates the meaning of each bit position in either of the GDI_O_OU or GDI_E_OU registers.

Table 14-3. GDI_x_OU Register

GDI_x_OU[0]	0: No connection between GIx[0] to GOx[0] 1: Allow GOx[0] to drive GIx[0]
GDI_x_OU[1]	0: No connection between GIx[1] to GOx[1] 1: Allow GOx[1] to drive GIx[1]
GDI_x_OU[2]	0: No connection between GIx[2] to GOx[2] 1: Allow GOx[2] to drive GIx[2]
GDI_x_OU[3]	0: No connection between GIx[3] to GOx[3] 1: Allow GOx[3] to drive GIx[3]
GDI_x_OU[4]	0: No connection between GIx[4] to GOx[4] 1: Allow GOx[4] to drive GIx[4]
GDI_x_OU[5]	0: No connection between GIx[0] to GOx[5] 1: Allow GOx[5] to drive GIx[5]
GDI_x_OU[6]	0: No connection between GIx[6] to GOx[6] 1: Allow GOx[6] to drive GIx[6]
GDI_x_OU[7]	0: No connection between GIx[7] to GOx[7] 1: Allow GOx[7] to drive GIx[7]

For additional information, refer to the GDI_O_OU register on page 146 and the GDI_E_OU register on page 147.

15. Array Digital Interconnect (ADI)

This chapter presents the Array Digital Interconnect (ADI). The digital PSoC array uses a scalable architecture that is designed to support from one to four digital PSoC rows, as defined in the Row Digital Interconnect (RDI) chapter on page 171. The digital PSoC array does not have any configurable interconnect; therefore, there are no associated registers in this chapter.

15.1 Architectural Description

The Array Digital Interconnect (ADI) for the CY8C24533, CY8C23533, CY8C23433CY8C24633 PSoC devices is shown in Figure 15-1. The ADI is not configurable; therefore, the information in this chapter is provided to improve the reader's understanding of the structure.

Figure 15-1. Digital PSoC Block Array Structure

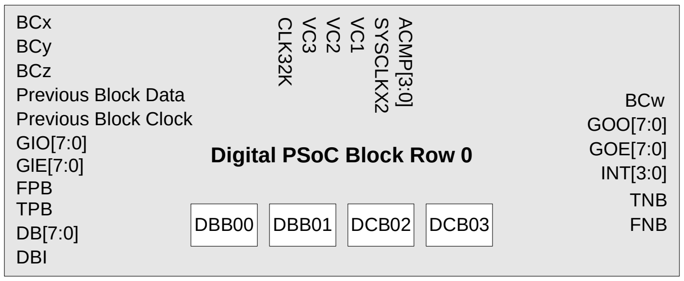


Figure 15-1 illustrates how all rows are connected to the same globals, clocks, and so on. The figure also illustrates how the broadcast clock nets (BCrowx) are connected.

The digital PSoC blocks in the digital array are arranged into rows and the ADI provides a regular interconnect architecture between the Global Digital Interconnect (GDI) and the Row Digital Interconnect (RDI). The most important aspect of the ADI and the digital PSoC rows is that all digital PSoC rows have the same connections to global inputs and outputs. The connections that make a row's position unique are explained as follows.

- **Register Address**: Rows and the blocks within them need to have unique register addresses.
- **Interrupt Priority**: Each digital PSoC block has its own interrupt priority and vector. A row's position in the array determines the relative priority of the digital PSoC blocks within the row. The lower the row number, the higher the interrupt priority, and the lower the interrupt vector address.
- **Broadcast**: Each digital PSoC row has an internal ***broadcast net*** that may be either driven internally, by one of the four digital PSoC blocks, or driven externally. In the case where the broadcast net is driven externally, the source may be any one of the other rows in the array. Therefore, depending on the row's position in the array, it has different options for driving its broadcast net.
- **Chaining Position**: Rows in the array form a string of digital blocks equal in length to the number of rows multiplied by four. The first block in the first row and the last block in the last row are not connected; therefore, the array does not form a loop. The first row in the array has its previous ***chaining*** inputs tied low. If there is a second row in the array, the next chaining outputs are connected to the next row. For the last row in the array, the next inputs are tied low.

16. Row Digital Interconnect (RDI)

This chapter explains the Row Digital Interconnect (RDI) and its associated registers. This chapter discusses a single digital PSoC block row. It does not discuss the functions, inputs, or outputs for individual digital PSoC blocks. Therefore, the information contained here is valid for 1 row configurations. Information about individual digital PSoC blocks is covered in the Digital Blocks chapter on page 177. For a complete table of the RDI registers, refer to the "Summary Table of the Digital Registers" on page 162. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

16.1 Architectural Description

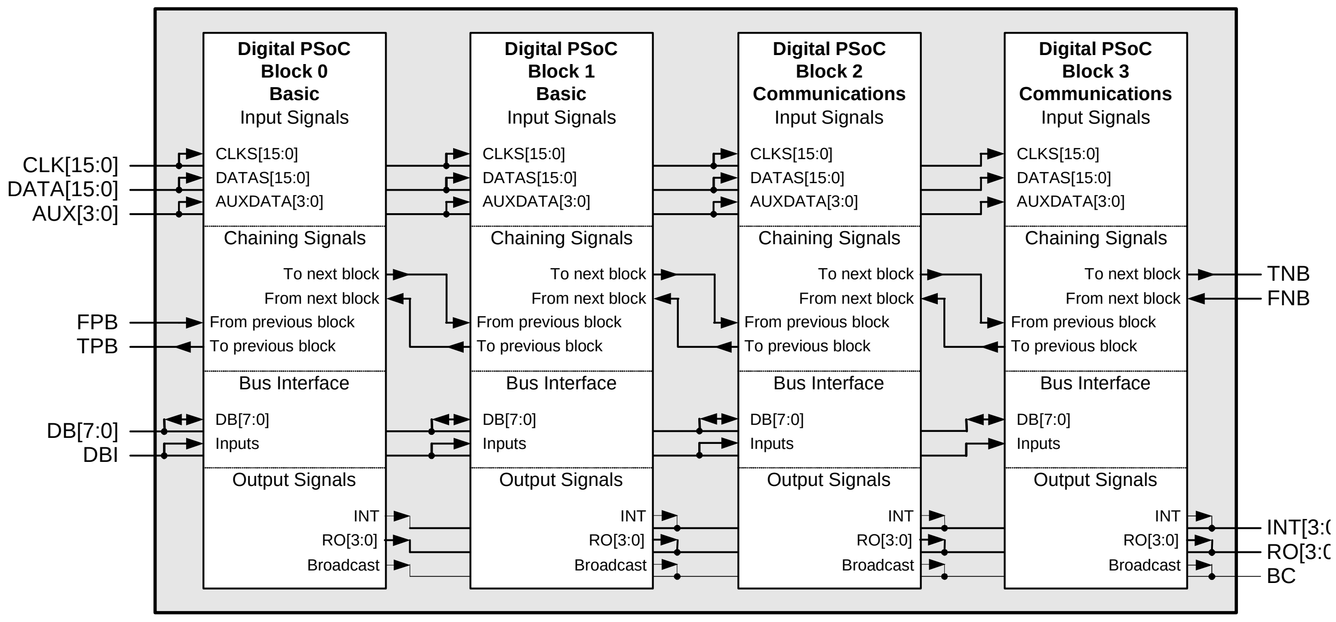
Many signals pass through the digital PSoC block row on their way to or from individual ***digital blocks***. However, only a small number of signals pass though configurable circuits on their way to and from digital blocks. The configurable circuits allow for greater flexibility in the connections between digital blocks and global buses. What follows is a discussion of the signals that are configurable by way of the registers listed in the "Register Definitions" on page 173.

In Figure 16-1, within a digital PSoC block row, there are four digital PSoC Blocks. The first two blocks are of the type basic (DBB).

The second two are of the type communication (DCB). This figure shows the connections between digital blocks within a row. Only the signals that pass outside the gray background box in Figure 16-1 are shown at the next level of hierarchy in Figure 16-2 on page 172.

In Figure 16-2, the detailed view shown in Figure 16-1 of the four PSoC block grouping, has been replaced by the box in the center of the figure labeled "4 PSoC Block Grouping." The rest of the configurable nature of the Row Inputs (RI), Row Outputs (RO), and Broadcast clock net (BC) is shown for the next level of hierarchy.

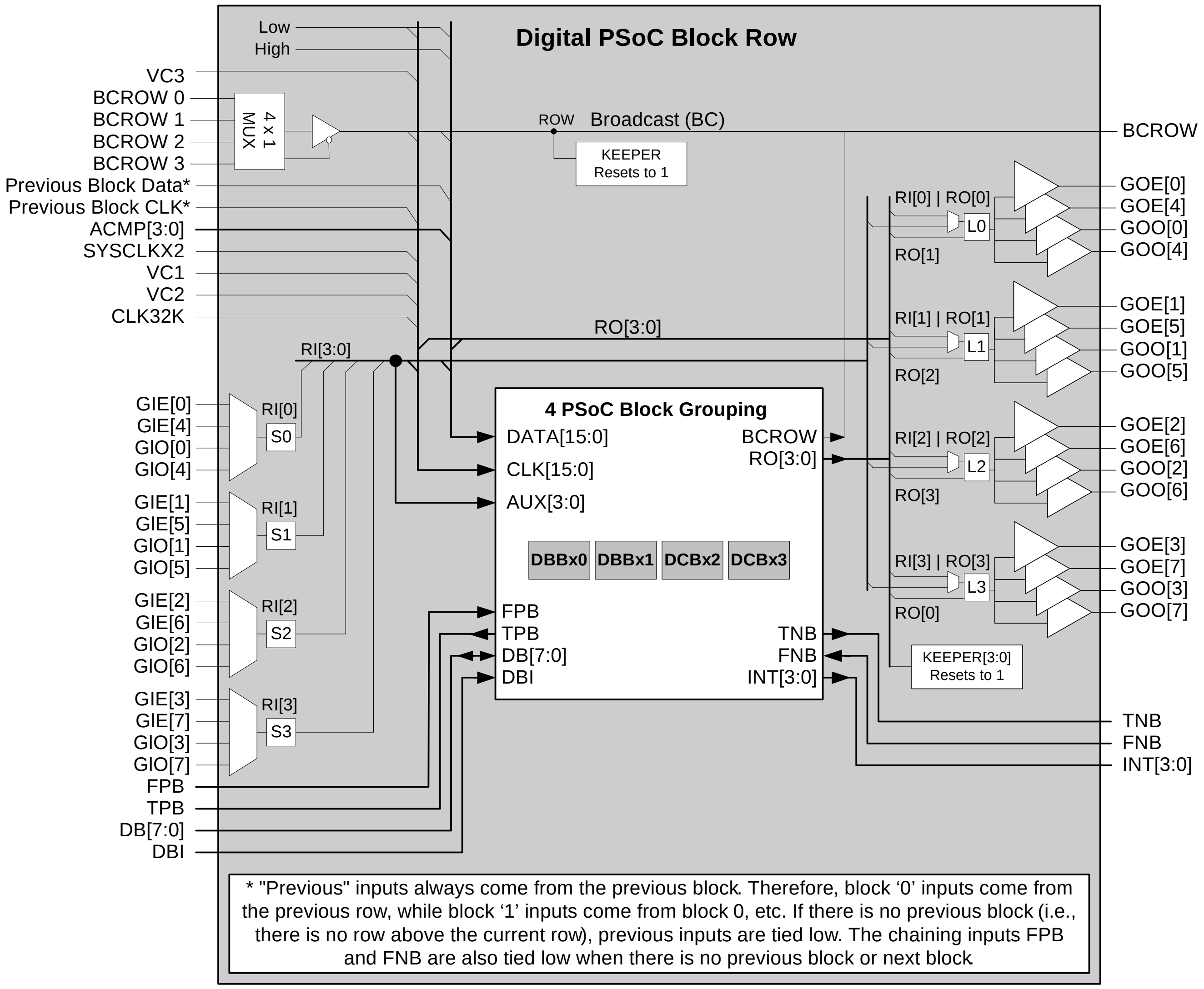
Figure 16-1. Detailed View of Four PSoC Block Grouping



As shown in Figure 16-2, there is a ***keeper*** connected to the row ***broadcast net*** and each of the row outputs. The keeper sets the value of these nets to ‘1’ on system reset and holds the value of the net should it become un-driven.

Notice on the left side of Figure 16-2 that global inputs (GIE[n] and GIO[n]) are inputs to 4-to-1 multiplexers. The output of these muxes are Row Inputs (RI[x]). Because there are four 4-to-1 muxes, each with a unique set of inputs, a row has access to every global input line in a PSoC device.

Figure 16-2. Digital PSoC Block Row Structure



16.2 Register Definitions

The following registers are associated with the Row Digital Interconnect (RDI) and are listed in address order. Each register description has an associated register table showing the bit structure for that register. For a complete table of RDI registers, refer to the "Summary Table of the Digital Registers" on page 162.

Only certain bits are accessible to be read or written. The bits that are grayed out throughout this manual are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of '0'.

The only configurable inputs to a digital PSoC block row are the Global Input Even and Global Input Odd 8-bit buses. The only configurable outputs from the digital PSoC block row are the Global Output Even and Global Output Odd 8-bit buses. Figure 16-2 on page 172 illustrates the relationships between global signals and row signals.

16.2.1 RDIxRI Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,B0h	RDI0RI							RI0[1:0]		RW : 00

LEGEND
x An "x" before the comma in the address field indicates that the register exists in both register banks.

The Row Digital Interconnect Row Input Register (RDIxRI) is used to control the input mux that determines which global inputs drive the row inputs.

The RDIxRI Register and the RDIxSYN Register are the only two registers that affect digital PSoC row input signals. All other registers are related to output signal configuration.

The RDIxRI register has select bits that control four muxes, where "x" denotes a place holder for the row index. Table 16-1 lists the meaning for each mux's four possible settings.

Bits 1 and 0: RI0[1:0]. These bits control the input mux for row 0.

Table 16-1. RDIxRI Register

RI0[1:0]	0h: GIE[0] 1h: GIE[4] 2h: GIO[0] 3h: GIO[4]

For additional information, refer to the RDIxRI register on page 87.

16.2.2 RDIxSYN Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,B1h	RDI0SYN								RI0SYN	RW : 00

LEGEND
x An "x" before the comma in the address field indicates that the register exists in both register banks.

The Row Digital Interconnect Synchronization Register (RDIxSYN) is used to control the input synchronization.

The RDIxRI Register and the RDIxSYN Register are the only two registers that affect digital PSoC row input signals. All other registers are related to output signal configuration.

By default, each row input is double synchronized to the SYSCLK (system clock), which runs at 24 MHz unless external clocking mode is enabled. However, a user may choose to disable this synchronization by setting the appropriate RIxSYN bit in the RDIxSYN register. Table 16-2 lists the bit meanings for each implemented bit of the RDIxSYN register.

Bit 0: RI0SYN. This bit controls the input synchronization for row 0.

Table 16-2. RDIxSYN Register

RI0SYN	0: Row input 0 is synchronized to SYSCLK 1: Row input 0 is passed without synchronization

For additional information, refer to the RDIxSYN register on page 88.

16.2.3 RDIxIS Register

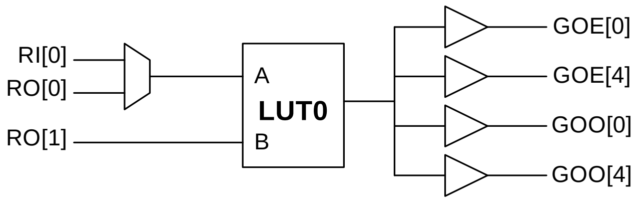
Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,B2h	RDI0IS			BCSEL[1:0]		IS3	IS2	IS1	IS0	RW : 00

LEGEND
x An "x" before the comma in the address field indicates that the register exists in both register banks.

The Row Digital Interconnect Input Select Register (RDIxIS) is used to configure the A inputs to the digital row LUTS and select a broadcast driver from another row, if present.

Each LUT has two inputs, where one of the inputs is configurable (Input A) and the other input (Input B) is fixed to a row output. Figure 16-3 presents an example of LUT configuration.

Figure 16-3. Example of LUT0 Configuration



These bits are the Input B for the ***look-up table (LUT)***. The configurable LUT input (Input A) chooses between a single row output and a single row input. Table 16-3 lists the options for each LUT in a row. The bits are labeled IS, meaning Input Select. The LUT's fixed input is always the RO[LUT number + 1], such as LUT0's fixed input is RO[1], LUT1's fixed input is RO[2], ..., and LUT3's fixed input is RO[0].

Bits 5 and 4: BCSEL[1:0]. These bits are used to determine which digital PSoC row drives the local broadcast net. If a row number is selected that does not exist, the broadcast net is driven to a logic 1 value. If any digital PSoC block in the local row has its DxBxFN[BCEN] bit set, the broadcast select is disabled. See the "DxBxxFN Registers" on page 193.

Bit 3: IS3. This bit controls the 'A' input of LUT 3.

Bit 2: IS2. This bit controls the 'A' input of LUT 2.

Bit 1: IS1. This bit controls the 'A' input of LUT 1.

Bit 0: IS0. This bit controls the 'A' input of LUT 0.

Table 16-3. RDIxIS Register Bits

BCSEL[1:0]	0: Row broadcast net driven by row 0 broadcast net.* 1: Row broadcast net driven by row 0 broadcast net.* 2: Row broadcast net driven by row 0 broadcast net.* 3: Row broadcast net driven by row 0 broadcast net.*
IS3	0: The 'A' input of LUT3 is RO[3] 1: The 'A' input of LUT3 is RI[3]
IS2	0: The 'A' input of LUT2 is RO[2] 1: The 'A' input of LUT2 is RI[2]
IS1	0:The 'A' input of LUT1 is RO[1] 1: The 'A' input of LUT1 is RI[1]
IS0	0: The 'A' input of LUT0 is RO[0] 1: The 'A' input of LUT0 is RI[0]
* When the BCSEL value is equal to the row number, the tri-state buffer that drives the row broadcast net from the input select mux is disabled, so that one of the row's blocks may drive the local row broadcast net. * Refer to Figure 16-2 on page 172. * If the row is not present in the part, the selection provides a logic 1 value.	

For additional information, refer to the RDIxIS register on page 89.

16.2.4 RDIxLTx Registers

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,B3h	RDI0LT0	LUT1[3:0]				LUT0[3:0]				RW : 00
x,B4h	RDI0LT1	LUT3[3:0]				LUT2[3:0]				RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Row Digital Interconnect Logic Table Register 0 and 1 (RDIxLT0 and RDIxLT1) are used to select the logic function of the digital row LUTS.

The outputs from a digital PSoC row are a bit more complicated than the inputs. Figure 16-2 on page 172 illustrates the output circuitry in a digital PSoC row. In the figure, find a block labeled Lx. This block represents a 2-input look-up table (LUT). The LUT allows the user to specify any one of 16 logic functions that should be applied to the two inputs.

The output of the logic function determines the value that may be driven on to the Global Output Even and Global Output Odd buses. Table 16-4 lists the relationship between a look-up table’s four configuration bits and the resulting logic function. Some users may find it easier to determine the proper configuration bits setting by remembering that the configuration’s bits represent the output column of a two-input logic truth table. Table 16-4 lists seven examples of the relationship between the LUT’s output column for a truth table and the LUTx[3:0] configuration bits. Figure 16-3 on page 174 presents an example of LUT configuration.

Bits 7 to 4: LUTx[3:0]. These configuration bits are for a row output LUT.

Bits 3 to 0: LUTx[3:0]. These configuration bits are for a row output LUT.

Table 16-4. Example LUT Truth Tables

A	B	AND	OR	$A+\overline{B}$	$A\&\overline{B}$	A	B	True
0	0	0	0	1	0	0	0	1
0	1	0	1	0	0	0	1	1
1	0	0	1	1	1	1	0	1
1	1	1	1	1	0	1	1	1
LUTx[3:0]		1h	7h	Bh	2h	3h	5h	Fh

Table 16-5. RDIxLTx Register

LUTx[3:0]	0h: 0000: FALSE 1h: 0001: A .AND. B 2h: 0010: A .AND. $\overline{B}$ 3h: 0011: A 4h: 0100: $\overline{A}$ .AND. B 5h: 0101: B 6h: 0110: A .XOR. B 7h: 0111: A .OR. B 8h: 1000: A .NOR. B 9h: 1001: A .XNOR. B Ah: 1010: $\overline{B}$ Bh: 1011: A .OR. $\overline{B}$ Ch: 1100: $\overline{A}$ Dh: 1101: $\overline{A}$ .OR. B Eh: 1110: A. NAND. B Fh: 1111: TRUE

For additional information, refer to the RDIxLT0 register on page 90 and the RDIxLT1 register on page 91.

16.2.5 RDIxROx Registers

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,B5h	RDI0RO0	GOO5EN	GOO1EN	GOE5EN	GOE1EN	GOO4EN	GOO0EN	GOE4EN	GOE0EN	RW : 00
x,B6h	RDI0RO1	GOO7EN	GOO3EN	GOE7EN	GOE3EN	GOO6EN	GOO2EN	GOE6EN	GOE2EN	RW : 00

LEGEND

x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Row Digital Interconnect Row Output Register 0 and 1 (RDIxRO0 and RDIxRO1) are used to select the global nets that the row outputs drive.

The final configuration bits for outputs from digital PSoC rows are in the two RDIxROx registers. These registers hold the 16 bits that can individually enable the tri-state buffers that connect to all eight of the Global Output Even lines and all eight of the Global Output Odd lines to the row LUTs.

The input to these tri-state drivers are the outputs of the row’s LUTs, as shown in Figure 16-2. This means that any row can drive any global output.

16.2.5.1 RDIxRO0 Register

Bits 7 to 4: GOxxEN. These configuration bits enable the tri-state buffers that connect to the Global Output Even lines for LUT 1.

Bits 3 to 0: GOxxEN. These configuration bits enable the tri-state buffers that connect to the Global Output Even lines for LUT 0.

For additional information, refer to the RDIxRO0 register on page 92.

16.2.5.2 RDIxRO1 Register

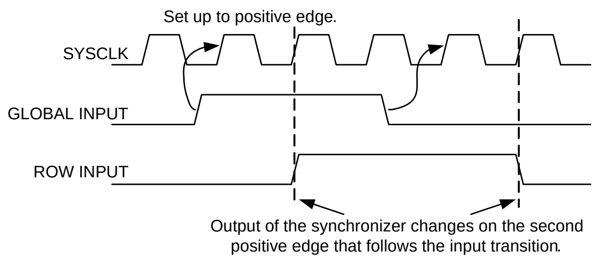
Bits 7 to 4: GOxxEN. These configuration bits enable the tri-state buffers that connect to the Global Output Even lines for LUT 3.

Bits 3 to 0: GOxxEN. These configuration bits enable the tri-state buffers that connect to the Global Output Even lines for LUT 2.

For additional information, refer to the RDIxRO1 register on page 93.

16.3 Timing Diagram

Figure 16-4. Optional Row Input Sync. to SYSCLK


Set up to positive edge.
SYSCLK
GLOBAL INPUT
ROW INPUT
Output of the synchronizer changes on the second positive edge that follows the input transition.

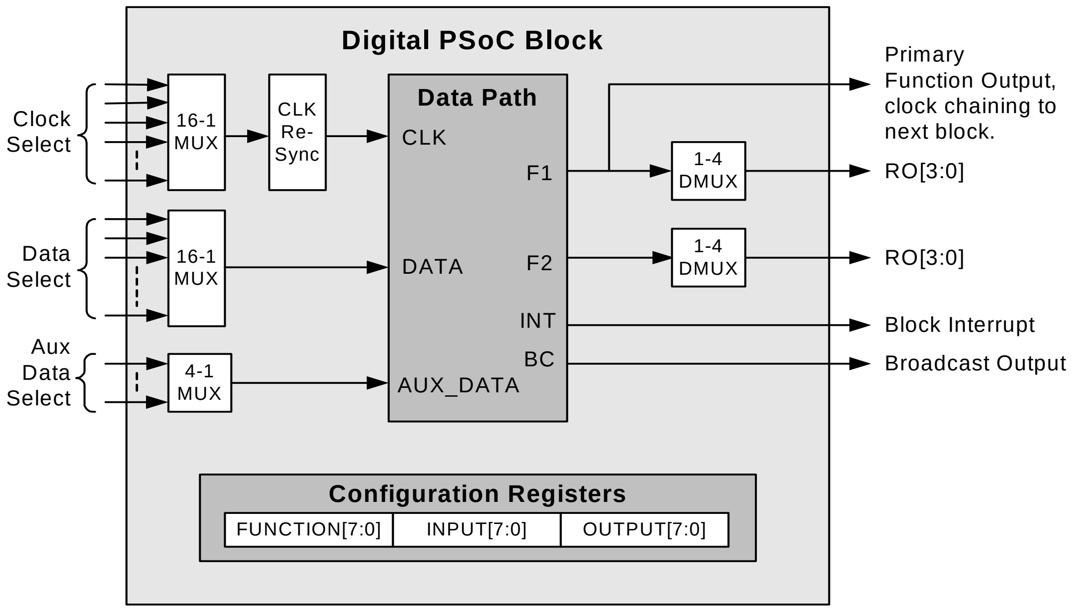
17. Digital Blocks

This chapter covers the configuration and use of the digital PSoC blocks and their associated registers. For a complete table of the Digital PSoC Block registers, refer to the “Summary Table of the Digital Registers” on page 162. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

17.1 Architectural Description

At the top level, the main components of the digital block are the data path, input multiplexers (muxes), output de-muxes, configuration registers, and chaining signals (see Figure 17-1).

Figure 17-1. Digital Blocks Top-Level Block Diagram



All digital PSoC blocks may be configured to perform any one of five basic functions: timer, counter, ***pulse width modulator (PWM)***, pseudo random sequence (PRS), or ***cyclic redundancy check (CRC)***. These functions may be used by configuring an individual PSoC block or chaining several PSoC blocks together to form functions that are greater than 8 bits. Digital communications PSoC blocks have two additional functions: master or slave SPI and a full duplex ***UART***.

Each digital PSoC block’s function is independent of all other PSoC blocks. Up to seven registers are used to determine the function and state of a digital PSoC block. These registers are discussed in the Register Definitions section. Digital PSoC block function registers end with FN. The individual bit settings for a block’s function register are listed in Table 17-14 on page 194. The input registers end with IN and its bit meanings are listed in Table 17-16 on page 195.

Finally, the block's outputs are controlled by the output register, which ends in OU.

Each digital PSoC block also has three data registers (DR0, DR1, and DR2) and one control register (CR0). The bit meanings for these registers are heavily function dependent and are discussed with each function's description.

In addition to seven registers that control the digital PSoC block's function and state, a separate interrupt mask bit is available for each digital PSoC block. Each digital PSoC block has a unique interrupt vector; and therefore, it can have its own interrupt service routine.

17.1.1 Input Multiplexers

Typically, each function has a clock and a data input that may be selected from a variety of sources. Each of these inputs is selected with a 16-to-1 input mux.

In addition, there is a 4-to-1 mux which provides an auxiliary input for the SPI slave function that requires three inputs: Clock, Data, and SS_ (unless the SS_ is forced active with the Aux IO Enable bit). The inputs to this mux are intended to be a selection of the row inputs.

17.1.2 Input Clock Resynchronization

Digital blocks allow a clock selection from one of 16 sources. Possible sources are the system clocks (VC1, VC2, VC3, SYSCLK, and SYSCLKX2), row inputs, and other digital block outputs. To manage clock ***skew*** and ensure that the interfaces between blocks meet timing in all cases, all digital block input clocks must be resynchronized to either SYSCLK or SYSCLKX2, which are the source clocks for all the PSoC device clocking. Also, SYSCLK or SYSCLKX2 may be used directly. The AUXCLK bits in the DxBxxOU register are used to specify the input synchronization. The following rules apply to the use of input clock resynchronization.

1. If the clock input is derived (for example, divided down) from SYSCLK, resynchronize to SYSCLK at the digital block. Most the PSoC device clocks are in this category. For example, VC1 and VC2, and the output of other blocks clocked by VC1 and VC2, or SYSCLK (for setting see Table 17-1).
2. If the clock input is derived from SYSCLKX2, resynchronize to SYSCLKX2. For example, VC3 clocked by SYSCLKX2 or other digital blocks clocked by SYSCLKX2 (for setting see Table 17-1).
3. Choose direct SYSCLK for clocking directly off of SYSCLK (for setting see Table 17-1).
4. Choose direct SYSCLKX2 (select SYSCLKX2 in the Clock Input field of the DxBxxIN register) for clocking directly off of SYSCLKX2.
5. Bypass synchronization. This should be a very rare selection; because if clocks are not synchronized, they may fail set up to CPU read and write commands. However, it is possible for an external pin to asynchronously clock a digital block. For example, if the user is willing to synchronize CPU interaction through interrupts or other techniques (setting 00 in AUXCLK). This setting is also required for blocks to remain active while in sleep.

The note below enumerates configurations that are not allowed, although the hardware does not prevent them. The clock dividers (VC1, VC2, and VC3) may not be configured in such a way as to create an output clock that is equal to SYSCLK or SYSCLKX2.

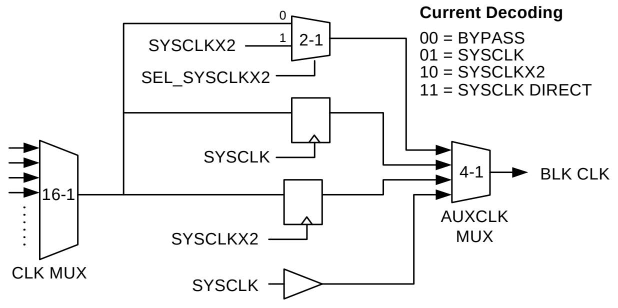
Note If the input clock frequency matches the frequency of the clock used for synchronization, the block never receives a clock (see Figure 17-2). With respect to SYSCLK, this can happen in the following cases:

- Using VC1 configured as divide by 1.
- Using VC2 with VC1 and VC2 both configured as divide by 1.
- Using VC3 divided by 1 with a source of VC1 divided by 1.
- Using VC3 divided by 1 with a source of VC2, where both VC1 and VC2 are divided by 1.
- Using VC3 divided by 1 with SYSCLK source.

In all of these cases, SYSCLK should be selected directly in the block. Similarly, if VC3 is configured as divide by one with a source of SYSCLKX2, then SYSCLKX2 should be selected to clock the block directly instead of VC3.

The clock resynchronizer is illustrated in Figure 17-2.

Figure 17-2. Input Clock Resynchronization



In sleep, SYSCLK is powered down, and therefore input synchronization is not available.

Table 17-1. AUXCLK Bit Selections

Code	Description	Usage
00	Bypass	Use this setting only when SYSCLKX2 (48 MHz) is selected. Other than this case, asynchronous clock inputs are not recommended. This setting is also required for blocks to remain active while in sleep.
01	Resynchronize to SYSCLK (24 MHz)	Use this setting for any SYSCLK-based clock. VC1, VC2, VC3 driven by SYSCLK, digital blocks with SYSCLK-based source clocks, broadcast bus with source based on SYSCLK, row input and row outputs with source based on SYSCLK.
10	Resynchronize to SYSCLKX2 (48 MHz)	Use this setting for any SYSCLKX2-based clock. VC3 driven by SYSCLKX2, digital blocks with SYSCLKX2-based source clocks, broadcast bus with source based on SYSCLKX2, row input and row outputs with source based on SYSCLKX2.
11	SYSCLK Direct	Use this setting to clock the block directly using SYSCLK. Note that this setting is not strictly related to clock resynchronization, but since SYSCLK cannot resync itself, it allows a direct skew controlled SYSCLK source.

17.1.2.1 Clock Resynchronization Summary

- Digital PSoC blocks have extremely flexible clocking configurations. To maintain reliable timing, input clocks must be resynchronized.
- The master clock for any clock in the system is either SYSCLK or SYSCLKX2. Determine the master clock for a given input clock and resynchronize to that clock.
- Do not use divide by 1 clocks derived from SYSCLK and SYSCLKX2. Use the direct SYSCLK or SYSCLKX2 clocking option available at the block.

17.1.3 Output De-Multiplexers

Most functions have two outputs: a primary and an auxiliary output, the meaning of which are function dependent. Each of these outputs may be driven onto the row output bus. Each de-mux is implemented with four tri-state drivers. There are two bits in the output register to select one of the four tri-state drivers and an additional bit to enable the selected driver.

17.1.4 Block Chaining Signals

Each digital block has the capability to be chained and to create functions with bit widths greater than eight. There are signals to propagate information, such as Compare, Carry, Enable, Capture and Gate, from one block to the next to implement higher precision functions. The selection made in the function register determines which signals are appropriate for the desired function. User modules that have been designed to implement digital functions, with greater than 8-bit width, automatically make the proper selections of the chaining signals, to ensure the correct information flow between blocks.

17.1.5 Input Data Synchronization

Any asynchronous input derived from an external source, such as a GPIO pin input, must be resynchronized through the row input before use into any digital block clock or data input. This is the default mode of operation (resynchronization on).

17.1.6 Timer Function

A timer consists of a period register, a ***synchronous*** down counter, and a capture/compare register, all of which are byte wide. When the timer is disabled and a period value is written into DR1, the period value is also loaded into DR0. When the timer is enabled, the counter counts down until positive terminal count (a count of 00h) is reached. On the next clock edge, the period is reloaded and, on subsequent clocks, counting continues. The terminal count signal is the primary function output. (Refer to the timing diagram for this function on page 198.) This can be configured as a full or half-clock cycle.

Hardware capture occurs on the positive edge of the data input. This event transfers the current count from DR0 to DR2. The captured value may then be read directly from DR2. A software capture function is equivalent to a hardware capture. A CPU read of DR0, with the timer enabled, triggers the same capture mechanism. The hardware and software capture mechanisms are OR'ed in the capture circuitry. Since the capture circuitry is positive edge sensitive, during an interval where the hardware capture input is high, a software capture is masked and does not occur.

The timer also implements a compare function between DR0 and DR2. The compare signal is the auxiliary function output. A limitation, in regards to the compare function, is that the capture and compare function both use the same register (DR2). Therefore, if a capture event occurs, it overwrites the compare value.

Mode bit 1 in the function register sets the compare type (DR0 <= DR2 or DR0 < DR2) and Mode bit 0 sets the interrupt type (terminal count or compare).

Timers may be chained in 8-bit lengths up to 32 bits.

17.1.6.1 Usability Exceptions

The following are usability exceptions for the timer function.

1. Capture operation is not supported at 48 MHz.
2. DR2 is not writeable when the timer is enabled.

17.1.6.2 Block Interrupt

The timer block has a selection of three interrupt sources. interrupt on terminal count (TC) and interrupt on compare may be selected in Mode bit 0 of the function register. The third interrupt source, interrupt on capture, may be selected with the Capture Interrupt bit in the control register.

- **Interrupt on Terminal Count**: The positive edge of terminal count (primary output) generates an interrupt for this block. The timing of the interrupt follows the TC pulse width setting in the control register.
- **Interrupt on Compare**: The positive edge of compare (auxiliary output) generates an interrupt for this block.
- **Interrupt on Capture**: Hardware or software capture generates an interrupt for this block. The interrupt occurs at the closing of the DR2 latch on capture.

17.1.7 Counter Function

A counter consists of a period register, a synchronous down counter, and a compare register. The counter function is identical to the timer function except for the following points:

- The data input is a counter gate (enable), rather than a capture input. Counters do not implement synchronous capture. The DR0 register in a counter should not be read when it is enabled.
- The compare output is the primary output and the terminal count (TC) is the auxiliary output (opposite of the timer).
- Terminal count output is full cycle only.

When the counter is disabled and a period value is written into DR1, the period value is also loaded into DR0. When the counter is enabled, the counter counts down until terminal count (a count of 00h) is reached. On the next clock edge, the period is reloaded and, on subsequent clocks, counting continues. (Refer to the timing diagram for this function on page 199.)

The counter implements a compare function between DR0 and DR2. The compare signal is the primary function output. Mode bit 1 sets the compare type (DR0 <= DR2 or DR0 < DR2) and Mode bit 0 sets the interrupt type (terminal count or compare).

The data input functions as a gate to counter operation. The counter only counts and reloads when the data input is asserted (logic 1). When the data input is negated (logic 0), counting (including the period reload) is halted.

Counters may be chained in 8-bit blocks up to 32 bits.

17.1.7.1 Usability Exceptions

The following is a usability exception for the counter function.

1. DR0 may only be read (to transfer DR0 data to DR2) when the block is disabled.

17.1.7.2 Block Interrupt

The counter block has a selection of two interrupt sources. interrupt on terminal count (TC) and interrupt on compare may be selected in Mode bit 0 of the function register.

- **Interrupt on Terminal Count**: The positive edge of terminal count (auxiliary output) generates an interrupt for this block. The timing of the interrupt follows the TC pulse width setting in the control register.
- **Interrupt on Compare**: The positive edge of compare (primary output) generates an interrupt for this block.

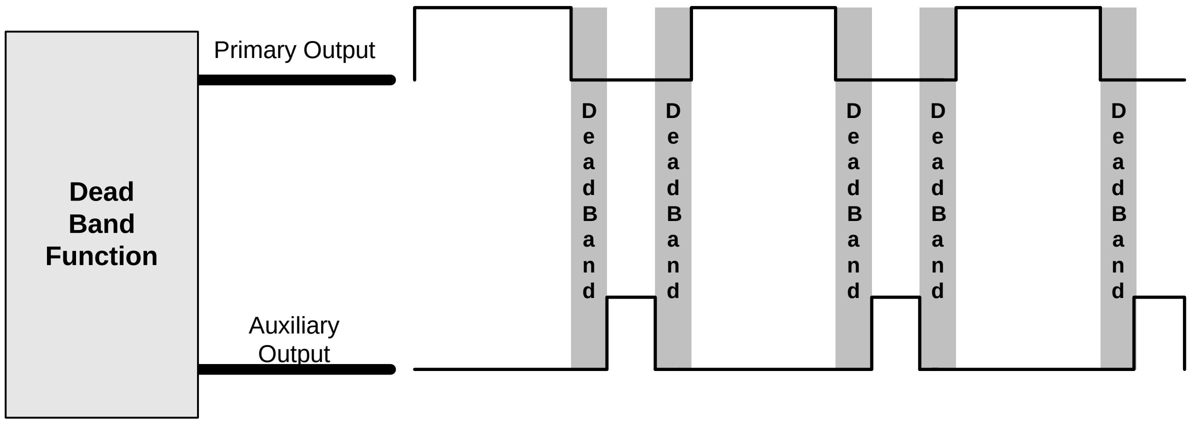
17.1.8 Dead Band Function

The dead band function generates output signals on both the primary and auxiliary outputs of the block, see Figure 17-3. Each of these outputs is one ***phase*** of a two-phase, non-overlapping clock generated by this function. The two clock phases are never high at the same time and the period between the clock phases is known as the ***dead band***. The width of the dead band time is determined by the value in the period register. This dead band function can be driven with a PWM as an input clock or it can be clocked directly by toggling a bit in software using the Bit Bang interface. If the clock source is a PWM, this makes a two output PWM with guaranteed non-overlapping outputs. An active asynchronous signal on the KILL data input disables both outputs immediately.

The PWM with the Dead Band User Module configures one or two blocks to create an 8- or 16-bit PWM and configures an additional block as the dead band function.

A dead band consists of a period register, a synchronous down counter, and a special dead band circuit. The DR2 register is only used to read the contents of DR0. As with the counter, when the dead band is disabled and a period value is written into DR1, the period value is also loaded into DR0. (Refer to the timing diagrams for this function on page 200.)

Figure 17-3. Dead Band Functional Overview



The dead band has two inputs: a PWM reference signal and a KILL signal. The PWM reference signal may be derived from one of two sources. By default, it is hardwired to be the primary output of the previous block. This previous block output is wired as an input to the 16-to-1 clock input mux. In the dead band case, as the previous block output is wired directly to the dead band reference input. If this mode is used, a PWM, or some other ***waveform*** generator, must be instantiated in the previous digital block. There is also an optional Bit Bang mode. In this mode, firmware toggles a register bit to generate a PWM reference; and therefore, the dead band may be used as a standalone block.

The KILL signal is derived from the data input signal to the block. Mode [1:0] is encoded as the Kill Type. In all cases when kill is asserted, the output is forced low immediately. Mode bits are encoded for kill options and are detailed in the following table.

Table 17-2. Dead Band KILL Options

Mode [1:0]	Description
00b	Synchronous Restart KILL mode. Internal state is reset and reference edges are ignored, until the KILL signal is negated.
01b	Disable KILL mode. Block is disabled. KILL signal must be negated and user must re-enable the block in firmware to resume operation.
10b	Asynchronous KILL mode. Outputs are low only for the duration that the KILL signal is asserted, subject to a minimum disable time between one-half to one and one-half clock cycles. Internal state is unaffected.
11b	Reserved.

When the block is initially enabled, both outputs are low. After enabling, a positive or negative edge of the incoming PWM reference enables the counter. The counter counts down from the period value to terminal count. At terminal count, the counter is disabled and the selected phase is asserted high. On the opposite edge of the PWM input, the output that was high is negated low and the process is repeated with the opposite phase. This results in the generation of a two phase non-overlapping clock matching the frequency and pulse width of the incoming PWM reference, but separated by a dead time derived from the period and the input clock.

There is a deterministic relationship between the incoming PWM reference and the output phases. The positive edge of the reference causes the primary output to be asserted to '1' and the negative edge of the reference causes the auxiliary output to be asserted to '1'.

17.1.8.1 Usability Exceptions

The following are usability exceptions for the dead band function.

1. The dead band function may not be chained.
2. Programming a dead band period value of 00h is not supported. The block output is undefined under this condition.
3. If the period (of either the ***high time*** or the ***low time*** of the reference input) is less than the programmed dead time, then the associated output phase is held low.
4. DR0 may only be read (to transfer DR0 data to DR2) when the block is disabled.
5. If the asynchronous KILL signal is being used in a given application, the output of the dead band cannot be connected directly to the input of another digital block in the same row. Since the kill is asynchronous, the digital block output must be resynchronized through a row input before using it as a digital block input.

17.1.8.2 Block Interrupt

The dead band block has one fixed interrupt source, which is the Phase 1 primary output clock. When the KILL signal is asserted, the interrupt follows the same behavior of the Phase 1 output with respect to the various KILL modes.

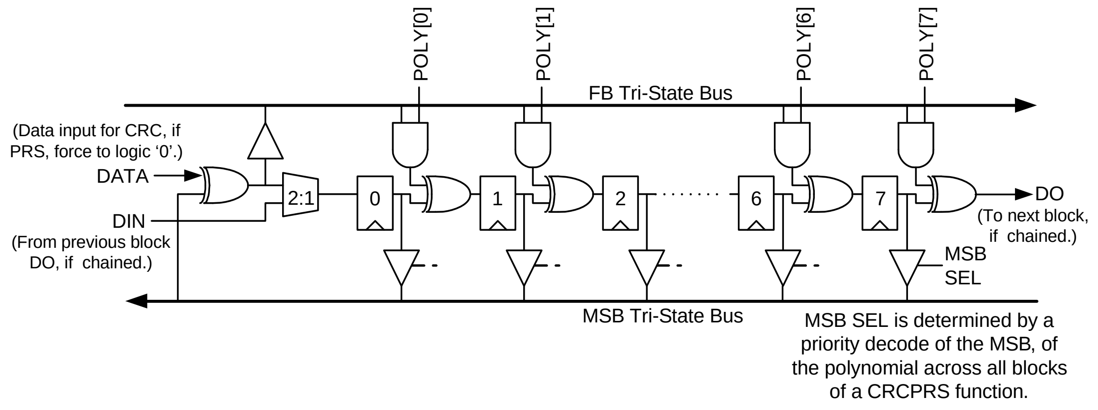
17.1.9 CRCPRS Function

A Cyclic Redundancy Check/Pseudo Random Sequence (CRCPRS) function consists of a polynomial register, a ***Linear Feedback Shift Register (LFSR)***, and a seed register. (See Figure 17-4 on page 182.) When the CRCPRS block is disabled and a ***seed value*** is written into DR2, the seed value is also loaded into DR0. When the CRCPRS is enabled, and synchronous clock and data are applied to the inputs, a CRC is computed on the ***serial*** data input stream. When the data input is forced to '0', then the block functions as a pseudo random sequencer (PRS) generator with the output data generated at the clock rate. The most significant bit (MSb) of the CRCPRS function is the primary output.

The CRCPRS has a selection of compare modes between DR0 and DR2. The default behavior of the compare is DR0==DR2. When the PRS function cycles through the seed value as one of the valid counts, the compare output is asserted high for one clock cycle. This is regarded as the epoch of the pseudo random sequence. The mode bits can be used to set other compare types. Setting Mode bit 0 to '1' causes the compare behavior to revert to DR0 <= DR2 or DR0 < DR2, depending upon Mode bit 1. The compare value is the auxiliary output. An interrupt is generated on compare true.

CRCPRS mode offers an optional pass function. By setting the Pass Mode bit in the CR0 register (bit 1), the CRCPRS function is overridden. In this mode, the data input is passed transparently to the primary output and an interrupt is generated on the rising of the data input. Similarly, the CLK input is passed transparently to the auxiliary output. This can only be used to pass signals to the global outputs. If the output of a pass function is needed as an input to another digital block, it must be resynchronized through the globals and row inputs.

Figure 17-4. CRCPRS LFSR Structure


POLY[0]
POLY[1]
POLY[6]
POLY[7]
FB Tri-State Bus
(Data input for CRC, if PRS, force to logic '0'.)
DATA
DIN
(From previous block DO, if chained.)
2:1
0
1
2
6
7
DO
(To next block, if chained.)
MSB SEL
MSB Tri-State Bus
MSB SEL is determined by a priority decode of the MSB, of the polynomial across all blocks of a CRCPRS function.

LSFR Structure

The LSFR (Linear Feedback Shift register) structure, as shown in Figure 17-4, is implemented as a modular ***shift*** register generator. The least significant block in the chain inputs the MSb and XORs it with the DATA input, in the case of CRC computation. For PRS computation, the DATA input is forced to logic 0 (by input selection); and therefore, the MSb bus is directly connected to the FB bus. In the case of a chained block, the data input (DIN) comes directly from the data output (DO) of the LFSR in the previous block. The MSb selection, derived from the priority decode of the polynomial, enables one of the tri-state drivers to drive the MSb bus.

Determining the CRC Polynomial

Computation of an n-bit result is generally specified by a polynomial with n+1 terms, the last of which is X_{16}, where

$$X_0 = 1 \quad \textbf{Equation 1}$$

As an example, the CRC-CCIT 16-bit polynomial is:

$$CRC - CCIT = X_{16} + X_{12} + X_5 + 1 \quad \textbf{Equation 2}$$

The CRCPRS hardware assumes the presence of the X_0 term; and therefore, this polynomial can be expressed in 16 bits as 1000100000010000 or 8810h. Two consecutive digital blocks may be allocated to perform this function, with 88h as the MS block polynomial (DR1) and 10h as the LS block polynomial value.

Determining the PRS Polynomial

Generally, PRS polynomials are selected from pre-computed reference tables. It is important to note that there are two common ways to specify a PRS polynomial: simple register configuration and modular configuration. In the simple method, a ***shift register*** is implemented with a reduction XOR of the MSb and feedback taps as input into the least significant bit.

In the modular method, there is an XOR operation implemented between each register bit and each tap point enables the XOR with the MSb for that given bit. The CRCPRS function implements the modular approach.

These are equivalent methods. However, there is a conversion that should be understood. If tables are specified in simple register format, then a conversion can be made to the modular format by subtracting each tap from the MS tap, as shown in the following example.

To implement a 7-bit PRS of length 127, one possible code is [7,6,4,2]s, which is in simple format. The modular format is [7,7-6,7-4,7-2]m or [7,1,3,5]m which is equivalent to [7, 5, 3, 1]. Determining the polynomial to program is similar to the CRC example above. Set a ***binary*** bit for each tap (with bit 0 of the register corresponding to tap 1). Therefore, the code [7,5,3,1] corresponds to 01010101 or 55h.

In both the CRC and PRS cases, an appropriate seed value should be selected. All ones for PRS, or all ones or all zeros for CRC are typical values. Note that a seed value of all zeros should not be used in a PRS function, because PRS counting is inhibited by this seed.

17.1.9.1 Usability Exceptions

Following is a usability exception for the CRCPRS function.

1. The polynomial register must only be written when the block is disabled.

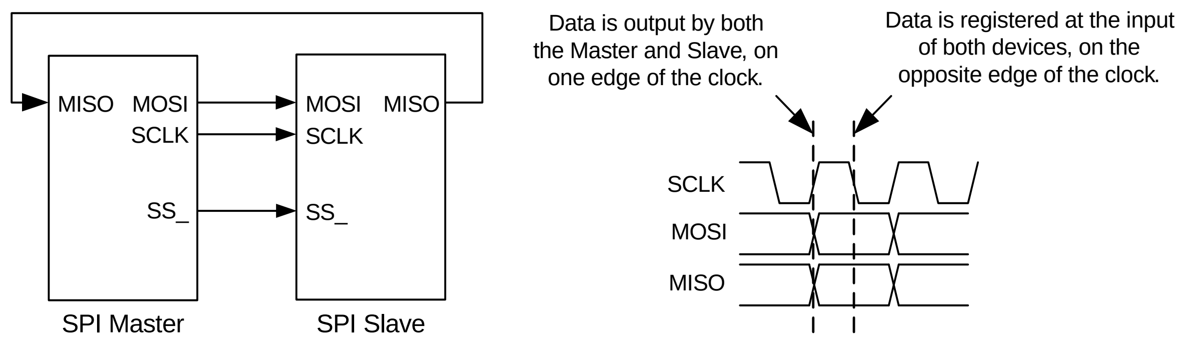
17.1.9.2 Block Interrupt

The CRCPRS block has one fixed interrupt source, which is the compare auxiliary output.

17.1.10 SPI Protocol Function

The Serial Peripheral Interface (SPI) is a Motorola™ specification for implementing full-duplex synchronous serial communication between devices. The 3-wire ***protocol*** uses both edges of the clock to enable synchronous communication, without the need for stringent set up and hold requirements. Figure 17-5 shows the basic signals in a simple connection.

Figure 17-5. Basic SPI Configuration


MISO
MOSI
SCLK
SS_
MOSI
MISO
SCLK
SS_
SPI Master
SPI Slave
Data is output by both the Master and Slave, on one edge of the clock.
Data is registered at the input of both devices, on the opposite edge of the clock.
SCLK
MOSI
MISO

A device can be a master or slave. A master outputs clock and data to the ***slave device*** and inputs slave data. A slave device inputs clock and data from the ***master device*** and outputs data for input to the master. The master and slave together are essentially a circular shift register, where the master generates the clocking and initiates data transfers.

A basic data transfer occurs when the master sends 8 bits of data, along with eight clocks. In any transfer, both master and slave transmit and receive simultaneously. If the master only sends data, the received data from the slave is ignored. If the master wishes to receive data from the slave, the master must send dummy bytes to generate the clocking for the slave to send data back.

17.1.10.1 SPI Protocol Signal Definitions

The SPI protocol signal definitions are located in Table 17-3. The use of the SS_ signal varies according to the capability of the slave device.

Table 17-3. SPI Protocol Signal Definitions

Name	Function	Description
MOSI	Master Out Slave In	Master data output.
MISO	Master In Slave Out	Slave data output.
SCLK	Serial Clock	Clock generated by the master.
SS_	Slave Select (active low)	This signal is provided to enable multi-slave connections to the MISO pin. The MOSI and SCLK pins can be connected to multiple slaves, and the SS_ input selects which slave receives the input data and drives the MISO line.

17.1.11 SPI Master Function

The SPI Master (SPIM) offers SPI operating modes 0-3. By default, the MSb of the data byte is shifted out first. An additional option can be set to reverse the direction and shift the data byte out LSb first. (Refer to the timing diagrams for this function on page 203.)

When configured for SPIM, DR0 functions as a shift register, with input from the DATA input (MISO) and output to the primary output F1 (MOSI). DR1 is the TX Buffer register and DR2 is the RX Buffer register.

The SPI protocol requires data to be registered at the device input, on the opposite edge of the clock that operates the output shifter. An additional register (RXD), at the input to the DR0 shift register, has been implemented for this purpose. This register stores received data for one-half cycle, before it is clocked into the shift register.

The SPIM controls ***data transmission*** between master and slave, because it generates the bit clock for internal clocking and for clocking the SPIS. The bit clock is derived from the CLK input selection. Since the PSoC system clock generators produce clocks with varying duty cycles, the SPIM divides the input CLK by two to produce a bit clock with a 50 percent duty cycle. This clock is gated, to provide the SCLK output on the auxiliary output, during byte transmissions.

There are 4 control bits and 4 status bits in the control register that provide for PSoC device interfacing and synchronization.

The SPIM hardware has no support for driving the Slave Select (SS_) signal. The behavior and use of this signal is application and PSoC-device dependent and, if required, must be implemented in firmware.

17.1.11.1 Usability Exceptions

The following are usability exceptions for the SPI protocol function.

1. The SPIM function may not be chained.
2. The MISO input must be resynchronized at the row inputs.
3. The DR2 (Rx Buffer) register is not writeable.

17.1.11.2 Block Interrupt

The SPIM block has a selection of two interrupt sources: Interrupt on TX Reg Empty (default) or interrupt on SPI Complete. Mode bit 1 in the function register controls the selection. These modes are discussed in detail in "SPIM Timing" on page 203.

If SPI Complete is selected as the block interrupt, the control register must be read in the interrupt routine so that this status bit is cleared; otherwise, no subsequent interrupts are generated.

17.1.12 SPI Slave Function

The SPI Slave (SPIS) offers SPI operating modes 0-3. By default, the MSb of the data byte is shifted out first. An additional option can be set to reverse the direction and shift the data byte out LSb first. (Refer to the timing diagrams for this function on page 206.)

When configured for SPI, DR0 functions as a shift register, with input from the DATA input (MOSI) and output to the primary output F1 (MISO). DR1 is the TX Buffer register and DR2 is the RX Buffer register.

The SPI protocol requires data to be registered at the device input, on the opposite edge of the clock that operates the output shifter. An additional register (RXD), at the input to the DR0 shift register, is implemented for this purpose. This register stores received data for one-half cycle before it is clocked into the shift register.

The SPIS function derives all clocking from the SCLK input (typically an external SPI master). This means that the master must initiate all transmissions. For example, to read a byte from the SPIS, the master must send a byte.

There are 4 control bits and 4 status bits in the control register that provide for PSoC device interfacing and synchronization.

In the SPIS, there is an additional data input, Slave Select (SS_), which is an ***active low*** signal. SS_ must be asserted to enable the SPIS to receive and transmit. SS_ has two high level functions: 1) To allow for the selection of a given slave in a multi-slave environment, and 2) To provide additional clocking for TX data queuing in SPI modes 0 and 1.

SS_ may be controlled from an external pin through a Row Input or by way of user firmware.

When SS_ is negated, the SPIS ignores any MOSI/SCLK input from the master. In addition, the SPIS ***state machine*** is reset, and the MISO output is forced to idle at logic 1. This allows for a wired-AND connection in a multi-slave environment. Note that if High Z output is required when the slave is not selected, this behavior must be implemented in firmware with IO writes to the port drive register.

17.1.12.1 Usability Exceptions

A usability exception for the SPI slave function.

1. The SPIS function may not be chained.

17.1.12.2 Block Interrupt

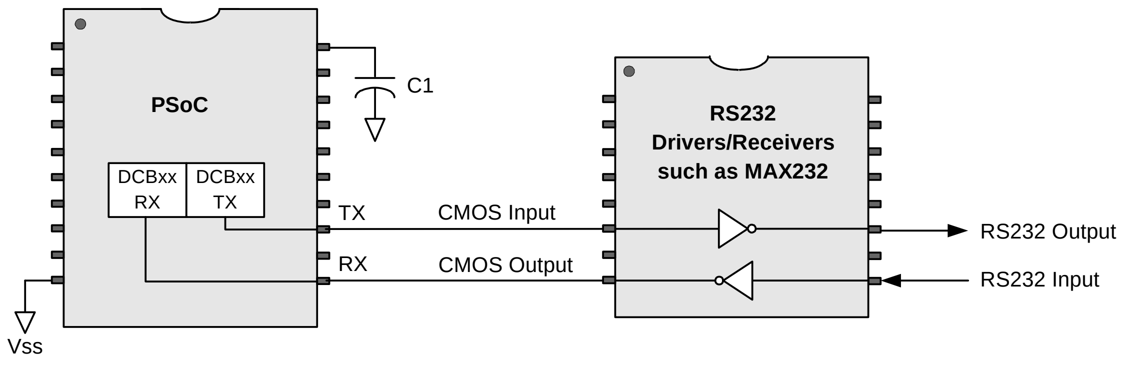
The SPIS block has a selection of two interrupt sources: Interrupt on TX Reg Empty (default) or interrupt on SPI Complete (same selection as the SPIM). Mode bit 1 in the function register controls the selection.

If SPI Complete is selected as the block interrupt, the control register must still be read in the interrupt routine so that this status bit is cleared; otherwise, no subsequent interrupts are generated.

17.1.13 Asynchronous Transmitter and Receiver Functions

The Asynchronous Transmitter and Receiver functions are illustrated in Figure 17-6.

Figure 17-6. Asynchronous Transmitter and Receiver Block Diagram



17.1.13.1 Asynchronous Transmitter Function

In the transmitter function, DR0 functions as a shift register, with no input and with the TXD serial ***data stream*** output to the primary output F1. DR1 is a TX Buffer register and DR2 is unused in this configuration. (Refer to the timing diagrams for this function on page 209.)

Unlike SPI, which has no output latency, the TXD output has one cycle of ***latency***. This is because a mux at the output must select which bits to shift out: the shift register data, framing bits, ***parity***, or mark bits. The output of this mux is registered to remove glitches. When the block is first enabled or when it is idle, a mark bit (logic 1) is output.

The ***clock generator*** is a free running divide-by-eight circuit. Although dividing the clock is not necessary for the transmitter function, the receiver function does require a divide by eight for input sampling. It is also done in the transmitter function, to allow the TX and RX functions to run off the same baud rate generator.

There are two formats supported: A 10-bit frame size including one start bit, eight data bits, and one ***stop bit*** or an 11-bit frame size including one start bit, eight data bits, one parity bit, and one stop bit.

The parity generator can be configured to output either even or odd parity on the eight data bits.

A write to the TX Buffer register (DR1) initiates a transmission and an additional byte can be buffered in this register, while transmission is in progress.

An additional feature of the transmitter function is that a clock, generated with set up and hold time for the data bits only, is output to the auxiliary output. This allows connection to a CRC generator or other digital blocks.

17.1.13.2 Usability Exceptions

The following is a usability exception for the transmitter function.

1. The transmitter function may not be chained.

17.1.13.3 Block Interrupt

The transmit block has a selection of two interrupt sources. Interrupt on TX Reg Empty (default) or interrupt on TX Complete. Mode bit 1 in the function register controls the selection.

If TX Complete is selected as the block interrupt, the control register must still be read in the interrupt routine so that this status bit is cleared; otherwise, no subsequent interrupts are generated.

17.1.13.4 Asynchronous Receiver Function

In the receiver function, DR0 functions as the serial data shift register with RXD input from the DATA input selection. DR2 is an RX Buffer register and DR1 is unused in this configuration. (Refer to the timing diagrams for this function on page 211.)

The clock generator and START detection are integrated. The clock generator is a divide by eight which, when the system is idle, is held in reset. When a START bit (logic 0) is detected on the RXD input, the reset is negated and a ***bit rate (BR)*** clock is generated, subsequently sampling the RXD input at the center of the bit time. Every subsequent START bit resynchronizes the clock generator to the incoming bit rate.

There are two formats supported: A 10-bit frame size including one start bit, eight data bits, and one stop bit or an 11-bit frame size including one start bit, eight data bits, one parity bit, and one stop bit.

The received data is an input to the parity generator. It is compared with a received parity bit, if this feature is enabled. The parity generator can be configured to output either even or odd parity on the eight data bits.

After eight bits of data are received, the byte is transferred from the DR0 shifter to the DR2 RX Buffer register.

An additional feature of the receiver function is that input data (RXD) and the synchronized clock are passed to the primary output and auxiliary output, respectively. This allows connection to a CRC generator or other digital block.

17.1.13.5 Usability Exceptions

The following are usability exceptions for the asynchronous receiver function.

1. The RXD input must be resynchronized through the row inputs.
2. DR2 is a read only register.

17.1.13.6 Block Interrupt

The receiver has one fixed interrupt source, which is the RX Reg Full status.

The RX Buffer register must always be read in the RX interrupt routine, regardless of error status, and so on, so that RX Reg Full status bit is cleared; otherwise, no subsequent interrupts are generated.

17.2 Register Definitions

The following registers are associated with the Digital Blocks and listed in address order. Note that there are two banks of registers associated with the PSoC device. Bank 0 encompasses the user registers (Data and Control registers, and Interrupt Mask registers) for the device and Bank 1 encompasses the Configuration registers for the device. Both are defined below. Refer to the "Bank 0 Registers" on page 49 and the "Bank 1 Registers" on page 123 for a quick reference of PSoC registers in address order.

Each register description that follows has an associated register table showing the bit structure for that register. The bits that are grayed out throughout this manual are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of '0'.

The Digital Block registers in this chapter are organized by function, as presented in Table 17-4. To reference timing diagrams associated with the digital block registers, see "Timing Diagrams" on page 198. For a complete table of digital block registers, refer to the "Summary Table of the Digital Registers" on page 162.

Data and Control Registers

The following table summarizes the Data and Control registers, by function type, for the digital blocks.

Table 17-4. Digital Block Data and Control Register Definitions

Function Type	DR0		DR1		DR2		CR0	
	Function	Access	Function	Access	Function	Access	Function	Access
Timer	Down Counter	R*	Period	W	Capture/Compare	RW	Control	RW
Counter	Down Counter	R*	Period	W	Compare	RW	Control	RW
Dead Band	Down Counter	R*	Period	W	N/A	N/A	Control	RW
CRCPRS	LFSR	R*	Polynomial	W	Seed	RW	Control	RW
SPIM	Shifter	N/A	TX Buffer	W	RX Buffer	R	Control/Status	RW**
SPIS	Shifter	N/A	TX Buffer	W	RX Buffer	R	Control/Status	RW**
TXUART	Shifter	N/A	TX Buffer	W	N/A	N/A	Control/Status	RW**
RXUART	Shifter	N/A	N/A	N/A	RX Buffer	R	Control/Status	RW**

LEGEND

* In Timer, Counter, Dead Band, and CRCPRS functions, a read of the DR0 register returns 00h and transfers DR0 to DR2.

** In the Communications functions, control bits are read/write accessible and status bits are read only accessible.

17.2.1 DxBxxDRx Registers

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	DxBxxDR0	Data[7:0]								# : 00
0,xxh	DxBxxDR1	Data[7:0]								W : 00
0,xxh	DxBxxDR2	Data[7:0]								# : 00

LEGEND

#	Access is bit specific. Refer to the register detail for additional information.
xx	An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 162.

The DxBxxDRx Registers are the digital blocks’ Data registers.

Bits 7 to 0: Data[7:0]. The Data registers and bits presented in this section encompass the DxBxxDR0, DxBxxDR1, and DxBxxDR2 registers. They are discussed according to which bank they are located in and then detailed in the tables that follow by function type.

For additional information, refer to the Register Details chapter for the following registers:

- DxBxxDR0 register on page 53.
- DxBxxDR1 register on page 54.
- DxBxxDR2 register on page 55.

17.2.1.1 Timer Register Definitions

There are three 8-bit Data registers and a 3-bit Control register. Table 17-5 explains the meaning of these registers in the context of timer operation. The Control register is described in section 17.2.2 DxBxxCR0 Register.

Note DR2 is not writeable when the timer is enabled.

Table 17-5. Timer Data Register Descriptions

Name	Function	Description
DR0	Count Value	Not directly readable or writeable. During normal operation, DR0 stores the current count of a synchronous down counter. When disabled, a write to the DR1 period register is also simultaneously loaded into DR0 from the data bus. When disabled, a read of DR0 returns 00h to the data bus and transfers the contents of DR0 to DR2. This transfer only occurs in the addressed block. When enabled, a read of DR0 returns 00h to the data bus and synchronously transfers the contents of DR0 to DR2. It operates simultaneously on the byte addressed and all higher bytes in a multi-block timer. Note that when the hardware capture input is high, the read of DR0 (software capture) is masked and does not occur. The hardware capture input must be low for a software capture to occur.
DR1	Period	Write only register. Data in this register sets the period of the count. The actual number of clocks counted is Period + 1. In the default one-half cycle terminal count mode (TC), a period value of 00h results in the primary output to be the inversion of the input clock. In the optional full cycle TC mode, a period of 00h gives a constant logic high on the primary output. When disabled, a write to this register also transfers the period value directly into DR0. When enabled, if the block frequency is 24 MHz or below, this register may be written to at any time, but the period is only reloaded into DR0 in the clock following a TC. If the block frequency is 48 MHz, the terminal count or compare interrupt should be used to synchronize the new period register write; otherwise, the counter could be incorrectly loaded.
DR2	Capture/ Compare	Read write register (see **Exception** below). DR2 has multiple functions in a timer configuration. It is typically used as a capture register, but also functions as a compare register. When enabled and a capture event occurs, the current count in DR0 is synchronously transferred into DR2. When enabled, the compare output is computed using the compare type (set in the function register mode bits) between DR0 and DR2. The result of the compare is output to the auxiliary output. When disabled, a read of DR0 transfers the contents of DR0 into DR2 for the addressed block only. **Exception**: When enabled, DR2 is not writeable.

17.2.1.2 Counter Register Definitions

There are three 8-bit Data registers and a 2-bit Control register. Table 17-6 explains the meaning of these registers in the context of the counter operation. Note that the descriptions of the registers are dependent on the enable/disable state of the block. This behavior is only related to the enable bit in the Control register, not the data input that provides the counter gate (unless otherwise noted).

Note DR0 may only be read (to transfer DR0 data to DR2) when the block is disabled.

Table 17-6. Counter Data Register Descriptions

Name	Function	Description
DR0	Count Value	Not directly readable or writeable. During normal operation, DR0 stores the current count of a synchronous down counter. When disabled, a write to the DR1 period register is also simultaneously loaded into DR0 from the data bus. When disabled or the data input (counter gate) is low, a read of DR0 returns 00h to the data bus and transfers the contents of DR0 to DR2. This register should not be read when the counter is enabled and counting.
DR1	Period	Write only register. Data in this register sets the period of the count. The actual number of clocks counted is Period + 1. A period of 00h gives a constant logic high on the auxiliary output. When disabled, a write to this register also transfers the period value directly into DR0. When enabled, if the block frequency is 24 MHz or below, this register may be written to at any time, but the period is only reloaded into DR0 in the clock following a TC. If the block frequency is 48 MHz, the terminal count or compare interrupt should be used to synchronize the new period register write; otherwise, the counter could be incorrectly loaded.
DR2	Compare	Read write register. DR2 functions as a compare register. When enabled, the compare output is computed using the compare type (set in the function register mode bits) between DR0 and DR2. The result of the compare is output to the primary output. When disabled or the data input (counter gate) is low, a read of DR0 transfers the contents of DR0 into DR2. DR2 may be written to when the function is enabled or disabled.

17.2.1.3 Dead Band Register Definitions

There are three 8-bit Data registers and a 3-bit Control register. Table 17-7 explains the meaning of these registers in the context of dead band operation.

Note DR0 may only be read (to transfer DR0 data to DR2) when the block is disabled.

Table 17-7. Dead Band Register Descriptions

Name	Function	Description
DR0	Count Value	Not directly readable or writeable. During normal operation, DR0 stores the current count of a synchronous down counter. When disabled, a write to the DR1 period register is also simultaneously loaded into DR0 from the data bus. When disabled, a read of DR0 returns 00h to the data bus and transfers the contents of DR0 to DR2.
DR1	Period	Write only register. Data in this register sets the period of the dead band count. The actual number of clocks counted is Period + 1. The minimum period value is 00h, which sets a dead band time of one clock. When disabled, a write to this register also transfers the period value directly into DR0. When enabled, if the block frequency is 24 MHz or below, this register may be written to at any time, but the period is only reloaded into DR0 in the clock following a terminal count (TC). If the block frequency is 48 MHz, the terminal count or compare interrupt should be used to synchronize the new period register write; otherwise, the counter could be incorrectly loaded.
DR2	Buffer	When disabled, a read of DR0 transfers the contents of DR0 into DR2.

17.2.1.4 CRCPRS Register Definitions

There are three 8-bit Data registers and one 2-bit Control register. Table 17-8 explains the meaning of these registers in the context of CRCPRS operation. Note that in the CRCPRS function, a write to the DR2 Seed register is also loaded simultaneously into DR0.

Table 17-8. CRCPRS Register Descriptions

Name	Function	Description
DR0	LFSR	Not directly readable or writeable. During normal operation, DR0 stores the state of a synchronous linear feedback shift register. When disabled, a write to the DR2 seed register is also simultaneously loaded into DR0 from the data bus. When disabled, a read of DR0 returns 00h to the data bus and transfers the contents of DR0 to DR2. This register should not be read while the block is enabled.
DR1	Polynomial	Write only register. Data in this register sets the polynomial for the CRC or PRS function. **Exception**: This register must only be written when the block is disabled.
DR2	Seed/Residue	Read write register. DR2 functions as a seed and residue register. When disabled, a write to this register also transfers the seed value directly into DR0. When enabled, DR2 may be written to at any time. The value written is used in the compare function. When enabled, the compare output is computed using the compare type (set in the function register mode bits) between DR0 and DR2. The result of the compare is output to the auxiliary output. When disabled, a read of DR0 transfers the contents of DR0 into DR2. This feature can be used to read out the residue, after a CRC operation is complete.

17.2.1.5 SPI Master Register Definitions

There are three 8-bit Data registers and one 8-bit Control/Status register. Table 17-9 explains the meaning of these registers in the context of SPIM operation.

Table 17-9. SPIM Data Register Descriptions

Name	Function	Description
DR0	Shifter	Not readable or writeable. During normal operation, DR0 implements a shift register for shifting serial data.
DR1	TX Buffer	Write only register. If no transmission is in progress and this register is written to, the data from this register (DR1) is loaded into the shift register (DR0), on the following clock edge, and a transmission is initiated. If a transmission is currently in progress, this register serves as a buffer for TX data. This register should only be written to when TX Reg Empty status is set and the write clears the TX Reg Empty status bit in the control register. When the data is transferred from this register (DR1) to the shift register (DR0), then TX Reg Empty status is set.
DR2	RX Buffer	Read only register. When a byte transmission/reception is complete, the data in the shifter (DR0) is transferred into the RX Buffer register and RX Reg Full status in the control register is set. A read from this register (DR2) clears the RX Reg Full status bit in the control register.

17.2.1.6 SPI Slave Register Definitions

There are three 8-bit Data registers and one 8-bit Control/Status register. Table 17-10 explains the meaning of these registers in the context of SPIS operation.

Table 17-10. SPIS Data Register Descriptions

Name	Function	Description
DR0	Shifter	Not readable or writeable. During normal operation, DR0 implements a shift register for shifting serial data.
DR1	TX Buffer	Write only register. This register should only be written to when TX Reg Empty status is set and the write clears the TX Reg Empty status bit in the control register. When the data is transferred from this register (DR1) to the shift register (DR0), then TX Reg Empty status is set.
DR2	RX Buffer	Read only register. When a byte transmission/reception is complete, the data in the shifter (DR0) is transferred into the RX Buffer register and RX Reg Full status in the control (CR0) register is set. A read from this register (DR2) clears the RX Reg Full status bit in the control register.

17.2.1.7 Transmitter Register Definitions

There are three 8-bit Data registers and one 5-bit Control/Status register. Table 17-11 explains the meaning of these registers in the context of transmitter operation.

Table 17-11. Transmitter Data Register Descriptions

Name	Function	Description
DR0	Shifter	Not readable or writeable. During normal operation, DR0 implements a shift register for shifting out serial data.
DR1	TX Buffer	Write only register. If no transmission is in progress and this register is written to, subject to the set up time requirement, the data from this register (DR1) is loaded into the shift register (DR0) on the following clock edge and a transmission is initiated. If a transmission is currently in progress, this register serves as a buffer for TX data. This register should only be written to when TX Reg Empty status is set and the write clears the TX Reg Empty status bit in the control (CR0) register. When the data is transferred from this register (DR1) to the shift register (DR0), then TX Reg Empty status is set.
DR2	NA	Not used in this function.

17.2.1.8 Receiver Register Definitions

There are three 8-bit Data registers and one 8-bit Control/Status register. Table 17-12 explains the meaning of these registers in the context of receiver operation.

Table 17-12. Receiver Data Register Descriptions

Name	Function	Description
DR0	Shifter	Not readable or writeable. During normal operation, DR0 implements a shift register for shifting in serial data from the RXD input.
DR1	NA	Not used in this function.
DR2	RX Buffer	Read only register. After 8 bits of data are received, the contents of the shifter (DR0) is transferred into the RX Buffer register and the RX Reg Full status is set. The RX Reg Full status bit in the control register is cleared when this register is read.

17.2.2 DxBxxCR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	DxBxxCR0	Function Control/Status bits for selected function[6:0]							Enable	# : 00

LEGEND

#	Access is bit specific. Refer to the register detail for additional information.
xx	An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 162.

The DxBxxCR0 Registers are the digital blocks’ Control registers.

Bits 7 to 1: Function Control/Status[6:0]. The bits for this register are described by function in Table 17-13. Refer to the “Summary Table of the Digital Registers” on page 162 for a complete description of bit functionality.

Bit 0: Enable. This bit is used to synchronously enable or disable the programmed function.

For additional information, refer to the DxBxxCR0 (Timer Control) register on page 56.

Table 17-13. DxBxxCR0 Control Register Descriptions

Function	Description
Timer	There are three bits in the control (CR0) register: one for enabling the block, one for setting the optional interrupt on capture, and one to select between one-half and a full clock for terminal count (TC) output.
Counter	One bit enable only.
Dead Band	There are three bits in the control (CR0) register: one bit for enabling the block, and two bits to enable and control Dead Band Bit-Bang mode. When Bit-Bang mode is enabled, the output of this register is substituted for the PWM reference. This register may be toggled by user firmware to generate PHI1 and PHI2 output clock with the programmed dead time. The options for Bit-Bang mode are as follows: 0 Function uses the previous clock primary output as the input reference. 1 Function uses the bit bang clock register as the input reference.
CRCPRS	There are two bits are used to enable operation.
SPIM	The SPI control (CR0) register contains both control and status bits. There are four control bits that are read/write: Enable, Clock Phase and Clock Polarity to set the mode, and LSb First, which controls bit ordering. There are two read only status bits: Overrun and SPI Complete. There are two additional read only status bits to indicate TX and RX buffer status.
SPIS	The SPI control (CR0) register contains both control and status bits. There are four control bits that are read/write: Enable, Clock Phase and Clock Polarity to set the mode, and LSb First, which controls bit ordering. There are two read only status bits: Overrun and SPI Complete. There are two additional read only status bits to indicate TX and RX buffer status.
TXUART	The transmitter control (CR0) register contains three control bits and two status bits. The control bits are Enable, Parity Enable, and Parity Type, and have read/write access. The status bits, TX Reg Empty and TX Complete, are read only.
RXUART	The receiver control (CR0) register contains both control and status bits. The three control bits are read/write: Enable, Parity Enable, and Parity Type. There are five read only status bits: RX Reg Full, RX Active, Framing Error, Overrun, and Parity Error.

Interrupt Mask Register

The following register is the interrupt mask register for the digital blocks.

17.2.3 INT_MSK1 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E1h	INT_MSK1					DCB03	DCB02	DBB01	DBB00	RW : 00

The Interrupt Mask Register 1 (INT_MSK1) is used to enable the individual sources' ability to create pending interrupts for digital blocks.

Bit 3: DCB03. Digital communications block interrupt enable for row 0 block 3.

Bit 2: DCB02. Digital communications block interrupt enable for row 0 block 2.

Bit 1: DBB01. Digital basic block interrupt enable for row 0 block 1.

Bit 0: DBB00. Digital basic block interrupt enable for row 0 block 0.

For additional information, refer to the INT_MSK1 register on page 105.

Configuration Registers

The configuration block contains three registers: Function (DxBxxFN), Input (DxBxxIN), and Output (DxBxxOU). The values in these registers should not be changed while the block is enabled. Note that the Digital Block Configuration registers are all located in bank 1 of the PSoC device's memory map.

17.2.4 DxBxxFN Registers

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	DxBxxFN	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]			RW : 00

LEGEND

xx An "x" after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the "Digital Register Summary" on page 162.

The Digital Basic/Communications Type B Block Function Registers (DxBxxFN) contain the primary Mode and Function bits that determine the function of the block.

All bits in these registers are common to all functions, except those specified in Table 17-15.

Bit 7: Data Invert. This bit inverts the selected data input.

Bit 6: BCEN. This bit enables the primary output of the block, to drive the row broadcast block. The BCEN bit is set independently in each block; and therefore, care must be taken to ensure that only one BCEN bit, in a given row, is enabled. However, if any of the blocks in a given row have the BCEN bit set, the input that allows the broadcast net from other rows to drive the given row's broadcast net is disabled (see Figure 16-2 on page 172).

Bit 5: End Single. This bit is used to indicate the last or most significant block in a chainable function. This bit must also be set if the chainable function only consists of a single block.

Bits 4 and 3: Mode[1:0]. The mode bits select the options available for the selected function. These bits should only be changed when the block is disabled.

Bits 2 to 0: Function[2:0]. The function bits configure the block into one of the available block functions (six for the communications block, four for the basic block).

For additional information, refer to the DxBxxFN register on page 127.

Table 17-14. DxBxxFN Function Registers

[7]: Data Invert	1 == Invert block's data input 0 == Do not invert block's data input
[6]: BCEN	1 == Enable 0 == Disable
[5]: End Single	1 == Block is not chained or is at the end of a chain 0 == Block is at the start of or in the middle of a chain
[4:3]: Mode	Function specific
[2:0]: Function	000b: Timer 001b: Counter 010b: CRCPRS 011b: Reserved 100b: Dead band for PWM 101b: UART (DCBxx blocks only) 110b: SPI (DCBxx blocks only) 111b: Reserved

Table 17-15. Digital Block Configuration Register Functional Descriptions

Function	Description
Timer	The mode bits in the timer block control the interrupt type and the compare type.
Counter	The mode bits in the counter block control the Interrupt type and the compare type (same as the timer function).
Dead Band	The mode bits are encoded as the kill type. See the table titled "Dead Band KILL Options" on page 181 for an explanation of kill options.
CRCPRS	The mode bits are encoded to determine the compare type.
SPIM	Mode bit 1 selects interrupt type. Mode bit 0 selects master or slave (for SPIM, it is '0').
SPIS	Mode bit 1 selects interrupt type. Mode bit 0 selects master or slave (for SPIS, it is '1').
TXUART	Mode bit 0 selects between transmitter and receiver (in this case Mode bit 0 is set to '1' for TX) and Mode bit 1 selects the interrupt type.
RXUART	Mode bit 0 selects between transmitter and receiver (in this case Mode bit 0 is set to '0' for RX) and Mode bit 1 selects the interrupt type.

17.2.5 DxBxxIN Registers

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	DxBxxIN	Data Input[3:0]				Clock Input[3:0]				RW : 00

LEGEND

xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 162.

The Digital Basic/Communications Type B Block Input Registers (DxBxxIN) are used to select the data and clock inputs.

These registers are common to all functional types, except the SPIS. The SPIS is unique in that it has three function inputs and one function output defined. Refer to the DxBxxOU registers.

The input registers are eight bits and consist of two 4-bit fields to control each of the 16-to-1 clock and data input muxes. The meaning of these fields depends on the external clock and data connections, which is context specific. See Table 17-16.

Bits 7 to 4:Data Input[3:0]. These bits control the data input.

Bits 3 to 0: Clock Input[3:0]. These bits control the clock input.

Table 17-16. Digital Block Input Definitions

Function	Inputs		
	DATA	CLK	Auxiliary
Timer	Capture	CLK	N/A
Counter	Enable	CLK	N/A
Dead Band	Kill	CLK	Reference *
CRCPRS	Serial Data **	CLK	N/A
SPIM	MISO	CLK	N/A
SPIS	MOSI	SCLK	SS_
Transmitter	N/A	8X Baud CLK	N/A
Receiver	RXD	8X Baud CLK	N/A

* The dead band reference input does not use the auxiliary input mux. It is hardwired to be the primary output of the previous block.

** For CRC computation, the input data is a serial data stream synchronized to the clock. For PRS mode, this input should be forced to logic 0.

For additional information, refer to the DxBxxIN register on page 129.

17.2.6 DxBxxOU Registers

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	DxBxxOU	AUXCLK		AUXEN	AUX IO Select[1:0]		OUTEN	Output Select[1:0]		RW : 00

LEGEND
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 162.

The Digital Basic/Communications Type B Block Output Registers (DxBxxOU) are used to control the connection of digital block outputs to the available row interconnect and control clock resynchronization.

When the selected function is SPI Slave (SPIS), the AUXEN and AUX IO bits change meaning, and select the input source and control for the Slave Select (SS_) signal.

The Digital Block Output register is common to all functional types, except the SPIS. The SPIS function is unique in that it has three function inputs and one function output defined. When the Aux IO Enable bit is '0', the Aux IO Select bits are used to select one of four inputs from the auxiliary data input mux to drive the SS_ input. Alternatively, when the Aux IO Enable bit is a '1', the SS_ signal is driven directly from the value of the Aux IO Select[0] bit. Thus, the SS_ input can be controlled in firmware, eliminating the need to use an additional GPIO pin for this purpose. Regardless of how the SS_ bit is configured, an SPIS block has the auxiliary row output drivers forced off; and therefore, the auxiliary output is not available in this block.

The following table enumerates the primary and auxiliary outputs that are defined for a given block function. Most functions have two outputs defined (the exception is the SPI slave, which has only one). One or both of these outputs can optionally be enabled for output. When output, these signals can be routed to other block inputs through row or global interconnect, or output to chip pins.

Table 17-17. Digital Block Output Definitions

Function	Outputs		
	Primary	Auxiliary	Interrupt
Timer	Terminal Count	Compare	Terminal Count or Compare
Counter	Compare	Terminal Count	Terminal Count or Compare
Dead Band	Phase 1	Phase 2	Phase 1
CRCPRS	MSB	Compare	Compare
SPIM	MOSI	SCLK	TX Reg Empty or SPI Complete
SPIS	MISO	N/A **	TX Reg Empty or SPI Complete
Transmitter	TXD	SCLK *	TX Reg Empty or TX Compete
Receiver	RXD	SCLK *	RX Reg Full

* The UART blocks generate an SPI mode 3 style clock that is only active during the data bits of a received or transmitted byte.

** In the SPIS, the field that is used to select the auxiliary output is used to control the auxiliary input to select the SS_.

Bits 7 and 6: AUXCLK. All digital block clock inputs must be resynchronized. The digital blocks have numerous selections for clocking. In addition to the system clocks, including VC1, VC2, and VC3, clocks generated by other digital blocks may be selected through row or global interconnect. To maintain the integrity of block timing, all clocks are resynchronized at the input to the digital block.

The two AUXCLK bits are used to enable the input clock resynchronization. When enabled, the input clock is resynchronized to the selected system clock, which occurs after the 16-to-1 multiplexing. The rules for selecting the value for this register are as follows:

- If the input clock is based on SYSCLK (for example, VC1, VC2, VC3 based on SYSCLK) or the output of other blocks whose clock source is based on SYSCLK, sync to SYSCLK.
- If the input clock is based on SYSCLKX2 (for example, VC3 based on SYSCLKX2) or another digital block clocked by SYSCLKX2 or a SYSCLKX2-based clock, sync to SYSCLKX2.
- If you want to clock the block at 24 MHz (SYSCLK), choose SYSCLK direct in the resynchronized bits (the 16-to-1 input clock selection is ignored).
- If you want to clock the block at 48 MHz (SYSCLKX2), choose SYSCLKX2 as the clock input selection and leave the resynchronized bits in bypass mode.

The following table summarizes the available selections of the AUXCLK bits.

Table 17-18. AUXCLK Bit Selections

Code	Description	Usage
00	Bypass	Use this selection only when SYSCLKX2 (48 MHz) is selected by the 16-to-1 clock multiplexer (see the DxBxxIN register).
01	Resynchronize to SYSCLK (24 MHz)	This is a typical selection. Use this setting for any SYSCLK-based clock: VC1, VC2, VC3 driven by SYSCLK, digital blocks with SYSCLK-based source clocks, broadcast bus with source based on SYSCLK, row input and row outputs with source based on SYSCLK.
10	Resynchronize to SYSCLKX2 (48 MHz)	Use this setting for any SYSCLKX2-based clock: VC3 driven by SYSCLKX2, digital blocks with SYSCLKX2-based source clocks, broadcast bus with source based on SYSCLKX2, row input and row outputs with source based on SYSCLKX2.
11	SYSCLK Direct	Use this setting to clock the block directly using SYSCLK. Note that this setting is not strictly related to clock resynchronization, since SYSCLK cannot resynchronize itself, it allows a direct skew controlled SYSCLK source.

Note Selecting VC1/1 or VC2/1 (when VC1 is 1), or VC3/1 when the input is SYSCLK, or SYSCLKX2 is not allowed.

Bit 5: AUXEN. The AUXEN bit enables the auxiliary output to be driven onto the selected row output. If the selected function is SPI slave, the meaning of this bit is different. The SPI slave does not have a defined auxiliary output, so this bit is used, in conjunction with the AUX IO Select bits to control the Slave Select input signal (SS_). When this bit is set, the SS_ input is forced active; and therefore, ***routing*** SS_ from an input pin is unnecessary.

Bits 4 and 3: AUX IO Select[1:0]. These two bits select one (out of the 4) row outputs to drive the auxiliary output onto. In SPI slave mode, these bits are used in conjunction with the AUXEN bit to control the Slave Select (SS_) signal.

In this mode, these two bits are used to select one of four row inputs for use as SS_. If no SS_ is required in a given application, the AUXEN bit can be used to force the SS_ input active; and therefore, routing SS_ in through a row input is not required.

Bit 2: OUTEN. This bit enables the primary output to be driven onto the selected row output.

Output Select[1:0]. These two bits indicate which of the four row outputs the primary output is driven onto.

For additional information, refer to the DxBxxOU register on page 131.

17.3 Timing Diagrams

The timing diagrams in this section are presented according to their functionality and are in the following order.

17.3.1 Timer Timing

Enable/Disable Operation. When the block is disabled, the clock is immediately gated low. All outputs are gated low, including the interrupt output. All internal states are reset to their configuration-specific reset state, except for DR0, DR1, and DR2 which are unaffected.

Terminal Count/Compare Operation. In the clock cycle following the count of 00h, the terminal count (TC) output is asserted. It is one-half cycle or a full cycle depending on the TC Pulse Width mode, as set in the block control register. If this block stands alone or is the least significant block in a chain, the Carry Out (CO) signal is also asserted. If the period is set to 00h and the TC Pulse Width mode is one-half cycle, the output is the inversion of the input clock.

The Compare (CMP) output is asserted in the cycle following the compare true and negated one cycle after compare false.

Multi-Block Terminal Count/Compare Operation. When timers are chained, the CO signal of a given block becomes the Carry In (CI) of the next most significant block in the chain. In a chained timer, the CO output indicates that block and all lower blocks are at 00h count. The CO is set up to the next positive edge of the clock, to enable the next higher block to count once for every terminal count (TC) of all lower blocks.

The terminal count out of a given block becomes the terminal count in of the next least significant block in the chain. The terminal count output indicates that the block and all higher blocks are at 00h count. The terminal count in/terminal count out chaining signals provide a way for the lower blocks to know when the upper blocks are at TC. Reload occurs when all blocks are at TC, which can be determined by CI, terminal count in, and the block zero detect. Example timing for a three block timer is shown in Figure 17-7.

The compare circuit compares registers DR0 <= DR2. (When Mode[1] = 1, the comparison is DR0 < DR2.)

Each block has an internal compare condition (DR0 compared to DR2), a chaining signal to the next block called CMPO, and the chaining signal from the previous block called CMPI. In any given block of a timer, the CMPO is used to generate the auxiliary output (primary output in the counter) with a one cycle clock delay.

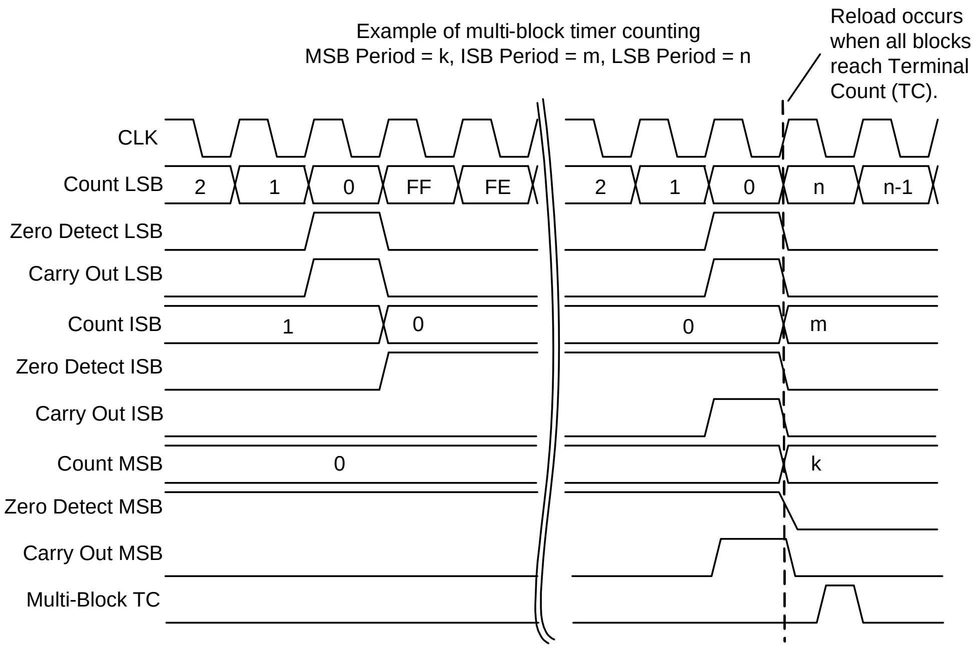
CMPO is generated from a combination of the internal compare condition and the CMPI input using the following rules:

1. For any given block, if DR0 < DR2, the CMPO condition is unconditionally asserted.
2. For any given block, if DR0 == DR2, CMPO is asserted only if the CMPI input to that block is asserted.
3. If the block is a start block, the effective CMPI depends on the compare type. If it is DR0 <= DR2, the effective CMPI input is '1'. If it is DR0 < DR2, the effective input is '0'.

Capture Operation. In the timer implementation, a rising edge of the data input or a CPU read of DR0 triggers a synchronous capture event. The result of this is to generate a latch enable to DR2 that loads the current count from DR0 into DR2. The latch enable signal is synchronized in such a way that it is not closing near an edge on which the count is changing.

A limitation is that capture does not work with the block clock of 48 MHz. (A fundamental limitation to timer capture operation is the fact the GPIO inputs are currently synchronized to the 24 MHz system clock).

Figure 17-7. Multi-Block Timing



17.3.2 Counter Timing

Enable/Disable Operation. See Timer Enable/Disable Operation ("Timer Function" on page 179).

Terminal Count/Compare Operation. See Timer Terminal Count/Compare Operation ("Timer Function" on page 179).

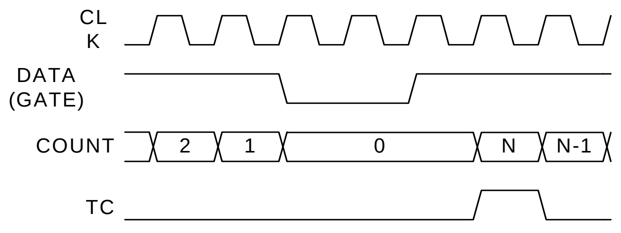
Multi-Block Operation. See Timer Multi-Block Terminal Count/Compare Operation ("Timer Function" on page 179).

Gate (Enable) Operation. The data input controls the counter enable. The transition on this enable must have at least one 24 MHz cycle of set up time to the block clock. This is ensured if internal or synchronized external inputs are used.

As shown in Figure 17-8, when the data input is negated (counting is disabled) and the count is 00h, the TC output stays low. When the data input goes high again, the TC occurs on the following input clock. When the block is disabled, the clock is immediately gated low.

All internal states are reset, except for DR0, DR1, and DR2, which are unaffected.

Figure 17-8. Counter Terminal Count Timing with Gate Disable



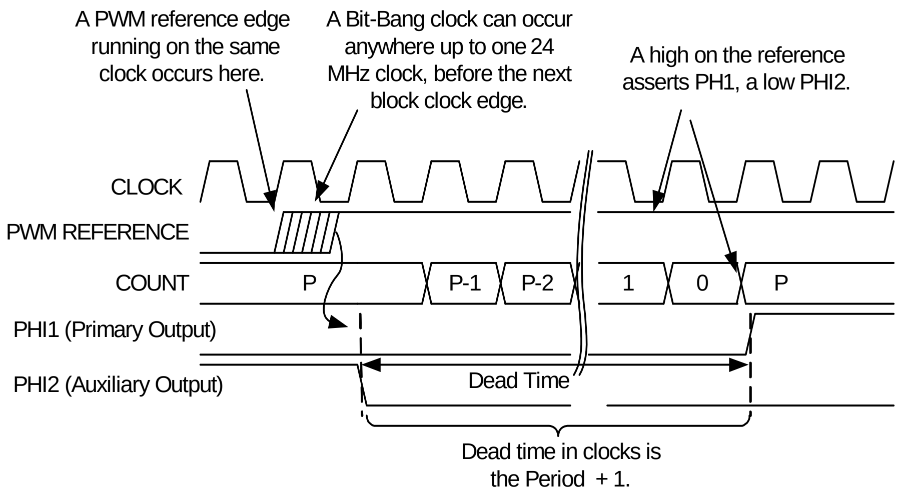
17.3.3 Dead Band Timing

Enable/Disable Operation. Initially both outputs are low. There are no critical timing requirements for enabling the block because dead band processing does not start until the first incoming positive or negative reference edge. In typical operation, it is recommended that the dead band block be enabled first, then the Pulse Width Modulator (PWM) generator block.

When the block is disabled, the clock is immediately gated low. All outputs are gated low, including the interrupt output. All internal states are reset to their configuration-specific reset state, except for DR0, DR1, and DR2 which are unaffected.

Normal Operation. Figure 17-9 shows typical dead band timing. The incoming reference edge can occur up to one 24 MHz system clock before the edge of the block clock. On the edge of the block clock, the currently asserted output is negated and the dead band counter is enabled. After Period + 1 clocks, the phase associated with the current state of the PWM reference is asserted (Reference High = Phase 1, Reference Low = Phase 2). The minimum dead time occurs with a period value of 00h and that dead time is one clock cycle.

Figure 17-9. Basic Dead Band Timing


A PWM reference edge running on the same clock occurs here.
A Bit-Bang clock can occur anywhere up to one 24 MHz clock, before the next block clock edge.
A high on the reference asserts PH1, a low PHI2.
CLOCK
PWM REFERENCE
COUNT
P
P-1
P-2
1
0
P
PHI1 (Primary Output)
PHI2 (Auxiliary Output)
Dead Time
Dead time in clocks is the Period + 1.

17.3.3.1 *Changing the PWM Duty Cycle*

Under normal circumstances, the dead band period is less than the minimum PWM high or low time. As an example, consider Figure 17-10 where the low of the PWM is four clocks, the dead band period is two clocks, and the high time of the PHI2 is two clocks.

Figure 17-10. DB High Time is PWM Width Minus DB Period

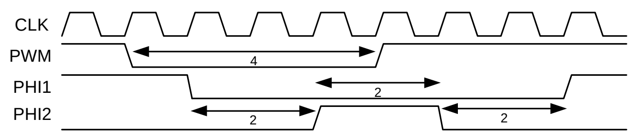
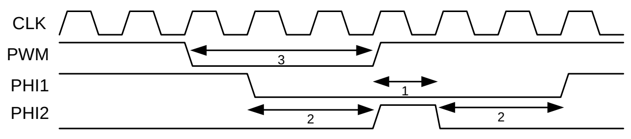


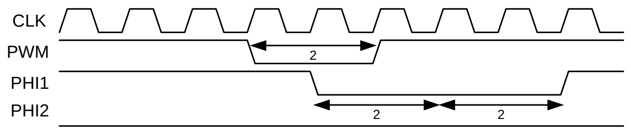
Figure 17-11 illustrates the reduction of the width of the PWM low time by one clock (to three clocks). The dead band period remains the same, but the high time for PHI2 is reduced by one clock (to one clock). Of course the opposite phase, PHI1, increases in length by one clock.

Figure 17-11. DB High Time is Reduced as PWM Width is Reduced



If the width of the PWM low time is reduced to a point where it is equal to the dead band period, the corresponding phase, PHI2, disappears altogether. Note that after the rising edge of the PWM, the opposite phase still has the programmed dead band. Figure 17-12 shows an example where the dead band period is two and the PWM width is two. In this case, the high time of PHI2 is zero clocks. Note that the phase 1 dead band time is still two clocks.

Figure 17-12. PWM Width Equal to Dead Band Period



In the case where the dead band period is greater than the high or low of the PWM reference, the output of the associated phase is not asserted high.

17.3.3.2 Kill Operation

It is assumed that the KILL input is not synchronized at the row input. (This is not a requirement; however, if synchronized, the KILL operation has up to two 24 MHz clock cycles latency, which is undesirable.) To support the restart modes, the negation of KILL is internally (in the block) synchronized to the 24 MHz system clock.

There are three KILL modes supported. In all cases, the KILL signal asynchronously forces the outputs to logic 0. The differences in the modes come from how dead band processing is restarted.

1. **Synchronous Restart Mode**: When KILL is asserted high, the internal state is held in reset and the initial dead band period is reloaded into the counter. While KILL is held high, incoming PWM reference edges are ignored. When KILL is negated, the next incoming PWM reference edge restarts dead band processing. See Figure 17-13.
2. **Asynchronous Restart Mode**: When KILL is asserted high, the internal state is not affected. When KILL is negated, the outputs are restored, subject to a minimum disable time between one-half and one and one-half clock cycle. See Figure 17-14.
3. **Disable Mode**: There is no specific timing associated with this mode. The block is disabled and the user must re-enable the function in firmware to continue processing.

Figure 17-13. Synchronous Restart KILL Mode

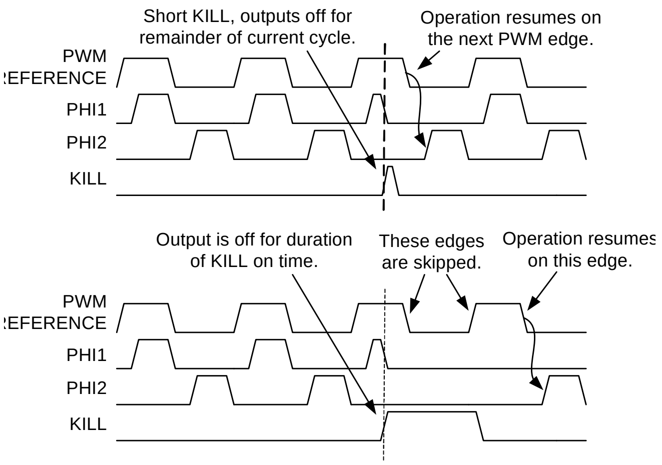
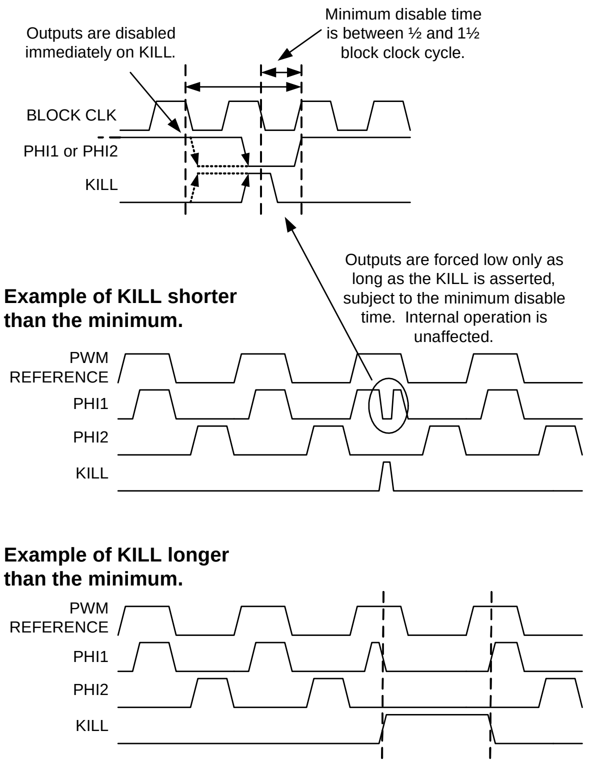

Short KILL, outputs off for remainder of current cycle.
Operation resumes on the next PWM edge.
PWM REFERENCE
PHI1
PHI2
KILL
Output is off for duration of KILL on time.
These edges are skipped.
Operation resumes on this edge.
PWM REFERENCE
PHI1
PHI2
KILL

Figure 17-14. Asynchronous Restart Kill Mode


Outputs are disabled immediately on KILL.
Minimum disable time is between ½ and 1½ block clock cycle.
BLOCK CLK
PHI1 or PHI2
KILL
Example of KILL shorter than the minimum.
Outputs are forced low only as long as the KILL is asserted, subject to the minimum disable time. Internal operation is unaffected.
PWM REFERENCE
PHI1
PHI2
KILL
Example of KILL longer than the minimum.
PWM REFERENCE
PHI1
PHI2
KILL

17.3.4 CRCPRS Timing

Enable/Disable Operation. Same as Timer Enable/Disable Operation ("Timer Timing" on page 198)

When the block is disabled, the clock is immediately gated low. All outputs are gated low, including the interrupt output. All internal states are reset to their configuration-specific reset state, except for DR0, DR1, and DR2, which are unaffected.

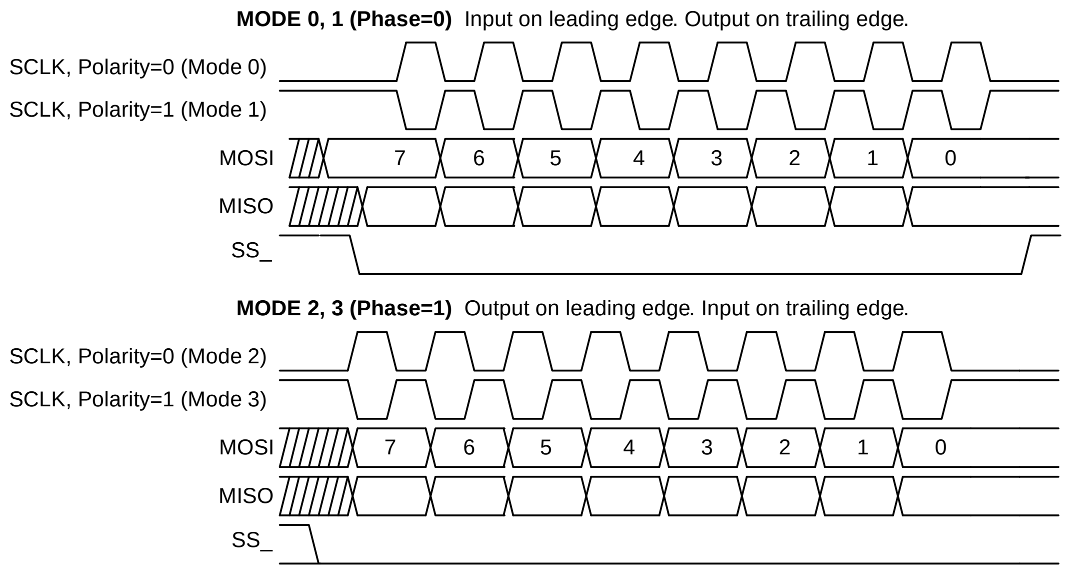
17.3.5 SPI Mode Timing

Figure 17-15 shows the SPI modes, which are typically defined as 0,1, 2, or 3. These mode numbers are an encoding of two control bits: Clock Phase and Clock Polarity.

Clock phase indicates the relationship of the clock to the data. When the clock phase is '0', it means that the data is registered as an input on the leading edge of the clock and the next data is output on the trailing edge of the clock. When the clock phase is '1', it means that the next data is output on the leading edge of the clock and that data is registered as an input on the trailing edge of the clock.

Clock polarity controls clock inversion. When clock polarity is set to '1', the clock ***idle state*** is high.

Figure 17-15. SPI Mode Timing


MODE 0, 1 (Phase=0) Input on leading edge. Output on trailing edge.
SCLK, Polarity=0 (Mode 0)
SCLK, Polarity=1 (Mode 1)
MOSI
7
6
5
4
3
2
1
0
MISO
SS_
MODE 2, 3 (Phase=1) Output on leading edge. Input on trailing edge.
SCLK, Polarity=0 (Mode 2)
SCLK, Polarity=1 (Mode 3)
MOSI
7
6
5
4
3
2
1
0
MISO
SS_

17.3.6 SPIM Timing

Enable/Disable Operation. As soon as the block is configured for SPIM, the primary output is the MSb or LSb of the shift register, depending on the LSb First configuration in bit 7 of the control register. The auxiliary output is '1' or '0' depending on the idle clock state of the SPI mode. This is the idle state.

When the SPIM is enabled, the internal reset is released on the divide-by-2 flip-flop and on the next positive edge of the selected input clock. This 1-bit divider transitions to a '1' and remains free running thereafter.

When the block is disabled, the SCLK and MOSI outputs revert to their idle state. All internal states are reset (including CR0 status) to their configuration-specific reset state, except for DR0, DR1, and DR2, which are unaffected.

Normal Operation. Typical timing for an SPIM transfer is shown in Figure 17-16 and Figure 17-17. The user initially writes a byte to transmit when TX Reg Empty status is true. If no transmission is currently in progress, the data is loaded into the shifter and the transmission is initiated. The TX Reg Empty status is asserted again and the user is allowed to write the next byte to be transmitted to the TX Buffer register. After the last bit is output, if TX Buffer data is available with one-half clock set up time to the next clock, a new byte transmission is initiated. An SPIM block receives a byte at the same time that it sends one. The SPI Complete or RX Reg Full can be used to determine when the input byte has been received.

Figure 17-16. Typical SPIM Timing in Mode 0 and 1

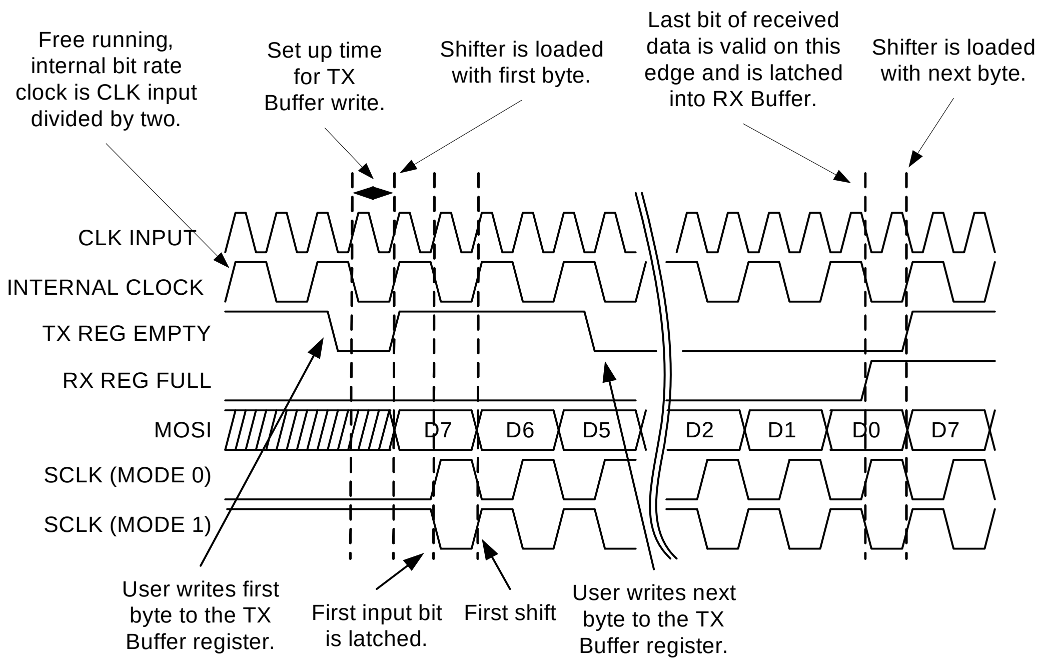
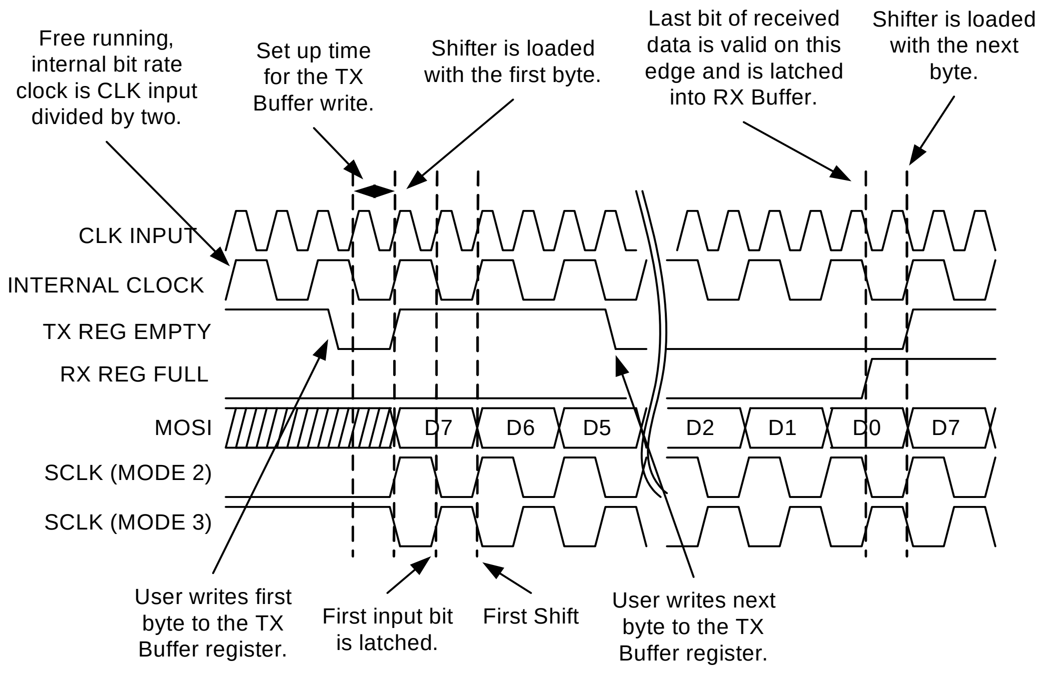

Free running, internal bit rate clock is CLK input divided by two.
Set up time for TX Buffer write.
Shifter is loaded with first byte.
Last bit of received data is valid on this edge and is latched into RX Buffer.
Shifter is loaded with next byte.
CLK INPUT
INTERNAL CLOCK
TX REG EMPTY
RX REG FULL
MOSI
SCLK (MODE 0)
SCLK (MODE 1)
D7
D6
D5
D2
D1
D0
D7
User writes first byte to the TX Buffer register.
First input bit is latched.
First shift
User writes next byte to the TX Buffer register.

Figure 17-17. Typical SPIM Timing in Mode 2 and 3


Free running, internal bit rate clock is CLK input divided by two.
Set up time for the TX Buffer write.
Shifter is loaded with the first byte.
Last bit of received data is valid on this edge and is latched into RX Buffer.
Shifter is loaded with the next byte.
CLK INPUT
INTERNAL CLOCK
TX REG EMPTY
RX REG FULL
MOSI
SCLK (MODE 2)
SCLK (MODE 3)
D7
D6
D5
D2
D1
D0
D7
User writes first byte to the TX Buffer register.
First input bit is latched.
First Shift
User writes next byte to the TX Buffer register.

Status Generation and Interrupts. There are four status bits in an SPI Block: TX Reg Empty, RX Reg Full, SPI Complete, and Overrun.

TX Reg Empty indicates that a new byte can be written to the TX Buffer register. When the block is enabled, this status bit is immediately asserted. This status bit is cleared when the user writes a byte of data to the TX Buffer register. TX Reg Empty is a control input to the state machine and, if a transmission is not already in progress, the assertion of this control signal initiates one. This is the default SPIM block interrupt. However, an initial interrupt is not generated when the block is enabled. The user must write a byte to the TX Buffer register and that byte must be loaded into the shifter before interrupts generated from the TX Reg Empty status bit are enabled.

RX Reg Full is asserted on the edge that captures the eighth bit of receive data. This status bit is cleared when the user reads the RX Buffer register (DR2).

SPI Complete is an optional interrupt and is generated when eight bits of data and clock have been sent. In modes 0 and 1, this occurs one-half cycle after RX Reg Full is set; because in these modes, data is latched on the leading edge of the clock and there is an additional one-half cycle remaining to complete that clock. In modes 2 and 3, this occurs at the same edge that the receive data is latched. This signal may be used to read the received byte or it may be used by the SPIM to disable the block after data transmission is complete.

Overrun status is set if RX Reg Full is still asserted from a previous byte when a new byte is about to be loaded into the RX Buffer register. Because the RX Buffer register is implemented as a latch, Overrun status is set one-half bit clock before RX Reg Full status.

See Figure and Figure 17-19 for status timing relationships.

Figure 17-18. SPI Status Timing for Modes 0 and 1

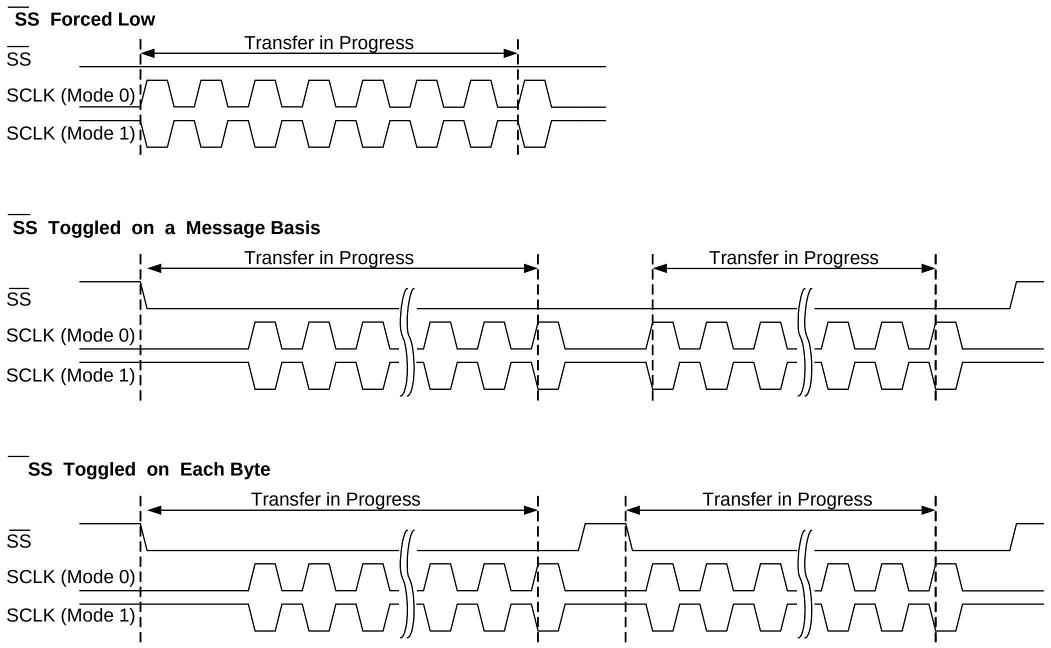
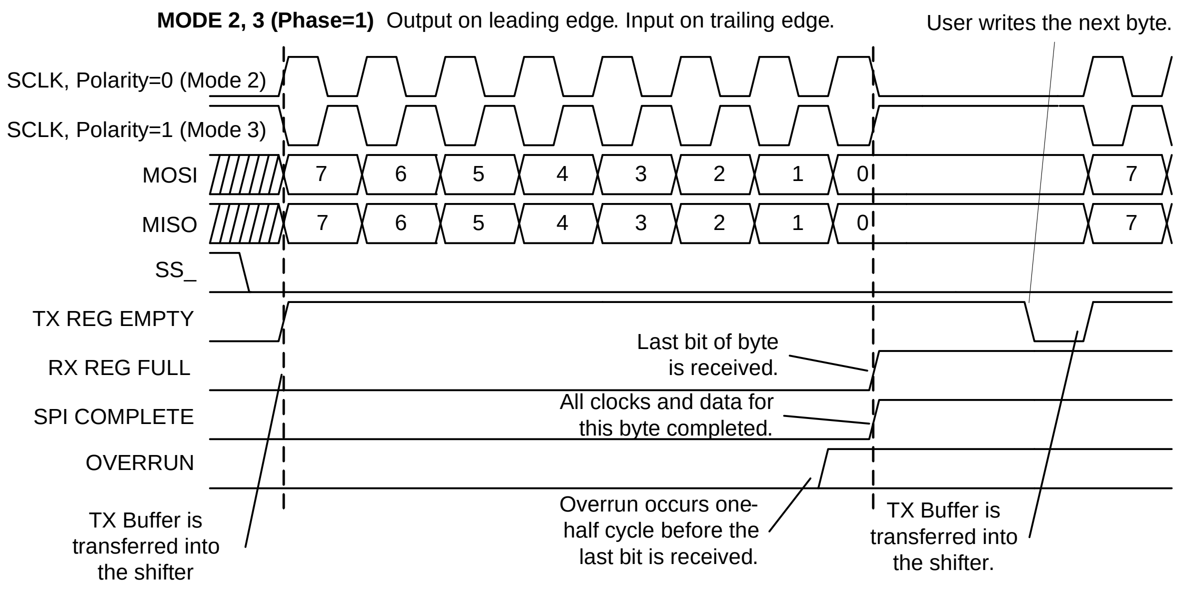


Figure 17-19. SPI Status Timing for Modes 2 and 3


MODE 2, 3 (Phase=1) Output on leading edge. Input on trailing edge.
User writes the next byte.
SCLK, Polarity=0 (Mode 2)
SCLK, Polarity=1 (Mode 3)
MOSI
7 6 5 4 3 2 1 0 7
MISO
7 6 5 4 3 2 1 0 7
SS_
TX REG EMPTY
RX REG FULL
SPI COMPLETE
OVERRUN
Last bit of byte is received.
All clocks and data for this byte completed.
Overrun occurs one-half cycle before the last bit is received.
TX Buffer is transferred into the shifter
TX Buffer is transferred into the shifter.

17.3.7 SPIS Timing

Enable/Disable Operation. As soon as the block is configured for SPI slave and before enabling, the MISO output is set to idle at logic 1. Both the enable bit must be set and the SS_ asserted (either driven externally or forced by firmware programming) for the block to output data. When enabled, the primary output is the MSb or LSb of the shift register, depending on the LSb First configuration in bit 7 of the control register. The auxiliary output of the SPIS is always forced into tri-state.

Since the SPIS has no internal clock, it must be enabled with set up time to any external master supplying the clock. Set up time is also required for a TX Buffer register write, before the first edge of the clock or the first falling edge of SS_, depending on the mode. This set up time must be assured through the protocol and an understanding of the timing between the master and slave in a system.

When the block is disabled, the MISO output reverts to its idle '1' state. All internal states are reset (including CR0 status) to their configuration-specific reset state, except for DR0, DR1, and DR2, which are unaffected.

Normal Operation. Typical timing for an SPIS transfer is shown in Figure 17-20 and Figure 17-21. If the SPIS is primarily being used as a receiver, the RX Reg Full (polling only) or SPI Complete (polling or interrupt) status may be used to determine when a byte has been received. In this way, the SPIS operates identically to the SPIM. However, there are two main areas in which the SPIS operates differently: 1) SPIS behavior related to the SS_ signal, and 2) TX data queuing (loading the TX Buffer register).

Figure 17-20. Typical SPIS Timing in Modes 0 and 1

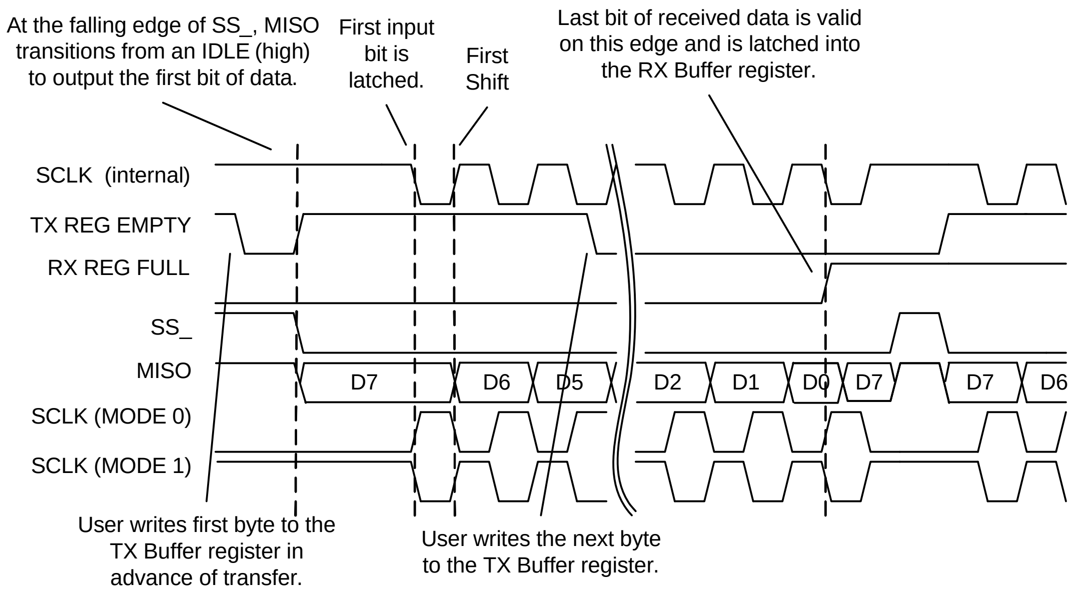
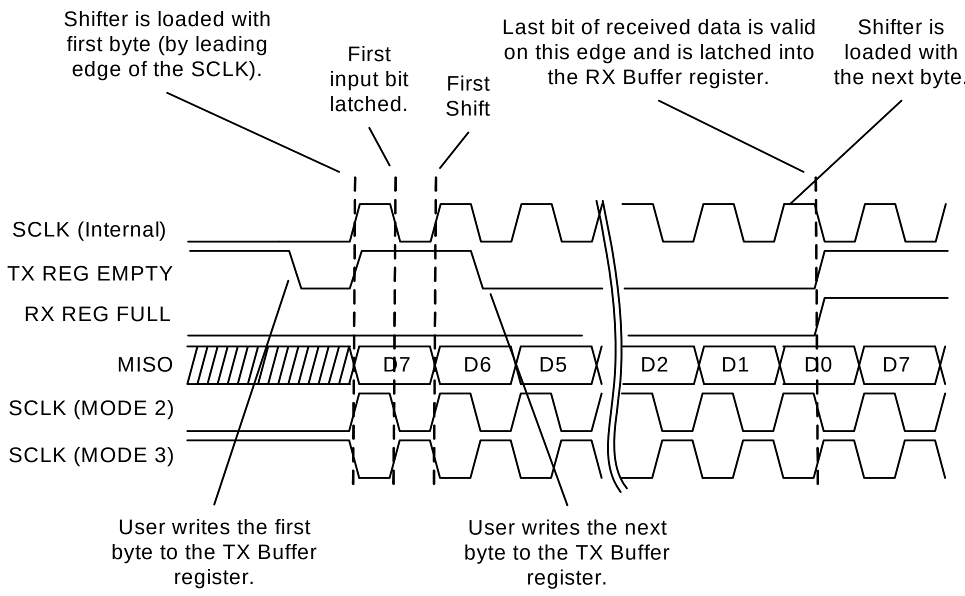

At the falling edge of SS_, MISO transitions from an IDLE (high) to output the first bit of data.
First input bit is latched.
First Shift
Last bit of received data is valid on this edge and is latched into the RX Buffer register.
SCLK (internal)
TX REG EMPTY
RX REG FULL
SS_
MISO
SCLK (MODE 0)
SCLK (MODE 1)
D7
D6
D5
D2
D1
D0
D7
D7
D6
User writes first byte to the TX Buffer register in advance of transfer.
User writes the next byte to the TX Buffer register.

Figure 17-21. Typical SPIS Timing in Modes 2 and 3


Shifter is loaded with first byte (by leading edge of the SCLK).
First input bit latched.
First Shift
Last bit of received data is valid on this edge and is latched into the RX Buffer register.
Shifter is loaded with the next byte.
SCLK (Internal)
TX REG EMPTY
RX REG FULL
MISO
D7
D6
D5
D2
D1
D0
D7
SCLK (MODE 2)
SCLK (MODE 3)
User writes the first byte to the TX Buffer register.
User writes the next byte to the TX Buffer register.

Slave Select (SS_, active low). Slave Select must be asserted to enable the SPIS for receive and transmit. There are two ways to do this:

1. Drive the auxiliary input from a pin (selected by the Aux IO Select bits in the output register). This gives the SPI master control of the slave selection in a multi-slave environment.
2. SS_ may be controlled in firmware with register writes to the output register. When Aux IO Enable = 1, Aux IO Select bit 0 becomes the SS_ input. This allows the user to save an input pin in single-slave environments.

When SS_ is negated (whether from an external or internal source), the SPIS state machine is reset and the MISO output is forced to idle at logic 1. In addition, the SPIS ignores any incoming MOSI/SCLK input from the master.

Status Generation and Interrupts. There are four status bits in the SPIS Block: TX Reg Empty, RX Reg Full, SPI Complete, and Overrun. The timing of these status bits is identical to the SPIM, with the exception of TX Reg Empty, which is covered in the section on TX data queuing.

Status Clear On Read. Refer to the same subsection in "SPIM Timing" on page 203.

TX Data Queuing. Most SPI applications call for data to be sent back from the slave to the master. Writing firmware to accomplish this requires an understanding of how the shift register is loaded from the TX Buffer register.

All modes use the following mechanism: 1) If there is no transfer in progress, 2) if the shifter is empty, and 3) if data is available in the TX Buffer register, the byte is loaded into the shifter.

The only difference between the modes is that the definition of "transfer in progress" is slightly different between modes 0 and 1, and modes 2 and 3.

Figure 17-22 illustrates TX data loading in modes 0 and 1. A transfer in progress is defined to be from the falling edge of SS_ to the point at which the RX Buffer register is loaded with the received byte. This means that in order to send a byte in the next transfer, it must be loaded into the TX Buffer register before the falling edge of SS_. This ensures a minimum set up time for the first bit, since the leading edge of the first SCLK must latch in the received data. If SS_ is not toggled between each byte or is forced low through the configuration register, the leading edge of SCLK is used to define the start of transfer. However, in this case, the user must provide the required set up time (one-half clock minimum before the leading edge), with a knowledge of system latencies and response times.

Figure 17-22. Mode 0 and 1 Transfer in Progress

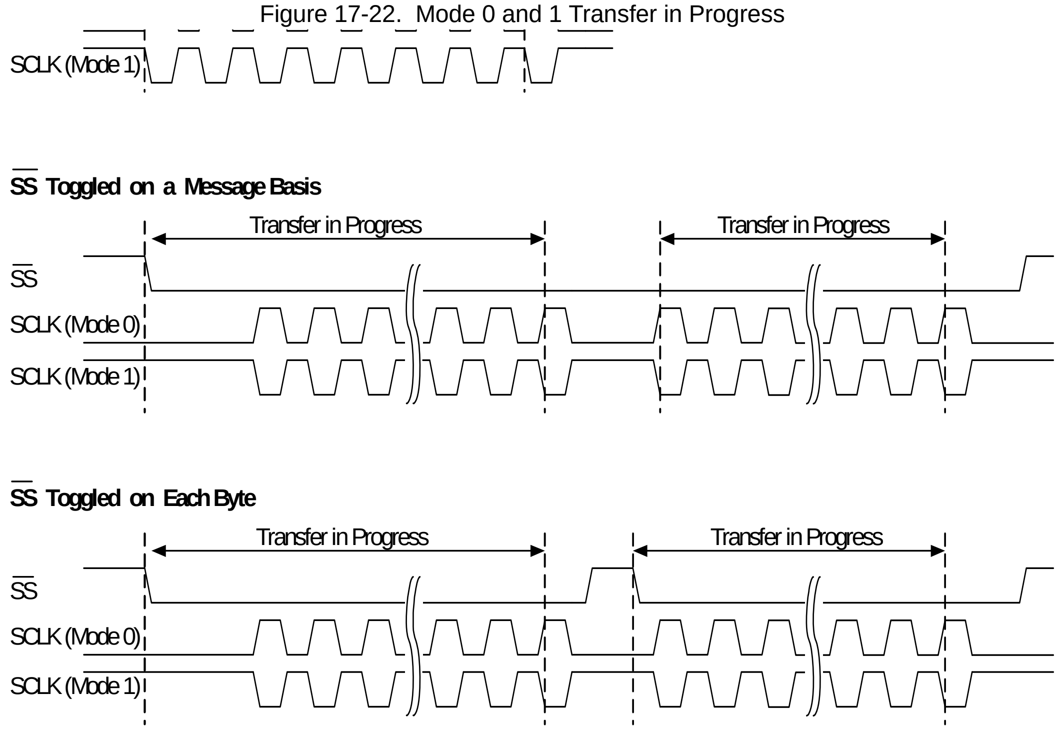
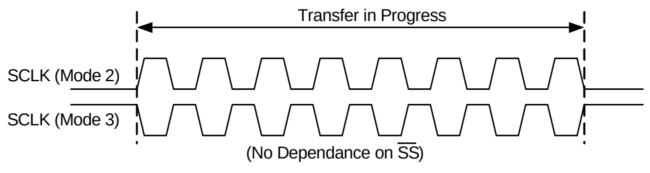


Figure 17-23 illustrates TX data loading in modes 2 and 3. In this case, there is no dependence on $\overline{SS}$ and a transfer in progress is defined to be from the leading edge of the first SCLK to the point at which the RX Buffer register is loaded with the received byte. Loading the shifter by the leading edge of the clock has the effect of providing the required one-half clock set up time, as the data is latched into the receiver on the trailing edge of the SCLK in these modes.

Figure 17-23. Mode 2 and 3 Transfer in Progress



17.3.8 Transmitter Timing

Enable/Disable Operation. As soon as the block is configured for the transmitter and before enabling, the primary output is set to idle at logic 1, the mark state. The output remains '1' until the block is enabled and a transmission is initiated. The auxiliary output also idles to '1', which is the idle state of the associated SPI mode 3 clock.

When the transmitter is enabled, the internal reset is released on the divide-by-eight clock generator circuit. On the next positive edge of the selected input clock, this 3-bit up counter circuit, which generates the bit clock with the MSb, starts counting up from 00h, and is free running thereafter.

When the block is disabled, the clock is immediately gated low. All internal states are reset (including CR0 status) to their configuration-specific reset state, except for DR0, DR1, and DR2, which are unaffected.

Transmit Operation. Transmission is initiated with a write to the TX Buffer register (DR1). The CPU write to this register is required to have one-half bit clock set up time for the data, to be recognized at the next positive internal bit clock edge. As shown in Figure 17-24, once the set up time is met, there is one clock of latency until the data is loaded into the shifter and the START bit is generated to the TXD (primary) output.

Figure 17-24. Typical Transmitter Timing

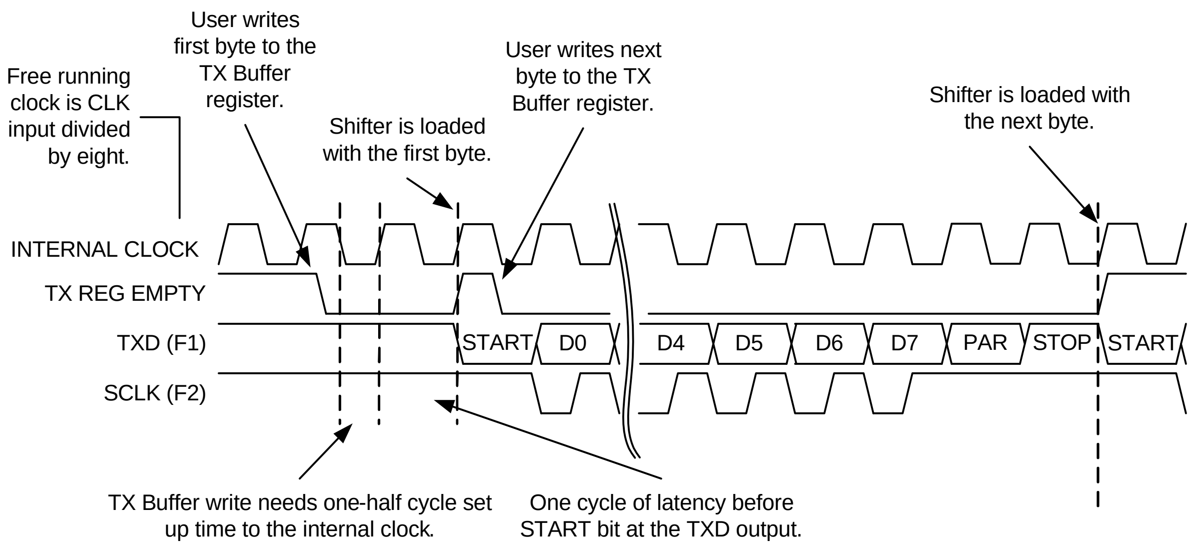
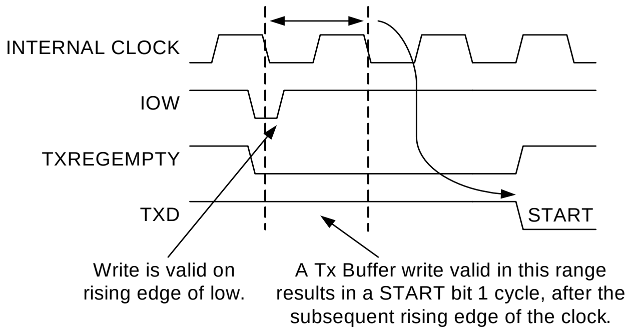

User writes first byte to the TX Buffer register.
Free running clock is CLK input divided by eight.
Shifter is loaded with the first byte.
User writes next byte to the TX Buffer register.
Shifter is loaded with the next byte.
INTERNAL CLOCK
TX REG EMPTY
TXD (F1)
SCLK (F2)
START
D0
D4
D5
D6
D7
PAR
STOP
START
TX Buffer write needs one-half cycle set up time to the internal clock.
One cycle of latency before START bit at the TXD output.

Figure 17-25 shows a detail of the Tx Buffer load timing. The data bits are shifted out on each of the subsequent clocks. Following the eighth bit, if parity is enabled, the parity bit is sent to the output. Finally, the STOP bit is multiplexed into the data stream. With one-half cycle set up to the next clock, if new data is available from the TX Buffer register, the next byte is loaded on the following clock edge and the process is repeated. If no data is available, a mark (logic 1) is output.

Figure 17-25. Tx Buffer Load Timing


INTERNAL CLOCK
IOW
TXREGEMPTY
TXD
START
Write is valid on rising edge of low.
A Tx Buffer write valid in this range results in a START bit 1 cycle, after the subsequent rising edge of the clock.

The SCLK (auxiliary) output has an SPI mode 3 clock associated with the data bits (for the mode 3 timing see Figure 17-15). During the mark (idle) and framing bits the SCLK output is high.

Status Generation. There are two status bits in the transmitter CR0 register: TX Reg Empty and TX Complete.

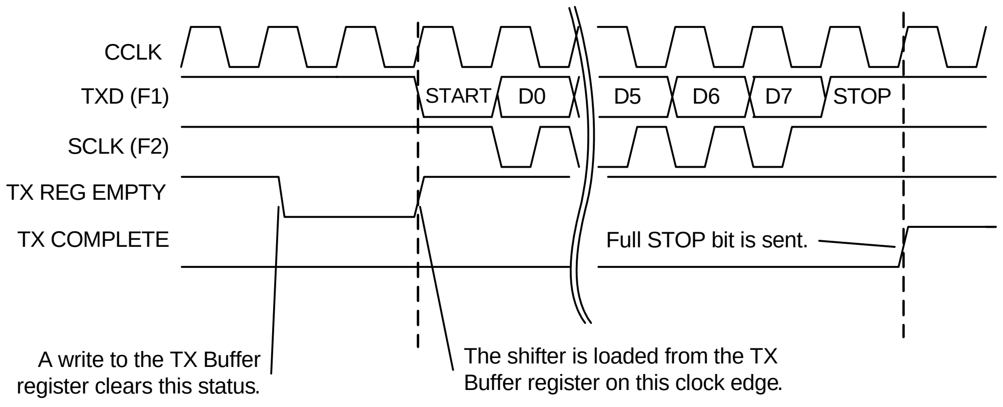
TX Reg Empty indicates that a new byte can be written to the TX Buffer register. When the block is enabled, this status bit is immediately asserted. This status bit is cleared when the user writes a byte of data to the TX Buffer register and set when the data byte in the TX Buffer register is transferred into the shifter. If a transmission is not already in progress, the assertion of this signal initiates one subject to the timing.

The default interrupt in the transmitter is tied to TX Reg Empty. However, an initial interrupt is not generated when the block is enabled. The user must write an initial byte to the TX Buffer register. That byte must be transferred into the shifter, before interrupts generated from the TX Reg Empty status bit are enabled. This prevents an interrupt from occurring immediately on block enable.

TX Complete is an optional interrupt and is generated when all bits of data and framing bits have been sent. It is cleared on a read of the CR0 register. This signal may be used to determine when it is safe to disable the block after data transmission is complete. In an interrupt-driven transmitter application, if interrupt on TX Complete is selected, the status must be cleared on every interrupt; otherwise, the status remains high and no subsequent interrupts are logged. See Figure 17-26 for timing relationships.

Status Clear On Read. Refer to the SPIM subsection in "SPIM Timing" on page 203.

Figure 17-26. Status Timing for the Transmitter


CCLK
TXD (F1)
START
D0
D5
D6
D7
STOP
SCLK (F2)
TX REG EMPTY
TX COMPLETE
Full STOP bit is sent.
A write to the TX Buffer register clears this status.
The shifter is loaded from the TX Buffer register on this clock edge.

17.3.9 Receiver Timing

Enable/Disable Operation. As soon as the block is configured for receiver and before enabling, the primary output is connected to the data input (RXD). This output continues to follow the input, regardless of enable state. The auxiliary output idles to '1', which is the idle state of the associated SPI mode 3 clock.

When the receiver is enabled, the internal clock generator is held in reset until a START bit is detected on the input. The block must be enabled with a set up time to the first START bit input.

When the block is disabled, the clock is immediately gated low. All internal states are reset (including CR0 status) to their configuration-specific reset state, except for DR0, DR1, and DR2 which are unaffected.

Receive Operation. A clock, which must be eight times the desired baud rate, is selected as the CLK input. This clock is an input to the RX block clock divider. When the receiver is idle, the clock divider is held in reset. As shown in Figure 17-27, reception is initiated when a START bit (logic 0) is detected on the RXD input.

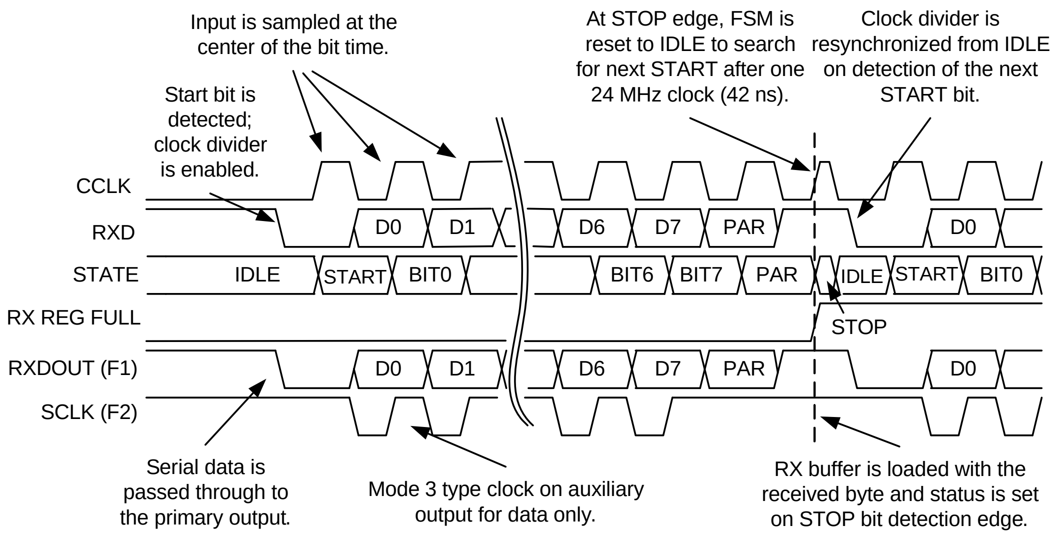
When this occurs, the reset is negated to the clock divider and the 3-bit counter starts an up-count. The block clock is derived from the MSb of this counter (corresponding to a count of four), which serves to sample each incoming bit at the nominal center point. This clock also sequences the state machine at the specified bit rate.

The sampled data is registered into an input flip-flop. This flip-flop feeds the DR0 shift register. Only data bits are shifted into the shift register.

At the STOP sample point, the block is immediately (within one cycle of the 24 MHz system clock) set back into an idle state. In this way, the clock generation circuit can immediately enable the search for the next START bit, thereby resynchronizing the bit clock with the incoming bit rate on every new data byte reception. The RX Reg Full status bit, as well as error status, is also set at the STOP sample point.

To facilitate connection to other digital blocks, the RXD input is passed directly to the RXDOUT (primary) output. The SCLK (auxiliary) output has an SPI mode 3 clock associated with the data bits (for mode 3 timing see Figure 17-27). During the mark (idle) and framing bits, the SCLK output is high.

Figure 17-27. Receiver Operation


Input is sampled at the center of the bit time.
At STOP edge, FSM is reset to IDLE to search for next START after one 24 MHz clock (42 ns).
Clock divider is resynchronized from IDLE on detection of the next START bit.
Start bit is detected; clock divider is enabled.
CCLK
RXD
STATE
RX REG FULL
RXDOUT (F1)
SCLK (F2)
D0
D1
D6
D7
PAR
IDLE
START
BIT0
BIT6
BIT7
STOP
Serial data is passed through to the primary output.
Mode 3 type clock on auxiliary output for data only.
RX buffer is loaded with the received byte and status is set on STOP bit detection edge.

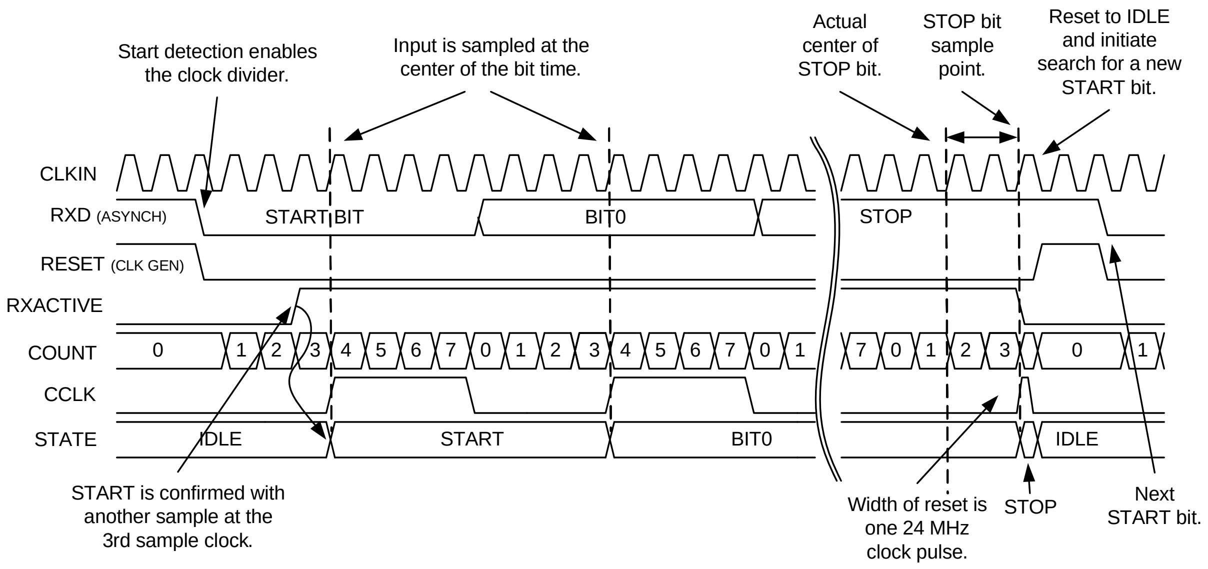
Clock Generation and Start Detection. The input clock selection is a free running, eight times over-sampling clock. This clock is used by the clock divider circuit to generate the block clock at the bit rate. As shown in Figure 17-28, the clock block is derived from the MSb of a 3-bit counter, giving a sample point as near to the center of the bit time as possible. This block clock is used to clock all internal circuits.

Since the RXD bit rate is asynchronous to the block bit clock, these clocks must be continually re-aligned. This is accomplished with the START bit detection.

When in IDLE state, the clock divider is held in reset. On START (when the input RXD transitions are detected as a logic 0), the reset is negated and the divider is enabled to count at the eight times rate. If the RXD input is still logic 0 after three samples of the input clock, the status RXACTIVE is asserted, which initiates a reception. If this sample of the RXD line is a logic 1, the input '0' transition is assumed to be a false start and the receiver remains in the idle state.

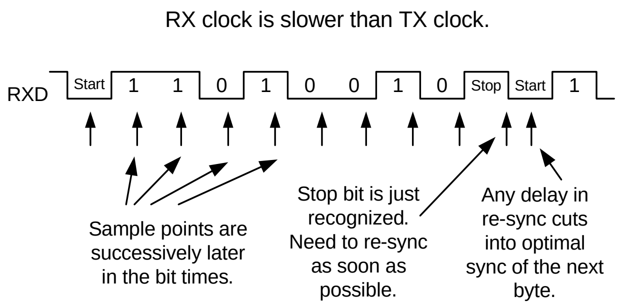
As shown in Figure 17-28, the internal bit clock (CCLK) runs slower than the external TX bit clock and the STOP bit is sampled later than the actual center point. After the STOP bit is sampled, the 24 MHz reset pulse forces the receiver back to an idle state. In this state, the next START bit search is initiated, resynchronizing the RX bit clock to the TX bit clock.

Figure 17-28. Clock Generation and Start Detection


Start detection enables the clock divider.
Input is sampled at the center of the bit time.
Actual center of STOP bit.
STOP bit sample point.
Reset to IDLE and initiate search for a new START bit.
CLKIN
RXD (ASYNCH)
START BIT
BIT0
STOP
RESET (CLK GEN)
RXACTIVE
COUNT
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7
0
1
7
0
1
2
3
0
1
CCLK
STATE
IDLE
START
BIT0
IDLE
START is confirmed with another sample at the 3rd sample clock.
Width of reset is one 24 MHz clock pulse.
STOP
Next START bit.

This resynchronization process (forcing the state back to idle) occurs regardless of the value of the STOP bit sample. It is important to reset as soon as possible, so that maximum performance can be achieved. Figure 17-29 shows an example where the RX block clock bit rate is slower than the external TX bit rate. The sample point shifts to successively later times. In the extreme case shown, the RX samples the STOP bit at the trailing edge. In this case, the receiver counts 9.5 bit times, while the transmitter counts 10 bit times. Therefore, for a 10-bit message, the maximum theoretical clock offset, for the message to be received correctly, is represented by one-half bit time or five percent. If the RX and TX clocks exceed this offset, a logic 0 may be sampled for the STOP bit. In this case, the Framing Error status is set.

Figure 17-29. Example RX Re-Synchronization


RX clock is slower than TX clock.
RXD
Start
1
1
0
1
0
0
1
0
Stop
Start
1
Sample points are successively later in the bit times.
Stop bit is just recognized. Need to re-sync as soon as possible.
Any delay in re-sync cuts into optimal sync of the next byte.

This theoretical maximum is degraded by the resynchronization time, which is fixed at approximately 42 ns. In a typical 115.2 Kbaud example, the bit time is 8.70 µs. In this case the new maximum offset is:

((4.35 ms - 42 ns) / 4.35 ms) x 5% or 4.95%

At slower baud rates, this value gets closer to the theoretical maximum of five percent.

Status Generation. There are five status bits in a receiver block: RX Reg Full, RX Active, Framing Error, Overrun, and Parity Error. All status bits, except RX Active and Overrun, are set synchronously on the STOP bit sample point.

RX Reg Full indicates a byte has been received and transferred into the RX Buffer register. This status bit is cleared when the user reads the RX Buffer register (DR2). The setting of this bit is synchronized to the STOP sample point. This is the earliest point at which the Framing Error status can be set; and therefore, error status is defined to be valid when RX Reg Full is set.

RX Active can be polled to determine if a reception is in progress. This bit is set on START detection and cleared on STOP detection. This bit is not ***sticky*** and there is no way for the user to clear it.

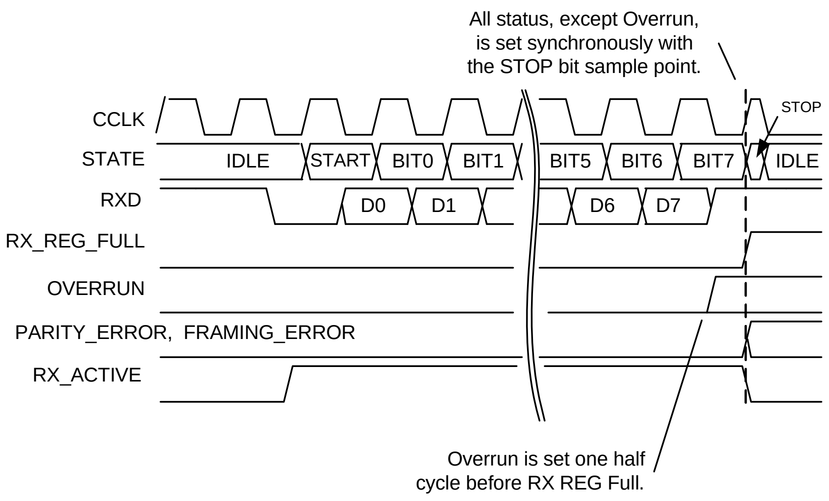
Framing Error status indicates that the STOP bit associated with a given byte was not received correctly (expecting a '1', but received a '0'). This typically occurs when the difference between the baud rates of the transmitter and receiver is greater than the maximum allowed.

Overrun occurs when there is a received data byte in the RX Buffer register and a new byte is loaded into the RX Buffer register before the user has had a chance to read the previous one. Because the RX Buffer register is actually a latch, Overrun status is set one-half cycle before RX Reg Full. This means that although the new data is not available, the previous data has been overwritten because the latch was opened.

Parity Error status indicates that resulting parity calculation on the received byte does not match the value of the parity bit that was transmitted. This status is set on the sample point of the STOP signal.

Status Clear On Read. Refer to the SPIM subsection in "SPIM Timing" on page 203.

Figure 17-30. Status Timing for Receiver


All status, except Overrun, is set synchronously with the STOP bit sample point.
CCLK
STATE
RXD
RX_REG_FULL
OVERRUN
PARITY_ERROR, FRAMING_ERROR
RX_ACTIVE
IDLE
START
BIT0
BIT1
BIT5
BIT6
BIT7
STOP
IDLE
D0
D1
D6
D7
Overrun is set one half cycle before RX REG Full.

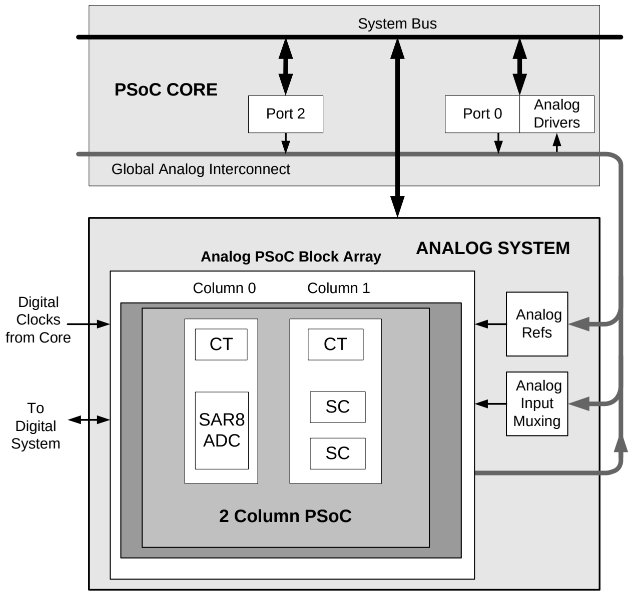
Section E: Analog System

The configurable Analog System section discusses the analog components of the PSoC device and the registers associated with those components. Note that the analog output drivers are described in the PSoC Core section, Analog Output Drivers chapter on page 13, because they are part of the core input and output signals. This section encompasses the following chapters:

Top-Level Analog Architecture

The figure below displays the top-level architecture of the PSoC's analog system. With the exception of the analog drivers, each component of the figure is discussed at length in this section. Analog drivers are discussed in detail within the PSoC Core section, in the Analog Output Drivers chapter on page 13.

PSoC Analog System



Interpreting the Analog Documentation

Information in this section covers the CY8C24533, CY8C23533, CY8C23433CY8C24633 PSoC device. The following table lists the resources available for the CY8C24533, CY8C23533, CY8C23433CY8C24633 (and related) PSoC devices. While reading the analog system section, keep in mind the number of analog columns in the CY8C24533, CY8C23533, CY8C23433CY8C24633 is 2.

PSoC Device Characteristics

PSoC Part Number	Digital IO (max)	Digital Rows	Digital Blocks	Analog Inputs	Analog Outputs	Analog Columns	Analog Blocks
CY8C24x23A	24	1	4	12	2	2	6
CY8C24533	26	1	4	12	2	2	4[a]
CY8C23533	26	1	4	12	2	2	4[a]
CY8C23433	26	1	4	12	2	2	4[a]
CY8C24633	25	1	4	12	2	2	4[b]

a. 2 CT, 2 SC, 1 SAR8 ADC.
b. 2 CT, 2 SC, 1 SAR8 ADC.

Application Description

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

A precision internal voltage reference provides accurate analog comparisons. A temperature sensor input is provided to the analog PSoC block array, supporting applications such as battery chargers and data acquisition, without requiring external components.

Defining the Analog Blocks

There are three analog PSoC block types: Continuous Time (CT) blocks, and Type C and Type D Switch Capacitor (SC) blocks. CT blocks provide continuous time analog functions. SC blocks provide switched capacitor analog functions.

Each of the analog blocks has many potential inputs and several outputs. The inputs to these blocks include ***analog signals*** from external sources, intrinsic analog signals driven from neighboring analog blocks, or various voltage reference sources.

The analog blocks are organized into columns. Each column contains one Continuous Time (CT) block, Type B (ACB); one Switched Capacitor (SC) block, Type C (ASC); and one Switched Capacitor block, Type D (ASD). However, the number of analog columns in a specific part can either be 1, 2, or 4 columns.

For the CY8C24533, CY8C23533, CY8C23433CY8C24633 devices, Analog Column 0 contains the SAR8 ADC block rather than the standard SC blocks.

The blocks in a particular column all run off the same clocking source. The blocks in a column also share some output bus resources. Refer to the Analog Interface, on page 219 for additional information.

There are three types of outputs from each analog block and additional two discrete outputs in the Continuous Time blocks.

1. The analog output bus (ABUS) is an analog bus resource that is shared by all of the analog blocks in a column. Only one block in a column can actively drive this bus at any one time, with the user having control of this output through register settings. This is the only analog output that can be driven directly to a pin.
2. The comparator bus (CBUS) is a digital bus resource that is shared by all of the analog blocks in a column. Only one block in a column can be actively driving this bus at any one time, with the user having control of this output through register settings.
3. The local outputs (OUT, GOUT, and LOUT in the Continuous Time blocks) are routed to neighbor blocks. The various input ***multiplexer (mux)*** connections (NMux, PMux, RBotMux, AMux, BMux, and CMux) all use the output bus from one block as their input.

Analog Functionality

The following is a sampling of the functions that operate within the capability of the analog PSoC blocks, using one analog PSoC block, multiple analog blocks, a combination of more than one *type* of analog block, or a combination of analog and digital PSoC blocks. Most of these functions are currently available as ***user modules*** in *PSoC Designer*. Others will be added in the future. Refer to the *PSoC Designer* software for additional information and the most up-to-date list of user modules.

- Delta-Sigma Analog-to-Digital Converters
- Successive Approximation Analog-to-Digital Converters
- 8-Bit Successive Approximation Analog-to-Digital Converter
- Incremental Analog-to-Digital Converters
- Digital to Analog Converters
- Programmable Gain/Loss Stage
- Analog Comparators
- Zero-Crossing Detectors
- Sample and Hold
- Low-Pass Filter
- Band-Pass Filter
- Notch Filter
- Amplitude Modulators
- Amplitude Demodulators
- Sine-Wave Generators
- Sine-Wave Detectors
- Sideband Detection
- Sideband Stripping
- Temperature Sensor
- Audio Output Drive
- DTMF Generator
- FSK Modulator
- Embedded Modem

By modifying registers, as described in this document, users can configure PSoC blocks to perform these functions and more. The philosophy of the analog functions supplied is as follows.

- Cost effective, single-ended configuration for reasonable speed and accuracy, providing a simple interface to most real-world analog inputs and outputs.
- Flexible, System-on-Chip programmability, providing variations in functions.
- Function specific, easily selected trade-offs of accuracy and resolution with speed, resources (number of analog blocks), and power dissipated for that application.

Analog Register Summary

The table below lists all the PSoC registers for the analog system in address order within their system resource configuration. The bits that are grayed out are reserved bits. If these bits are written, they should always be written with a value of ‘0’. The naming conventions for the SC and CT registers and their arrays of PSoC blocks are detailed in their respective table title rows.

Analog PSoC arrays are 2 column or 1 column devices. The CY8C24533, CY8C23533, CY8C23433CY8C24633 are 2 column devices.

Summary Table of the Analog Registers

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
ANALOG INTERFACE REGISTERS (page 227)										
0,64h	CMP_CR0			COMP[1:0]]				AINT[1:0]		# : 00
0,65h	ASY_CR		SARCNT[2:0]			SARSIGN	SARCOL[1]		SYNCEN	RW : 00
0,66h	CMP_CR1			CLDIS[1]	CLDIS[0]					RW : 00
0,E6h	DEC_CR0			IGEN[1:0]		ICLKS0	DCOL[1:0]		DCLKS0	RW : 00
0,E7h	DEC_CR1	ECNT	IDEC	ICLKS3	ICLKS2	ICLKS1	DCLKS3	DCLKS2	DCLKS1	RW : 00
1,60h	CLK_CR0					AColumn1[1:0]		AColumn0[1:0]		RW : 00
1,61h	CLK_CR1		SHDIS	ACLK1[2:0]			ACLK0[2:0]			RW : 00
1,63h	AMD_CR0						AMOD0[2:0]			RW : 00
1,66h	AMD_CR1						AMOD1[2:0]			RW : 00
1,67h	ALT_CR0	LUT1[3:0]				LUT0[3:0]				RW : 00
ANALOG INPUT CONFIGURATION REGISTERS (page 243)										
0,60h	AMX_IN					ACI1[1:0]		ACI0[1:0]		RW : 00
1,62h	ABF_CR0	ACol1Mux		ABUF1EN		ABUF0EN		Bypass	PWR	RW : 00
ANALOG REFERENCE REGISTER (page 246)										
0,63h	ARF_CR		HBE	REF[2:0]			PWR[2:0]			RW : 00

CONTINUOUS TIME PSoC BLOCK REGISTERS (page 251)										
x,74h	ACB01CR3					LPCMPEN	CMOUT	INSAMP	EXGAIN	RW : 00
x,75h	ACB01CR0	RTapMux[3:0]				Gain	RTopMux	RBotMux[1:0]		RW : 00
x,76h	ACB01CR1	AnalogBus	CompBus	NMux[2:0]			PMux[2:0]			RW : 00
x,77h	ACB01CR2	CPhase	CLatch	CompCap	TMUXEN	TestMux[1:0]		PWR[1:0]		RW : 00
SWITCHED CAPACITOR PSoC BLOCK REGISTERS (page 258)										
Switched Capacitor Block Registers, Type C (page 259)										
x,94h	ASCxxCR0	FCap	ClockPhase	ASign	ACap[4:0]					RW : 00
x,95h	ASC21CR1	ACMux[2:0]			BCap[4:0]					RW : 00
x,96h	ASC21CR2	AnalogBus	CompBus	AutoZero	CCap[4:0]					RW : 00
x,97h	ASC21CR3	ARefMux[1:0]		FSW1	FSW0	BMuxSC[1:0]		PWR[1:0]		RW : 00
Switched Capacitor Block Registers, Type D (page 262)										
x,84h	ASD11CR0	FCap	ClockPhase	ASign	ACap[4:0]					RW : 00
x,85h	ASD11CR1	AMux[2:0]			BCap[4:0]					RW : 00
x,86h	ASD11CR2	AnalogBus	CompBus	AutoZero	CCap[4:0]					RW : 00
x,87h	ASD11CR3	ARefMux[1:0]		FSW1	FSW0	BSW	BMuxSD	PWR[1:0]		RW : 00

Summary Table of the Analog Registers *(continued)*

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
SAR8 ADC PSoC BLOCK REGISTERS (page 266)										
0,67h	SARADC_DL	Data[7:0]								RW : 00
0,69h	SARADC_CR0		ADC Channel[3:0]				Data Ready	Start/Busy	ADCEN	# : 00
0,6Ah	SARADC_CR1	PWRSELADC	PWRSELR2R				Align Source[1:0]		Align Enable	RW : 0
1,A8h	SARADC_TRS	DCB03_HL[1:0]		DCB02_HL[1:0]		DBB01_HL[1:0]		DBB00_HL[1:0]		RW : 0
1,A9h	SARADC_TRCL	CMP_L[7:0]								RW : 00
1,AAh	SARADC_TRCH	CMP_H[7:0]								RW : 00
1,ABh	SARADC_CR2	Test Enable	Free Run	Scale Size [2:0]			ADC Clock [2:0]			# : 0
1,ACh	SARADC_LCR	DA_L[7:0]								RW : 00

LEGEND

x	An “x” before the comma in the address field indicates that this register can be accessed or written to no matter what bank is used.
#	Access is bit specific. Refer to the Register Details chapter on page 47 for additional information.
R	Read register or bit(s).
W	Write register or bit(s).

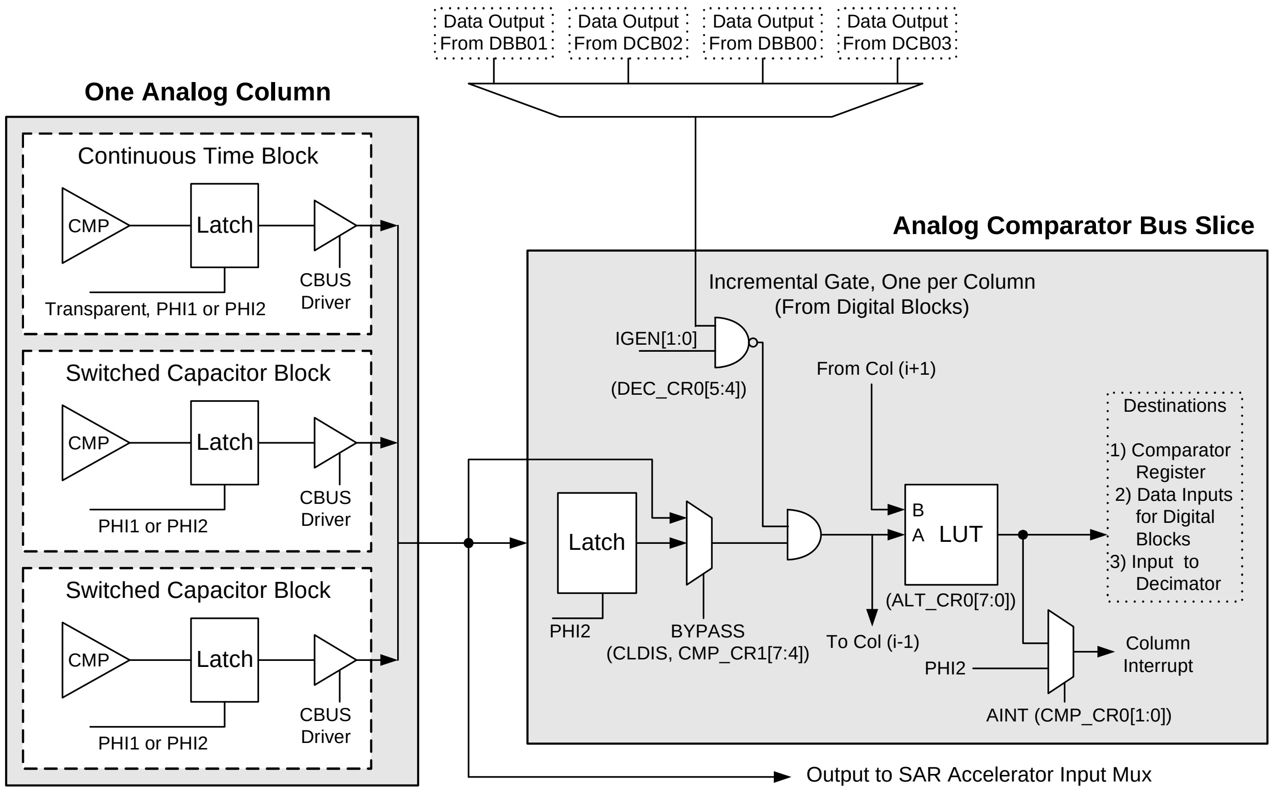
18. Analog Interface

This chapter explains the Analog Interface and its associated registers. The analog system interface is a collection of system level interfaces to the analog array and analog reference block. For a complete table of the analog interface registers, refer to the “Summary Table of the Analog Registers” on page 217. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

18.1 Architectural Description

Figure 18-1 displays the top-level diagram of the PSoC device’s analog interface system. This diagram displays Analog Column 1 for the CY8C24533, CY8C23533, CY8C23433CY8C24633 devices. Analog Column 0 for this device includes one CT (Continuous Time) block and the SAR8 ADC block, rather than the two standard SC (Switched Capacitor) blocks.

Figure 18-1. Analog Comparator Bus Slice



18.1.1 Analog Data Bus Interface

The Analog Data Bus Interface isolates the analog array and analog system interface registers from the CPU system data bus, to reduce bus loading. Transceivers are implemented on the system data bus to isolate the analog data bus from the system data bus. This creates a local analog data bus.

18.1.2 Analog Comparator Bus Interface

Each analog column has a dedicated comparator bus associated with it. Every analog PSoC block has a comparator output that can drive this bus. However, only one analog block in a column can actively drive the comparator bus for a column at any one time. The output on the comparator bus drives into the digital blocks as a data input. It also serves as an input to the decimator, as an interrupt input, and is available as read only data in the Analog Comparator Control register (CMP_CR0).

Figure 18-1 illustrates one column of the comparator bus. In the Continuous Time (CT) analog blocks, the CPhase and CLatch bits of CT Block 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 (SC) analog blocks, the output on the comparator bus is always latched. The ClockPhase bit in SC Block Control Register 0 determines the phase on which this data is latched and available.

The comparator bus is latched before it is available, to either drive the digital blocks, interrupt, decimator, or for it to be read in the CMP_CR0 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 CMP_CR0 register is described in the "CMP_CR0 Register" on page 227. There is also an option to force the latch in each column into a transparent mode by setting bits in the CMP_CR1 register.

As shown in Figure 18-1, the comparator bus output is gated by the primary output of a selected digital block. This feature is used to precisely control the integration period of an incremental ADC. Any digital block can be used to drive the gate signal. This selection may be made with the ICLKS bits in registers DEC_CR0 and DEC_CR1. This function may be enabled on a column-by-column basis, by setting the IGEN bits in the DEC_CR0 register.

The analog comparator bus output values can be modified or combined with another analog comparator bus through the Analog ***look-up table (LUT)*** function. The LUT takes two inputs, A and B, and provides a selection of 16 possible logic functions for those inputs. The LUT A and B inputs for each column comparator output is shown in the following table.

Table 18-1. A and B Inputs for Each Column Comparator LUT Output

Comparator LUT Output	A	B
4 Column PSoCs		
Column 0	ACMP0	ACMP1
Column 1	ACMP1	ACMP2
Column 2	ACMP2	ACMP3
Column 3	ACMP3	ACMP0
2 Column PSoCs		
Column 0	ACMP0	ACMP1
Column 1	ACMP1	0
Column 2	0	0
Column 3	0	ACMP0
1 Column PSoCs		
Column 0	ACMP0	0
Column 1	0	0
Column 2	0	0
Column 3	0	ACMP0

The LUT configuration is set in one control register, ALT_CR0. Each selection for each column is encoded in four bits. The function value corresponding to the bit encoding is shown in the following table.

Table 18-2. RDIxLTx Register

LUTx[3:0]	0h: 0000: FALSE 1h: 0001: A .AND. $\underline{B}$ 2h: 0010: A .AND. B 3h: 0011: $\underline{A}$ 4h: 0100: A .AND. B 5h: 0101: B 6h: 0110: A .XOR. B 7h: 0111: A .OR. B 8h: 1000: A .NOR. B 9h: 1001: $\underline{A}$ .XNOR. B Ah: 1010: B Bh: 1011: $\underline{A}$ .OR. $\overline{B}$ Ch: 1100: $\underline{A}$ Dh: 1101: A .OR. B Eh: 1110: A. NAND. B Fh: 1111: TRUE

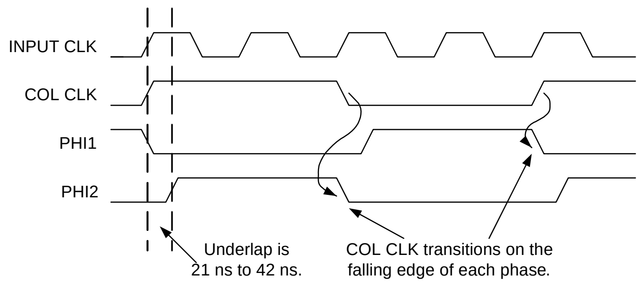
18.1.3 Analog Column Clock Generation

The analog array switched capacitor blocks require a two-phase, non-overlapping clock. The switched capacitor blocks are arranged two to a column (a third block in the column is a continuous time block).

An analog column clock generator is provided for each column and is shared among the blocks in that column. The input clock source for each column clock generator is selectable according to the CLK_CR0 register. It is important to note that regardless of the clock source selected, the output frequency of the column clock generator is the input frequency divided by four. There are four selections for each column: V1, V2, ACLK0, and ACLK1. The V1 and V2 clock signals are global system clocks. Programming options for these system clocks can be accessed in the OSC_CR1 register. Each of the ACLK0 and ACLK1 clock selections are driven by a selection of digital block outputs. The settings for the digital block selection are located in register CLK_CR1 .

The timing for analog column clock generation is shown in Figure 18-2. The dead band time between two phases of the clock is designed to be a minimum of 21 ns.

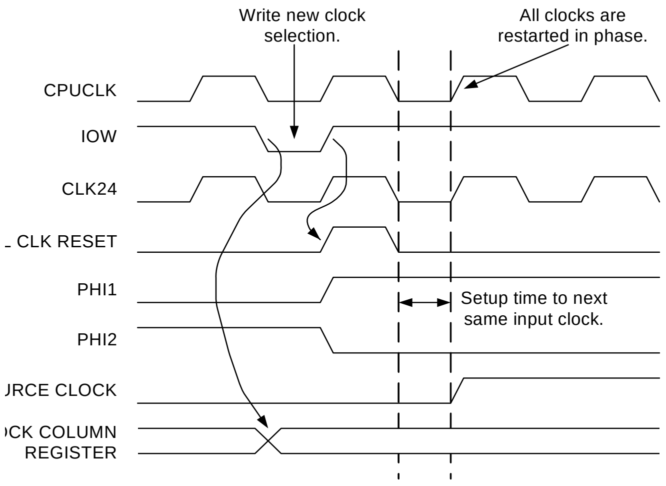
Figure 18-2. Two Phase Non-Overlapping Clock Generation


INPUT CLK
COL CLK
PHI1
PHI2
Underlap is 21 ns to 42 ns.
COL CLK transitions on the falling edge of each phase.

18.1.3.1 Column Clock Synchronization

When analog signals are routed between blocks in adjacent columns, it is important that the clocks in these columns are synchronized in phase and frequency. Frequency synchronization may be achieved by selecting the same input source to two columns. However, there is a special feature of the column clock interface logic that provides a resynchronization of clock phase. This function is activated on any IO write to either the Column Clock Selection register (CLK_CR0) or the Reference Calibration Clock register (RCL_CR). A write to either of these registers initiates a synchronous reset of the column clock generators, restarting all clocks to a known state. This action causes all columns with the same selected input frequency to be in phase. Writing these registers should be avoided during critical analog processing, as column clocks are all re-initialized and thus a discontinuity in PHI1/PHI2 clocking occurs.

Figure 18-3. Column Clock Resynchronize on an IO Write


Write new clock selection.
All clocks are restarted in phase.
CPUCLK
IOW
CLK24
CLK RESET
PHI1
Setup time to next same input clock.
PHI2
RCE CLOCK
CK COLUMN REGISTER

18.1.4 Decimator and Incremental ADC Interface

The Decimator and Incremental ADC Interface provides hardware support and signal routing for analog-to-digital conversion functions, specifically the Delta Signal ADC and the Incremental ADC. The control signals for this interface are split between two registers: DEC_CR0 and DEC_CR1.

18.1.4.1 Decimator

The Decimator is a hardware block used to perform digital processing on the analog block outputs.

The DCLKS0 and DCLKS1 bits, which are split between the DEC_CR0 and DEC_CR1 registers, are used to select a source for the decimator output latch enable. The decimator is typically run autonomously over a given period. The length of this period is set in a timer block that runs in conjunction with the analog processing. At the terminal count of this timer, the primary output goes high for one clock cycle. This pulse is translated into the decimator output latch enable signal, which transfers data from the internal accumulators to an output buffer. The terminal count also causes an interrupt and the CPU may read this output buffer at any time between one latch event and the next.

18.1.4.2 Incremental ADC

The analog interface has support for the incremental ADC operation through the ability to gate the analog comparator outputs. This gating function is required in order to precisely control the digital integration period that is performed in a digital block, as part of the function. A digital block pulse width modulator (PWM) is used as a source to provide the gate signal. Only one source for the gating signal can be selected. However, the gating can be applied independently to any of the column comparator outputs.

The ICLKS bits, which are split between the DEC_CR0 and DEC_CR1 registers, are used to select a source for the incremental gating signal. The four IGEN bits are used to independently enable the gating function on a column-by-column basis.

18.1.5 Analog Modulator Interface (Mod Bits)

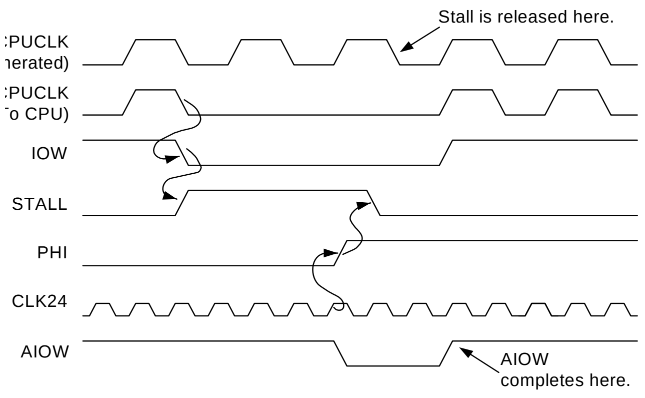
The Analog ***Modulator*** Interface provides a selection of signals that are routed to any of the four analog array ***modulation*** control signals. There is one modulation control signal for each Type C Analog Switched Capacitor block in every analog column. There are eight selections, which include the analog comparator bus outputs, two global outputs, and a digital block broadcast bus. The selections for all columns are identical and are contained in the AMD_CR0 and AMD_CR1 registers. The Mod bit is XOR'ed with the switched capacitor block ***sign bit*** (ASign in ASCxxCR0) to provide dynamic control of that bit.

18.1.6 Analog Synchronization Interface (Stalling)

For high precision analog operation, it is necessary to precisely time when updated register values are available to the analog PSOC blocks. The optimum time to update values in Switched Capacitor registers is at the beginning of the PHI1 active period. Depending on the relationship between the CPU CLK and the analog column clock, the CPU IO write cycle can occur at any 24 MHz master clock boundary in the PHI1 or PHI2 cycle. Register values may be written at arbitrary times; however, glitches may be apparent at analog outputs. This is because the capacitor value is changing when the circuit is designed to be settling.

The SYNCEN bit in the Analog Synchronization Control register (ASY_CR) is designed to address this problem. When the SYNCEN bit is set, an IO write instruction to any Switched Capacitor register is blocked at the interface and the CPU stalls. On the subsequent rising edge of PHI1, the CPU stall is released, allowing the IO write to be performed at the destination analog register. This mode synchronizes the IO write action to perform at the optimum point in the analog cycle, at the expense of CPU ***bandwidth***. Figure 18-4 shows the timing for this operation.

Figure 18-4. Synchronized Write to a DAC Register



As an alternative to stalling, the source for the analog column interrupts is set as the falling edge of the PHI2 clock. This configuration synchronizes the CPU to perform the IO write after the PHI2 phase is completed, which is equivalent to the start of PHI1.

18.2 PSoC Device Distinctions

The DEC_CR1 register's bit 7 (ECNT) is only available in PSoC devices with a type 1 decimator and is reserved in PSoC devices with a type 2 decimator.

18.3 Application Description

18.3.1 SAR Hardware Acceleration

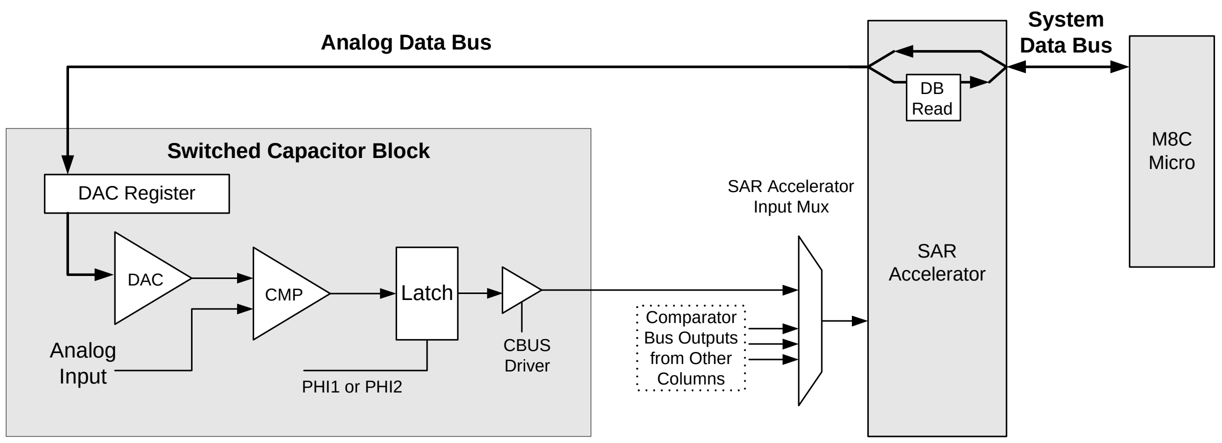
The Successive Approximation Register (SAR) ***algorithm*** is a binary search on the Digital-to-Analog Converter (DAC) code that best matches the input voltage being measured. The first step is to take an initial guess at mid-scale, which effectively splits the range by half. The DAC output value is then compared to the input voltage. If the guess is too low, a result bit is set for that binary position and the next guess is set at mid-scale of the remaining upper range. If the guess is too high, a result bit is cleared and the next guess is set at mid-scale of the remaining lower range. This process is repeated until all bits are tested. The resulting DAC code is the value that produces an output voltage closest to the input voltage. This code should be within one LSb of the input voltage.

The successive approximation analog-to-digital algorithm requires the following building blocks: a DAC, a comparator, and a method or apparatus to sequence successive writes to the DAC based on the comparator output. The SAR hardware accelerator represents a trade off between a fully automatic hardware sequencing approach and a pure firmware approach.

18.3.1.1 Architectural Description

The architectural description for the SAR hardware accelerator is illustrated in Figure 18-5.

Figure 18-5. SAR Hardware Accelerator



As shown in Figure 18-5, the SAR accelerator hardware is interfaced to the analog array through the comparator output and the analog array data bus. To create DAC output, values are written directly to the ACAP field in the DAC register. To facilitate the sequencing of the DAC writes in the SAR algorithm, the M8C is programmed to do a sequence of READ, MODIFY, and WRITE instructions. This is an atomic operation that consists of an IO read (IOR) followed closely by an IO write (IOW). One example of an assembly level instruction is as follows.

```
        OR reg[DAC_REG],0
```

The effect of this instruction is to read the DAC register and follow it closely in time by a write back. The OR instruction does not modify the read data (it is OR'ed with '0'). The CPU does not need to do any additional computation in conjunction with this procedure. The SAR hardware transparently does the data modification during the read portion of the cycle. The only purpose for executing this instruction is to initiate a read that is modified by the SAR hardware, then to follow up with a write that transfers the data back to the DAC register.

During each IO read operation, the SAR hardware overrides two bits of the data:

- To correct the previous bit guess based on the current comparator value.
- To set the next guess (next least significant bit).

The CPU latches this SAR modified data, OR's it with '0' (no CPU modification), and writes it back to the DAC register. A counter in the SAR hardware is used to decode which bits are being operated on in each cycle. In this way, the capability of the CPU and the IOR/IOW control lines are used to implement the read and write.

Use the SAR accelerator hardware to make the decisions and to control the values written, achieving the optimal level of performance for the current system.

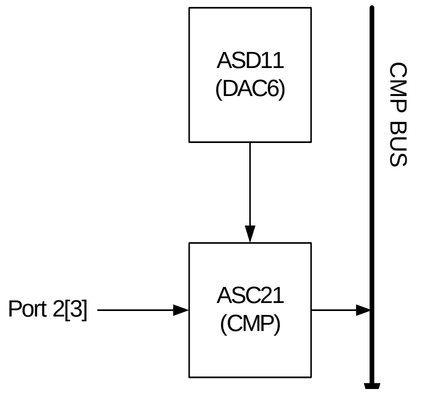
The SAR hardware is designed to process six bits of a result in a given sequence. A higher resolution SAR is implemented with multiple passes.

18.3.1.2 Application Description

There are a number of ways to map a SAR6 module into the analog array. A SAR6 can be created from 1 SC block, 2 SC blocks, or 1 SC block and 1 CT block. In the following example, the programming, the clock selection, connectivity, inputs, of a two block SAR6 is demonstrated.

This type of SAR6 is made up of 1 SC block that operates as a DAC6, and 1 SC block that operates as a voltage summer and comparator. The 2 block SAR6 is placed in column 1 as shown in Figure 18-6.

Figure 18-6. SAR6 Module Example



The programming for the DAC6 block is as follows:

```
CR0:  mov reg[ASD11CR0], a0h
        // Full Feedback, ACap Value = >
        // Start with Sign = 1
CR1:  mov reg[ASD11CR1], 40h
        // Select REFHI for DAC function
CR2:  mov reg[ASD11CR2], a0h
        // OBUS ON, Auto-Zero ON
CR3:  mov reg[ASD11CR3], 33h
        // Feedback ON, Power ON
```

The programming for the SUMMING/COMPARATOR block is as follows:

```
CR0:  mov reg[ASC21CR0], bfh
        // Full Feedback, Sign = 1, ACap = 31
CR1:  mov reg[ASC21CR1], 3fh
        // A Input = P2_3, BCap = 31
CR2:  mov reg[ASC21CR2], 60h
        // Cmp Bus ON, Auto Zero ON
CR3:  mov reg[ASC21CR3], 17h
        // Feedback OFF, B Input = North
```

Firmware Support Examples

In addition to the use of the OR instruction to sequence the algorithm, there are some minimal set up requirements. The SAR control bits are in the ASY_CR register. The definition of these bits as related to the SAR are as follows.

Bits [2:1] Column Select for the SAR Comparator Input

The DAC portion of the SAR can reside in any of the appropriate positions in the analog PSoC block array. However, once the COMPARATOR block is positioned (and it is possible to have the DAC and COMPARATOR in the same block), this should be the column selected.

Bit [3] Sign Selection

This bit optionally inverts the comparator input to the SAR accelerator. It must be set based on the type of PSoC block configuration selected. Some typical examples are listed in Table 18-3.

Table 18-3. Example SAR Configuration

Configuration	Description	Sign
SAR6 – 2 block	1 DAC6, 1 COMP (could be CT)	0
SAR6 – 1 block	1 for both DAC6 and COMP	1
MS SAR10 – 3 blocks	1 DAC9, 1 COMP (could be CT) (when processing MS DAC block)	0
LS SAR10 – 3 blocks	1 DAC9, 1 COMP (could be CT) (when processing LS DAC block)	1

Bits [6:4] SAR Count Value

These three bits are used to initialize a 3-bit counter to sequence the 6 bits of the SAR algorithm. Typically, the user initializesx this register to ‘6’. When these bits are any value other than ‘0’, an IOR command to an SC block is assumed to be part of a SAR sequence.

Assuming the comparator bus output is programmed for column 0, a typical firmware sequence is as follows.

```
mov  reg[ASY_CR], 60h // SAR count value=6,
                      // Sign=0, Col=0
or reg[ASD11CR0], 0 // Check sign, set bit 4
or reg[ASD11CR0], 0 // Check bit 4, set bit 3
or reg[ASD11CR0], 0 // Check bit 3, set bit 2
or reg[ASD11CR0], 0 // Check bit 2, set bit 1
or reg[ASD11CR0], 0 // Check bit 1, set bit 0
or reg[ASD11CR0], 0 // Check bit 0
```

SAR6 Calculation Example

This example assumes an input voltage level (VIn) of 3.0V on the PSoC input pin. The selection is made of ± VREF for the DAC references. Assuming VREF = 1.25, the input range is from 1.25 to 3.75 volts. The 6-bit DAC yields a sign magnitude result with 64 discrete values, thus giving 39 mV of resolution over the input range.

With 3.0V input, the expected magnitude of the result is (3.0-2.5)/1.25 * 32 = 12.8. The expected sign of the result is ‘0’, meaning positive; therefore, the result is Sign=0, Magnitude=12 or 13. The error in this basic SAR algorithm is always less than one LSb in the final result.

Table 18-4 shows the sequence of calculations which correspond to the six OR instructions.

The final result of the computation is:

Sign = 1 and Magnitude = 011000 or 12.

To represent the true sign of the input voltage, you must invert the sign of the result from the DAC register. Therefore, the result becomes Sign = 0, Magnitude = 12, which is (3.75 – 2.5)/32 * 12 + 2.5 = 2.96875. The error is 31.25 mV, or less that one LSb of 39 mV.

Table 18-4. SAR Sequence Example

Step	Current ACap	VIn	VDac	VSum	Comparator Bus (CMP)	New ACap	Comment
1	100000	3.0	2.5	2.75	0	110000	**Keep** the sign bit and set bit 4.
2	110000	3.0	1.875	2.4375	1	101000	**Overshoot**, clear bit 4, set bit 3.
3	101000	3.0	2.1875	2.59375	0	101100	**Keep** bit 3, and set bit 2.
4	101100	3.0	2.03125	2.515625	0	101110	**Keep** bit 2, and set bit 1.
5	101110	3.0	1.953125	2.4765625	1	101101	**Overshoot**, clear bit 1, set bit 0.
6	101101	3.0	1.992188	2.496094	1	101100	**Overshoot**, clear bit 0
	101100	3.0	2.03125	2.515625	0	101100	Final Result

Notes

1. VSum is the voltage at the summing node, that is, the input to the comparator.
2. VDac is the voltage generated by the DAC block from the ACap value.
3. When VSum > AGND, CMP = 0; when VSum < AGND, CMP = 1.
4. CMP = 0 means keep the bit (undershoot); CMP = 1 means clear the bit (overshoot).
5. Start with Sign = 1 (configuration programming), equivalent to setting that bit to test.

As shown in Table 18-4, the value of the result from Step 5, Magnitude = 13, is closer to the actual value of 12.8. This demonstrates that even though it is possible that the resulting code could be closer to the actual value, in the SAR algorithm there is no provision to detect this. The result is a maximum theoretical error of less than one LSb.

Implementing Higher Resolution SARs

It is straightforward to implement higher resolution SARs using the SAR hardware accelerator. For example, to create an 11-bit SAR, 3 blocks are allocated: 2 SC blocks to make a DAC9 and one SC or CT block for summing and compare.

To get the results of the most significant (MS) block, which is the first 6 bits (Sign and 5 bits of Magnitude), the firmware sequencing proceeds exactly as in the previous SAR6 example.

The trick with the least significant (LS) block of the DAC9 is to get the sign right. For the output to be correct, the sign of the LS block of a DAC9 should be opposite to that of the MS block (since it is connected through an inverting input to the MS block).

There are two possible ways to handle this.

1. In firmware, one can manually compute what the SIGN bit should be from the result in the MS block and write it to the LS block. Then the SAR count value should be set to 5 instead of 6 to skip the sign bit check.
2. An interesting property of the SAR algorithm is that the resulting voltage at the summing node after the first 6 steps (MS block processing) is going to be the same polarity (above or below AGND) as the input voltage. The reason for this is that, by definition, if the polarity of the summing voltage is opposite to that of the input voltage, this triggers a clear of the previous bit set. Since, also by definition, the final result of the summing voltage is less than one LSb from AGND, clearing the LSb results in a summing voltage of the same polarity as the input voltage.

According to number 2 above, the sign bit of the LS block can be handled exactly as the SIGN bit of the MS block, just another OR instruction. This sequence is then appended on the above MS processing sequence (substituting the LS DAC block address for <LS_CR0>). Note that the meaning of the comparator is inverted by setting the SIGN bit in the ASYNC Control register. This is because the LS block is inverted with respect to the MS block.

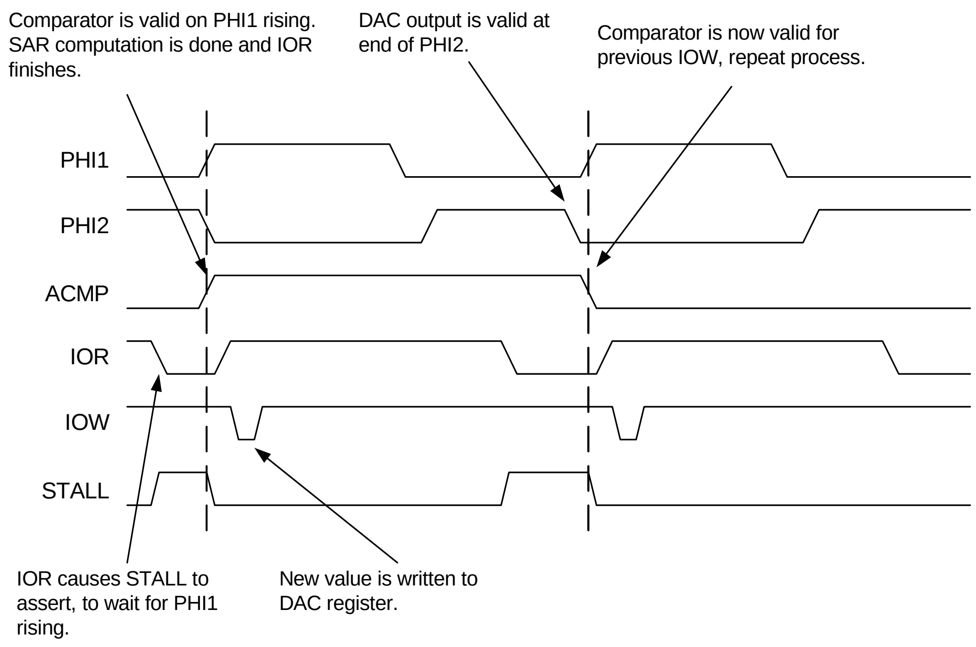
```
mov  reg[ASY_CR], 68h// SAR count value=6,
                     // Sign=1, Col=0

or reg[<LS_CR0>], 0  // Check sign, set bit 4
or reg[<LS_CR0>], 0  // Check bit 4, set bit 3
or reg[<LS_CR0>], 0  // Check bit 3, set bit 2
or reg[<LS_CR0>], 0  // Check bit 2, set bit 1
or reg[<LS_CR0>], 0  // Check bit 1, set bit 0
or reg[<LS_CR0>], 0  // Check bit 0
```

18.3.1.3 SAR Timing

Another important function of the SAR hardware is to synchronize the IO read (the point at which the comparator value is used to make the SAR decision) to when the analog comparator bus is valid. Under normal conditions, this point is at the rising edge of PHI1 for the previous compute cycle. When the OR instruction is executed in the CPU, a few CPU clocks cycle into the instruction and an IOR signal is asserted to initiate a read of the DAC register. The SAR hardware then stalls the CPU clock, for one 24 MHz clock cycle after the rising edge of PHI1. When the stall is released, the IO read completes and is immediately followed by an IO write. In this sequence of events, the DAC register is written with the new value within a few CPU clocks after PHI1.

The rising edge of PHI1 is also the optimal time to write the DAC register for maximum settling time. The timing from the positive edge of PHI1 to the start of the IO write is 4.5 clocks, which at 24 MHz is 189 ns. If the analog clock is running at 1 MHz, this allows over 300 ns for the DAC output and comparator to settle.

Figure 18-7. General SAR Timing


Comparator is valid on PHI1 rising. SAR computation is done and IOR finishes.
DAC output is valid at end of PHI2.
Comparator is now valid for previous IOW, repeat process.
PHI1
PHI2
ACMP
IOR
IOW
STALL
IOR causes STALL to assert, to wait for PHI1 rising.
New value is written to DAC register.

18.4 Register Definitions

The following registers are associated with the Analog Interface and are listed in address order. Each register description has an associated register table showing the bit structure for that register. For a complete table of analog interface registers, refer to the “Summary Table of the Analog Registers” on page 217.

The bits that are grayed out throughout this manual are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of ‘0’.

18.4.1 CMP_CR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,64h	CMP_CR0			COMP[1:0]				AINT[1:0]		# : 00

#: Access is bit specific. Refer to the Register Details chapter on page 47.

The Analog Comparator Bus Register 0 (CMP_CR0) is used to poll the analog column comparator bits and select column interrupts.

This register contains two fields: COMP and AINT. By default, the interrupt is the comparator bit. A rising edge on a comparator bit causes an interrupt to be registered. However, if a bit in this field is set, the interrupt input for that column is derived from the falling edge of PHI2 clock for that column (that is, the falling edge of PHI2 leaves a rising interrupt signal). Firmware can use this capability to synchronize to the current column clock.

Bits 7 to 4: COMP[x]. These bits are the read only bits corresponding to the comparator bits in each analog column. They are synchronized to the column clock, and thus may be reliably polled by the CPU.

Bits 3 to 0: AINT[x]. These bits select the interrupt source for each column, as the input to the interrupt controller.

For additional information, refer to the CMP_CR0 register on page 66.

18.4.2 ASY_CR Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,65h	ASY_CR			SARCNT[2:0]		SARSIGN	SARCOL[1:0]		SYNCEN	RW : 00

The Analog Synchronization Control Register (ASY_CR) is used to control SAR operation, except for bit 0, SYNCEN.

SYNCEN is associated with analog register write stalling and is described in “Analog Synchronization Interface (Stalling)” on page 222.

The SAR hardware accelerator is a block of specialized hardware designed to sequence the SAR algorithm for efficient analog-to-digital conversion. A SAR ADC is implemented conceptually with a DAC of the desired precision and a comparator. This functionality is configured from one or more PSoC blocks. For each conversion, the firmware initializes the ASY_CR register and sets the SIGN bit of the DAC as the first guess in the algorithm. A sequence of OR instructions (read, modify, write) to the ASxxxCR0 register is then executed. Each of these OR instructions causes the SAR hardware to read the current state of the comparator, checking the validity of the previous guess. It either clears it or leaves it set, accordingly. The next LSb in the DAC register is also set as the next guess. Six OR instructions complete the conversion of a 6-bit DAC.

The resulting DAC code, which matches the input voltage to within one LSb, is then read back from the ASxxxCR0 register.

Bits 6 to 4: SARCNT[2:0]. These bits are the SAR count value and are used to initialize a three-bit counter to sequence the six bits of the SAR algorithm. Typically, the user initializes this register to ‘6’. When these bits are any value other than ‘0’, a register read command to an SC block is assumed to be part of a SAR sequence.

Assuming the comparator bus output is programmed for column 0, a typical firmware sequence is as follows.

```
mov  reg[ASY_CR], 60h  // SAR count value=6,
                       // Sign=0, Col=0
or reg[ASD11CR0], 0    // Check sign, set bit 4
or reg[ASD11CR0], 0    // Check bit 4, set bit 3
or reg[ASD11CR0], 0    // Check bit 3, set bit 2
or reg[ASD11CR0], 0    // Check bit 2, set bit 1
or reg[ASD11CR0], 0    // Check bit 1, set bit 0
or reg[ASD11CR0], 0    // Check bit 0
```

Bit 3: SARSIGN. This bit is the SAR sign selection and optionally inverts the comparator input to the SAR accelerator. It must be set based on the type of PSoC block configuration selected. Table 18-5 lists some typical examples.

Table 18-5. Typical PSoC Block Configurations

Configuration	Description	Sign
SAR6 – 2 blocks	1 DAC6, 1 COMP (could be CT)	0
SAR6 – 1 block	DAC6 and COMP in 1 block	1
MS SAR10 – 3 blocks	1 DAC9, 1 COMP (could be CT) (when processing MS DAC block)	0

Bits 2 and 1: SARCOL[1:0]. These bits are the column select for the SAR comparator input. The DAC portion of the SAR can reside in any of the appropriate positions in the analog PSoC block array. However, once the COMPARATOR block is positioned (and it is possible to have the DAC and COMPARATOR in the same block), this position should be the column selected.

Bit 0: SYNCEN. This bit is to synchronize CPU data writes to Switched Capacitor (SC) block operation in the analog array. The SC block clock is selected in the CLK_CR0 register. The selected clock source is divided by four and the output is a pair of two-phase, non-overlapping clocks: PHI1 and PHI2. There is an optimal time, with respect to the PHI1 and PHI2 clocks, to change the capacitor configuration in the SC block, which is typically the rising edge of PHI1. This is normally the time when the input branch capacitor is charging.

When this bit is set, any write to an SC block register is stalled until the rising edge of the next PHI1 clock phase, for the column associated with the SC block address. The stalling operation is implemented by suspending the CPU clock. No CPU activity occurs during the stall, including interrupt processing. Therefore, the effect of stalling on CPU throughput must be considered.

For additional information, refer to the ASY_CR register on page 67.

18.4.3 CMP_CR1 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,66h	CMP_CR1			CLDIS[1]	CLDIS[0]					RW : 00

The Analog Comparator Bus Register 1 (CMP_CR1) is used to override the analog column comparator synchronization.

Bits 5 to 4: CLDIS[x]. When these bits are set, the given column is not synchronized to PHI2 in the analog interface.

This capability is typically used to allow a continuous time comparator result to propagate directly to the interrupt controller during sleep. Since the master clocks (except the 32 kHz clock) are turned off during sleep, the synchronizer must be bypassed.

For additional information, refer to the CMP_CR1 register on page 68.

18.4.4 DEC_CR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E6h	DEC_CR0			IGEN[1:0]		ICLKS0	DCOL[1:0]		DCLKS0	RW : 00

The Decimator Control Register 0 (DEC_CR0) contains control bits to access hardware support for both the Incremental ADC and the DELISG ADC.

Bits 5 to 4: IGEN[1:0]. For incremental support, these bits select which column comparator bit is gated by the output of a digital block. The output of that digital block is typically a PWM signal; the high time of which corresponds to the ADC conversion period. This ensures that the comparator output is only processed for the precise conversion time. The digital block selected for the gating function is controlled by ICLKS0 in this register, and ICLKS3, ICLKS2 and ICLKS1 bits in the DEC_CR1 register.

Bit 3: ICLKS0. In conjunction with ICLKS1, ICLKS2, and ICLKS3 in the DEC_CR1 register, these bits select up to one of 16 digital blocks (depending on the PSoC device resources) to provide the gating signal for an incremental ADC conversion.

Bits 2 and 1: DCOL[1:0]. The DELSIG ADC uses the hardware decimator to do a portion of the post processing computation on the comparator signal. DCOL[1:0] selects the column source for the decimator data (comparator bit) and clock input (PHI clocks).

Bit 0: DCLKS0. The decimator requires a timer signal to sample the current decimator value to an output register that may subsequently be read by the CPU. This timer period is set to be a function of the DELSIG conversion time and may be selected from up to one of eight digital blocks (depending on the PSoC device resources) with DCLKS0 in this register and DCLKS3, DCLKS2, and DCLKS1 in the DEC_CR1 register.

For additional information, refer to the DEC_CR0 register on page 110.

18.4.5 DEC_CR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E7h	DEC_CR1	ECNT	IDEC	ICLKS3	ICLKS2	ICLKS1	DCLKS3	DCLKS2	DCLKS1	RW : 00

The Decimator Control Register 1 (DEC_CR1) is used to configure the decimator prior to using it.

Bit 7: ECNT. The ECNT bit is a mode bit that controls the operation of the decimator hardware block. By default, the decimator is set to a double integrate function, for use in hardware DELSIG processing. When the ECNT bit is set, the decimator block converts to a single integrate function. This gives the equivalent of a 16-bit counter suitable for use in hardware support for an Incremental ADC function.

The DEC_CR1 register's bit 7 (ECNT) is only available in PSoC devices with a type 1 decimator and is reserved in PSoC devices with a type 2 decimator.

Bit 6: IDEC. Any function using the decimator requires a digital block timer to sample the current decimator value. Normally, the positive edge of this signal causes the decimator output to be sampled. However, when the IDEC bit is set, the negative edge of the selected digital block input causes the decimator value to be sampled.

Bits 5 to 0: ICLKSx and DCLKSx. The ICLKS3, ICLKS2, ICLKS1, DCLKS3, DCLKS2, and DCLKS1 bits in this register select the digital block sources for Incremental and DELSIGN ADC hardware support (see the DEC_CR0 register).

For additional information, refer to the DEC_CR1 register on page 111.

18.4.6 CLK_CR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,60h	CLK_CR0					AColumn1[1:0]		AColumn0[1:0]		RW : 0

The Analog Clock Source Control Register 0 (CLK_CR0) is used to select the clock source for an individual analog column.

An analog column clock generator is provided for each column. The bits in this register select the source for each column clock generator. Regardless of the source selected, the input clock is divided by four to generate the PHI1/PHI2 non-overlapping clocks for the column.

There are four selections for each clock: VC1, VC2, ACLK0, and ACLK1. VC1 and VC2 are the programmable global system clocks. ACLK0 and ACLK1 sources are each selected from one up to eight digital block outputs (functioning as clock generators).

Bits 3 and 2: AColumn1[1:0]. These bits select the source for analog column 1.

Bits 1 and 0: AColumn0[1:0]. These bits select the source for analog column 0.

For additional information, refer to the CLK_CR0 register on page 133.

18.4.7 CLK_CR1 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,61h	CLK_CR1		SHDIS	ACLK1[2:0]			ACLK0[2:0]			RW : 00

The Analog Clock Source Control Register 1 (CLK_CR1) is used to select the clock source for an individual analog column.

Bit 6: SHDIS. The SHDIS bit functions as follows. 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).

The following are the exceptions: 1) If the ClockPhase bit in ASCxx_CR0 (for the SC block in question) is set to ‘1’, then the output is enabled if the analog bus output is enabled during both PHI1 and PHI2. 2) If the SHDIS signal is set in bit 6 of the Analog Clock Source Control register, then sample and hold operation is disabled for all columns and all enabled outputs of SC blocks are connected to their respective output buses, for the entire period of their respective PHI2s.

Bits 5 to 0: ACLKx[2:0]. There are two 3-bit fields in this register that can select up to one of eight digital blocks (depending on the PSoC device resources), to function as the clock source for ACLK0 and ACLK1. ACLK0 and ACLK1 are alternative clock inputs to the analog column clock generators (see the CLK_CR0 register above).

For additional information, refer to the CLK_CR1 register on page 134.

18.4.8 AMD_CR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,63h	AMD_CR0						AMOD0[2:0]			RW : 00

The Analog Modulation Control Register 0 (AMD_CR0) is used to select the modulator bits used with each column.

The MODBIT is an input into a Switched Capacitor C Type block only and is XOR'ed with the currently programmed value of the ASIGN bit in the CR0 register for that SC block. This allows the ACAP sign bit to be dynamically modulated by hardware signals. Three bits for each column allow a one of eight selection for the MODBIT. Sources include any of the analog column comparator buses, two global buses, and one broadcast bus. The default for this function is zero or off.

Bits 2 to 0: AMOD0[2:0]. These bits control the selection of the MODBITs for analog column 0.

For additional information, refer to the AMD_CR0 register on page 136.

18.4.9 AMD_CR1 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,66h	AMD_CR1						AMOD1[2:0]			RW : 00

The Analog Modulation Control Register 1 (AMD_CR1) is used to select the modulator bits used with each column.

The MODBIT is an input into a Switched Capacitor Type C block only and is XOR'ed with the currently programmed value of the ASIGN bit in the CR0 register for that SC block. This allows the ACAP sign bit to be dynamically modulated by hardware signals. Three bits for each column allow a one of eight selection for the MODBIT. Sources include any of the analog column comparator buses, two global buses, and one broadcast bus. The default for this function is zero or off.

Bits 2 to 0: AMOD1[2:0]. These bits control the selection of the MODBITs for analog column 1.

For additional information, refer to the AMD_CR1 register on page 137.

18.4.10 ALT_CR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,67h	ALT_CR0	LUT1[3:0]				LUT0[3:0]				RW : 00

The Analog LUT Control Register 0 (ALT_CR0) is used to select the logic function.

A one of 16 look-up table (LUT) is applied to the outputs of each column comparator bit and optionally a neighbor bit to implement two input logic functions.

Table 18-1 shows the available functions, where the A input applies to the selected column and the B input applies to the next most significant neighbor column. Column 0 settings apply to combinations of column 0 and column 1. Column 1 settings apply to combinations of column 1 and column 2, where B=0 for one column PSoC devices.

Bits 7 to 4: LUT1[3:0]. These bits control the selection of the LUT 1 logic functions that may be selected for the analog comparator bits in column 0 and column 1.

Bits 3 to 0: LUT0[3:0]. These bits control the selection of LUT 0 logic functions that may be selected for the analog comparator bits in column 0 and column 1.

For additional information, refer to the ALT_CR0 register on page 138.

19. Analog Array

This chapter presents the Analog Array, which has no registers directly associated with it. This chapter is important because it discusses the block and column level interconnects that exist in the analog PSoC array.

19.1 Architectural Description

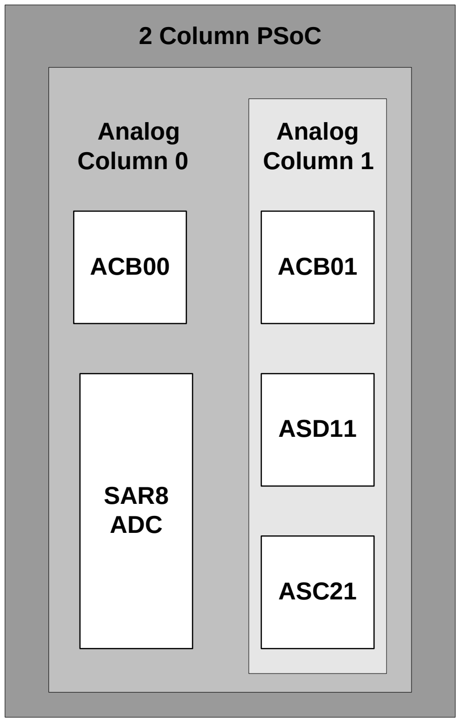
The analog array is designed to allow interaction between PSoC devices without modifying projects, except for resource limitations.

The figures that follow illustrate the analog multiplexer (mux) connections for the various PSoC devices, which vary depending on column availability.

Figure 19-1 displays the various analog arrays for the CY8C24533, CY8C23533, CY8C23433CY8C24633 PSoC device family. Analog column 1 has 3 analog blocks associated with it. Analog column 0 has 1 analog block and 1 SAR8 ADC block associated with it.

In the figures throughout this chapter, shading and call outs portray the different column configurations that are available in a PSoC device.

Figure 19-1. Array of Analog PSoC Blocks for CY8C24533, CY8C23533, CY8C23433CY8C24633



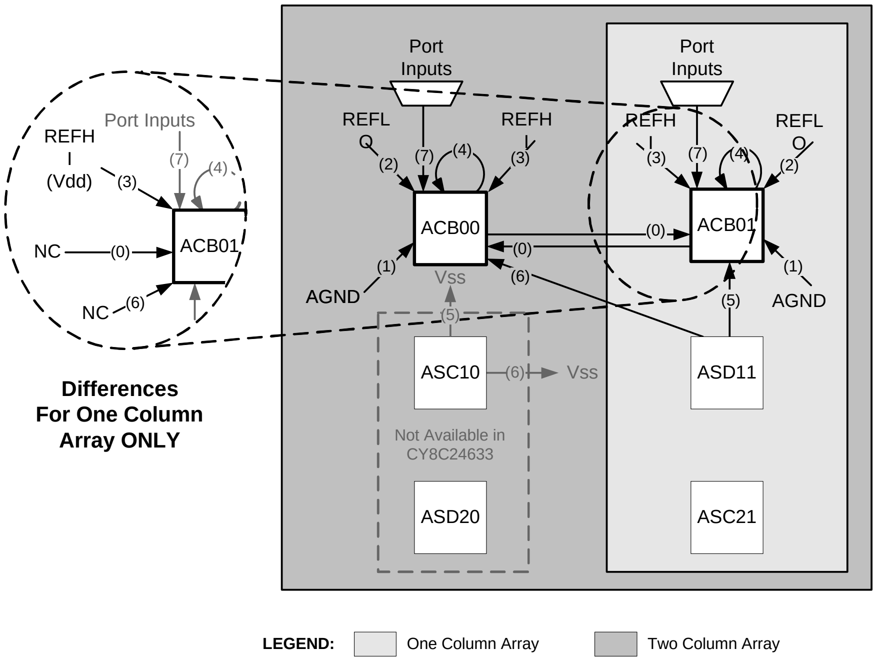
19.1.1 NMux Connections General Overview

The NMux is an 8-to-1 mux, which determines the source for the inverting (also called negative) input of Continuous Time PSoC blocks. These blocks are named ACB00 and ACB01. More details on the Continuous Time PSoC blocks are available in the chapter Continuous Time PSoC Block, on page 249. The NMux connections are described in detail in the ACBxxCR1 register on page 76, bits NMux[2:0].

The numbers in Figure 19-2, which are associated with each arrow, are the corresponding NMux select line values for the data in the NMux portion of the register. The call out names in the figure show nets selected for each NMux value.

For one column PSoC devices, the figure view is expanded in a circular area to the left of the main diagram, where black call outs and arrows signify exclusive one column functionality and gray call outs and arrows signify commonality with four and two column PSoC devices.

Figure 19-2. NMux Connections



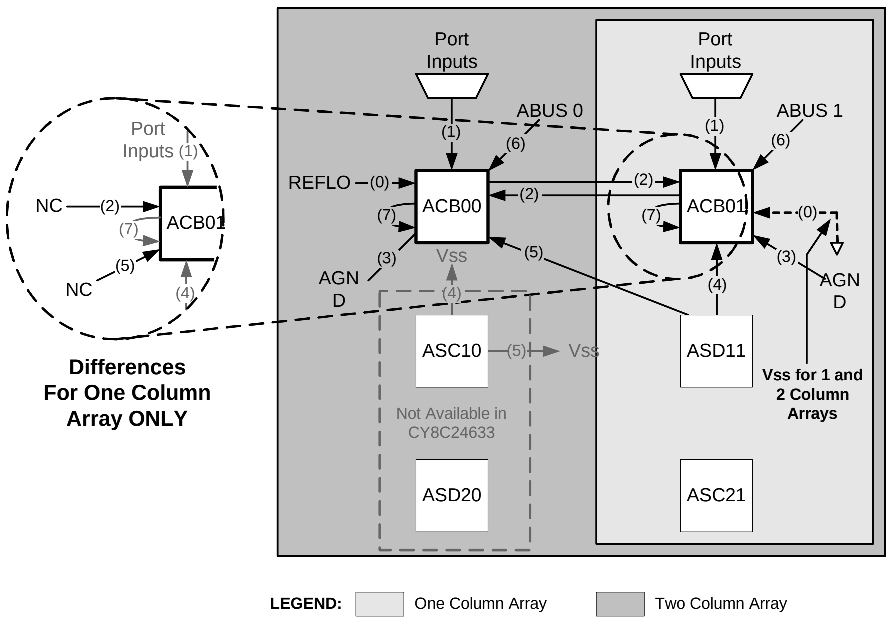
19.1.2 PMux Connections General Overview

The PMux is an 8-to-1 mux, which determines the source for the non-inverting (also called positive) input of Continuous Time PSoC blocks. These blocks are named ACB00 and ACB01. More details on the Continuous Time PSoC blocks are available in the chapter Continuous Time PSoC Block, on page 249. The PMux connections are described in detail in the ACBxxCR1 register on page 76, bits PMux[2:0].

The numbers in Figure 19-3, which are associated with each arrow, are the corresponding PMux select line values for the data in the PMux portion of the register. The call out names in the figure show nets selected for each PMux value.

For one column PSoC devices, the figure view is expanded in a circular area to the left of the main diagram, where black call outs and arrows signify exclusive one column functionality, and gray call outs and arrows signify commonality with four and two column PSoC devices.

Figure 19-3. PMux Connections



19.1.3 RBotMux Connections General Overview

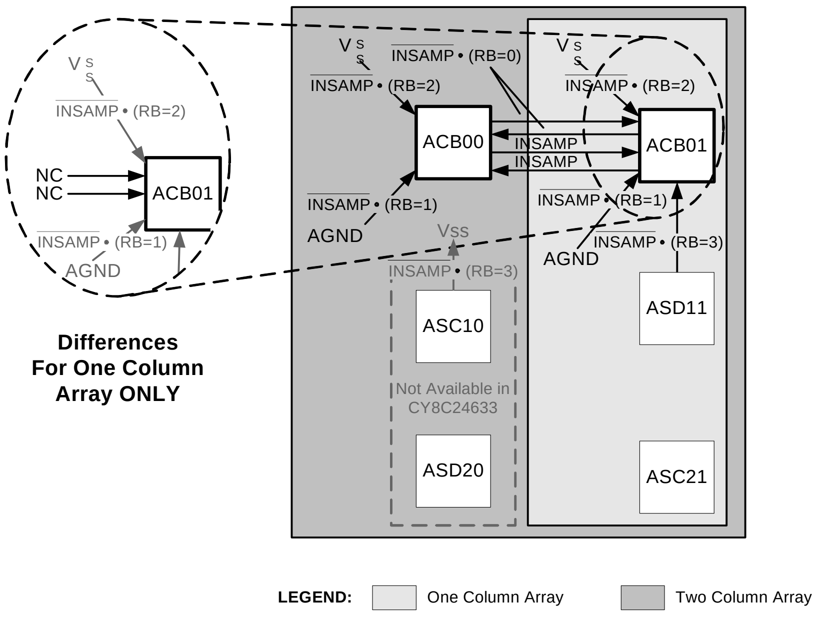
The RBotMux connections in the figure below are the mux inputs for the bottom of the resistor string, see Figure 22-1 on page 250. The RBotMux connections are used in the Continuous Time PSoC blocks. These blocks are named ACB00 and ACB013. The RBotMux connections are described in detail in the ACBxxCR0 register on page 74, bits RBotMux[1:0].

The numbers in Figure 19-4, which are associated with each arrow, are the corresponding RBotMux select line values for the data in the RBotMux portion of the register. The call out names in the figure show nets selected for each RBotMux value.

The logic statements in Figure 19-4 are the RBotMux connections that are selected by the combination of the RBotMux bits (ACB0xCR0 bits 1 and 0) and the INSAMP bit (ACB0xCR3 bit 1). For example, the RBotMux selects a connection to AGND, if the INSAMP bit is low and the RBotMux bits are 01b. This is shown in the figure as the logic statement $INSAMP \cdot (RB = 1)$.

For one column PSoC devices, the figure view is expanded in a circular area to the left of the main diagram, where black call outs and arrows signify exclusive one column functionality, and gray call outs and arrows signify commonality with four and two column PSoC devices.

Figure 19-4. RBotMux Connections



Vss
$\overline{INSAMP}$ • (RB=2)
NC
NC
ACB01
$\overline{INSAMP}$ • (RB=1)
AGND
Differences For One Column Array ONLY
Vss
$\overline{INSAMP}$ • (RB=0)
$\overline{INSAMP}$ • (RB=2)
ACB00
INSAMP
INSAMP
ACB01
$\overline{INSAMP}$ • (RB=1)
AGND
Vss
$\overline{INSAMP}$ • (RB=3)
ASC10
Not Available in CY8C24633
ASD20
$\overline{INSAMP}$ • (RB=1)
$\overline{INSAMP}$ • (RB=3)
AGND
ASD11
ASC21
LEGEND:
One Column Array
Two Column Array

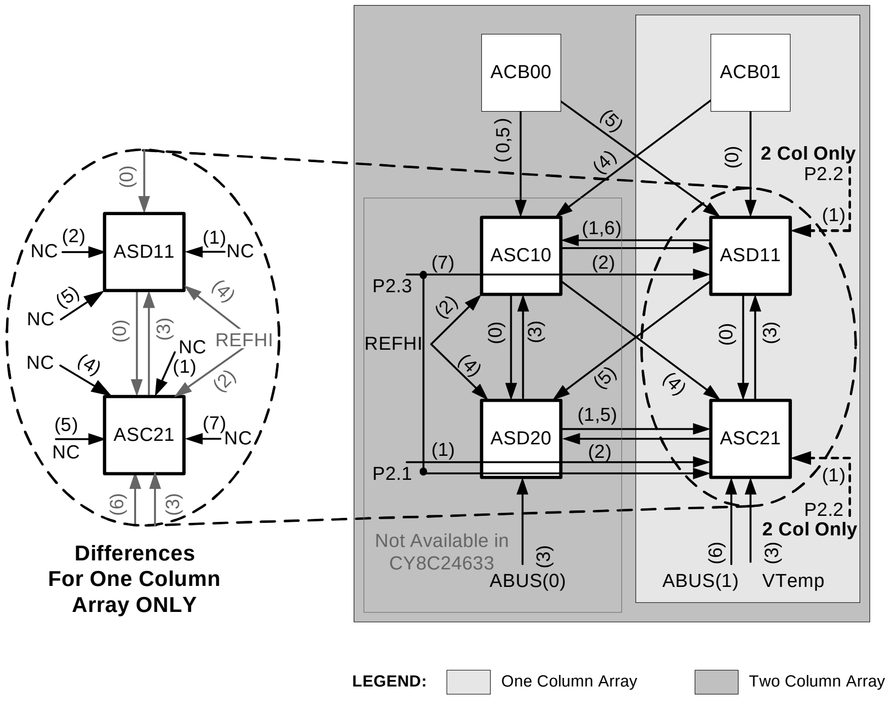
19.1.4 AMux Connections General Overview

The AMux connections in the figure below are the mux inputs for controlling both the A and C capacitor branches. The high order bit, ACMux[2], selects one of two inputs for the C branch, which is used to control both the AMux and CMux. (See the A inputs in Figure 23-1 on page 256 and Figure 23-2 on page 257.) The AMux connections are used in the Switched Capacitor PSoC blocks. These blocks are named ASC10, ASD11, ASD20, and ASC21. The AMux connections are described in detail in the ASCxxCR1 register on page 84, bits ACMux[2:0], and ASDxxCR1 register on page 80, bits AMux[2:0].

The numbers in Figure 19-5, which are associated with each arrow, are the corresponding AMux select line values for the data in the ACMux portion of the register. The call out names in the figure show nets selected for each AMux value.

For one column PSoC devices, the figure view is expanded in a circular area to the left of the main diagram, where black call outs and arrows signify exclusive one column functionality, and gray call outs and arrows signify commonality with four and two column PSoC devices.

Figure 19-5. AMux Connections

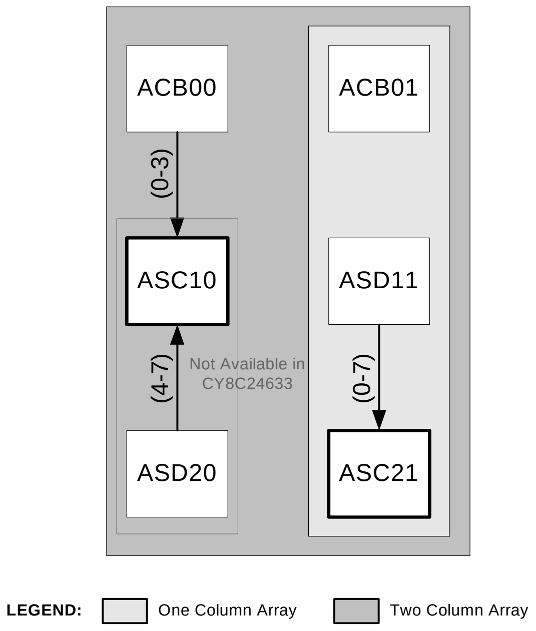


19.1.5 CMux Connections General Overview

The CMux connections in the figure below are the mux inputs for controlling the C capacitor branches. The high order bit, ACMux[2], selects one of two inputs for the C branch, which is used to control both the AMux and CMux. (See the C inputs in Figure 23-1 on page 256.) The CMux connections are used in the Switched Capacitor PSoC blocks. These blocks are named ASC10 and ASC21.

The CMux connections are described in detail in the ASCxxCR1 register on page 84, bits ACMux[2:0]. The numbers in the figure, which are associated with each arrow, are the corresponding CMux select line values for the data in the CMux portion of the register. The call out names in the figure show nets selected for each CMux value.

Figure 19-6. CMux Connections



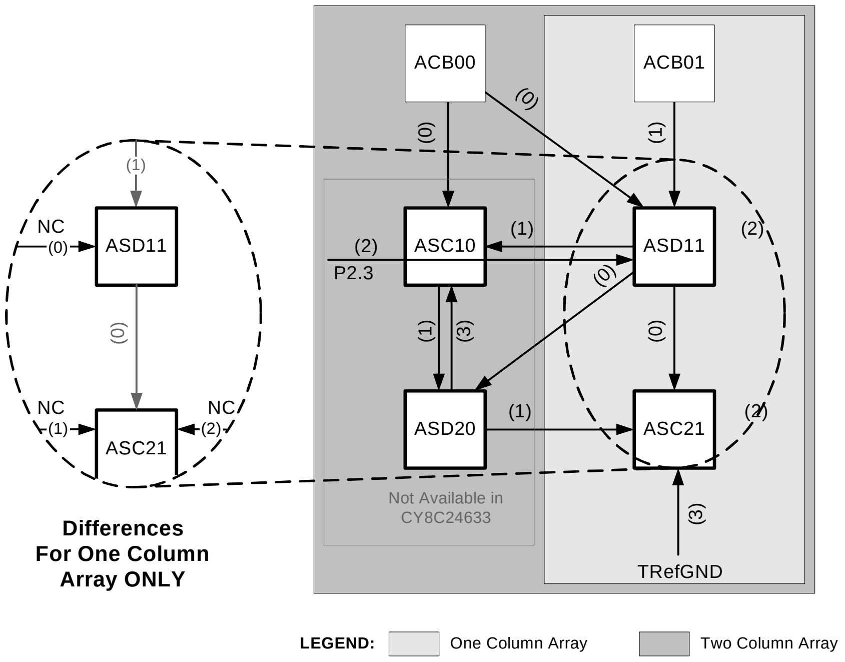
19.1.6 BMux SC/SD Connections General Overview

The BMux SC/SD connections in the figure below are the mux inputs for controlling the B capacitor branches. (See Figure 23-1 on page 256 and Figure 23-2 on page 257.) The BMux SC/SD connections are used in the Switched Capacitor PSoC blocks. These blocks are named ASC10, ASD11, ASD20, and ASC21. The BMux connections are described in detail in the ASCxxCR3 register on page 86, bits BMuxSC[1:0], and ASDxxCR3 register on page 82, bit BMuxSD[2].

The numbers in Figure 19-7, which are associated with each arrow, are the corresponding BMux select line values for the data in the BMux portion of the register. The call out names in the figure show nets selected for each BMux value.

For one column PSoC devices, the figure view is expanded in a circular area to the left of the main diagram, where black call outs and arrows signify exclusive one column functionality, and gray call outs and arrows signify commonality with four and two column PSoC devices.

Figure 19-7. BMux SC/SD Connections



19.1.7 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. However, 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. Refer to the "Analog Comparator Bus Interface" on page 220 in the Analog Interface chapter for more information.

19.2 Temperature Sensing Capability

A temperature-sensitive voltage, derived from the bandgap sensing on the die, is buffered and available as an analog input into the Analog Switch Cap Type C block, ASC21. Temperature sensing allows protection of device operating ranges for fail-safe applications. Temperature sensing, combined with a long sleep timer interval (to allow the die to approximate ***ambient temperature***), gives an approximate ambient temperature for data acquisition and battery charging applications. The user may also calibrate the internal temperature rise based on a known current consumption. The temperature sensor input to the ASC21 block is labeled VTemp and its associated ground reference is labeled TRefGND.

20. Analog Input Configuration

This chapter discusses the Analog Input Configuration and its associated registers. For a complete table of analog input configuration registers, refer to the "Summary Table of the Analog Registers" on page 217. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

20.1 Architectural Description

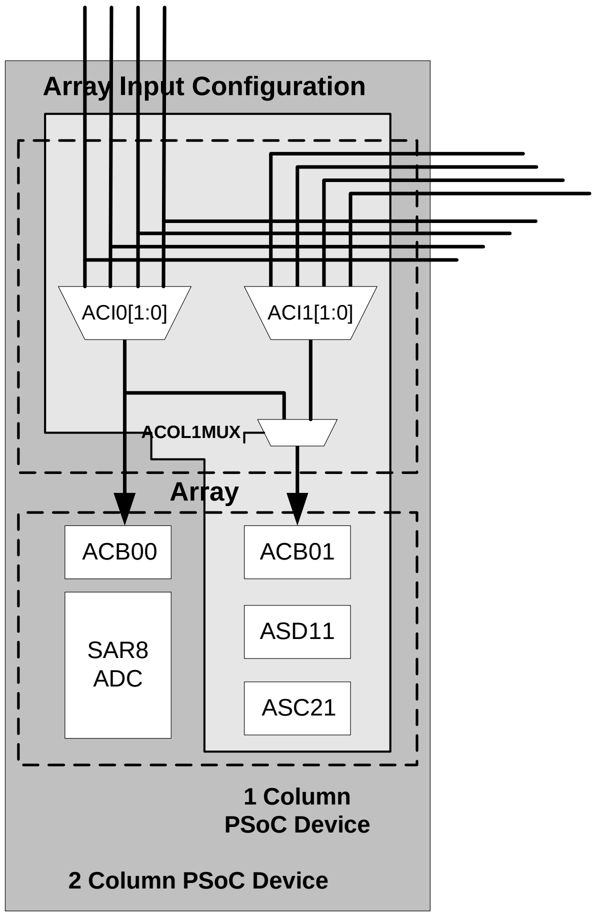
The CY8C24533, CY8C23533, CY8C23433CY8C24633 PSoC devices use the 2 Column PSoC Device analog input configuration and arrays as illustrated in Figure 20-1.

Figure 20-2 presents a more detailed view of the analog column configuration for the CY8C24533, CY8C23533, CY8C23433CY8C24633 PSoC devices, along with analog driver and pin specifics.

The input multiplexer (mux) maps device inputs (package pins) to analog array columns, based on bit values in the AMX_IN and ABF_CR0 registers.

Refer to the analog block diagram on the following page, to view the various analog input configurations. The CY8C24533, CY8C23533, CY8C23433CY8C24633 PSoC devices have two analog drivers used to output analog values on port pins P0[5] and P0[3].

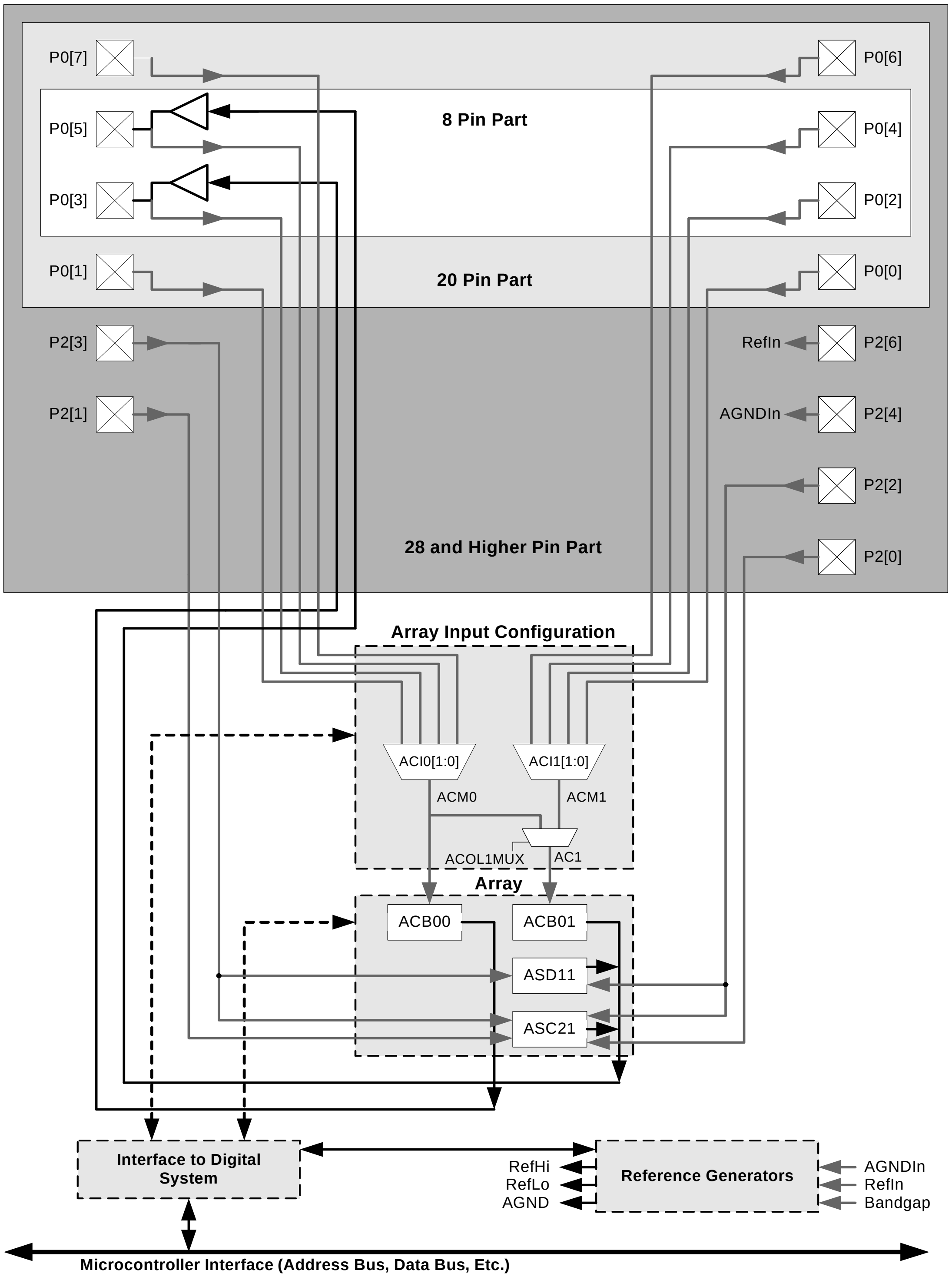
Figure 20-1. Analog Input Configuration Column Overview



20.1.1 Two Column Analog Input Configuration

The two column analog input configuration is detailed in Figure 20-2, along with the analog driver and pin specifics.

Figure 20-2. Two Column PSoC Analog Pin Block Diagram



20.2 Register Definitions

The following registers are associated with Analog Input Configuration and are listed in address order. Each register description has an associated register table showing the bit structure for that register. For a complete table of the analog input configuration registers, refer to the “Summary Table of the Analog Registers” on page 217.

Only certain bits are accessible to be read or written. The bits that are grayed out throughout this manual are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of ‘0’.

20.2.1 AMX_IN Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,60h	AMX_IN					ACI1[1:0]		ACI0[1:0]		RW : 00

The Analog Input Select Register (AMX_IN) controls the analog muxes that feed signals in from port pins into the analog column.

Bits 3 to 0: ACIx[1:0].

For two column PSoC devices, the ACI1[1:0] and ACI0[1:0] bits control the analog muxes that feed signals in from port pins into the analog column.

The analog column can have up to eight port bits connected to its muxed input. ACI1 and ACI0 are used to select among even and odd pins. The AC1Mux bit field controls the bits for those muxes and is located in the Analog Output Buffer Control register (ABF_CR0). There are up to two additional analog inputs that go directly into the Switched Capacitor PSoC blocks.

For additional information, refer to the AMX_IN register on page 64.

20.2.2 ABF_CR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,62h	ABF_CR0	ACol1Mux		ABUF1EN		ABUF0EN		Bypass	PWR	RW : 00

The Analog Output Buffer Control Register 0 (ABF_CR0) controls analog input muxes from Port 0 and the output buffer amplifiers that drive column outputs to device pins.

Bit 7: ACol1MUX.

A mux selects the output of column 0 input mux or column 1 input mux. When set, this bit sets the column 1 input to column 0 input mux output.

Bit 5: ABUF1EN

Enables the analog output buffer for Analog Column 1 (Pin P0[5]). A ‘0’ disables the analog output buffer, a ‘1’ enables.

Bit 3: ABUF0EN

Enables the analog output buffer for Analog Column 0 (Pin P0[3]). (1 Column: AGND). A ‘0’ disables the analog output buffer, a ‘1’ enables

Bit 1: Bypass.

Bypass mode connects the analog output driver input directly to the output. When this bit is set, all analog output drivers are in bypass mode. This is a high impedance connection used primarily for measurement and calibration of internal references. Use of this feature is not recommended for customer designs.

Bit 0: PWR. This bit is used to set the power level of the analog output drivers. When this bit is set, all of the analog output drivers are in a High Power mode.

For additional information, refer to the ABF_CR0 register on page 135.

21. Analog Reference

This chapter discusses the Analog Reference generator and its associated register. The reference generator establishes a set of three internally fixed reference voltages for AGND, RefHi, and RefLo. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

21.1 Architectural Description

The PSoC device is a single supply part, with no negative voltage available or applicable. Figure 21-1shows the analog reference control schematic.

Analog ground (AGND) is constructed near mid-supply. This ground is routed to all analog blocks and separately buffered within each block. Note that there may be a small offset voltage between buffered analog grounds. RefHi and RefLo signals are generated, buffered, and routed to the analog blocks. RefHi and RefLo are used to set the conversion range (that is, span) of ***analog-to-digital (ADC)*** and ***digital-to-analog (DAC)***) converters. RefHi and RefLo can also be used to set thresholds in comparators the two column PSoC devices.

The reference array supplies voltage to all blocks and current to the Switched Capacitor blocks. At higher block clock rates, there is increased reference current demand; the reference power should be set equal to the highest power level of the analog blocks used.

Figure 21-1. Analog Reference Structure

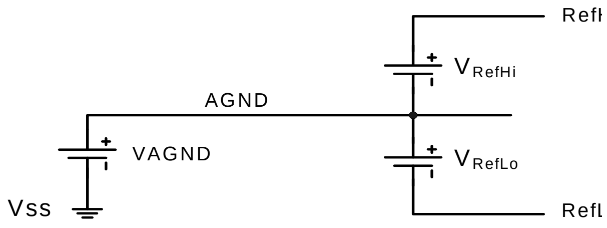
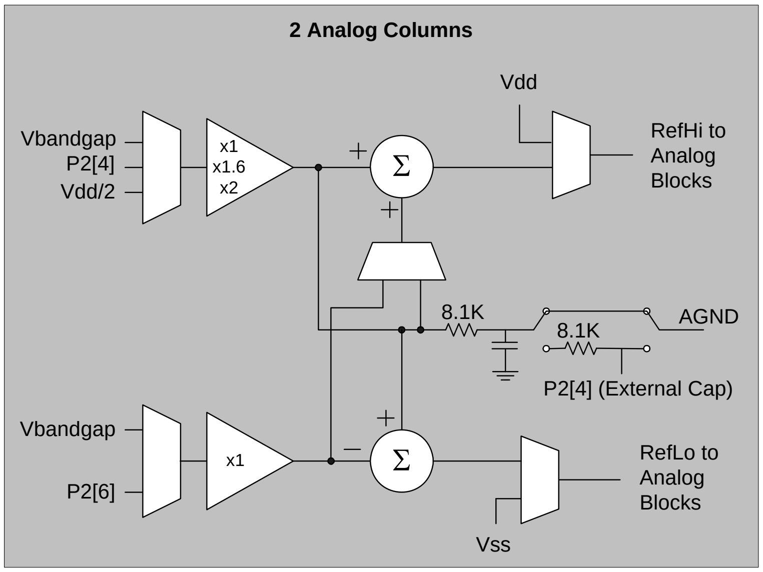


Figure 21-2. Analog Reference Control Schematic



21.2 Register Definitions

The following register is associated with the Analog Reference. For a complete table of all analog registers, refer to the “Summary Table of the Analog Registers” on page 217.

The register description below has an associated register table showing the bit structure. Only certain bits are accessible to be read or written. The bits that are grayed out throughout this manual are reserved bits and are not detailed in the register description that follows. Reserved bits should always be written with a value of ‘0’.

21.2.1 ARF_CR Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,63h	ARF_CR		HBE	REF[2:0]			PWR[2:0]			RW : 00

The Analog Reference Control Register (ARF_CR) is used to configure various features of the configurable analog references.

Note The external bypass capacitor bit 6 (AGNDBYP) in the Bandgap Trim register (BDG_TR: 1, EAh) controls the external bypass capacitor. The default value is zero, which disables this function (see Figure 21-2). The figure shows the two switches in the AGND path in their default state. If bit 6 is set, then the P2[4] IO should be tri-stated and an external capacitor connect from P2[4] to Vss.

Bit 6: HBE. This bit controls the ***bias*** level for all the opamps. It operates with the power setting in each block, to set the parameters of that block. Most applications benefit from the low bias level. At high bias, the analog block opamps have a faster slew rate, but slightly less voltage swing and higher power.

Bits 5 to 3: REF[2:0]. REF (AGND, RefHi, and RefLo) sets the analog array reference control, selecting specific combinations of voltage for analog ground and references. Many of these reference voltages are based on the precision internal reference, a silicon bandgap operating at 1.30 volts. This reference has good thermal stability and power supply rejection.

Alternatively, the power supply can be scaled to provide analog ground and references; this is particularly useful for signals that are ratiometric to the power supply voltage. See Table 21-2.

User supplied external precision references can be connected to Port 2 inputs (available on 28 pin and larger parts). This is useful in setting reference for specific customer applications, such as a ±1.00 V (from AGND) ADC scale. References derived from Port 2 inputs are limited to the same output voltage range as the opamps in the analog blocks.

Bits 2 to 0: PWR[2:0]. PWR controls the bias current and bandwidth for all of the opamps in the analog reference block. PWR also provides on/off control in various rows of the analog array.

Table 21-1. Analog Array Power Control Bits

PWR[2:0]	CT Row	Both SC Rows	REF Bias
000b	Off	Off	Off
001b	On	Off	Low
010b	On	Off	Medium
011b	On	Off	High
100b	Off	Off	Off
101b	On	On	Low
110b	On	On	Medium
111b	On	On	High

For additional information, refer to the ARF_CR register on page 65.

Table 21-2. REF[2:0]: AGND, RefHi, and RefLo Operating Parameters

REF [2:0]	AGND		RefHi		RefLo		Notes
	Source	Voltage	Source	Voltage	Source	Voltage	
000b	Vdd/2	2.5 V 1.65 V	Vdd/2+Vbg	3.8 V 2.95 V	Vdd/2-Vbg	1.2 V 0.35 V	5.0 V System 3.3 V System
001b	P2[4]	2.2 V	P2[4]+P2[6]	3.2 V	P2[4]-P2[6]	1.2 V	User Adjustable. Example: P2[4]=2.2V and P2[6]=1.0V
010b	Vdd/2	2.5 V 1.65 V	Vdd	5.0 V 3.3 V	Vss	0.0 V 0.0 V	5.0 V System 3.3 V System
011b	2*Vbg	2.6 V	3*Vbg	3.9 V	1*Vbg	1.3 V	Not for 3.3 V Systems
100b	2*Vbg	2.6 V	2*Vbg+P2[6]	3.6 V	2*Vbg-P2[6]	1.6 V	P26 < Vdd - 2.6. Example: P2[6]=1.0V
101b	P2[4]	2.2 V	P2[4]+Vbg	3.5 V	P2[4]-Vbg	0.9 V	User Adjustable. Example: P2[4]=2.2V 1.3 < P2[4] < Vdd -1.3
110b	Vbg	1.30 V	2*Vbg	2.6 V	Vss	0	5.0 V System 3.3 V System
111b	1.6*Vbg	2.08 V	3.2*Vbg	4.16 V	Vss	0	Not for 3.3 V Systems

22. Continuous Time PSoC Block

This chapter discusses the Analog Continuous Time PSoC Block and its associated registers. This block supports programmable ***gain*** or ***attenuation*** opamp circuits; instrumentation amplifiers, using two CT blocks (differential gain); and modest response-time analog comparators. For a complete table of the Continuous Time PSoC Block registers, refer to the “Summary Table of the Analog Registers” on page 217. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

22.1 Architectural Description

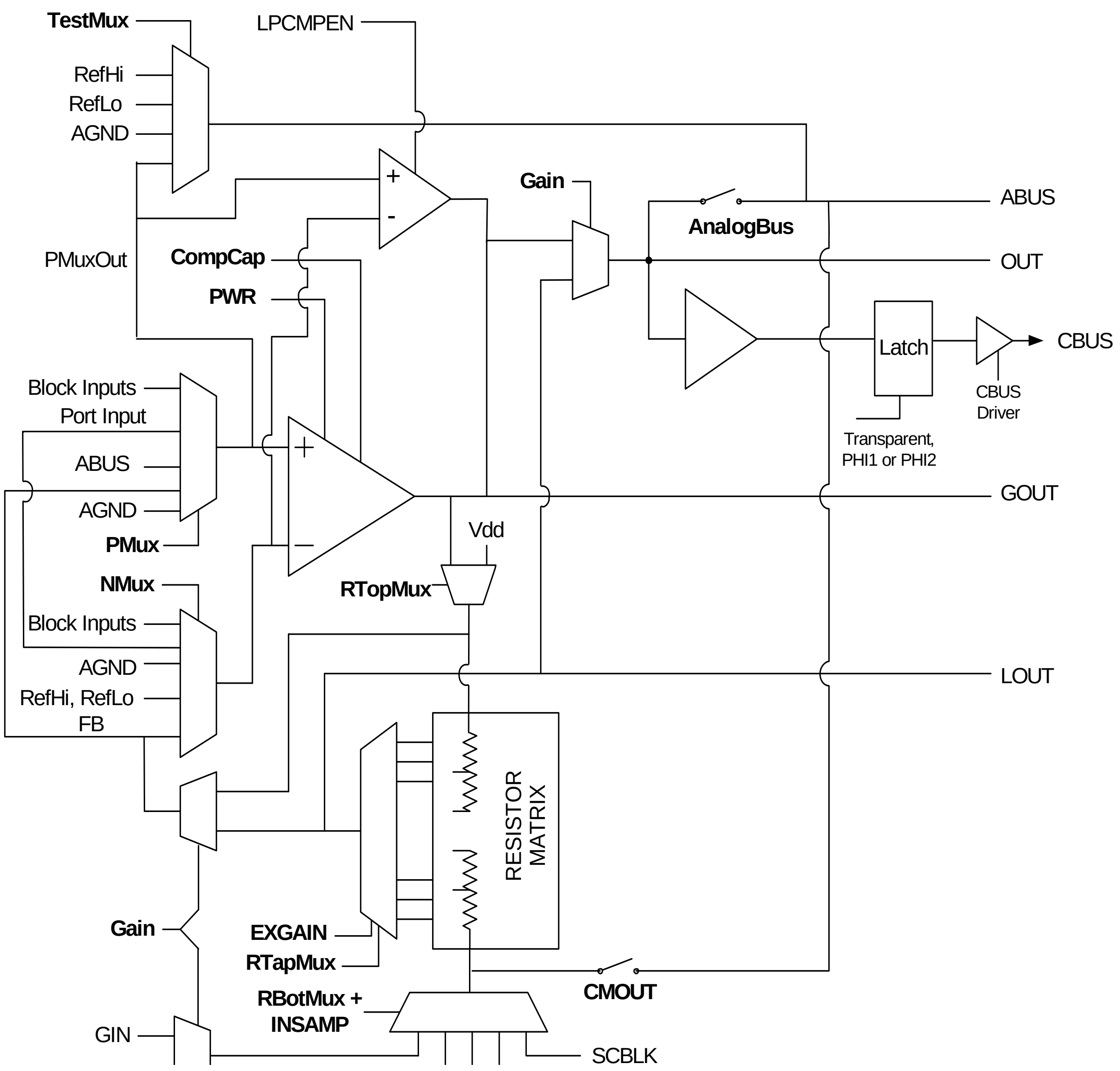
The Analog Continuous Time blocks are built around a rail-to-rail input and output, low offset, low ***noise*** opamp. There are several analog multiplexers (muxes) controlled by register bit settings in the control registers that determine the signal topology inside the block. There is also a precision resistor string located in the feedback path of the opamp, which is controlled by register bit settings.

The block also contains a low power comparator, connected to the same inputs and outputs as the main amplifier. This comparator is useful for providing a digital compare output in low power sleep modes, when the main amplifier is powered off.

There are three discrete outputs from this block. These outputs connect to the following buses:

1. The analog output bus (ABUS), which is an analog bus resource shared by all of the analog blocks in the analog column. This signal may also be routed externally through an output buffer.
2. The comparator bus (CBUS), which is a digital bus resource shared by all of the analog blocks in the analog column.
3. The local output buses (OUT, GOUT, and LOUT), which are routed to neighboring blocks. GOUT and LOUT refer to the gain/loss mode configuration of the block and connect to GIN/LIN inputs of neighboring blocks.

Figure 22-1. Analog Continuous Time Block Diagram



22.2 Register Definitions

The following registers are associated with the Continuous Time (CT) PSoC Block and are listed in address order. Each register description has an associated register table showing the bit structure for that register. For a complete table of the CT PSoC Block registers, refer to the “Summary Table of the Analog Registers” on page 217.

Only certain bits are accessible to be read or written. The bits that are grayed out throughout this manual are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of ‘0’.

In the tables below, an “x” before the comma in the address field (in the "Add." column) indicates that the register exists in both register banks. The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index and n=column index. Therefore, ACB01CR2 is a register for an analog PSoC block in row 0 column 1.

22.2.1 ACBxxCR3 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,70h	ACB00CR3					LPCMPEN	CMOUT	INSAMP	EXGAIN	RW : 00
x,74h	ACB01CR3					LPCMPEN	CMOUT	INSAMP	EXGAIN	RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Analog Continuous Time Type B Block Control Register 3 (ACBxxCR3) is one of four registers used to configure a type B continuous time PSoC block.

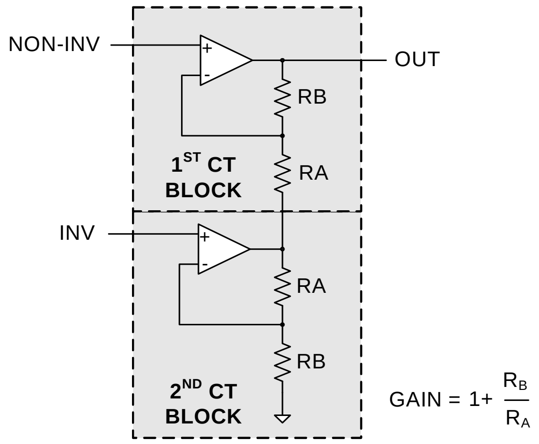
The analog array can be used to build two different forms of instrumentation amplifiers. Two continuous time blocks combine to make the two-opamp instrumentation amplifier illustrated in Figure 22-2.

Two continuous time blocks and one switched capacitor block combine to make a three-opamp instrumentation amplifier (see Figure 22-3).

The three-opamp instrumentation amplifier handles a larger common mode input range but takes more resources. Bit 2 (CMOUT) and bit 1 (INSAMP) control switches are involved in the three-opamp instrumentation amplifier.

Bit 3: LPCMPEN. Each continuous time block has a low power comparator connected in ***parallel*** with the block’s main opamp/comparator. The low power comparator is used in applications where low power is more important than low noise and low offset. The low power comparator operates when the LPCMPEN bit is set high. Since the main opamp/comparator’s output is connected to the low power comparator’s output, only one of the comparators should be active at a particular time. The main opamp/comparator is powered down by setting ACBxxCR2: PWR[1:0] to 00b, or setting ARF_CR: PWR[2:0] to x00b. The low power comparator is unaffected by the PWR bits in the ACBxxCR2 and ARF_CR registers.

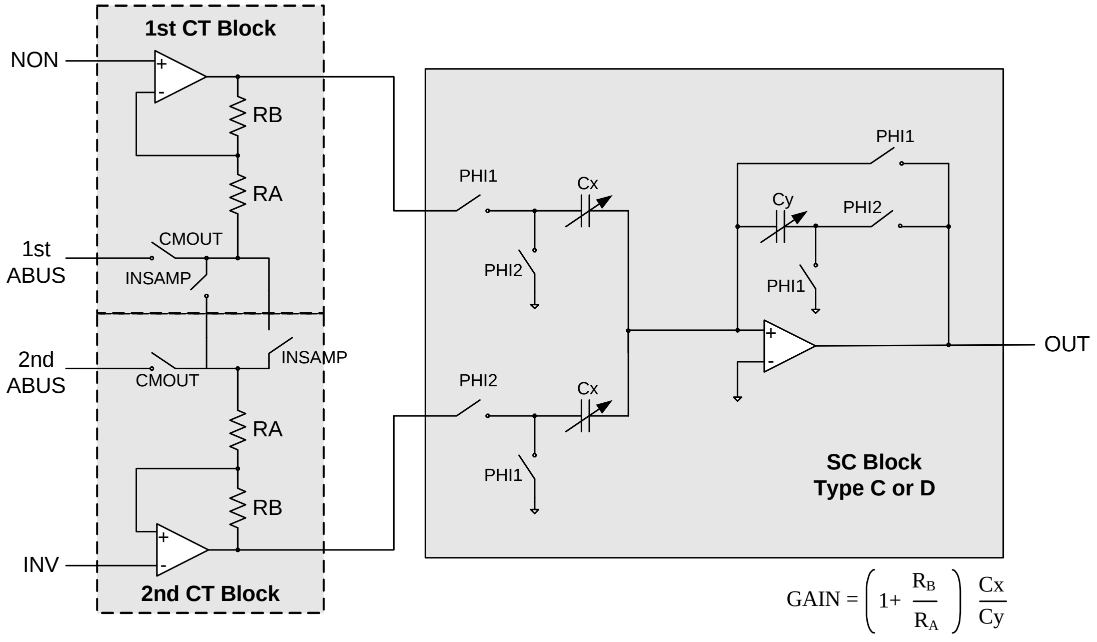
Figure 22-2. Two-Opamp Instrumentation Amplifier



Bit 2: CMOUT. If this bit is high, then the node formed by the connection of the resistors, between the continuous time blocks, is connected to that continuous time block’s ABUS. This node is the common mode of the inputs to the instrumentation amplifier. The CMOUT bit is optional for the three-opamp instrumentation amplifier.

Bit 1: INSAMP. This bit is used to connect the resistors of two continuous time blocks as part of a three-opamp instrumentation amplifier. The INSAMP bit must be high for the three-opamp instrumentation amplifier (see Figure 22-3).

Figure 22-3. Three-Opamp Instrumentation Amplifier

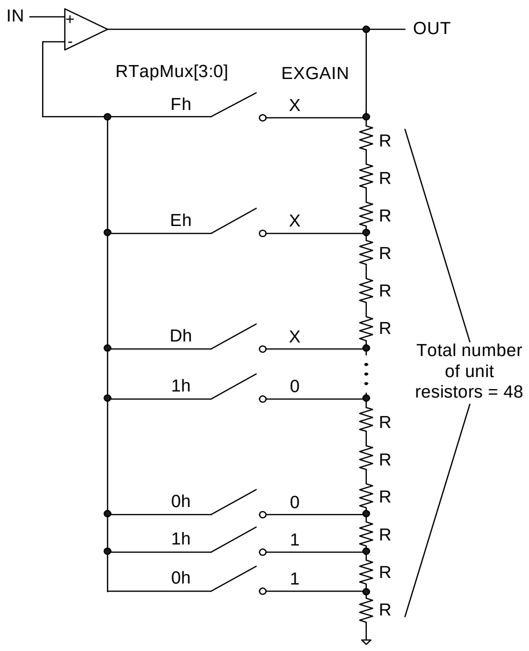


$$\text{GAIN} = \left(1+\frac{R_B}{R_A}\right)\frac{Cx}{Cy}$$

Bit 0: EXGAIN. The continuous time block's resistor tap is specified by the value of ACBxxCR3 EXGAIN, combined with the value of ACBxxCR0 RtapMux[3:0]. For RtapMux values from 02h through 15h, the EXGAIN bit has no effect on which tap is selected. (See the ACBxxCR0 register for details.) The EXGAIN bit enables additional resistor tap selections for RtapMux = 01h and RtapMux = 00h (see Figure 22-4).

For additional information, refer to the ACBxxCR3 register on page 73.

Figure 22-4. CT Block in Gain Configuration



22.2.2 ACBxxCR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,71h	ACB00CR0	RTapMux[3:0]				Gain	RTopMux	RBotMux[1:0]		RW : 00
x,75h	ACB01CR0	RTapMux[3:0]				Gain	RTopMux	RBotMux[1:0]		RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Analog Continuous Time Type B Block Control Register 0 (ACBxxCR0) is one of four registers used to configure a type B continuous time PSoC block.

Bits 7 to 4: RTapMux[3:0]. These bits, in combination with the EXGAIN bit 0 in the ACBxxCR3 register, select the tap of the resistor string.

Bit 3: Gain. This bit controls whether the resistor string is connected around the opamp as for gain (tap to inverting opamp input) or for loss (tap to output of the block). Note that setting Gain alone does not guarantee a gain or loss block. Routing of the ends of the resistor string determine this.

Bit 2: RTopMux. This bit controls the top end of the resistor string, which can either be connected to Vdd or to the opamp output.

Bits 1 and 0: RBotMux[1:0]. These bits, in combination with the INSAMP bit 1 in the ACBxxCR3 register, control the connection of the bottom end of the resistor string.

For additional information, refer to the ACBxxCR0 register on page 74.

22.2.3 ACBxxCR1 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,72h	ACB00CR1	AnalogBus	CompBus	NMux[2:0]			PMux[2:0]			RW : 00
x,76h	ACB01CR1	AnalogBus	CompBus	NMux[2:0]			PMux[2:0]			RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Analog Continuous Time Type B Block Control Register 1 (ACBxxCR1) is one of four registers used to configure a type B continuous time PSoC block.

Bit 7: AnalogBus. This bit controls the analog output bus (ABUS). A CMOS switch connects the opamp output to the analog bus.

Bit 6: CompBus. This bit controls a tri-state buffer that drives the comparator logic. If no block in the analog column is driving the comparator bus, it is driven low externally to the blocks.

Bits 5 to 3: NMux[2:0]. These bits control the multiplexing of inputs to the inverting input of the opamp. There are seven input choices from outside the block, plus the internal feedback selection from the resistor string top.

Bits 2 to 0: PMux[2:0]. These bits control the multiplexing of inputs to the non-inverting input of the opamp. There are seven input choices from outside the block, plus the internal feedback selection from the resistor string top.

For additional information, refer to the ACBxxCR1 register on page 76.

22.2.4 ACBxxCR2 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,73h	ACB00CR2	CPhase	CLatch	CompCap	TMUXEN	TestMux[1:0]		PWR[1:0]		RW : 00
x,77h	ACB01CR2	CPhase	CLatch	CompCap	TMUXEN	TestMux[1:0]		PWR[1:0]		RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Analog Continuous Time Type B Block Control Register 2 (ACBxxCR2) is one of four registers used to configure a type B continuous time PSoC block.

Bit 7: CPhase. This bit controls which internal clock phase the comparator data is latched on.

Bit 6: CLatch. This bit controls whether the latch is active or if it is always transparent.

Bit 5: CompCap. This bit controls whether or not the compensation capacitor is enabled in the opamp. By not switching in the compensation capacitance, a much faster response is obtained if the amplifier is used as a comparator.

Bit 4: TMUXEN. If the TMUXEN bit is high, then the value of TestMux[1:0] determines which test mux input is connected to the ABUS for that particular continuous time block. If the TMUXEN bit is low, then none of the test mux inputs are connected to the ABUS regardless of the value of TestMux[1:0].

Bits 3 and 2: TextMux[1:0]. These bits select which signal is connected to the analog bus.

Bits 1 and 0: PWR[1:0]. Power is encoded to select one of three power levels or power down (off). The blocks power up in the off state. Combined with the Turbo mode, this provides six power levels. Turbo mode is controlled by the HBE bit of the Analog Reference Control register (ARF_CR).

For additional information, refer to the ACBxxCR2 register on page 78.

23. Switched Capacitor PSoC Block

This chapter presents the Analog Switched Capacitor Block and its associated registers. The analog Switched Capacitor (SC) blocks are built around a low offset, low noise operational amplifier. For a complete table of the Switched Capacitor PSoC Block registers, refer to the "Summary Table of the Analog Registers" on page 217. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

23.1 Architectural Description

The Analog Switched Capacitor blocks are built around a rail-to-rail, input and output, low offset and low noise opamp. (Refer to Figure 23-1 and Figure 23-2.) There are several analog multiplexers (muxes) controlled by register bit settings in the control registers that determine the signal topology inside the block. There are four user-selectable capacitor arrays inside this block connected to the opamp.

There are four analog arrays. Three of the four arrays are input arrays and are labeled A Cap Array, B Cap Array, and C Cap Array. The fourth array is the feedback path array and is labeled F Cap Array. All arrays have user-selectable unit values: one array is in the feedback path of the opamp and three arrays are in the input path of the opamp. Analog muxes, controlled by bit settings in control registers, set the capacitor topology inside the block. A group of muxes are used for the signal processing and switch synchronously to clocks PHI1 and PHI2, with behavior that is modified by control register settings. There is also an analog comparator that converts the opamp output (relative to the local analog ground) into a digital signal.

There are two types of Analog Switched Capacitor blocks called Type C and Type D. Their primary differences relate to connections of the C Cap Array and the block's position in a two-pole filter section. The Type D block also has greater flexibility in switching the B Cap Array.

There are three discrete outputs from this block. These outputs connect to the following buses:

1. The analog output bus (ABUS), which is an analog bus resource shared by all of the analog blocks in the analog column. This signal may also be routed externally through the output buffer. The ABUS of each column has a 1.4 pF capacitor to GND. This capacitor may be used to hold a sampled value on the ABUS net. Although there is only one capacitor per column, it is shown in both Figure 23-1 and Figure 23-2 to allow visualization of the sample and hold function. See the description of the ClockPhase bit in the ASCxxCR0 and ASDxxCR0 registers in section 23.3 Register Definitions.
2. The comparator bus (CBUS), which is a digital bus resource shared by all of the analog blocks in the analog column.
3. The local output bus (OUT), which is an analog node, is routed to neighboring block inputs.

Figure 23-1. Analog Switch Cap Type C PSoC Blocks

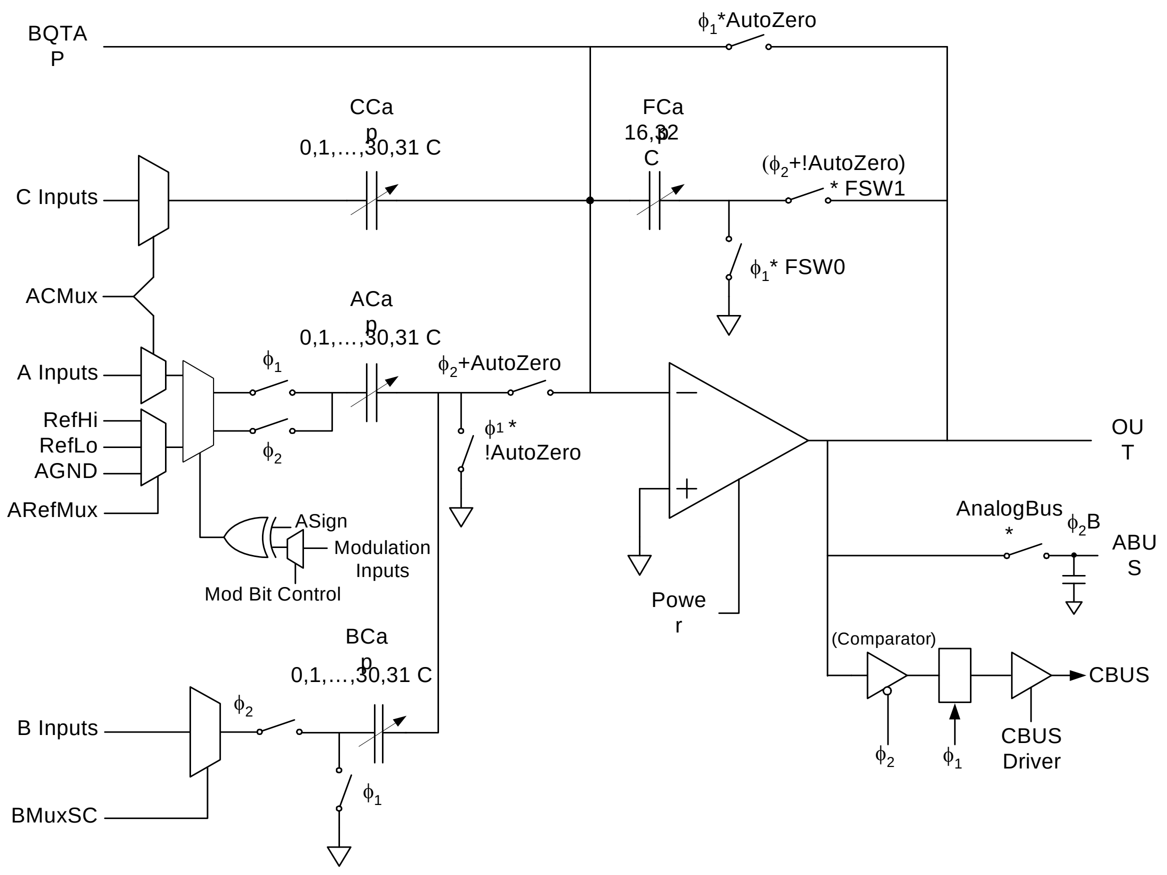
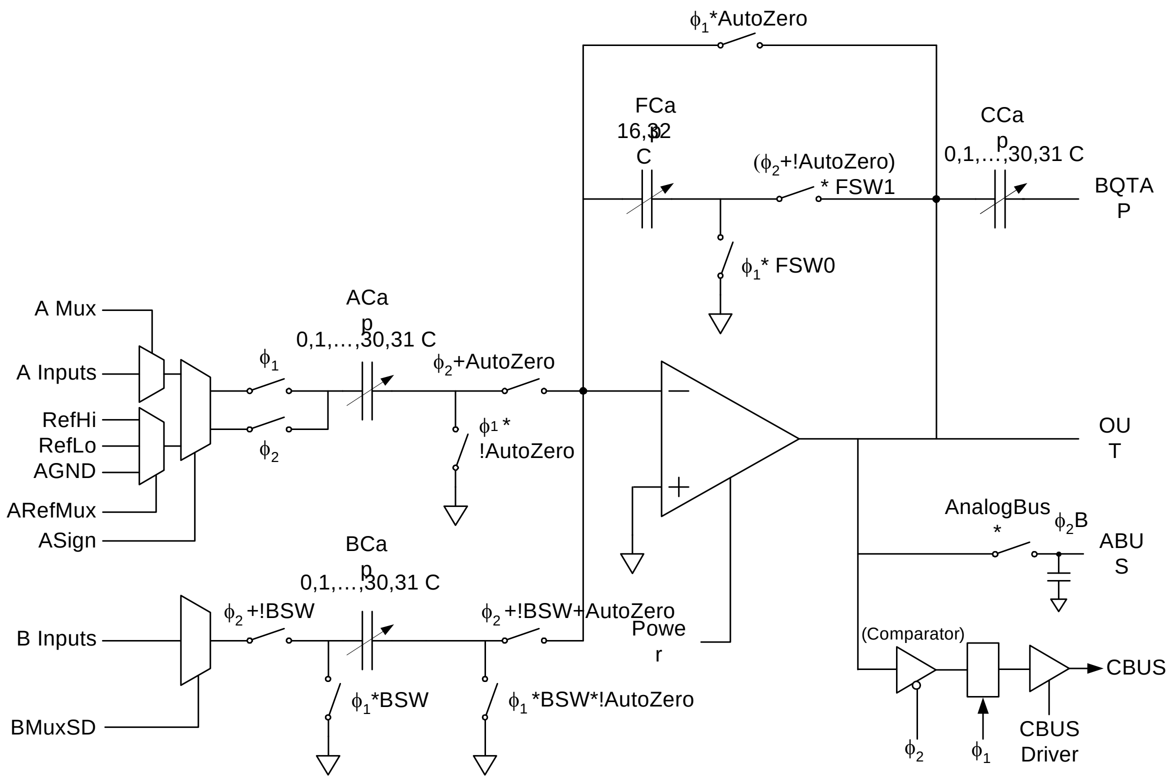


Figure 23-2. Analog Switch Cap Type D PSoC Blocks



23.2 Application Description

The analog Switched Capacitor (SC) blocks support Delta-Sigma, Successive Approximation, and Incremental Analog-to-Digital Conversion, Capacitor DACs, and SC filters. They have three input arrays of binary-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.

The non-switched capacitor node is labeled “BQTAP” in the figure above. The local connection of BQTAP is between horizontal neighboring SC blocks within an analog bi-column. Since the input of SC Block C (ASCxx) has this additional switched capacitor, it is configured for the input stage of such a switched capacitor bi-quad ***filter***. When followed by an SC Block D (ASDxx) integrator, this combination of blocks can be used to provide a full universal two-pole switched capacitor bi-quad filter.

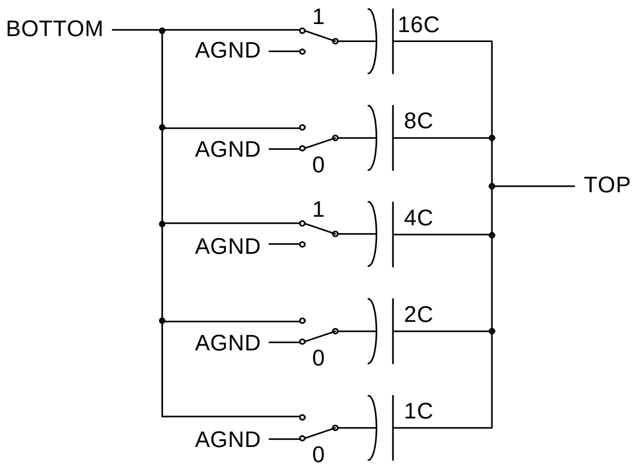
23.3 Register Definitions

The following registers are associated with the Switched Capacitor (SC) PSoC Block and are listed in address order. Each register description has an associated register table showing the bit structure for that register. For a complete table of SC PSoC Block registers, refer to the “Summary Table of the Analog Registers” on page 217.

Only certain bits are accessible to be read or written. The bits that are grayed out throughout this manual are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of ‘0’.

Figure 23-3 applies to the ACap, BCap, and CCap functionality for the capacitor registers. The *X*Cap field is used to store the binary encoded value for capacitor *X*, where *X* can be A (ACap), B (BCap), or C (CCap), in both the ASCxxCRx and ASDxx-CRx registers. Figure 23-3 illustrates the switch settings for the example ACap[4:0]=14h=10100b=20d.

Figure 23-3. Example Switch Capacitor Settings



Analog Switch Cap Type C PSoC Block Control Registers

In the tables below, an “x” before the comma in the address field (in the "Add." column) indicates that the register exists in both register banks. The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index and n=column index. Therefore, ASC21CR2 is a register for an analog PSoC block in row 2 column 1.

23.3.1 ASCxxCR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,94h	ASCxxCR0	FCap	ClockPhase	ASign	ACap[4:0]					RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Analog Switch Cap Type C Block Control Register 0 (ASCxxCR0) is one of four registers used to configure a type C switch capacitor PSoC block.

Bit 7: FCap. This bit controls the size of the switched feedback capacitor in the integrator.

Bit 6: ClockPhase. This bit 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 the Control 2 register.

This bit is the ClockPhase select that inverts the clock 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). The 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 Source Control register, then sample and hold operation is disabled for all columns and all enabled outputs of SC blocks are connected to their respective output buses for the entire period of their respective PHI2s.

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. The rising edge of the other clock phase causes the value to come out (output point event). This bit determines which clock phase triggers the capture point event, and the other clock triggers the output point event. The value output to the comparator bus remains stable between output point events.

Bit 5: ASign. This bit controls the switch phasing of the switches on the bottom plate of the ACap capacitor. The bottom plate samples the input or the reference.

Bits 4 to 0: ACap[4:0]. The ACap bits set the value of the capacitor in the A path.

For additional information, refer to the ASCxxCR0 register on page 83.

23.3.2 ASCxxCR1 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,95h	ASC21CR1	ACMux[2:0]			BCap[4:0]					RW : 00

LEGEND

x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Analog Switch Cap Type C Block Control Register 1 (ASCxxCR1) is one of four registers used to configure a type C switch capacitor PSoC block.

Bits 7 to 5: ACMUX[2:0]. These bits control 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.

Bits 4 to 0: BCap[4:0]. The BCap bits set the value of the capacitor in the B path.

For additional information, refer to the ASCxxCR1 register on page 84.

23.3.3 ASCxxCR2 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,96h	ASC21CR2	AnalogBus	CompBus	AutoZero	CCap[4:0]					RW : 00

LEGEND

x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Analog Switch Cap Type C Block Control Register 2 (ASCxxCR2) is one of four registers used to configure a type C switch capacitor PSoC block.

Bit 7: AnalogBus. This bit gates the output to the analog column bus (ABUS). The output on the ABUS is affected by the state of the ClockPhase bit in the Control 0 register. If AnalogBus is set to ‘0’, the output to the analog column bus is tri-stated. If AnalogBus is set to ‘1’, the signal that is output to the analog column bus is selected by the ClockPhase bit. If the ClockPhase bit is ‘0’, the block output is gated by sampling clock on the last part of PHI2. If the ClockPhase bit is ‘1’, the block output continuously drives the ABUS.

Bit 6: CompBus. This bit controls the output to the column comparator bus (CBUS). Note that if the CBUS 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.

Bit 5: AutoZero. This bit controls the shorting of the output to the inverting input of the opamp. When shorted, the opamp is basically a follower. The output is the opamp 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 opamp. If AutoZero is enabled, then the pair of switches is active. AutoZero also affects the function of the FSW1 bit in the Control 3 register.

Bits 4 to 0: CCap[4:0]. The CCap bits set the value of the capacitor in the C path.

For additional information, refer to the ASCxxCR2 register on page 85.

23.3.4 ASCxxCR3 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,97h	ASC21CR3	ARefMux[1:0]		FSW1	FSW0	BMuxSC[1:0]		PWR[1:0]		RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Analog Switch Cap Type C Block Control Register 3 (ASCxxCR3) is one of four registers used to configure a type C switch capacitor PSoC block.

Bits 7 and 6: ARefMux[1:0]. These bits select the reference input of the A capacitor branch.

Bit 5: FSW1. This bit is used to control a switch in the integrator capacitor path. It connects the output of the opamp to the integrating cap. The state of the feedback switch is affected by the state of the AutoZero bit in the Control 2 register. 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.

Bit 4: FSW0. This bit is used to control a switch in the integrator capacitor path. It connects the output of the opamp to analog ground.

Bits 3 and 2: BMuxSC[1:0]. These bits control the muxing to the input of the B capacitor branch.

Bits 1 and 0: PWR[1:0]: The power bits serve as encoding for selecting one of four power levels. The block always powers up in the off state.

For additional information, refer to the ASCxxCR3 register on page 86.

Analog Switch Cap Type D PSoC Block Control Registers

In the tables below, an “x” before the comma in the address field (in the "Add." column) indicates that the register exists in both register banks. The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index and n=column index. Therefore, ASD01CR0 is a register for an analog PSoC block in row 0 column 1.

23.3.5 ASDxxCR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,84h	ASD11CR0	FCap	ClockPhase	ASign	ACap[4:0]					RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Analog Switch Cap Type D Block Control Register 0 (ASDxxCR0) is one of four registers used to configure a type D switch capacitor PSoC block.

Bit 7: FCap. This bit controls the size of the switched feedback capacitor in the integrator.

Bit 6: ClockPhase. This bit 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 the Control 2 register.

This bit is the ClockPhase select that inverts the clock 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). The 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 buses, for the entire period of their respective PHI2s.

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. The rising edge of the other clock phase causes the value to come out (output point event). This bit determines which clock phase triggers the capture point event, and the other clock triggers the output point event. The value output to the comparator bus remains stable between output point events.

Bit 5: ASign. This bit controls the switch phasing of the switches on the bottom plate of the A capacitor. The bottom plate samples the input or the reference.

Bits 4 to 0: ACap[4:0]. The ACap bits set the value of the capacitor in the A path.

For additional information, refer to the ASDxxCR0 register on page 79.

23.3.6 ASDxxCR1 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,85h	ASD11CR1	AMux[2:0]			BCap[4:0]					RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Analog Switch Cap Type D Block Control Register 1 (ASDxxCR1) is one of four registers used to configure a type D switch capacitor PSoC block.

Bits 7 to 5: AMux[2:0]. These bits control the input muxing for the A capacitor branch.

Bits 4 to 0: BCap[4:0]. The BCap bits set the value of the capacitor in the B path.

For additional information, refer to the ASDxxCR1 register on page 80.

23.3.7 ASDxxCR2 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,86h	ASD11CR2	AnalogBus	CompBus	AutoZero	CCap[4:0]					RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Analog Switch Cap Type D Block Control Register 2 (ASDxxCR2) is one of four registers used to configure a type D switch capacitor PSoC block.

Bit 7: AnalogBus. This bit gates the output to the analog column bus (ABUS). The output on the ABUS is affected by the state of the ClockPhase bit in the Control 0 Register. If AnalogBus is set to ‘0’, the output to the ABUS 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 the last part of PHI2. If the ClockPhase bit is ‘1’, the block ClockPhase continuously drives the ABUS.

Bit 6: CompBus. This bit controls the output to the column comparator bus (CBUS). Note that if the CBUS 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.

Bit 5: AutoZero. This bit controls the shorting of the output to the inverting input of the opamp. When shorted, the opamp is basically a follower. The output is the opamp 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 opamp. If AutoZero is enabled, then the pair of switches is active. AutoZero also affects the function of the FSW1 bit in the Control 3 register.

Bits 4 to 0: CCap[4:0]. The CCap bits set the value of the capacitor in the C path.

For additional information, refer to the ASDxxCR2 register on page 81.

23.3.8 ASDxxCR3 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,87h	ASD11CR3	ARefMux[1:0]		FSW1	FSW0	BSW	BMuxSD	PWR[1:0]		RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Analog Switch Cap Type D Block Control Register 3 (ASDxxCR3) is one of four registers used to configure a type D switch capacitor PSoC block.

Bits 7 and 6: ARefMux[1:0]. These bits select the reference input of the A capacitor branch.

Bit 5: FSW1. This bit is used to control a switch in the integrator capacitor path. It connects the output of the opamp to the integrating cap. The state of the switch is affected by the state of the AutoZero bit in the Control 2 register. 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.

Bit 4: FSW0. This bit is used to control a switch in the integrator capacitor path. It connects the output of the opamp to analog ground.

Bit 3: BSW. This bit 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.

Bit 2: BMuxSD. This bit controls muxing to the input of the B capacitor branch. The B branch can be switched or unswitched.

Bits 1 and 0: PWR[1:0]. The power bits serve as encoding for selecting one of four power levels. The block always powers up in the off state.

For additional information, refer to the ASDxxCR3 register on page 82.

24. SAR8 ADC PSoC Block

This chapter briefly discusses the SAR8 ADC PSoC Block and its associated registers. For a complete table of the SAR8 ADC registers, refer to the "Summary Table of the Analog Registers" on page 217. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

24.1 Architectural Description

The CY8C24533, CY8C23533, CY8C23433CY8C24633 include a high performance, configurable 8-bit analog-to-digital converter (ADC) with eight multiplexed analog input channels and two inside analog channels. The ADC uses a successive approximation technique to convert the analog voltage levels from up to ten different sources. The analog input channels of the ADC are available at Port 0.

24.1.1 Features

- Successive approximation functionality support
- 8-bit resolution
- Eight primary input analog channels and two inside analog channels, which come from two CT blocks
- Support for right side scale feature when reading conversion result data. Scale size is programmed as 0 to 6 bits. This feature is useful in digital filter functions.
- Single conversion
- Free running conversion
- Selectable conversion request trigger
- Programmable sample time
- Programmable clock divider
- Cancel/restart feature for running conversions
- Support to auto align/trigger at any point of PWM, timer, or counter cycle
- Automatic entry into low power mode after every conversion in single conversion mode
- Reference voltage can be pulled out by P1[7]

24.2 Register Definitions

The following registers are associated with the SAR8 ADC PSoC Block and are listed in address order. The register descriptions below have an associated register table showing the bit structure.

The bits that are grayed out in the register tables are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of '0'. For a complete table of the SAR8 ADC registers, refer to the "Summary Table of the Analog Registers" on page 217.

24.2.1 SARADC_DL Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,67h	SARADC_DL	Data[7:0]								RW : 00

The SAR8 ADC Conversion Low Byte Results Register (SARADC_DL) is the raw conversion data register of the SAR8 ADC.

Bits 7 to 0: Data[7:0]. The read out data of this register is dependant on the right side scale bits' setting, which are defined in the SARADC_CR2 register. The scale size setting affects the final read out results, but not the ADC conversion results in this register.

For additional information, refer to the SARADC_DL register on page 69.

24.2.2 SARADC_CR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,69h	SARADC_CR0		ADC Channel[3:0]				Data Ready	Start/Busy	ADCEN	# : 0

The SAR8 ADC Control Register 0 (SARADC_CR0) is used to control normal ADC operation and show ADC status.

Bits 6 to 3: ADC Channel[3:0].

Table 24-1. ADC Channel

Bit Option	Description
0000	P0[0]
0001	P0[1]
0010	P0[2]
0011	P0[3]
0100	P0[4]
0101	P0[5]
0110	P0[6]
0111	P0[7]
1000	CT1
1001	CT2

Bit 2: Data Ready. When '0', new conversion data is not ready or the conversion data has already been read out. Writing '1' to Start/Busy bit automatically clears the bit to '0'. When '1', ADC has newest conversion data, which has never been read out.

Bit 1: Start/Busy. When '0', ADC has finished the operation. If firmware writes a '1' to this bit it means that firmware triggers the ADC to perform the sample and conversion from the next system clock cycle. If this bit is already '1' (in ADC conversion mode), the new write of '1' forces the ADC to cancel the ongoing conversion and restart one new conversion from the next system clock cycle. When the ADC is set to free running, this signal means start a new ADC conversion. The ADC then automatically starts the new conversion after it finishes the previous conversion. When '1', ADC is busy.

Bit 0: ADCEN. When '0', disable ADC operation. The ADC automatically enters low power mode right after it finishes the conversion, even in Enable mode. If the ADC has been disabled, it does not perform any operation. When '1', Enable ADC operation.

For additional information, refer to the SARADC_CR0 register on page 70.

24.2.3 SARADC_CR1 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,6Ah	SARADC_CR1	PWRSELADC	PWRSELR2R				Align Source [1:0]		Align Enable	RW : 0

The SAR8 ADC Control Register 1 (SARADC_CR1) is used to control normal ADC operation and show ADC status.

There may be a need in motor control applications to automatically trigger the ADC to execute one analog-to-digital conversion at a certain time during the PWM period in order to obtain best performance.

In PSoC devices, the PWM function is implemented in few PSoC digital blocks so that the ADC control block can monitor the digital block status (there are 4 digital blocks in the CY8C24533, CY8C23533, CY8C23433CY8C24633) and perform analog-to-digital conversions per user settings.

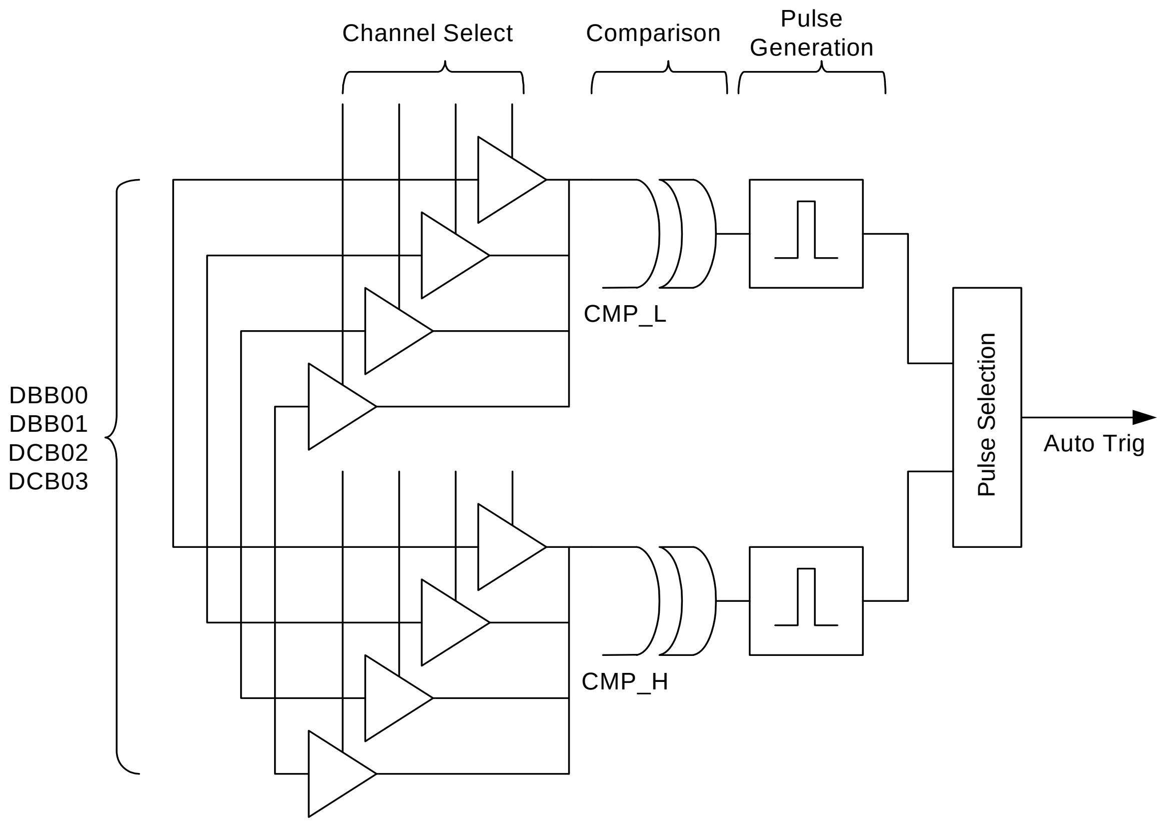
PWMs can be 8 bits or 16 bits. The core of PWM functionality is a down-counter that is implemented in the CY8C24533, CY8C23533, CY8C23433CY8C24633 digital blocks. So we pull out the DR0 register values from the digital blocks to execute a data comparison in order to decide when an analog-to-digital conversion should be performed.

We define a 16-bit comparator in cases when the PWM is 16 bits. Note that the digital block enable signal is also used in auto-trigger generation.

We separate the 16-bit comparator into two 8-bit channels (H-channel and L-channel) for maximum flexibility. Every digital block can drive both channels at the same time but each channel can only be driven by one digital block. The lower digital blocks have higher priority than the higher digital blocks in order to output to the high or low channel. The channel data keeps 0s if it is not driven.

The selected DR0 data are compared with the pre-set values in the registers. The auto align circuit sends out a half SYSCLK cycle 'trigger' signal when it sees a low-to-high transition in comparison results. There is no trigger signal if the relevant digital block is disabled. There is no trigger signal even if a digital block enable signal goes high at the same time the comparison results are true.

Figure 24-1. Digital Block Channel ADC Auto Trigger



Bit 7: PWRSELADC. When ‘0’, ADC analog block obtains power supply from inside VPWR. When ‘1’, ADC analog block obtains power supply from P3[0]. P3[0] must supply no less than 3 mA current.

Bit 6: PWRSELR2R. When ‘0’, ADC R-2R reference generation block obtains power supply from inside VPWR. P3[0] must be less than or equal to VPWR and ADC R2R reference generation block power supply must be less than or equal to that of the ADC analog block. When ‘1’, ADC R2R reference generation block obtains power supply from P3[0] .P3[0] must supply no less than 0.3 mA current.

Bits 2 and 1: Align Source[1:0]. When ‘00‘, low and high channels are completely independent. Both can trigger ADC. Each is 8-bit accuracy. When ‘01’, only low channel triggers ADC. It is 8-bit accuracy. When ‘10‘, only high channel triggers ADC. It is 8-bit accuracy. When ‘11’, high and low channels combine together to form a 16-bit trigger source. It is 16-bit accuracy.

Bit 0: Align Enable. When ‘0‘, auto align function is disabled. When ‘1’, auto align function is enabled.

For additional information, refer to the SARADC_CR1 register on page 71.

24.2.4 SARADC_TRS Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,A8h	SARADC_TRS	DCB03_HL[1:0]		DCB02_HL[1:0]		DBB01_HL[1:0]		DBB00_HL[1:0]		RW : 0

The SAR8 ADC Auto Align/Trigger Source Register (SARADC_TRS) is where the ADC auto align/trigger source is set.

This register is where to select any digital blocks in digital ROW0 as the align/trigger source for ADC conversion. You can also select two adjacent blocks in digital ROW0 as the align/trigger source for ADC conversion. For example, if there is one PWM8 in block DBB0x and the system needs the ADC auto aligned with the PWM pulse, select the DBB0x as auto-align/trigger source to the ADC. Setting the SARADC_TRCL register, triggers the ADC start conversion at any point during one PWM cycle. You can also select two adjacent blocks as auto-align/trigger source to align with a 16-bit PWM, timer, or counter. Be sure that the block you select is the same block you place the user module with which you want the ADC to align.

Bits 7 and 6: DCB03_HL[1:0].

Table 24-2. DCB03_HL

Bit	Bit Option	Description
7	0	DCB03 DR0 is not driving the channel high
7	1	DCB03 DR0 is driving the channel high
6	0	DCB03 DR0 is not driving the channel low
6	1	DCB03 DR0 is driving the channel low.

Bits 5 and 4: DCB02_HL[1:0].

Table 24-3. DCB02_HL

Bit	Bit Option	Description
5	0	DCB02 DR0 is not driving the channel high
5	1	DCB02 DR0 is driving the channel high
4	0	DCB02 DR0 is not driving the channel low
4	1	DCB02 DR0 is driving the channel low.

Bits 3 and 2: DBB01_HL[1:0].

Table 24-4. DBB01_HL

Bit	Bit Option	Description
3	0	DBB01 DR0 is not driving the channel high
3	1	DBB01 DR0 is driving the channel high
2	0	DBB01 DR0 is not driving the channel low
2	1	DBB01 DR0 is driving the channel low.

Bits 1 and 0: DBB00_HL[1:0].

Table 24-5. DBB00_HL

Bit	Bit Option	Description
1	0	DBB00 DR0 is not driving the channel high
1	1	DBB00 DR0 is driving the channel high
0	0	DBB00 DR0 is not driving the channel low
0	1	DBB00 DR0 is driving the channel low.

Note The digital block with the smaller number has higher priority to output to the related channel. The circuitry guarantees only the highest priority bit to be set to 1 when the user enables multiple blocks to one channel. The user can only see one bit set for one channel if they read back the register data.

For additional information, refer to the SARADC_TRS register on page 139.

24.2.5 SARADC_TRCL Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,A9h	SARADC_TRCL	CMP_L[7:0]								RW : 00

The SAR8 ADC Low Channel Comparator Data Register. (SARADC_TRCL) is the ADC auto align/trigger comparator data register.

A trigger occurs when the low channel trigger is enabled, a selected digital block is enabled, and the selected digital block's DR0 data is equal to the data of this register.

Bits 7 to 0: CMP_L[7:0]. The comparator data for low channel trigger.

Note The low channel is enabled when auto align is enabled and align type is 00, 01, or 11.

For additional information, refer to the SARADC_TRCL register on page 140.

24.2.6 SARADC_TRCH Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,AAh	SARADC_TRCH	CMP_H[7:0]								RW : 00

The SAR8 ADC High Channel Comparator Data Register (SARADC_TRCH) is the ADC auto align/trigger comparator data register.

A trigger occurs when the high channel trigger is enabled, a selected digital block is enabled, and the selected digital block's DR0 data is equal to the data of this register.

Bits 7 to 0: CMP_H[7:0]. The comparator data for high channel trigger.

Note The high channel is enabled when auto align is enabled and align type is 00, 10, or 11.

For additional information, refer to the SARADC_TRCH register on page 141.

24.2.7 SARADC_CR2 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,ABh	SARADC_CR2	Test Enable	Free Run	Scale Size[2:0]			ADC Clock[2:0]			RW : 0

The SAR8 ADC Control Register 2 (SARADC_CR2) is used to control ADC settings.

Bit 7: Test Enable. When Test Enable is set to ‘0’, ADC is in user mode. When Test Enable is set to ‘1’, ADC is in test mode.

Bit 6: Free Run. When Free Run is set to ‘0’, ADC is in one-shot mode. When Free Run is set to ‘1’, ADC is in free running mode.

Bits 5 to 3: Scale Size[2:0].

Table 24-6. Scale Size

Bit Option	Description
000	ADC raw results are directly read out
001	ADC raw results are divided by 2 when read out
010	ADC raw results are divided by 4 when read out
011	ADC raw results are divided by 8 when read out
100	ADC raw results are divided by 16 when read out
101	ADC raw results are divided by 32 when read out
110	ADC raw results are divided by 64 when read out

This feature does not affect the raw data in the ADC raw results register. It only affects the data read out to the MCU.

Bits 2 to 0: ADC Clock[2:0].

Table 24-7. ADC Clock

Bit Option	Description
000	ADC clock is SYSCLK
001	ADC clock is SYSCLK/2
010	ADC clock is SYSCLK/4
011	ADC clock is SYSCLK/8
100	ADC clock is SYSCLK/16
101	ADC clock is SYSCLK/32
110	ADC clock is SYSCLK/64

ADC sample rate is ADC clock divided by 8, and its conversion time is ADC clock period multiplied by 8. The maximum ADC clock speed should be no more than 6 MHz. The ADC automatically goes into Test mode after conversion, so for one-shot mode, the low-speed ADC clock consumes more power than the high-speed ADC clock. In free running mode, the low-speed ADC clock consumes less power than the high-speed ADC clock.

For additional information, refer to the SARADC_CR2 register on page 142.

24.2.8 SARADC_LCR Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,ACh	SARADC_LCR	DA_L[7:0]								RW : 00

The SAR8 ADC Reference Voltage Generator Control Register (SARADC_LCR) is the ADC DA register used for reference voltage generation control. It is write only in Test mode.

Bits 7 to 0: DA_L[7:0]. The low byte control for reference voltage.

For additional information, refer to the SARADC_LCR register on page 143.

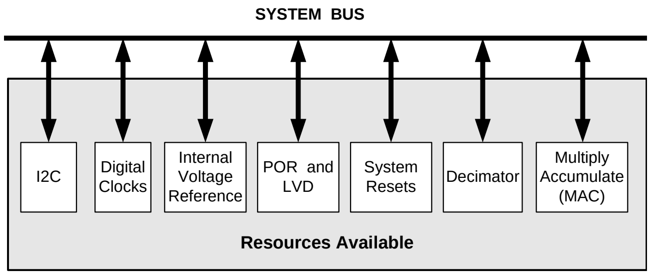
Section F: System Resources

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

Top-Level System Resources Architecture

The figure below displays the top-level architecture of the PSoC's system resources. Each component of the figure is discussed at length in this section.

CY8C24533, CY8C23533, CY8C23433CY8C24633 PSoC Device System Resources



Interpreting the System Resources Documentation

The following table lists the resources available for the CY8C24533, CY8C23533, CY8C23433CY8C24633 (and related PSoC devices).

Availability of System Resources for PSoC Devices

PSoC Part Number	Digital Clocks	I2C	Internal Voltage Ref	POR and LVD	System Resets	Decimator	Multiply Accumulate	SAR8 ADC
CY8C24423A	✓	✓	✓	✓	✓	T1	1	
CY8C24533	✓	✓	✓	✓	✓	T1	1	✓
CY8C23533	✓	✓	✓	✓	✓	T1	1	✓
CY8C23433	✓	✓	✓	✓	✓	T1	1	✓
CY8C24633	✓	✓	✓	✓	✓	T1	1	✓

System Resources Register Summary

The table below lists all the PSoC registers for the system resources, in address order, within their system resource configuration. The bits that are grayed out are reserved bits. If these bits are written, they should always be written with a value of ‘0’.

Note that the CY8C24533, CY8C23533, CY8C23433CY8C24633 are 1 digital row and 2 analog column devices.

Summary Table of the System Resource Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
DIGITAL CLOCK REGISTERS (page 279)										
0,DAh	INT_CLR0	VC3	Sleep	GPIO			Analog 1	Analog 0	V Monitor	RW : 00
0,E0h	INT_MSK0	VC3	Sleep	GPIO			Analog 1	Analog 0	V Monitor	RW : 00
1,DDh	OSC_GO_EN	SLPINT	VC3	VC2	VC1	SYSCLKX2	SYSCLK	CLK24M	CLK32K	RW : 00
1,DEh	OSC_CR4							VC3 Input Select[1:0]		RW : 00
1,DFh	OSC_CR3	VC3 Divider[7:0]								RW : 00
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00
1,E1h	OSC_CR1	VC1 Divider[3:0]				VC2 Divider[3:0]				RW : 00
1,E2h	OSC_CR2	PLLGAIN					EXTCLKEN	RSVD	SYSCLKX2 DIS	RW : 00
MULTIPLY ACCUMULATE (MAC) REGISTERS (page 287)										
0,E8h	MUL0_X	Data[7:0]								W : XX
0,E9h	MUL0_Y	Data[7:0]								W : XX
0,EAh	MUL0_DH	Data[7:0]								R : XX
0,EBh	MUL0_DL	Data[7:0]								R : XX
0,ECh	MAC0_X/ ACC0_DR1	Data[7:0]								RW : 00
0,EDh	MAC0_Y/ ACC0_DR0	Data[7:0]								RW : 00
0,EEh	MAC0_CL0/ ACC0_DR3	Data[7:0]								RW : 00
0,EFh	MAC0_CL1/ ACC0_DR2	Data[7:0]								RW : 00
DECIMATOR REGISTERS (page 293)										
0,E4h	DEC_DH	Data High Byte[7:0]								RC : XX
0,E5h	DEC_DL	Data Low Byte[7:0]								RC : XX
0,E6h	DEC_CR0	IGEN[3:0]				ICLKS0	DCOL[1:0]		DCLKS0	RW : 00
0,E7h	DEC_CR1	ECNT	IDEC	ICLKS3	ICLKS2	ICLKS1	DCLKS3	DCLKS2	DCLKS1	RW : 00
I2C REGISTERS (page 298)										
0,D6h	I2C_CFG		PSelect	Bus Error IE	Stop IE	Clock Rate[1:0]		Enable Master	Enable Slave	RW : 00
0,D7h	I2C_SCR	Bus Error	Lost Arb	Stop Status	ACK	Address	Transmit	LRB	Byte Complete	R : 00
0,D8h	I2C_DR	Data[7:0]								RW : 00
0,D9h	I2C_MSCR					Bus Busy	Master Mode	Restart Gen	Start Gen	R : 00

Summary Table of the System Resource Registers *(continued)*

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
INTERNAL VOLTAGE REFERENCE REGISTER (page 312)										
1,EAh	BDG_TR		AGNDBYP	TC[1:0]		V[3:0]				RW : 00
SYSTEM RESET REGISTERS (page 314)										
0,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW	ECO EX		IRAMDIS	# : 00
0,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	# : XX
POR AND LVD REGISTERS (page 320)										
1,E3h	VLT_CR	SMP		PORLEV[1:0]		LVDTBEN	VM[2:0]			RW : 00
1,E4h	VLT_CMP						PUMP	LVD	PPOR	R : 00

LEGEND

X	The value after power on reset is unknown.
C	Clearable register or bits.
R	Read register or bit(s).
W	Write register or bit(s).
#	Access is bit specific. Refer to the Register Details chapter on page 47 for additional information.

25. Digital Clocks

This chapter discusses the Digital Clocks and their associated registers. It serves as an overview of the clocking options available in the PSoC devices. For detailed information on specific oscillators, see the individual oscillator chapter in the section called "PSoC Core" on page 31. For a complete table of the digital clock registers, refer to the "Summary Table of the System Resource Registers" on page 272. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

25.1 Architectural Description

The PSoC M8C core has a large number of clock sources that increase the flexibility of the PSoC device, as listed in Table 25-1 and illustrated in Figure 25-1.

Table 25-1. System Clocking Signals and Definitions

Signal	Definition
SYSCLKX2	Twice the frequency of SYSCLK.
SYSCLK	Either the direct output of the Internal Main Oscillator or the direct input of the EXTCLK pin while in external clocking mode.
CPUCLK	SYSCLK is divided down to one of eight possible frequencies to create CPUCLK which determines the speed of the M8C. See OSC_CR0 in the Register Definitions section of this chapter.
VC1	SYSCLK is divided down to create Variable Clock 1 (VC1). See OSC_CR1 in the Register Definitions section of this chapter. Division range is from 1 to 16.
VC2	VC1 is divided down to create Variable Clock 2 (VC2). See OSC_CR1 in the Register Definitions section of this chapter. Division range is from 1 to 16.
VC3	Divides down either SYSCLK, VC1, VC2, or SYSCLKX2 to create Variable Clock 3 (VC3). Division range is from 1 to 256. See OSC_CR3 and OSC_CR4 in the Register Definitions section of this chapter.
CLK32K	Either the Internal Low Speed Oscillators or the External Crystal Oscillators output. See OSC_CR0 in the Register Definitions section of this chapter.
CLK24M	The internally generated 24 MHz clock by the IMO. By default, this clock drives SYSCLK; however, an external clock may be used by enabling EXTCLK mode. Also, the IMO may be put into a slow mode using the SLIMO bit which changes the speed of the IMO and the CLK24M to either 6 MHz or 12 MHz in some PSoC devices.
SLEEP	One of four sleep intervals may be selected from 1.95 ms to 1 second. See OSC_CR0 in the Register Definitions section of this chapter.

25.1.1 Internal Main Oscillator

The Internal Main Oscillator (IMO) is the foundation upon which almost all other clock sources in the PSoC device are based. The default mode of the IMO creates a 24 MHz reference clock that is used by many other circuits in the PSoC device. The IMO may also be configured to operate in a PLL mode where the oscillator is locked to a precision 32.768 kHz crystal reference. The PSoC device has an option to replace the IMO with an externally supplied clock that becomes the base for all of the clocks the IMO normally serves. The internal base clock net is called SYSCLK and may be driven by either the IMO or an external clock (EXTCLK).

Whether the external clock or the internal main oscillator is selected, all PSoC device functions are clocked from a derivative of SYSCLK or are resynchronized to SYSCLK. All external asynchronous signals (through row inputs), as well as the selected 32.768 kHz crystal oscillator, are resynchronized to SYSCLK for use in the digital PSoC blocks.

Some PSoC devices contain the option to lower the internal oscillator's system clock from 24 MHz to 6 MHz. See the "Architectural Description" on page 15, in the Internal Main Oscillator chapter, for more information.

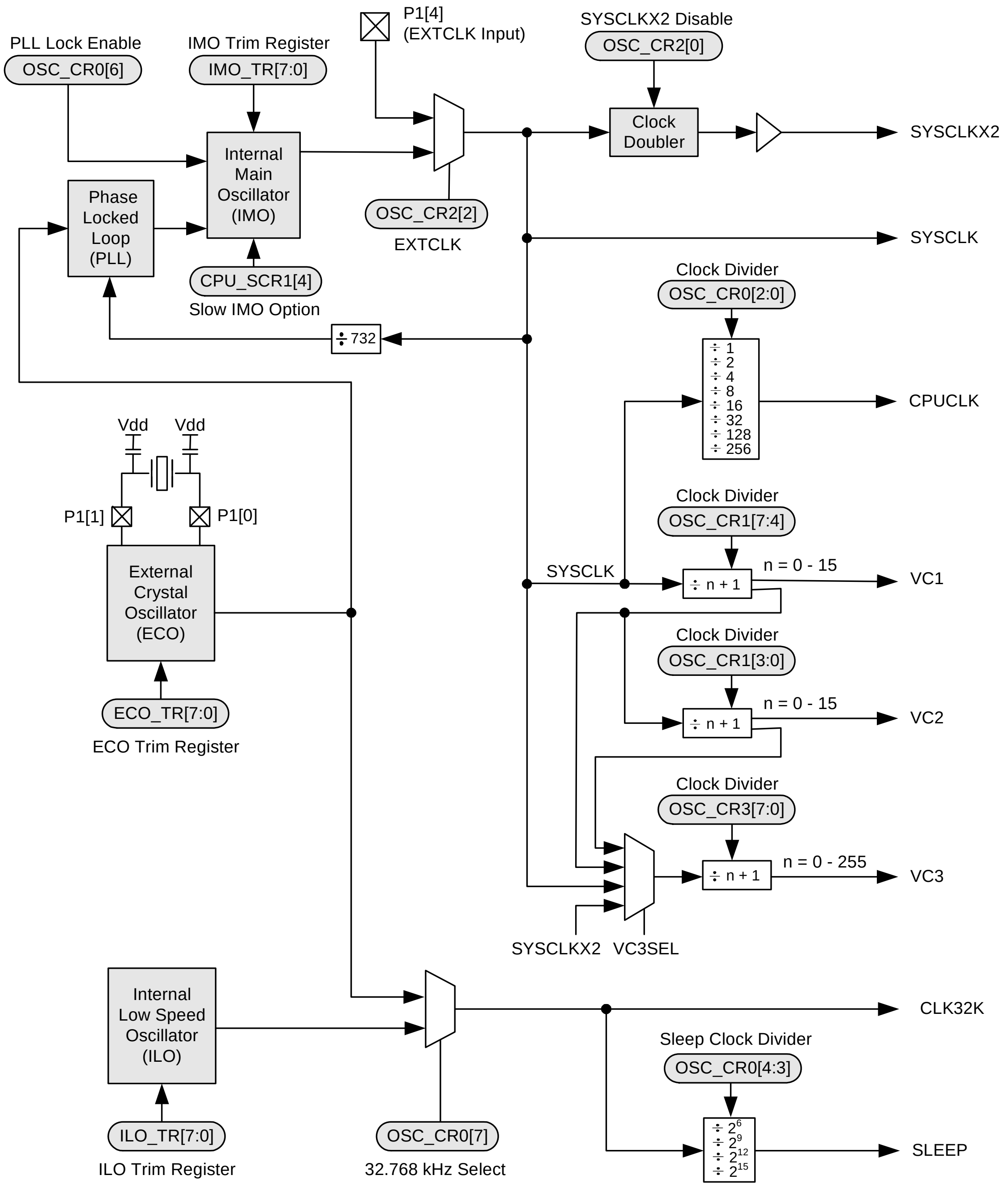
The IMO is discussed in detail in the chapter "Internal Main Oscillator (IMO)" on page 15.

25.1.2 Internal Low Speed Oscillator

The Internal Low Speed Oscillator (ILO) is always on unless the device is operating off an external crystal. The ILO is available as a general clock, but is also the clock source for the sleep and watchdog timers.

The ILO is discussed in detail in the chapter "Internal Low Speed Oscillator (ILO)" on page 19.

Figure 25-1. Overview of PSoC Clock Sources



25.1.3 32.768 kHz Crystal Oscillator

The PSoC may be configured to use an external crystal. The crystal oscillator is discussed in detail in the chapter "External Crystal Oscillator (ECO)" on page 21.

25.1.4 External Clock

The ability to replace the 24 MHz internal main oscillator (IMO), as the device master system clock (SYSCLK) with an externally supplied clock, is a feature in the PSoC device (see Figure 25-1).

Pin P1[4] is the input pin for the external clock. This pin was chosen because it is not associated with any special features such as analog IO, crystal, or In System Serial Programming (ISSP). It is also not physically close to either the P1[0] and P1[1] crystal pins. If P1[4] is selected as the external clock source, the Drive mode of the pin must be set to High Z (not High Z analog).

The user is able to supply an external clock with a frequency between 1 MHz and 24 MHz. The reset state of the EXTCLKEN bit is '0'; and therefore, the device always boots up under the control of the IMO. There is no way to start the system from a reset state with the external clock.

When the EXTCLKEN bit is set, the external clock becomes the source for the internal clock tree, SYSCLK, which drives most PSoC device clocking functions. All external and internal signals, including the 32 kHz clock, whether derived from the internal low speed oscillator (ILO) or the crystal oscillator, are synchronized to this clock source.

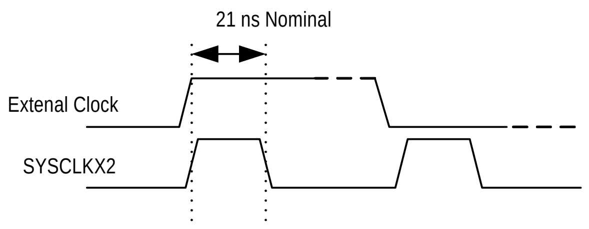
Note that there is no glitch protection in the device for an external clock. Ensure that the external clock is glitch-free. See device datasheet for the clock specifications.

25.1.4.1 Clock Doubler

One of the blocks driven by the system clock is the clock doubler circuit that drives the SYSCLKX2 output. This doubled clock, which is 48 MHz when the IMO is the selected clock (at 24 MHz), may be used as a clock source for the digital PSoC blocks. When the external clock is selected, the SYSCLKX2 signal is still available and serves as a doubler for whatever frequency is input on the external clock pin.

Following the specification for the external clock input ensures that the internal circuitry of the digital PSoC blocks, which is clocked by SYSCLKX2, meets timing requirements. However, since the doubled clock is generated from both edges of the input clock, clock jitter is introduced if the duty cycle deviates greatly from 50 percent. Also, the high time of the clock out of the doubler is fixed at 21 ns, so the duty cycle of SYSCLKX2 is proportional to the inverse of the frequency, as shown in Figure 25-2. Regardless of the input frequency, the high period of SYSCLKX2 is 21 ns nominal.

Figure 25-2. Operation of the Clock Doubler



25.1.4.2 Switch Operation

Switching between the IMO and the external clock may be done in firmware at any time and is transparent to the user. Since all PSoC device resources run on clocks derived from or synchronized to SYSCLK, when the switch is made, analog and digital functions may be momentarily interrupted.

Switch timing depends on whether the CPU clock divider is set for divide by 1, or divide by 2 or greater. In the case where the CPU clock divider is set for divide by 2 or greater, as shown in Figure 25-3, the setting of the EXTCLKEN bit occurs shortly after the rising edge of SYSCLK. The SYSCLK output is then disabled after the next falling edge of SYSCLK, but before the next rising edge. This ensures a glitch-free transition and provides a full cycle of set up time from SYSCLK to output disable. Once the current clock selection is disabled, the enable of the newly selected clock is double synchronized to that clock. After synchronization, on the subsequent negative edge, SYSCLK is enabled to output the newly selected clock.

In the 24 MHz case, as shown in Figure 25-4, the assertion of IOW_ and thus the setting of the EXTCLKEN bit occurs on the falling edge of SYSCLK. Since SYSCLK is already low, the output is immediately disabled. Therefore, the set up time from SYSCLK to disable is one-half SYSCLK.

Figure 25-3. Switch from IMO to the External Clock with a CPU Clock Divider of Two or Greater

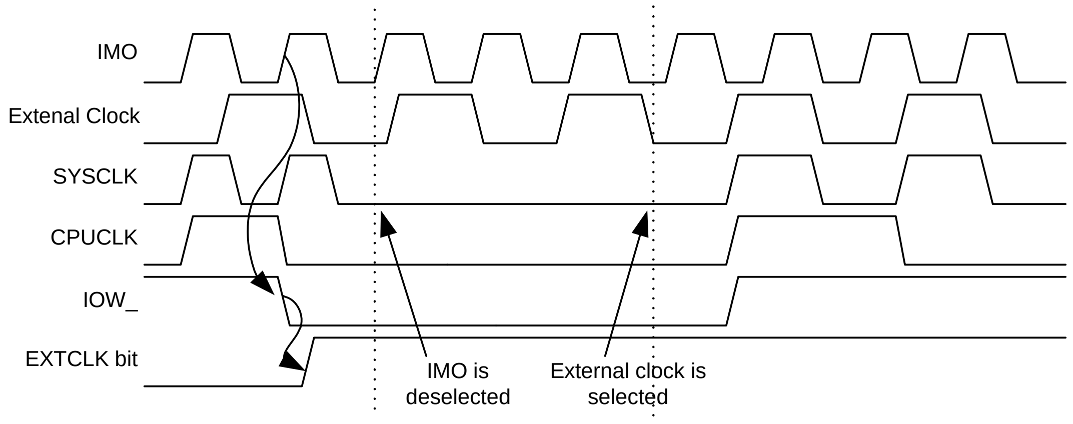
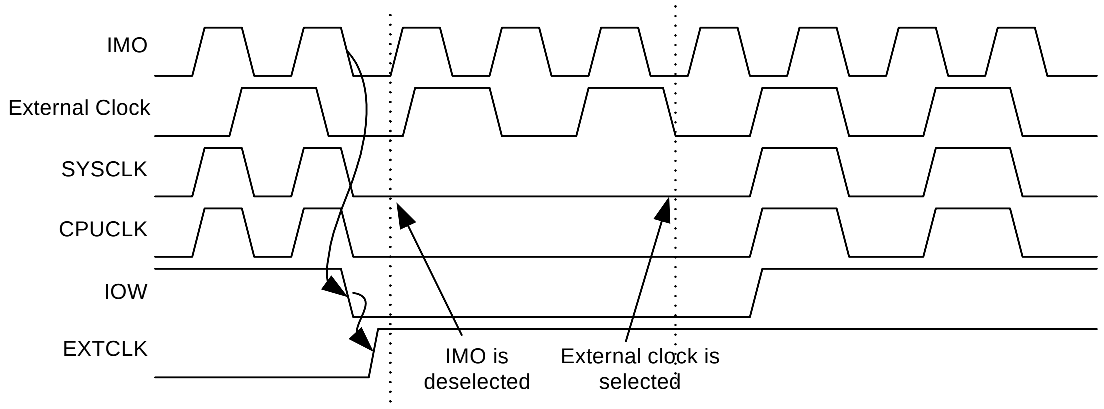


Figure 25-4. Switch from IMO to External Clock with the CPU Running with a CPU Clock Divider of One



25.2 PSoC Device Distinctions

The PSoC device distinctions that apply to the digital clocks are listed as follows.

- In PSoC devices with a part number of CY8C24x23 or CY8C22x13, bit 7 of the OSC_GO_EN register is reserved. However, in PSoC devices with a part number of CY8C24533, CY8C23533, CY8C23433CY8C24633 or CY8C24x23A, bit 7 is not reserved.

25.3 Register Definitions

The following registers are associated with the Digital Clocks and are listed in address order. Each register description has an associated register table showing the bit structure for that register. For a complete table of digital clock registers, refer to the "Summary Table of the System Resource Registers" on page 272.

Only certain bits are accessible to be read or written, such as the INT_CLR0 and INT_MSK0 registers that are analog column dependent. The bits in the tables that are grayed out throughout this manual are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of '0'.

25.3.1 INT_CLR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,DAh	INT_CLR0	VC3	Sleep	GPIO	SAR8 ADC		Analog 1	Analog 0	V Monitor	RW : 00

The Interrupt Clear Register 0 (INT_CLR0) is used to enable the individual interrupt sources' ability to clear posted interrupts.

Bit 7: VC3. The digital clocks only use bit 7 of the INT_CLR0 register for the VC3 clock. This bit controls the VC3 clock interrupt status.

Bits 6 to 0. The INT_CLR0 register holds bits that are used by several different resources. For a full discussion of the INT_CLR0 register, see the INT_CLRx Registers in the Interrupt Controller chapter on page 61.

For additional information, refer to the INT_CLR0 register on page 99.

25.3.2 INT_MSK0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E0h	INT_MSK0	VC3	Sleep	GPIO	SAR8 ADC		Analog 1	Analog 0	V Monitor	RW : 00

The Interrupt Mask Register 0 (INT_MSK0) is used to enable the individual sources' ability to create pending interrupts.

Bit 7: VC3. The digital clocks only use bit 7 of the INT_CLR0 register for the VC3 clock. This bit controls the VC3 clock interrupt enable.

Bits 6 to 0. The INT_MSK0 register holds bits that are used by several different resources. For a full discussion of the INT_MSK0 register, see the INT_MSKx Registers in the Interrupt Controller chapter on page 61.

For additional information, refer to the INT_MSK0 register on page 104.

25.3.3 OSC_GO_EN Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,DDh	OSC_GO_EN	SLPINT	VC3	VC2	VC1	SYSCLKX2	SYSCLK	CLK24M	CLK32K	RW : 00

The Oscillator to Global Outputs Enable Register (OSC_GO_EN) is used to enable tri-state buffers that connect specific system clocks to specific global output even nets.

The OSC_GO_EN register holds eight bits which independently enable a tri-state buffer to drive a clock onto a global net. In all cases, the clock is driven onto one of the nets in the Global Output Even (GOE) bus. In all cases, these bits should only be set and the resulting clock signal on the global be used when the clock frequency is less than or equal to the maximum ***switching*** frequency of the global buses (12 MHz). Therefore, bits 2 and 3 are only useful when the PSoC device is in external clocking mode and bit 1 may never be used.

Note See "PSoC Device Distinctions" on page 278 for more information about this register.

Bit 7: SLPINT. This bit provides the option to connect the sleep interrupt signal to GOE[7]. This is useful in real-time clock applications where very low power is required. By driving the sleep interrupt to a global, it is then routed to a digital PSoC block. The digital PSoC block then counts several sleep interrupts before generating its own interrupt, which is used to bring the PSoC device out of the sleep state.

Bit 6: VC3. This bit enables the driving of the VC3 clock onto GOE[6].

Bit 5: VC2. This bit enables the driving of the VC3 clock onto GOE[5].

Bit 4: VC1. This bit enables the driving of the VC3 clock onto GOE[4].

Bit 3: SYSCLKX2. This bit enables the driving of the SYSCLKX2 clock onto GOE[3].

Bit 2: SYSCLK. This bit enables the driving of the SYSCLK clock onto GOE[2].

Bit 1: CLK24M. This bit enables the driving of the 24 Mhz clock onto GOE[1].

Bit 0: CLK32K. This bit enables the driving of the 32 kHz clock onto GOE[0].

For additional information, refer to the OSC_GO_EN register on page 148.

25.3.4 OSC_CR4 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,DEh	OSC_CR4							VC3 Input Select[1:0]		RW : 00

The Oscillator Control Register 4 (OSC_CR4) selects the input clock to variable clock 3 (VC3).

Bits 1 and 0: VC3 Input Select [1:0]. The VC3 clock net is the only clock net with the ability to generate an interrupt. The input clock of VC3 comes from a configurable source. As shown in Figure 25-1 on page 276, a 4-to-1 mux determines the clock that is used in the input to the VC3 divider. The mux allows either the 48 MHz, 24 MHz, VC1, or VC2 clocks to be used as the input clock to the divider. Because the selection of a clock for the VC3 divider is performed by a simple 4-to-1 mux, ***runt pulses*** and glitches may be injected to the VC3 divider when the OSC_CR4[1:0] bits are changed. Care should be taken to ensure that blocks using the VC3 clock are either disabled when OSC_CR4[1:0] is changed or not sensitive to glitches. Unlike the VC1 and VC2 clock dividers, the VC3 clock divider is 8-bits wide. Therefore, there are 256 valid divider values as indicated by Table 25-3.

Remember that even though the VC3 divider has four choices for the input clock, none of the choices have fixed frequencies for all device configurations. Both the 24 MHz and 48 MHz clocks may have very different frequencies if an external clock is in use. Also, the divider values for the VC1 and VC2 inputs to the mux must be considered.

Table 25-2. OSC_CR4[1:0] Bits: VC3

Bits	Multiplexer Output
00b	SYSCLK
01b	VC1
10b	VC2
11b	SYSCLKX2

For additional information, refer to the OSC_CR4 register on page 149.

25.3.5 OSC_CR3 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,DFh	OSC_CR3	VC3 Divider[7:0]								RW : 00

The Oscillator Control Register 3 (OSC_CR3) selects the divider value for variable clock 3 (VC3).

Bits 7 to 0: VC3 Divider[7:0]. As an example of the flexibility of the clocking structure in PSoC devices, consider a device that is running off of an externally supplied clock at a frequency of 93.7 kHz. This clock value may be divided by the VC1 divider to achieve a VC1 clock net frequency of 5.89 kHz. The VC2 divider could reduce the frequency by another factor of 16, resulting in a VC2 clock net frequency of 366.02 Hz. Finally, the VC3 divider may choose VC2 as its input clock and divide by 256, resulting in a VC3 clock net frequency of 1.43 Hz.

Table 25-3. OSC_CR3[7:0] Bits: VC3 Divider Value

Bits	Divider Source Clock			
	SYSCLKX2	SYSCLK	VC1	VC2
00h	SYSCLKX2	SYSCLK	VC1	VC2
01h	SYSCLKX2 / 2	SYSCLK / 2	VC1 / 2	VC2 / 2
02h	SYSCLKX2 / 3	SYSCLK / 3	VC1 / 3	VC2 / 3
03h	SYSCLKX2 / 4	SYSCLK / 4	VC1 / 4	VC2 / 4
...	...	...	...	...
FCh	SYSCLKX2 / 253	SYSCLK / 253	VC1 / 253	VC2 / 253
FDh	SYSCLKX2 / 254	SYSCLK / 254	VC1 / 254	VC2 / 254
FEh	SYSCLKX2 / 255	SYSCLK / 255	VC1 / 255	VC2 / 255
FFh	SYSCLKX2 / 256	SYSCLK / 256	VC1 / 256	VC2 / 256

The VC3 clock net can generate a system interrupt. Once the input clock and the divider value for the VC3 clock are chosen, only one additional step is needed to enable the interrupt; the VC3 mask bit must be set in register INT_MSK0[7]. Once the VC3 mask bit is set, the VC3 clock generates pending interrupts every number of clock periods equal to the VC3 divider register value plus one. Therefore, if the VC3 divider register's value is 05h (divide by 6), an interrupt occurs every six periods of the VC3's input clock. Another example is if the divider value is 00h (divide by one), an interrupt is generated on every period of the VC3 clock. The VC3 mask bit only controls the ability of a posted interrupt to become pending. Because there is no enable for the VC3 interrupt, VC3 interrupts always post. See the Interrupt Controller chapter on page 61 for more information on posting and pending.

For additional information, refer to the OSC_CR3 register on page 150.

25.3.6 OSC_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00

The Oscillator Control Register 0 (OSC_CR0) is used to configure various features of internal clock sources and clock nets.

Bit 7: 32k Select. By default, the 32 kHz clock source is the Internal Low Speed Oscillator (ILO). Optionally, the 32.768 kHz External Crystal Oscillator (ECO) may be selected.

Bit 6: PLL Mode. This bit is the only bit that directly influences the PLL. When set, it enables the PLL. The EXTCLK bit should be set low during PLL operation.

Bit 5: No Buzz. Normally, when the Sleep bit is set in the CPU_SCR register, all PSoC device systems are powered down, including the bandgap reference. However, to facilitate the detection of POR and LVD events at a rate higher than the sleep interval, the bandgap circuit is powered up periodically (for about 60 µs) at the Sleep System Duty Cycle (set in ECO_TR), which is independent of the sleep interval and typically higher. When the No Buzz bit is set, the Sleep System Duty Cycle value is overridden and the bandgap circuit is forced to be on during sleep. This results in faster response to an LVD or POR event (continuous detection as opposed to periodic), at the expense of slightly higher average sleep current.

Bits 4 and 3: Sleep[1:0]. The available sleep interval selections are shown in Table 25-4. Remember that when the ILO is the selected 32 kHz clock source, sleep intervals are approximate.

Table 25-4. Sleep Interval Selections

Sleep Interval OSC_CR[4:3]	Sleep Timer Clocks	Sleep Period (nominal)	Watchdog Period (nominal)
00b (default)	64	1.95 ms	6 ms
01b	512	15.6 ms	47 ms
10b	4096	125 ms	375 ms
11b	32,768	1 sec	3 sec

Bits 2 to 0: CPU Speed[2:0]. The PSoC M8C operates over a range of CPU clock speeds (Table 25-5), allowing the M8C's performance and power requirements to be tailored to the application.

The reset value for the CPU speed bits is zero. Therefore, the default CPU speed is one-eighth of the clock source. The internal main oscillator is the default clock source for the CPU speed circuit; therefore, the default CPU speed is 3 MHz. See "External Clock" on page 277 for more information on the supported frequencies for externally supplied clocks.

The CPU frequency is changed with a write to the OSC_CR0 register. There are eight frequencies generated from a power-of-two divide circuit, which are selected by a 3-bit code. At any given time, the CPU 8-to-1 clock mux is selecting one of the available frequencies, which is resynchronized to the 24 MHz master clock at the output.

Regardless of the CPU speed bit's setting, if the actual CPU speed is greater than 12 MHz, the 24 MHz operating requirements apply. An example of this scenario is a device that is configured to use an external clock, which is supplying a frequency of 20 MHz. If the CPU speed register's value is 0x03, the CPU clock is 20 MHz. Therefore, the supply voltage requirements for the device are the same as if the part was operating at 24 MHz off of the internal main oscillator. The operating voltage requirements are not relaxed until the CPU speed is at 12.0 MHz or less.

Some devices support the slow IMO option, as discussed in the IMO chapter in the "Architectural Description" on page 15. This offers an option to lower both system and CPU clock speed in order to save power.

Table 25-5. OSC_CR0[2:0] Bits: CPU Speed

Bits	6 MHz Internal Main Oscillator *	24 MHz Internal Main Oscillator	External Clock
000b	750 kHz	3 MHz	EXTCLK/ 8
001b	1.5 MHz	6 MHz	EXTCLK/ 4
010b	3 MHz	12 MHz	EXTCLK/ 2
011b	6 MHz	24 MHz	EXTCLK/ 1
100b	375 kHz	1.5 MHz	EXTCLK/ 16
101b	187.5 kHz	750 kHz	EXTCLK/ 32
110b	93.7 kHz	187.5 kHz	EXTCLK/ 128
111b	23.4 kHz	93.7 kHz	EXTCLK/ 256

* For PSoC devices that support the slow IMO option, see the "Architectural Description" on page 15.

An automatic protection mechanism is available for systems that need to run at peak CPU clock speed but cannot guarantee a high enough supply voltage for that clock speed. See the LVDTBEN bit in the "VLT_CR Register" on page 320 for more information.

For additional information, refer to the OSC_CR0 register on page 151.

25.3.7 OSC_CR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E1h	OSC_CR1	VC1 Divider[3:0]				VC2 Divider[3:0]				RW : 00

The Oscillator Control Register 1 (OSC_CR1) selects the divider value for variable clocks 1 and 2 (VC1 and VC2).

Bits 7 to 4: VC1 Divider[3:0]. The VC1 clock net is one of the variable clock nets available in the PSoC M8C. The source for the VC1 clock net is a simple 4-bit divider. The source for the divider is the 24 MHz system clock; however, if the device is configured to use an external clock, the input to the divider is the external clock. Therefore, the VC1 clock net is not always the result of dividing down a 24 MHz clock. The 4-bit divider that controls the VC1 clock net may be configured to divide, using any integer value between 1 and 16. Table 25-6 lists all values for the VC1 clock net.

Bits 3 to 0: VC2 Divider[3:0]. The VC2 clock net is one of the variable clock nets available in the PSoC M8C. The source for the VC2 clock net is a simple 4-bit divider. The source for the divider is the VC1 clock net. The 4-bit divider that controls the VC2 clock net may be configured to divide, using any integer value between 1 and 16. Table 25-7 lists all values for the VC2 clock net.

Table 25-6. OSC_CR1[7:4] Bits: VC1 Divider Value

Bits	Divider Source Clock	
	Internal Main Oscillator at 24 MHz	External Clock
0h	24 MHz	EXTCLK / 1
1h	12 MHz	EXTCLK / 2
2h	8 MHz	EXTCLK / 3
3h	6 MHz	EXTCLK / 4
4h	4.8 MHz	EXTCLK / 5
5h	4 MHz	EXTCLK / 6
6h	3.43 MHz	EXTCLK / 7
7h	3 MHz	EXTCLK / 8
8h	2.67 MHz	EXTCLK / 9
9h	2.40 MHz	EXTCLK / 10
Ah	2.18 MHz	EXTCLK / 11
Bh	2.00 MHz	EXTCLK / 12
Ch	1.85 MHz	EXTCLK / 13
Dh	1.71 MHz	EXTCLK / 14
Eh	1.6 MHz	EXTCLK / 15
Fh	1.5 MHz	EXTCLK / 16

Table 25-7. OSC_CR1[3:0] Bits: VC2 Divider Value

Bits	Divider Source Clock	
	Internal Main Oscillator	External Clock
0h	(24 / (OSC_CR1[7:4]+1)) / 1	(EXTCLK / (OSC_CR1[7:4]+1)) / 1
1h	(24 / (OSC_CR1[7:4]+1)) / 2	(EXTCLK / (OSC_CR1[7:4]+1)) / 2
2h	(24 / (OSC_CR1[7:4]+1)) / 3	(EXTCLK / (OSC_CR1[7:4]+1)) / 3
3h	(24 / (OSC_CR1[7:4]+1)) / 4	(EXTCLK / (OSC_CR1[7:4]+1)) / 4
4h	(24 / (OSC_CR1[7:4]+1)) / 5	(EXTCLK / (OSC_CR1[7:4]+1)) / 5
5h	(24 / (OSC_CR1[7:4]+1)) / 6	(EXTCLK / (OSC_CR1[7:4]+1)) / 6
6h	(24 / (OSC_CR1[7:4]+1)) / 7	(EXTCLK / (OSC_CR1[7:4]+1)) / 7
7h	(24 / (OSC_CR1[7:4]+1)) / 8	(EXTCLK / (OSC_CR1[7:4]+1)) / 8
8h	(24 / (OSC_CR1[7:4]+1)) / 9	(EXTCLK / (OSC_CR1[7:4]+1)) / 9
9h	(24 / (OSC_CR1[7:4]+1)) / 10	(EXTCLK / (OSC_CR1[7:4]+1)) / 10
Ah	(24 / (OSC_CR1[7:4]+1)) / 11	(EXTCLK / (OSC_CR1[7:4]+1)) / 11
Bh	(24 / (OSC_CR1[7:4]+1)) / 12	(EXTCLK / (OSC_CR1[7:4]+1)) / 12
Ch	(24 / (OSC_CR1[7:4]+1)) / 13	(EXTCLK / (OSC_CR1[7:4]+1)) / 13
Dh	(24 / (OSC_CR1[7:4]+1)) / 14	(EXTCLK / (OSC_CR1[7:4]+1)) / 14
Eh	(24 / (OSC_CR1[7:4]+1)) / 15	(EXTCLK / (OSC_CR1[7:4]+1)) / 15
Fh	(24 / (OSC_CR1[7:4]+1)) / 16	(EXTCLK / (OSC_CR1[7:4]+1)) / 16

For additional information, refer to the OSC_CR1 register on page 152.

25.3.8 OSC_CR2 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E2h	OSC_CR2	PLLGAIN					EXTCLKEN	RSVD	SYSCLKX2 DIS	RW : 00

The Oscillator Control Register 2 (OSC_CR2) is used to configure various features of internal clock sources and clock nets.

Bit 7: PLLGAIN. This is the only bit in the OSC_CR2 register that directly influences the PLL. When set, this bit keeps the PLL in a Low Gain mode. If this bit is held low, the lock time is less than 10 ms. If this bit is held high, the lock time is on the order of 50 ms. After lock is achieved, it is recommended that this bit be forced high to decrease the jitter on the output. If longer lock time is tolerable, the PLLGAIN bit can be held high all the time.

Bit 2: EXTCLKEN. When the EXTCLKEN bit is set, the external clock becomes the source for the internal clock tree, SYSCLK, which drives most PSoC device clocking functions. All external and internal signals, including the 32 kHz clock, whether derived from the internal low speed oscillator (ILO) or the crystal oscillator, are synchronized to this clock source.

If an external clock is enabled, PLL mode should be off. The external clock input is located on port P1[4]. When using this input, the pin Drive mode should be set to High Z (not High Z analog).

Bit 1: RSVD. Reserved bit - This bit should always be 0.

Bit 0: SYSCLKX2DIS. When set, the Internal Main Oscillator's doubler is disabled. This results is a reduction of overall device power, on the order of 1 mA. It is advised that any application that does not require this doubled clock should have it turned off.

For additional information, refer to the OSC_CR2 register on page 153.

26. Multiply Accumulate (MAC)

This chapter presents the Multiply Accumulate (MAC) and its associated registers. The MAC block is a fast 8-bit multiplier or a fast 8-bit multiplier with 32-bit accumulate. For a complete table of the MAC registers, refer to the "Summary Table of the System Resource Registers" on page 272. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

26.1 Architectural Description

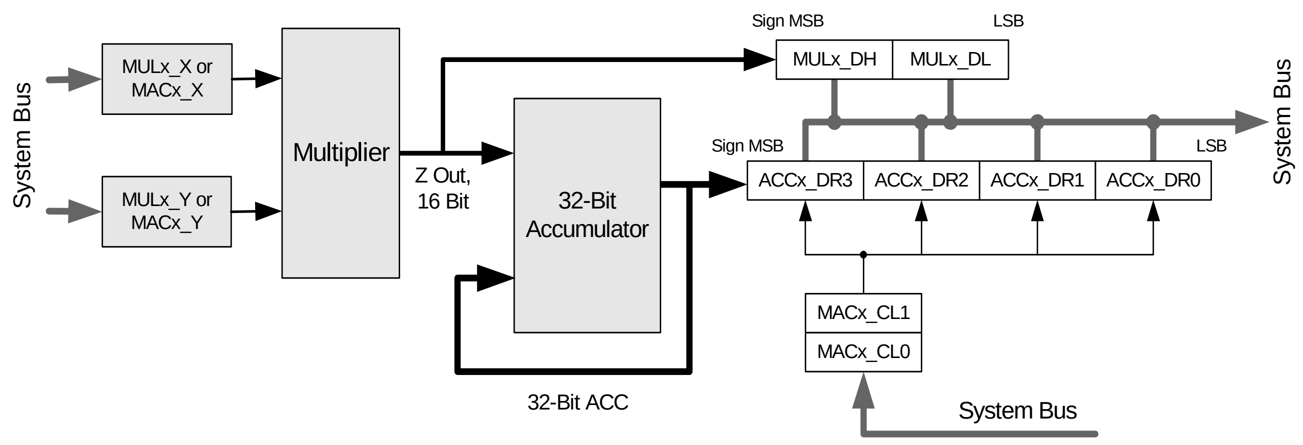
The MAC is a register-based system resource. Its only interface is the system bus; therefore, there are no special clocks or enables that are required to be sourced from digital or analog PSoC blocks.

The architectural presentation of the MAC is illustrated in Figure 26-1.

Table 26-1. MAC Availability

PSoC Part Number	Number of MAC Blocks
CY8C24x23A	1
CY8C24533	1
CY8C23533	1
CY8C23433	1
CY8C24633	1

Figure 26-1. MAC Block Diagram



26.2 Application Description

26.2.1 Multiplication with No Accumulation

For simple multiplication, the MAC block accepts two 8-bit signed numbers as the multiplicands for a multiply operation. The product of the multiplication is stored in a 16-bit signed form. Up to four registers are involved with simple multiplication: MULx_X, MULx_Y, MULx_DH, and MULx_DL.

To execute a multiply, simply write a value to either the MULx_X or MULx_Y registers. Immediately after the write of the multiplicand, the product is available at registers MULx_DH and MULx_DL. After reset of the part at power up or after an external reset, the MAC registers are not reset to zero. Therefore, after the write of the first multiplicand, the product is indeterminate. After the write of the second multiplicand, the product registers are updated with the product of the first and second multiplicands (assuming one of the writes was to MULx_X and the other was to MULx_Y). Multiplication is associative so the order in which you write to X and Y does not matter.

26.2.2 Accumulation After Multiplication

Accumulation of products is a feature that is implemented on top of simple multiplication. When using the MAC to accumulate the products of successive multiplications, two 8-bit signed values are used for input. The product of the multiplication is accumulated as a 32-bit signed value.

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 choosing of input register. The multiply function occurs immediately whenever the MULx_X or the MULx_Y multiplier input registers are written, and the result is available in the MULx_DH and MULx_DL multiplier result registers, as discussed in the 26.2.1 Multiplication with No Accumulation section. The multiply/accumulate function is executed whenever there is a write to the MACx_X or the MACx_Y multiply/accumulate input registers; the result is available in the ACCx_DR3, ACCx_DR2, ACCx_DR1, and ACCx_DR0 accumulator result registers. A write to the MULx_X or MACx_X registers is input as the X value to both the multiply and multiply/accumulate functions. A write to the MULx_Y or MACx_Y registers is input as the Y value to both the multiply and multiply/accumulate functions. A write to the MACx_CL0 or MACx_CL1 registers clears the value in the four accumulate registers.

To clear the accumulated products, simply write to either of the MACx_CLx registers.

26.3 Register Definitions

The following registers are associated with the MAC PSoC Block and are listed in address order. Each register description has an associated register table showing the bit structure for that register. The 'X' in the Access column of some register tables signify that the value after power on reset is unknown. For a complete table of the MAC registers, refer to the "Summary Table of the System Resource Registers" on page 272.

26.3.1 MULx_X Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E8h	MUL0_X	Data[7:0]								W : XX

LEGEND
X The value after power on reset is unknown.

The Multiply Input X Register (MULx_X) is one of two multiplicand registers for the signed 8-bit multiplier in the PSoC MAC.

Bits 7 to 0: Data[7:0]. The multiply X (MULx_X) register is one of two multiplicand registers for the signed 8-bit multiplier in the PSoC MAC. When these write only registers are written, the product of the written value and the current value of the MULx_X registers are calculated.

For additional information, refer to the MULx_X register on page 112.

26.3.2 MULx_Y Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E9h	MUL0_Y	Data[7:0]								W : XX

LEGEND
X The value after power on reset is unknown.

The Multiply Input Y Register (MULx_Y) is one of two multiplicand registers for the signed 8-bit multiplier in the PSoC MAC.

Bits 7 to 0: Data[7:0]. The multiply Y (MULx_Y) register is one of two multiplicand registers for the signed 8-bit multiplier in the PSoC MAC. When these write only registers are written, the product of the written value and the current value of the MULx_Y registers are calculated.

For additional information, refer to the MULx_Y register on page 113.

26.3.3 MULx_DH Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,EAh	MUL0_DH	Data[7:0]								R : XX

LEGEND
X The value after power on reset is unknown.

The Multiply Result High Byte Register (MULx_DH) holds the most significant byte of the 16-bit product.

Bits 7 to 0: Data[7:0]. The product of the multiply operation on the MULx_X and MULx_Y registers is stored as a signed 16-bit value. The read only multiply data high (MUL0_DH and MUL1_DH) registers hold the most significant byte of the 16-bit product.

For additional information, refer to the MULx_DH register on page 114.

26.3.4 MULx_DL Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,EBh	MUL0_DL	Data[7:0]								R : XX

LEGEND
X The value after power on reset is unknown.

The Multiply Result Low Byte Register (MULx_DL) holds the least significant byte of the 16-bit product.

Bits 7 to 0: Data[7:0]. The product of the multiply operation on the MULx_X and MULx_Y registers is stored as a signed 16-bit value. The read only multiply data low (MUL0_DL and MUL1_DL) registers hold the least significant byte of the 16-bit product.

For additional information, refer to the MULx_DL register on page 115.

26.3.5 MACx_X/ACCx_DR1 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,ECh	MAC0_X/ ACC0_DR1	Data[7:0]								RW : 00

The Accumulator Data Register 1 (MACx_X/ACCx_DR1) is the multiply accumulate X register and the second byte of the accumulated value.

Bits 7 to 0: Data[7:0]. This register performs two distinct functions; therefore, two names are used to refer to the same address. When the address is written, a multiply operation with accumulation is performed. The multiply accumulate X (MACx_X) register is one of the two multiplicand registers for the signed 8-bit multiply with accumulate operation. When this register is written, the product of the written value and the current value of the MACx_Y register is calculated, then that product is added to the 32-bit accumulator's value. When this address is read, the accumulator's data register 1 is read. This register holds the second of four bytes used to hold the accumulator's value. This byte is the most significant of the lower 16 bits of the accumulator's value.

For additional information, refer to the MACx_X/ACCx_DR1 register on page 116.

26.3.6 MACx_Y/ACCx_DR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,EDh	MAC0_Y/ ACC0_DR0	Data[7:0]								RW : 00

The Accumulator Data Register 0 (MACx_Y/ACCx_DR0) is the multiply accumulate Y register and the first byte of the accumulated value.

Bits 7 to 0: Data[7:0]. This register performs two distinct functions; therefore, two names are used to refer to the same address. When the address is written, a multiply operation with accumulation is performed. The multiply accumulate Y (MACx_Y) register is one of the two multiplicand registers for the signed 8-bit multiply with accumulate operation. When this register is written, the product of the written value and the current value of the MACx_X register is calculated, then that product is added to the 32-bit accumulator's value. When this address is read, the accumulator's data register 0 is read. This register holds the least significant of four bytes used to hold the accumulator's value.

For additional information, refer to the MACx_Y/ACCx_DR0 register on page 117.

26.3.7 MACx_CL0/ACCx_DR3 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,EEh	MAC0_CL0/ ACC0_DR3	Data[7:0]								RW : 00

The Accumulator Data Register 3 (MACx_CL0/ACCx_DR3) is an accumulator clear register and the fourth byte of the accumulated value.

Bits 7 to 0: Data[7:0]. This register performs two distinct functions; therefore, two names are used to refer to the same address. When the address is written with any value, all 32 bits of the accumulator are reset to zero. When this address is read, the accumulator's data register 3 is read. This register holds the most significant of four bytes used to hold the accumulator's value.

For additional information, refer to the MACx_CL0/ ACCx_DR3 register on page 118.

26.3.8 MACx_CL1/ACCx_DR2 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,EFh	MAC0_CL1/ ACC0_DR2	Data[7:0]								RW : 00

The Accumulator Data Register 2 (MACx_CL1/ACCx_DR2) is an accumulator clear register and the third byte of the accumulated value.

Bits 7 to 0: Data[7:0]. This register performs two distinct functions; therefore, two names are used to refer to the same address. When the address is written with any value, all 32 bits of the accumulator are reset to zero. When this address is read, the accumulator's data register 2 is read. This register holds the third of four bytes used to hold the accumulator's value. This byte is the least significant of the upper 16 bits of the accumulator's value.

For additional information, refer to the MACx_CL1/ ACCx_DR2 register on page 119.

27. Decimator

This chapter explains the PSoC Decimator block and its associated registers. The decimator block is a hardware assist for digital signal processing applications. The decimator is used for delta-sigma ADCs and incremental ADCs. For a complete table of the decimator registers, refer to the “Summary Table of the System Resource Registers” on page 272. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

27.1 Architectural Description

27.1.1 Decimator Block

The decimator block used by theCY8C24533, CY8C23533, CY8C23433 CY8C24633 device family performs either a single or double integration of the discrete-time, discrete-amplitude signal applied to the data input pin of the block. The integrated value is up to 16 bits long and read or cleared by way of a register interface.

Because the data input to the decimator is only one bit, the input signal's amplitude can only be one of two values:

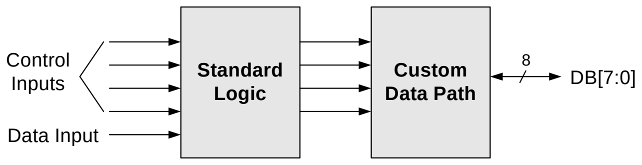
Table 27-1. Input Signal Amplitude

Encoding	Weight
0	-1
1	+1

Because the input signal is a discrete-time signal, the weight of each encoding is analogous to the area under the signal for that instant in time. Therefore, to integrate the signal, the sum of the weights must be calculated over a period of time. When the decimator is configured as a single integrator, this is exactly what happens. For each period of the input clock, the current area (integral value) is either increased by one (weight = +1, encoding = 1) or decreased by one (weight = -1, encoding = 0).

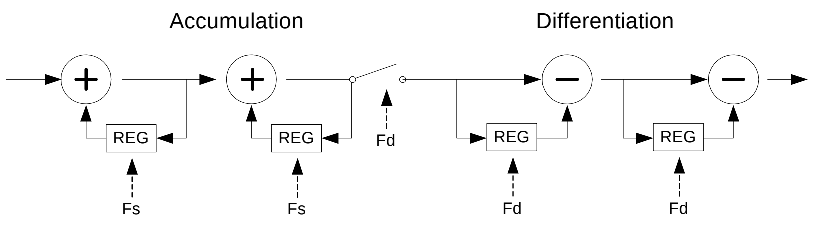
The major functional units within the decimator block are illustrated below.

Figure 27-1. Decimator Architecture



The decimator is divided into two major functional units: a logic block composed of standard logic cells and a custom digital data path block. The logic block interfaces to all of the decimator block pins, except for the data bus. The logic block takes the decimator's inputs and creates the necessary control input to the custom block. The custom block is where the adding and storing of accumulated values occurs. The custom block also interfaces to the data bus.

Figure 27-2. The principle of operation of a Sinc2 decimation filter is inferred in Figure 27-2 and Equation 1. The decimator's custom data path follows the Accumulation stage of Figure 27-2, in principle. The Differentiation is accomplished with external firmware in user modules.Sinc^2 Filter Block Diagram



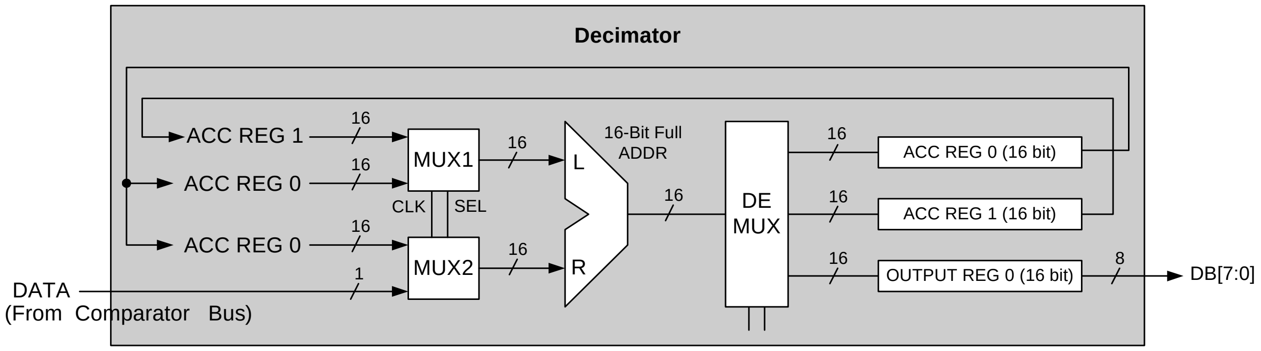
H(z) = Transfer function of Sinc^N filter with a decimation rate of M

$$H(z) = (1/M)^N (1-Z^{-M})^N (1/ (1-Z^{-1}))^N$$

Sinc^2 Transfer Function **Equation 1**

There are three 16-bit internal registers in the type 1 decimator: A0, A1, and AB (see Figure 27-3). The A0 register is used to store the 16-bit sum from the Data + A0 calculation. The A1 register is used to store the 16-bit sum from the A0 + A1 calculation. The AB register is the register that is readable by way of the data bus. The A0 and A1 registers are not accessible from outside the block.

Figure 27-3. Decimator Custom Data Path



27.2 Register Definitions

The following registers are associated with the Decimator and listed in address order. Each register description has an associated register table showing the bit structure for that register. The bits that are grayed out in the tables are reserved bits and are not detailed in the register description that follows. Reserved bits should always be written with a value of '0'. For a complete table of decimator registers, refer to the "Summary Table of the System Resource Registers" on page 272.

27.2.1 DEC_DH Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E4h	DEC_DH	Data High Byte[7:0]								RC : XX

LEGEND
C: Clearable register or bits.
X: The value for power after reset is unknown.

The Decimator Data High Register (DEC_DH) is a dual purpose register used to read the high byte of the decimator's output or clear the decimator.

Bits 7 to 0: Data High Byte[7:0]When the register is read, the most significant byte of the 16-bit decimator value is returned. Depending on how the decimator is configured, this value is either the result of the second integration or the high byte of the 16-bit counter.

The second function of the DEC_DH register is activated whenever the register is written: That function is to clear the decimator value. When the DEC_DH register is written, the decimator's value is cleared regardless of the value written. Either the DEC_DH or DEC_DL registers may be written to clear the decimator's value. Note that this register does not reset to 00h. The DEC_DH register resets to an indeterminate value.

For additional information, refer to the DEC_DH register on page 108.

27.2.2 DEC_DL Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E5h	DEC_DL	Data Low Byte[7:0]								RC : XX

LEGEND
C: Clearable register or bits.
X: The value for power after reset is unknown.

The Decimator Data Low Register (DEC_DL) is a dual purpose register used to read the low byte of the decimator's output or clear the decimator.

Bits 7 to 0: Data Low Byte[7:0]. When the register is read, the most significant byte of the 16-bit decimator value is returned. Depending on how the decimator is configured, this value is either the result of the second integration or the high byte of the 16-bit counter.

The second function of the DEC_DL register is activated whenever the register is written: That function is to clear the decimator value. When the DEC_DL register is written, the decimator's value is cleared regardless of the value written. Either the DEC_DH or DEC_DL registers may be written to clear the decimator's value. Note that this register does not reset to 00h. The DEC_DL register resets to an indeterminate value.

For additional information, refer to the DEC_DL register on page 109.

27.2.3 DEC_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E6h	DEC_CR0			IGEN[1:0]		ICLKS0	DCOL[1:0]		DCLKS0	RW : 00

The Decimator Control Register 0 (DEC_CR0) contains control bits to access hardware support for both the Incremental ADC and the DELISG ADC.

Bits 5 to 4: IGEN[1:0]. For incremental support, the upper four bits, IGEN[3:0], select which column comparator bit is gated by the output of a digital block. The output of that digital block is typically a PWM signal; the high time of which corresponds to the ADC conversion period. This ensures that the comparator output is only processed for the precise conversion time. The digital block selected for the gating function is controlled by ICLKS0 in this register, and ICLKS3, ICLKS2, and ICLKS1 bits in the DEC_CR1 register.

Bit 3: ICLKS0. In conjunction with the ICLKS1, ICLKS2, and ICLKS3 bits in the DEC_CR1 register, these bits select up to 1 of 16 digital blocks (depending on PSoC device resources) to provide the gating signal for an incremental ADC conversion.

Bits 2 and 1: DCOL[1:0]. The DELSIG ADC uses the hardware decimator to do a portion of the post processing computation on the comparator signal. DCOL[1:0] selects the column source for the decimator data (comparator bit) and clock input (PHI clocks).

Bit 0: DCLKS0. The decimator requires a timer signal to sample the current decimator value to an output register that may subsequently be read by the CPU. This timer period is set to be a function of the DELSIG conversion time and may be selected from up to one of eight digital blocks (depending on the PSoC device resources) with DCLKS0 in this register and DCLKS3, DCLKS2, and DCLKS1 in the DEC_CR1 register.

For additional information, refer to the DEC_CR0 register on page 110.

27.2.4 DEC_CR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E7h	DEC_CR1	ECNT	IDEC	ICLKS3	ICLKS2	ICLKS1	DCLKS3	DCLKS2	DCLKS1	RW : 00

The Decimator Control Register 1 (DEC_CR1) is used to configure the decimator prior to using it.

Bit 7: ECNT. The ECNT bit is a mode bit that controls the operation of the decimator hardware block. By default, the decimator is set to a double integrate function, for use in hardware DELSIG processing. When the ECNT bit is set, the decimator block converts to a single integrate function. This gives the equivalent of a 16-bit counter suitable for use in hardware support for an Incremental ADC function.

This bit is only available in PSoC devices with a type 1 decimator .

Bit 6: IDEC. Any function using the decimator requires a digital block timer to sample the current decimator value. Normally, the positive edge of this signal causes the decimator output to be sampled. However, when the IDEC bit is set, the negative edge of the selected digital block input causes the decimator value to be sampled.

Bits 5 to 0: ICLKSx and DCLKSx. The ICLKS3, ICLKS2, ICLKS1, DCLKS3, DCLKS2, and DCLKS1 bits in this register select the digital block sources for Incremental and DELSIGN ADC hardware support (see the DEC_CR0 register).

For additional information, refer to the DEC_CR1 register on page 111.

28. I^{2}C

This chapter explains the I^{2}C™ block and its associated registers. The I2C communications block is a serial processor designed to implement a complete I2C slave or master. For a complete table of the I2C registers, refer to the "Summary Table of the System Resource Registers" on page 272. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

28.1 Architectural Description

The I2C communications block is a serial to parallel processor, designed to interface the PSoC device to a two-wire I2C serial communications bus. To eliminate the need for excessive M8C microcontroller intervention and overhead, the block provides I2C-specific support for status detection and generation of framing bits.

The I2C block directly controls the data (SDA) and clock (SCL) signals to the external I2C interface, through connections to two dedicated GPIO pins. The PSoC device firmware interacts with the block through IO (input/output) register reads and writes, and firmware synchronization is implemented through polling and/or interrupts.

PSoC I2C features include:

- Master/slave, transmitter/receiver operation
- Byte processing for low CPU overhead
- Interrupt or polling CPU interface
- Master clock rates: 50K, 100K, 400K
- Multi-master clock synchronization
- Multi-master mode arbitration support
- 7- or 10-bit addressing (through firmware support)
- SMBus operation (through firmware support)

Hardware functionality provides basic I2C control, data, and status primitives. A combination of hardware support and firmware command sequencing provides a high degree of flexibility for implementing the required I2C functionality.

Hardware limitations in regards to I2C are as follows:

1. There is no hardware support for automatic address comparison. When Slave mode is enabled, every slave address causes the block to interrupt the PSoC device and possibly stall the bus.
2. Since receive and transmitted data are not buffered, there is no support for automatic receive acknowledge. The M8C microcontroller must intervene at the boundary of each byte and either send a byte or ACK received bytes.

The I2C block is designed to support a set of primitive operations and detect a set of status conditions specific to the I2C protocol. These primitive operations and conditions are manipulated and combined at the firmware level to support the required data transfer modes. The CPU sets up control options and issue commands to the unit through IO writes and obtains status through IO reads and interrupts.

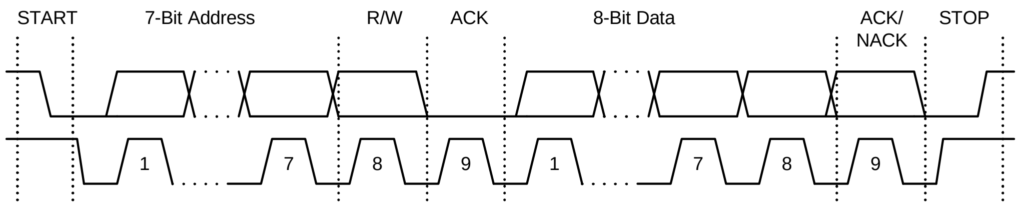
The block operates as either a slave, a master, or both. When enabled in Slave mode, the unit is always listening for a start condition, or sending or receiving data. Master mode works in conjunction with Slave mode. The master supplies the ability to generate the start or stop condition and determine if other masters are on the bus. For Mult-Master mode, clock synchronization is supported. If Master mode is enabled and Slave mode is not enabled, the block does not generate interrupts on externally generated start conditions.

28.1.1 Basic I^{2}C Data Transfer

Figure 28-1 shows the basic form of data transfers on the I2C bus with a 7-bit address format. (For a more detailed description, see the Phillips Semiconductors' I^{2}C™ Specification, version 2.1.)

A start condition (generated by the master) is followed by a data byte, consisting of a 7-bit slave address (there is also a 10-bit address mode) and a Read/Write (RW) bit. The RW bit sets the direction of data transfer. The addressed slave is required to acknowledge (ACK) the bus by pulling the data line low during the ninth bit time. If the ACK is received, the transfer proceeds and the master transmits or receives an indeterminate number of bytes, depending on the RW direction. If the slave does not respond with an ACK for any reason, a stop condition is generated by the master to terminate the transfer or a restart condition may be generated for a retry attempt.

Figure 28-1. Basic I^{2}C Data Transfer with 7-Bit Address Format



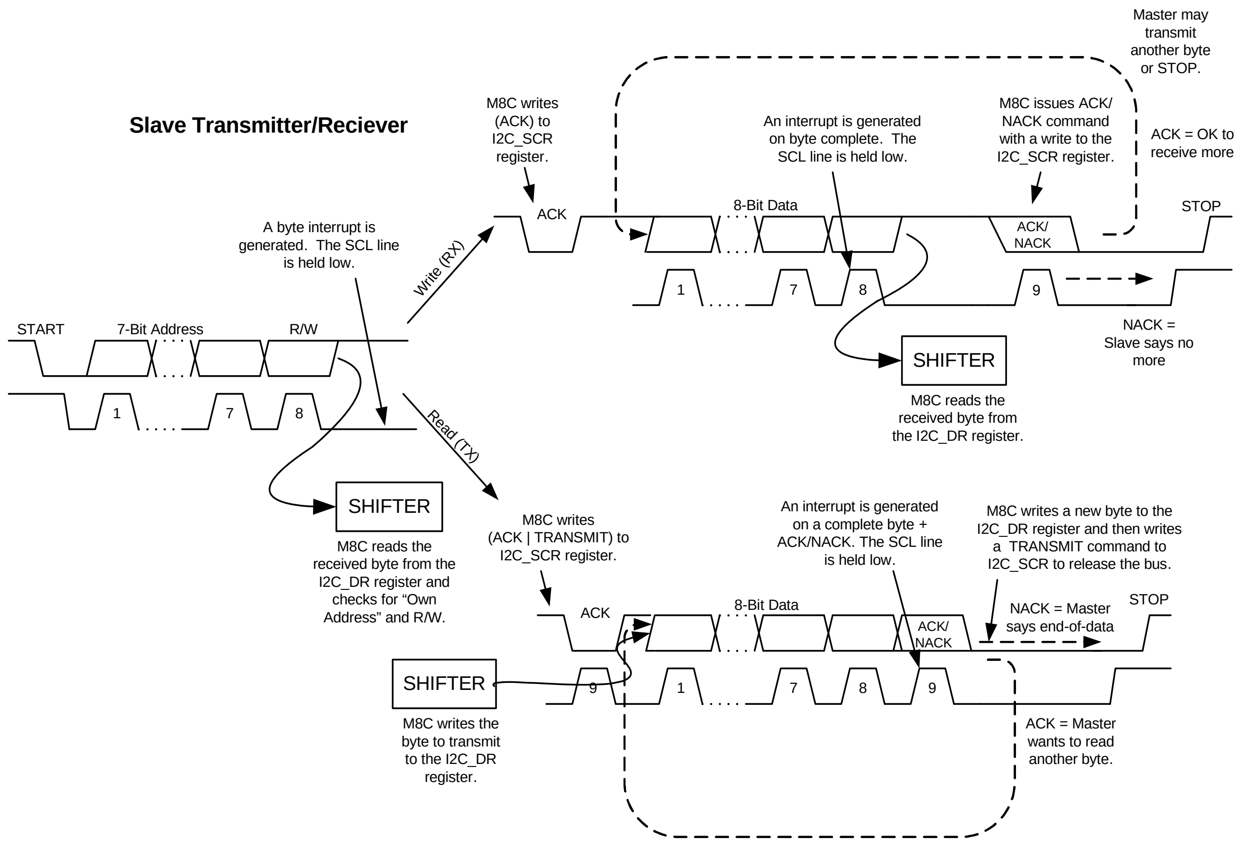
28.2 Application Description

28.2.1 Slave Operation

Assuming Slave mode is enabled, it is continually listening to or on the bus for a start condition. When detected, the transmitted Address/RW byte is received and read from the I2C block by firmware. At the point where eight bits of the address/RW byte are received, a byte complete interrupt is generated. On the following low of the clock, the bus is stalled by holding the SCL line low, until the PSoC device has a chance to read the address byte and compare it to its own address. It issues an ACK or NACK command based on that comparison.

If there is an address match, the RW bit determines how the PSoC device sequences the data transfer in Slave mode, as shown in the two branches of Figure 28-2. I2C handshaking methodology (slave holds the SCL line low to "stall" the bus) is used as necessary, to give the PSoC device time to respond to the events and conditions on the bus. Figure 28-2 is a graphical representation of a typical data transfer from the slave perspective.

Figure 28-2. Slave Operation


Slave Transmitter/Reciever
Master may transmit another byte or STOP.
M8C writes (ACK) to I2C_SCR register.
An interrupt is generated on byte complete. The SCL line is held low.
M8C issues ACK/ NACK command with a write to the I2C_SCR register.
ACK = OK to receive more
ACK
8-Bit Data
ACK/ NACK
STOP
A byte interrupt is generated. The SCL line is held low.
Write (RX)
NACK = Slave says no more
SHIFTER
M8C reads the received byte from the I2C_DR register.
START
7-Bit Address
R/W
Read (TX)
SHIFTER
M8C reads the received byte from the I2C_DR register and checks for "Own Address" and R/W.
M8C writes (ACK | TRANSMIT) to I2C_SCR register.
An interrupt is generated on a complete byte + ACK/NACK. The SCL line is held low.
M8C writes a new byte to the I2C_DR register and then writes a TRANSMIT command to I2C_SCR to release the bus.
ACK
8-Bit Data
ACK/ NACK
NACK = Master says end-of-data
STOP
SHIFTER
M8C writes the byte to transmit to the I2C_DR register.
ACK = Master wants to read another byte.
1
7
8
9

28.2.2 Master Operation

To prepare for a Master mode transaction, the PSoC device determines if the bus is free. This is done by polling the Bus-Busy status. If busy, interrupts are enabled to detect a stop condition. Once it is determined that the bus is available, firmware writes the address byte into the I2C_DR register and sets the Start Gen bit in the I2C_MSCR register.

If the slave sub-unit is not enabled, the block is in Master Only mode. In this mode, the unit does not generate interrupts or stall the I2C bus on externally generated start conditions.

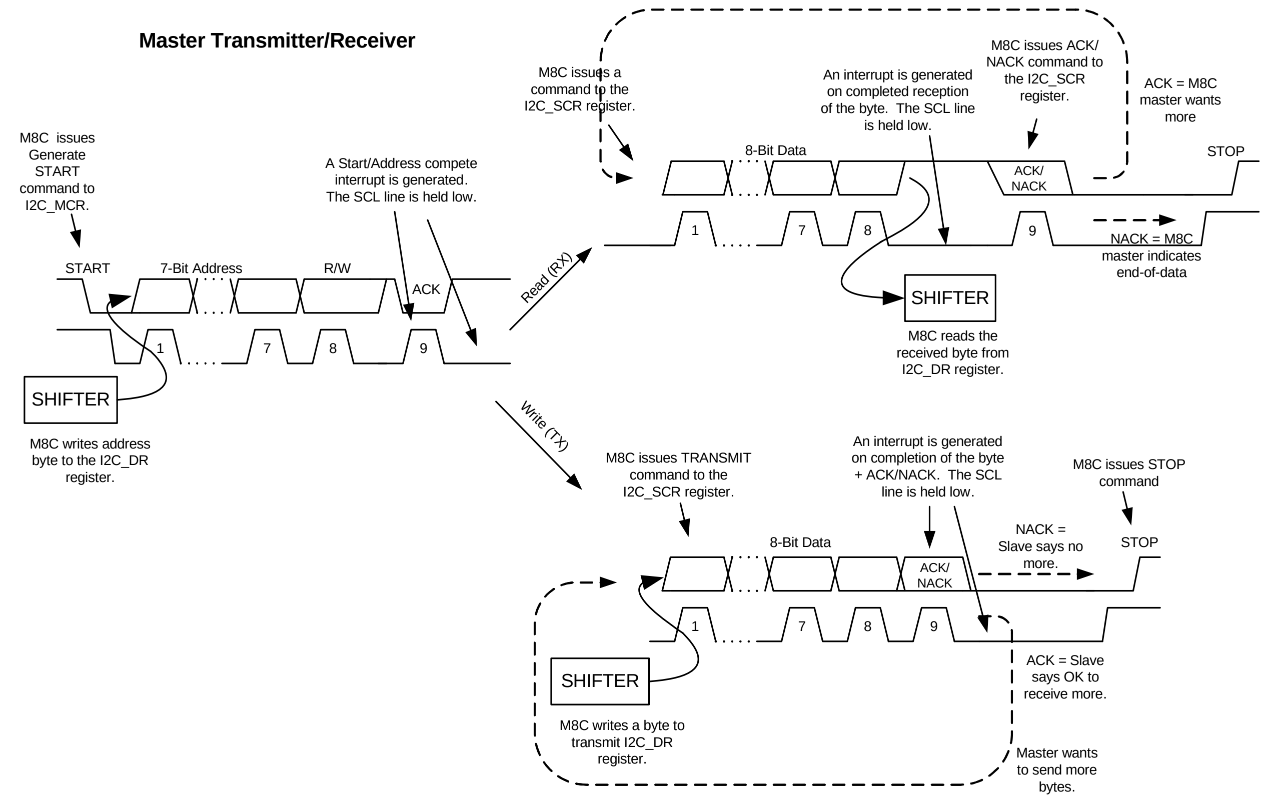
In a multi-master environment there are two additional outcomes possible:

1. The PSoC device is too late to reserve the bus as a master, and another master has generated a start and sent an Address/RW byte. In this case, the unit as a master fails to generate a start and is forced into Slave mode. The start is pending and eventually occurs at a later time when the bus becomes free.

   When the interrupt occurs in Slave mode, the PSoC device determines that the Start command was unsuccessful by reading the I2C_MSCR register Start bit, which is reset on successful start from this unit as master. If this bit is still a ‘1’ on the Start/Address interrupt, it means that the unit is operating in Slave mode. In this case, the data register has the master’s address data.
2. If another master starts a transmission at the same time as this unit, arbitration occurs. If this unit loses the arbitration, the LostArb status bit is set. In this case, the block releases the bus and switches to slave operation. When the Start/Address interrupt occurs, the data register has the winning master’s address data.

Figure 28-3 is a graphical representation of a typical data transfer from the master perspective.

Figure 28-3. Master Operation


Master Transmitter/Receiver
M8C issues Generate START command to I2C_MCR.
START
7-Bit Address
R/W
ACK
1
7
8
9
SHIFTER
M8C writes address byte to the I2C_DR register.
A Start/Address compete interrupt is generated. The SCL line is held low.
Read (RX)
Write (TX)
M8C issues a command to the I2C_SCR register.
An interrupt is generated on completed reception of the byte. The SCL line is held low.
M8C issues ACK/NACK command to the I2C_SCR register.
ACK = M8C master wants more
8-Bit Data
ACK/NACK
STOP
NACK = M8C master indicates end-of-data
M8C reads the received byte from I2C_DR register.
M8C issues TRANSMIT command to the I2C_SCR register.
An interrupt is generated on completion of the byte + ACK/NACK. The SCL line is held low.
M8C issues STOP command
NACK = Slave says no more.
ACK = Slave says OK to receive more.
Master wants to send more bytes.
M8C writes a byte to transmit I2C_DR register.

28.3 Register Definitions

The following registers are associated with I2C and are listed in address order. Each register description has an associated register table showing the bit structure for that register. The bits in the tables that are grayed out are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of '0'. For a complete table of I2C registers, refer to the "Summary Table of the System Resource Registers" on page 272.

28.3.1 I2C_CFG Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D6h	I2C_CFG		PSelect	Bus Error IE	Stop IE	Clock Rate[1:0]		Enable Master	Enable Slave	RW : 00

The I2C Configuration Register (I2C_CFG) is used to set the basic operating modes, baud rate, and selection of interrupts.

The bits in this register control baud rate selection and optional interrupts. The values are typically set once for a given configuration. The bits in this register are all RW.

Table 28-1. I2C_CFG Configuration Register

Bit	Access	Description	Mode
6	RW	I2C Pin Select 0 = P1[7], P1[5] 1 = P1[1], P1[0]	Master/ Slave
5	RW	Bus Error IE Bus error interrupt enable. 0 = Disabled. 1 = Enabled. An interrupt is generated on the detection of a Bus Error.	Master Only
4	RW	Stop IE Stop interrupt enable. 0 = Disabled. 1 = Enabled. An interrupt is generated on the detection of a Stop Condition.	Master/ Slave
3:2	RW	Clock Rate 00 = 100K Standard Mode 01 = 400K Fast Mode 10 = 50K Standard Mode 11 = Reserved	Master/ Slave
1	RW	Enable Master 0 = Disabled 1 = Enabled	Master/ Slave
0	RW	Enable Slave 0 = Disabled 1 = Enabled	Master/ Slave

Bit 6: PSelect. With the default value of zero, the I2C pins are P1[7] for clock and P1[5] for data. When this bit is set, the pins for I2C switch to P1[1] for clock and P1[0] for data. This bit may not be changed while either the Enable Master or Enable Slave bits are set. However, the PSelect bit may be set at the same time as the enable bits. The two sets of pins that may be used on I2C are not equivalent. The default set, P1[7] and P1[5], are the preferred set. The alternate set, P1[1] and P1[0], are provided so that I2C may be used with 8-pin PSoC parts.

If In-System Serial Programming (ISSP) is used and the alternate I2C pin set is also used, it is necessary to take into account the interaction between the PSoC Test Controller and the I2C bus. The interface requirements for ISSP should be reviewed to ensure that they are not violated.

Even if ISSP is not used, pins P1[1] and P1[0] respond differently to a POR or XRES event than other IO pins. After an XRES event, both pins are pulled down to ground by going into the resistive zero Drive mode, before reaching the High Z Drive mode.After a POR event, P1[0] drives out a one, then goes to the resistive zero state for some time, and finally reaches the High Z Drive mode state. After POR, P1[1] goes into a resistive zero state for a while, before going to the High Z Drive mode.

Another issue with selecting the alternate I2C pins set is that these pins are also the crystal pins. Therefore, a crystal may not be used when the alternate I2C pin set is selected.

Bit 5: Bus Error IE (Interrupt Enable). This bit controls whether the detection of a bus error generates an interrupt. A bus error is typically a misplaced start or stop.

This is an important interrupt with regards to master operation. When there is a misplaced start or stop on the I2C bus, all slave devices (including this device, if Slave mode is enabled) reset the bus interface and synchronize to this signal. However, when the hardware detects a bus error in Master Mode operation, the device releases the bus and transitions to an idle state. In this case, a master operation in progress never has any further status or interrupts associated with it. Therefore, the master may not be able to determine the status of that transaction. An immediate bus error interrupt informs the master that this transfer did not succeed.

Bit 4: Stop IE (Interrupt Enable). When this bit is set, a master or slave can interrupt on stop detection. The status bit associated with this interrupt is the Stop Status bit in the Slave Status and Control register. When the Stop Status bit transitions from ‘0’ to ‘1’, the interrupt is generated. It is important to note that the Stop Status bit is not automatically cleared. Therefore, if it is already set, no new interrupts are generated until it is cleared by firmware.

Bits 3 and 2: Clock Rate[1:0]. These bits offer a selection of three sampling and bit rates. All block clocking is based on the SYSCLK input, which is nominally 24 MHz (unless the PSoC device is in external clocking mode). The sampling rate and the baud rate are determined as follows:

- Sample Rate = SYSCLK/Pre-scale Factor
- Baud Rate = 1/(Sample Rate x Samples per Bit)

The nominal values, when using the internal 24 MHz oscillator, are shown in Table 28-2.

Table 28-2. I^{2}C Clock Rates

Clock Rate [1:0]	I2C Mode	SYSCLK Pre-scale Factor	Samples per Bit	Internal Sampling Freq./Period (24 MHz)	Master Baud Rate (nominal)	Start/Stop Hold Time (8 clocks)
00b	Standard	/16	16	1.5 MHz/667 ns	93.75 kHz	5.3 µs
01b	Fast	/4	16	6 MHz/167 ns	375 kHz	1.33 µs
10b	Standard	/16	32	1.5 MHz/667 ns	46.8 kHz	10.7 µs
11b	Reserved					

When clocking the input with a frequency other than 24 MHz (for example, clocking the PSoC device with an external clock), the baud rates and sampling rates scale accordingly. Whether the block works in a Standard Mode or Fast Mode system depends on the sample rate. The sample rate must be sufficient to resolve bus events, such as start and stop conditions. (See the Phillips Semiconductors’ I^{2}C™ Specification, version 2.1, for minimum start and stop hold times.)

Bit 1: Enable Master. When this bit is set, the Master Status and Control register is enabled (otherwise it is held in reset) and I2C transfers can be initiated in Master mode. When the master is enabled and operating, the block clocks the I2C bus at one of three baud rates, defined in the Clock Rate register. When operating in Master mode, the hardware is multi-master capable, implementing both clock synchronization and arbitration. If the Slave Enable bit is not set, the block operates in Master Only mode. All external start conditions are ignored (although the Bus Busy status bit still keeps track of bus activity). Block enable synchronizes to the SYSCLK clock input (see “Timing Diagrams” on page 304).

Bit 0: Enable Slave. When the slave is enabled, the block generates an interrupt on any start condition and an address byte that it receives, indicating the beginning of an I2C transfer. When operating as a slave, the block is clocked from an external master. Therefore, the block works at any frequency up to the maximum defined by the currently selected clock rate. The internal clock is only used in Slave mode to ensure that there is adequate set up time from data output to the next clock on the release of a slave stall. When the Enable Slave and Enable Master bits are both ‘0’, the block is held in reset and all status cleared. See Figure 28-3 for a description of the interaction between the Master/Slave Enable bits. Block enable synchronizes to the SYSCLK clock input (see “Timing Diagrams” on page 304).

Table 28-3. Enable Master/Slave Block Operation

Enable Master	Enable Slave	Block Operation
No	No	Disabled: The block is disconnected from the GPIO pins, P1[5] and P1[7]. (The pins may be used as general purpose IO.) When either the master or slave is enabled, the GPIO pins are under control of the I2C hardware and are unavailable. All internal registers (except I2C_CFG) are held in reset.
No	Yes	Slave Only Mode: Any external start condition causes the block to start receiving an address byte. Regardless of the current state, any start resets the interface and initiates a receive operation. Any stop causes the block to revert to an idle state The I2C_MSCR register is held in reset.
Yes	No	Master Only Mode: External Start conditions are ignored in this mode. No Byte Complete interrupts on external traffic are generated, but the Bus Busy status bit continues to capture Start and Stop status, and thus may be polled by the master to determine if the bus is available. Full multi-master capability is enabled, including clock synchronization and arbitration. The block generates a clock based on the setting in the Clock Rate register
Yes	Yes	Master/Slave Mode: Both master and slave are operational in this mode. The block may be addressed as a slave, but firmware may also initiate Master mode transfers. In this configuration, when a master loses arbitration during an address byte, the hardware reverts to Slave mode and the received byte generates a slave address interrupt.

For additional information, refer to the I2C_CFG register on page 94.

28.3.2 I2C_SCR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D7h	I2C_SCR	Bus Error	Lost Arb	Stop Status	ACK	Address	Transmit	LRB	Byte Complete	# : 00

LEGEND

# Access is bit specific. Refer to Table 28-4 for detailed bit descriptions.

The I2C Status and Control Register (I2C_SCR) is used by both master and slave to control the flow of data bytes and to keep track of the bus state during a transfer.

This register contains status bits, for determining the state of the current I2C transfer, and control bits, for determining the actions for the next byte transfer. At the end of each byte transfer, the I2C hardware interrupts the M8C microcontroller and stalls the I2C bus on the subsequent low of the clock, until the PSoC device intervenes with the next command. This register may be read as many times as necessary; but on a subsequent write to this register, the bus stall is released and the current transfer continues.

There are six status bits: Byte Complete, LRB, Address, Stop Status, Lost Arb, and Bus Error. These bits have Read/Clear (R/C) access, which means that they are set by hardware but may be cleared by a write of ‘0’ to the bit position. Under certain conditions, status is cleared automatically by the hardware. These cases are noted in Table 28-4.

There are two control bits: Transmit and ACK. These bits have RW access and may be cleared by hardware.

Bit 7: Bus Error. The Bus Error status detects misplaced start or stop conditions on the bus. These may be due to noise, rogue devices, or other devices that are not yet synchronized with the I2C bus traffic. According to the I2C specification, all compatible devices must reset their interface on a received start or stop. This is a natural thing to do in Slave mode, because a start initiates an address reception and a stop idles the slave. In the case of a master, this event forces the master to release the bus and idle. However, since a master does not respond to external start or stop conditions, an immediate interrupt on this event allows the master to continue to keep track of the bus state.

A bus error is defined as follows. A start is only valid if the block is idle (master or slave) or a slave receiver is ready to receive the first bit of a new byte after an ACK. Any other timing for a start condition causes the Bus Error bit to be set. A stop is only valid if the block is idle or a slave receiver is ready to receive the first bit of a new byte after an ACK. Any other timing for a stop condition causes the Bus Error bit to be set.

Table 28-4. I2C_SCR Status and Control Register

Bit	Access	Description	Mode
7	RC	Bus Error 1 = A misplaced start or stop condition was detected. This status bit must be cleared by firmware with a write of ‘0’ to the bit position. It is never cleared by the hardware.	Master Only
6	RC	Lost Arb 1 = Lost Arbitration. This bit is set immediately on lost arbitration; however, it does not cause an interrupt. This status may be checked after the following Byte Complete interrupt. Any start detect automatically clears this bit.	Master Only
5	RC	Stop Status 1 = A stop condition was detected. This status bit must be cleared by firmware with a write of ‘0’ to the bit position. It is never cleared by the hardware.	Master/ Slave
4	RW	ACK: Acknowledge Out 0 = NACK the last received byte. 1 = ACK the last received byte. This bit is automatically cleared by hardware on the following Byte Complete event.	Master/ Slave
3	RC	Address 1 = The transmitted or received byte is an address. This status bit must be cleared by firmware with a write of ‘0’ to the bit position.	Master/ Slave
2	RW	Transmit 0 = Receive Mode. 1 = Transmit Mode. This bit is set by firmware to define the direction of the byte transfer. Any start detect automatically clears this bit.	Master/ Slave
1	RC	LRB: Last Received Bit The value of the ninth bit in a transmit sequence, which is the acknowledge bit from the receiver. 0 = Last transmitted byte was ACK’ed by the receiver. 1 = Last transmitted byte was NACK’ed by the receiver. Any start detect automatically clears this bit.	Master/ Slave
0	RC	Byte Complete Transmit Mode: 1 = 8 bits of data have been transmitted and an ACK or NACK has been received. Receive Mode: 1 = 8 bits of data have been received. Any start detect automatically clears this bit.	Master/ Slave

Bit 6: Lost Arb. This bit is set when I2C bus contention is detected, during a Master mode transfer. Contention occurs when a master is writing a ‘1’ to the SDA output line and reading back a ‘0’ on the SDA input line at the given sampling point. When this occurs, the block immediately releases the SDA, but continues clocking to the end of the current byte. On the resulting byte interrupt, firmware can determine that arbitration was lost to another master by reading this bit.

The sequence occurs differently between master transmitter and master receiver. As a transmitter, the contention occurs on a data bit. On the subsequent Byte Complete interrupt, the Lost Arbitration status is set. In receiver mode, the contention occurs on the ACK bit. The master that NACK’ed the last reception loses the arbitration. However, the hardware shifts in the next byte in response to the winning master’s ACK, so that a subsequent Byte Complete interrupt occurs. At this point, the losing master can read the Lost Arbitration status. Contention is checked only at the eight data bit sampling point and one ACK bit sampling point.

Bit 5: Stop Status. Stop Status is set on detection of an I2C stop condition. This bit is sticky, which means that it remains set until a ‘0’ is written back to it by the firmware. This bit may only be cleared if the Byte Complete status is set. If the Stop Interrupt Enable bit is set, an interrupt is also generated on stop detection. It is never automatically cleared.

Using this bit, a slave can distinguish between a previous stop or restart on a given address byte interrupt. In Master mode, this bit may be used in conjunction with the Stop IE bit, to generate an interrupt when the bus is free. However, in this case, the bit must have previously been cleared prior to the reception of the stop in order to cause an interrupt.

Bit 4: ACK. This control bit defines the acknowledge data bit that is transmitted out in response to a received byte. When receiving, a Byte Complete interrupt is generated after the eighth data bit is received. On the subsequent write to this register to continue (or terminate) the transfer, the state of this bit determines the next bit of data that is transmitted. It is ***active high***. A ‘1’ sends an ACK and a ‘0’ sends a NACK.

A master receiver normally terminates a transfer by writing a ‘0’ (NACK) to this bit. This releases the bus and automatically generates a stop condition. A slave receiver may also send a NACK to inform the master that it cannot receive any more bytes.

Bit 3: Address. This bit is set when an address has been received. This consists of a start or restart, and an address byte. This bit applies to both master and slave.

In Slave mode, when this status is set, firmware reads the received address from the data register and compares it with its own address. If the address does not match, the firmware writes a NACK indication to this register. No further interrupts occur until the next address is received. If the address does match, firmware must ACK the received byte, then Byte Complete interrupts are generated on subsequent bytes of the transfer.

This bit is also sets when address transmission is complete in Master mode. If a lost arbitration occurs during the transmission of a master address (indicated by the Lost Arb bit), the block reverts to Slave mode if enabled. This bit then signifies that the block is being addressed as a slave.

If Slave mode is not enabled, the Byte Complete interrupt still occurs to inform the master of lost arbitration.

Bit 2: Transmit. This bit sets the direction of the shifter for a subsequent byte transfer. The shifter is always shifting in data from the I2C bus, but a write of ‘1’ enables the output of the shifter to drive the SDA output line. Since a write to this register initiates the next transfer, data must be written to the data register prior to writing this bit. In receive mode, the previously received data must have been read from the data register before this write. In Slave mode, firmware derives this direction from the RW bit in the received slave address. In Master mode, the firmware decides on the direction and sets it accordingly.

This direction control is only valid for data transfers. The direction of address bytes is determined by the hardware, depending on the Master or Slave mode.

The master transmitter terminates a transfer by writing a zero to the Transmit bit. This releases the bus and automatically sends a stop condition, or a stop/start or restart, depending on the I2C_MSCR control bits.

Bit 1: LRB (Last Received Bit). This is the last received bit in response to a previously transmitted byte. In transmit mode, the hardware sends a byte from the data register and clock in an acknowledge bit from the receiver. On the subsequent byte complete interrupt, firmware checks the value of this bit. A ‘0’ is the ACK value and a ‘1’ is a NACK value. The meaning of the LRB depends on the current operating mode.

Master Transmitter:

‘0’: ACK. The slave has accepted the previous byte. The master sends another byte by first writing the byte to the I2C_DR register and then setting the Transmit bit in the I2C_SCR register. Optionally, the master clears the Transmit bit in the I2C_SCR register. This automatically sends a stop. If the start or restart bits are set in the I2C_MSCR register, the stop is followed by a start or restart.

‘1’: NACK. The slave cannot accept any more bytes. A stop is automatically generated by the hardware on the subsequent write to the I2C_SCR register (regardless of the value written). However, a stop/start or restart condition may also be generated, depending on whether firmware has set the Start or Restart bits in the I2C_MSCR register.

Slave Transmitter:

‘0’: ACK. The master wants to read another byte. The slave loads the next byte into the I2C_DR register and sets the transmit bit in the I2C_SCR register to continue the transfer.

‘1’: NACK. The master is done reading bytes. The slave reverts to IDLE state on the subsequent I2C_SCR write (regardless of the value written).

Bit 0: Byte Complete. The I2C hardware operates on a byte basis. In transmit mode, this bit is set and an interrupt is generated at the end of nine bits (the transmitted byte + the received ACK). In receive mode, the bit is set after the eight bits of data are received. When this bit is set, an interrupt is generated at these data sampling points, which are associated with the SCL input clock rising (see details in the Timing section). If the PSoC device responds with a write back to this register before the subsequent falling edge of SCL (which is approximately one-half bit time), the transfer continues without interruption. However, if the PSoC device is unable to respond within that time, the hardware holds the SCL line low, stalling the I2C bus. In both Master and Slave mode, a subsequent write to the I2C_SCR register releases the stall.

For additional information, refer to the I2C_SCR register on page 95.

28.3.3 I2C_DR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D8h	I2C_DR	Data[7:0]								RW : 00

The I2C Data Register (I2C_DR) provides read/write access to the Shift register.

Bits 7 to 0: Data[7:0]. This register is not buffered; and therefore, writes and valid data reads only occur at specific points in the transfer. These cases are outlined as follows.

- **Master or Slave Receiver** – Data in the I2C_DR register is only valid for reading, when the Byte Complete status bit is set. Data bytes must be read from the register before writing to the I2C_SCR register, which continues the transfer.
- **Master Start or Restart** – Address bytes must be written in I2C_DR before the Start or Restart bit is set in the I2C_MSCR register, which causes the start or restart to generate and the address to shift out.
- **Master or Slave Transmitter** – Data bytes must be written to the I2C_DR register before the transmit bit is set in the I2C_SCR register, which causes the transfer to continue.

For additional information, refer to the I2C_DR register on page 97.

28.3.4 I2C_MSCR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D9h	I2C_MSCR					Bus Busy	Master Mode	Restart Gen	Start Gen	R : 00

The I2C Master Status and Control Register (I2C_MSCR) implements I2C framing controls and provides Bus Busy status.

Bit 3: Bus Busy. This read only bit is set to ‘1’ by any start condition and reset to ‘0’ by a stop condition. It is polled by firmware to determine when a bus transfer is initiated.

Bit 2: Master Mode. This bit indicates that the device is operating as a master. It is set in the detection of this block’s start condition and reset in the detection of the subsequent stop condition.

Bit 1: Restart Gen. This bit is only used at the end of a master transfer (as noted in Other Cases 1 and 2 of the Start Gen bit). If an address is loaded into the data register and this bit is set prior to NACKing (master receiver) or resetting the transmit bit (master transmitter), or after a master transmitter is NACK’ed by the slave, a restart condition is generated followed by the transmission of the address byte.

Bit 0: Start Gen. Before setting this bit, firmware must write the address byte to send into the I2C_DR register. When this bit is set, the start condition is generated followed immediately by the transmission of the address byte. (No control in the I2C_SCR register is needed for the master to initiate a transmission; the direction is inherently “transmit.”) The bit is automatically reset to ‘0’ after the start has been generated.

There are three possible outcomes as a result of setting the Start Gen bit.

1. The bus is free and the start condition is generated successfully. A Byte Complete interrupt is generated after the start and the address byte transmitted. If the address is ACK'ed by the receiver, the firmware then proceeds to send data bytes.
2. The start command is too late. Another master in a multi-master environment has generated a valid start and the bus is busy. The resulting behavior depends upon whether Slave mode is enabled.

   Slave mode is enabled: A start and address byte interrupt is generated. When reading the I2C_MSCR register, the master sees that the Start Gen bit is still set and that the I2C_SCR register has the Address bit set, indicating that the block is addressed as a slave.

   Slave mode is not enabled: The Start Gen bit remains set and the start is queued, until the bus becomes free and the start condition is subsequently generated. An interrupt is generated at a later time, when the start and address byte have been transmitted.
3. The start is generated, but the master loses arbitration to another master in a multi-master environment. The resulting behavior depends upon whether Slave mode is enabled.

   Slave mode is enabled: A start and address byte interrupt is generated. When reading the I2C_MSCR, the master sees that the Start Gen bit cleared, indicating that the start was generated. However, the Lost Arb bit is set in the I2C_SCR register. The Address status is also set, indicating that the block has been addressed as a slave. The firmware then ACKs or NACKs the address to continue the transfer.

   Slave mode is not enabled: A start and address byte interrupt is generated. The Start Gen bit is cleared and the Lost Arb bit is set. The hardware waits for command input, stalling the bus if necessary. In this case, the master clears the I2C_SCR register, to release the bus and allow the transfer to continue, and the block idles.

Other cases where the Start bit is used to generate a Start condition are as follows.

1. When a master is finished with a transfer, a NACK is written to the I2C_SCR register (in the case of the master receiver) or the transmit bit is cleared (in case of a master transmitter). Normally, the action frees the stall and generates a stop condition. However, if the Start bit is set and an address is written into the data register prior to the I2C_SCR write, a stop, followed immediately by a start (minimum bus free time), is generated. In this way, messages may be chained.
2. When a master transmitter is NACK'ed, an automatic stop condition is generated on the subsequent I2C_SCR write. However, if the Start Gen bit has previously been set, the stop is immediately followed by a start condition.

Table 28-5. I2C_MSCR Master Status/Control Register

Bit	Access	Description	Mode
3	R	Bus Busy This bit is set to '1' when any start condition is detected and reset to '0' when a stop condition is detected.	Master Only
2	R	Master Mode This bit is set to '1' when a start condition, generated by this block, is detected and reset to '0' when a stop condition is detected.	Master Only
1	RW	Restart Gen 1 = Generate a Restart condition. This bit is cleared by hardware when the start generation is complete.	Master Only
0	RW	Start Gen 1 = Generate a start condition and send a byte (address) to the I2C bus. This bit is cleared by hardware when the start generation is complete.	Master Only

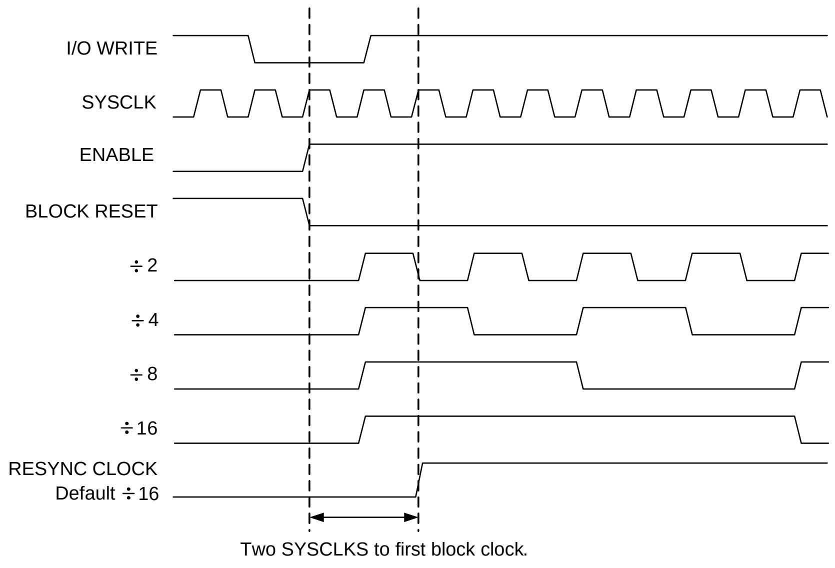
For additional information, refer to the I2C_MSCR register on page 98.

28.4 Timing Diagrams

28.4.1 Clock Generation

Figure 28-4 illustrates the I2C input clocking scheme. The SYSCLK pin is an input into a four-stage ripple divider that provides the baud rate selections. When the block is disabled, all internal state is held in a reset state. When either the Master or Slave Enable bits in the I2C_CFG register are set, the reset is synchronously released and the clock generation is enabled. Two taps from the ***ripple divider*** are selectable (/4, /16) from the clock rate bits in the I2C_CFG register. If any of the two divider taps is selected, that clock is resynchronized to SYSCLK. The resulting clock is routed to all of the synchronous elements in the design.

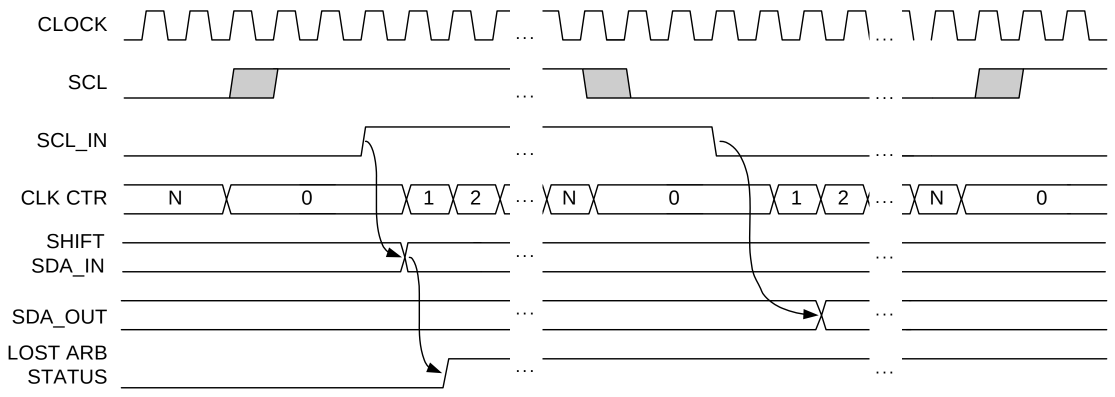
Figure 28-4. I^{2}C Input Clocking



28.4.2 Basic Input/Output Timing

Figure 28-5 illustrates basic input output timing that is valid for both 16 times sampling and 32 times sampling. For 16 times sampling, N=4; and for 32 times sampling, N=12. N is derived from the half-bit rate sampling of 8 and 16 clocks, respectively, minus the input latency of three (count of 4 and 12 correspond to 5 and 13 clocks).

Figure 28-5. Basic Input/Output Timing



28.4.3 Status Timing

Figure 28-6 illustrates the interrupt timing for Byte Complete, which occurs on the positive edge of the ninth clock (byte + ACK/NACK) in transmit mode and on the positive edge of the eighth clock in receive mode. There is a maximum of three cycles of latency, due to the input synchronizer/filter circuit. As shown, the interrupt occurs on the clock following a valid SCL positive edge input transition (after the synchronizers). The Address bit is set with the same timing, but only after a slave address has been received. The LRB (Last Received Bit) status is also set with the same timing, but only on the ninth bit after a transmitted byte.

Figure 28-6. Byte Complete, Address, LRB Timing

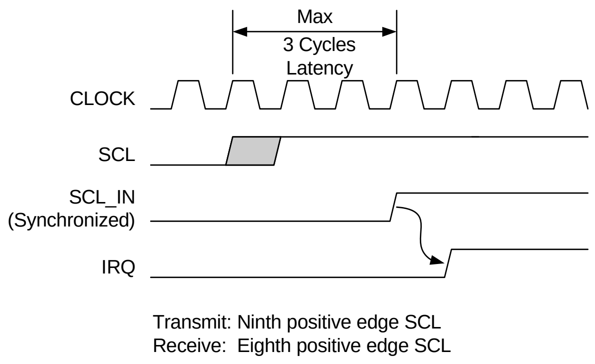

Max
3 Cycles
Latency
CLOCK
SCL
SCL_IN
(Synchronized)
IRQ
Transmit: Ninth positive edge SCL
Receive: Eighth positive edge SCL

Figure 28-7 shows the timing for Stop Status. This bit is set (and the interrupt occurs) two clocks after the synchronized and filtered SDA line transitions to a '1', when the SCL line is high.

Figure 28-7. Stop Status and Interrupt Timing

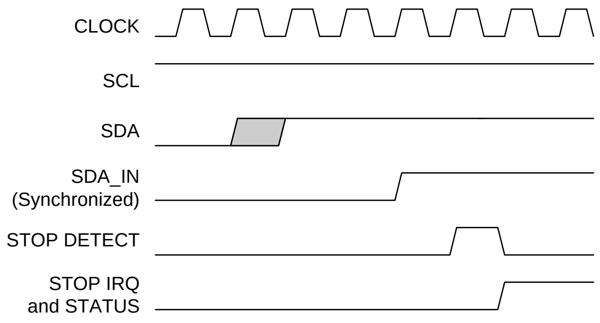
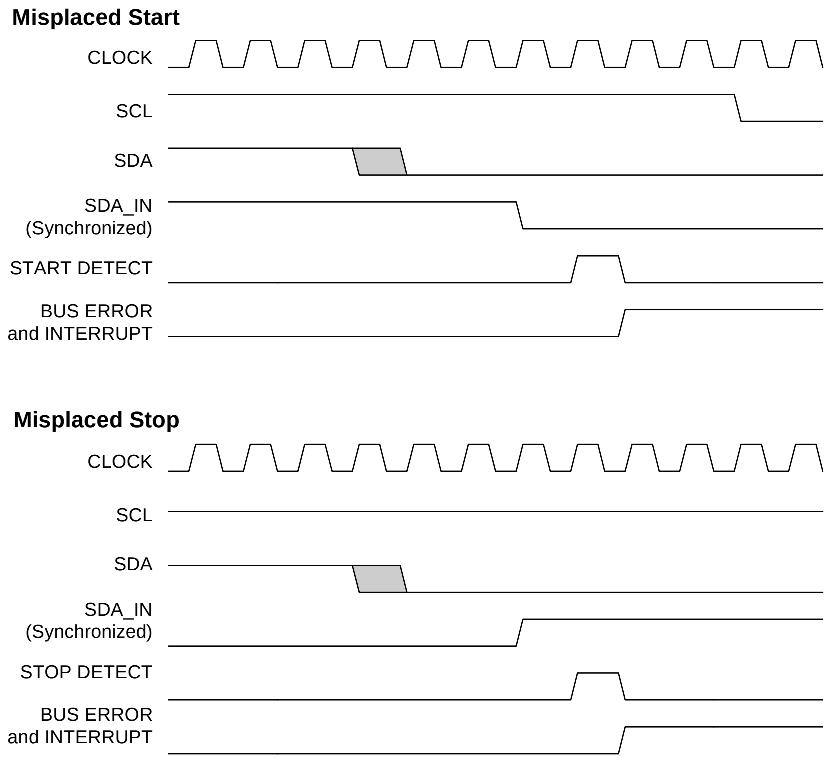


Figure 28-8 illustrates the timing for bus error interrupts. Bus Error status (and Interrupt) occurs one cycle after the internal Start or Stop Detect (two cycles after the filtered and synced SDA input transition).

Figure 28-8. Bus Error Interrupt Timing



28.4.4 Master Start Timing

When firmware writes the Start Gen command, hardware resynchronizes this bit to SYSCLK, to ensure a minimum of a full SYSCLK of set up time to the next clock edge. When the start is initiated, the SCL line is left high for 6/14 clocks (corresponding to 16/32 times sampling rates). During this initial SCL high period, if an external start is detected, the start sequence is aborted and the block returns to an IDLE state. However, on the next stop detection, the block automatically initiates a new Start sequence.

Figure 28-9. Basic Master Start Timing

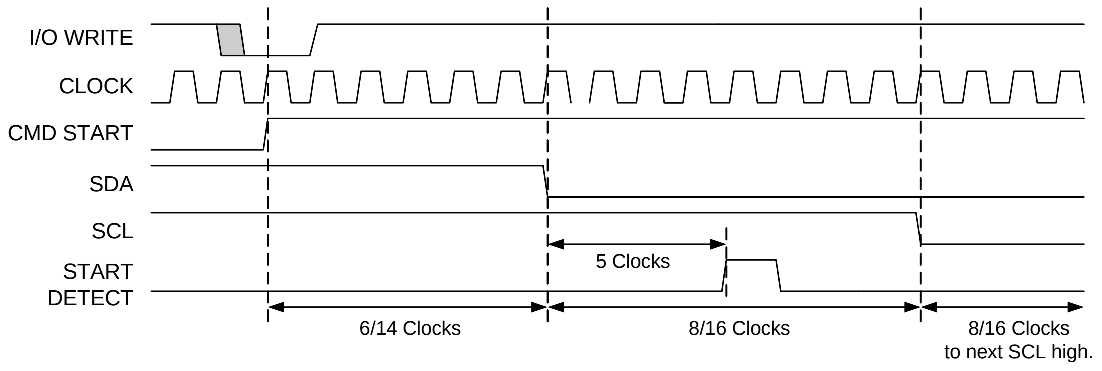


Figure 28-10. Start Timing with a Pending Start

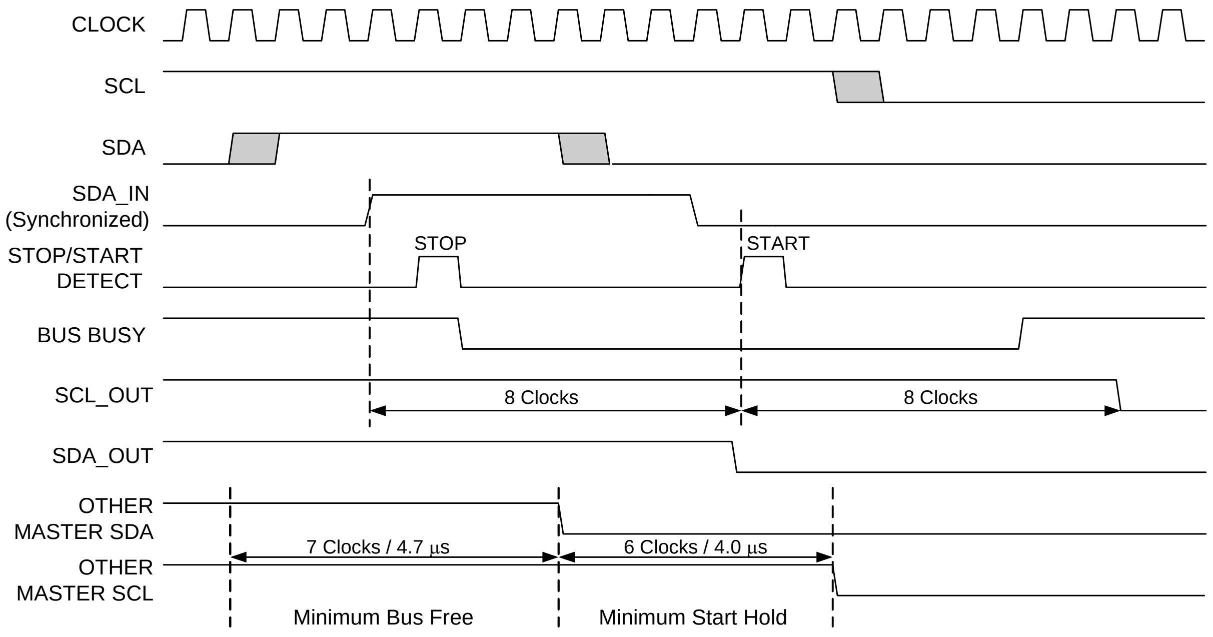
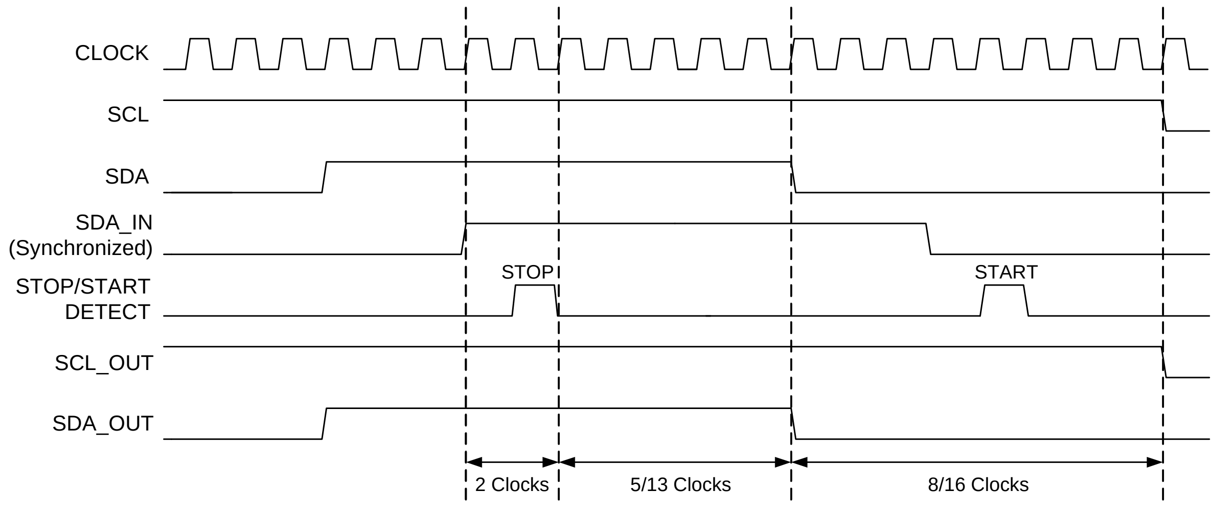


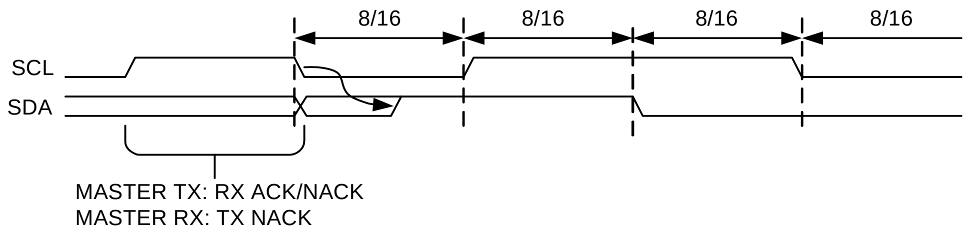
Figure 28-11. Master Stop/Start Chaining



28.4.5 Master Restart Timing

Figure 28-12 shows the Master Restart timing. After the ACK/NACK bit, the clock is held low for a half-bit time (8/16 clocks corresponding to the 16 or 32 times sampling rates), during which time the data is allowed to go high, then a valid start is generated in the following 3 half-bit times as shown.

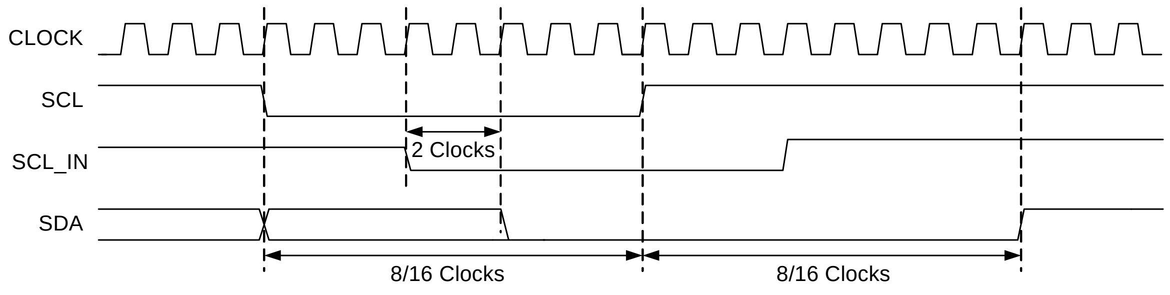
Figure 28-12. Master Restart Timing



28.4.6 Master Stop Timing

Figure 28-13 shows basic Master Stop timing. In order to generate a stop, the SDA line is first pulled low, in accordance with the basic SDA output timing. Then, after the full low of SCL is completed and the SCL line is pulled high, the SDA line remains low for a full 1 half-bit time before it is pulled high to signal the stop.

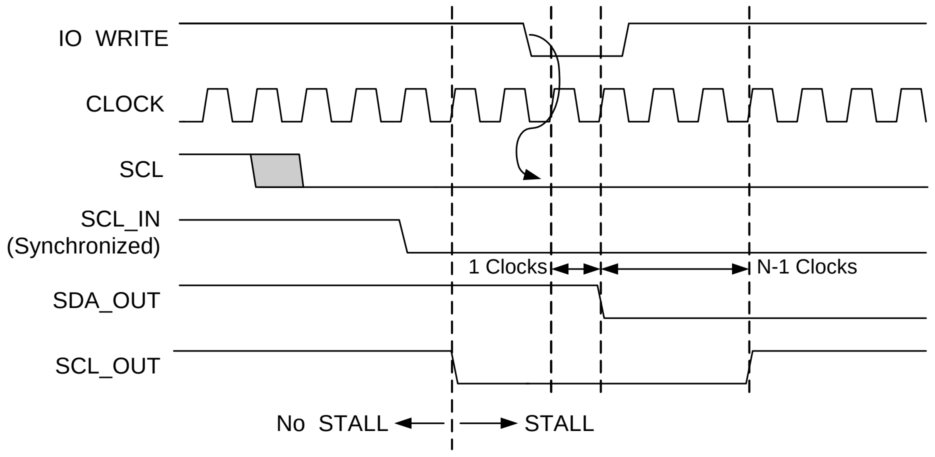
Figure 28-13. Master Stop Timing



28.4.7 Master/Slave Stall Timing

When a Byte Complete interrupt occurs, the PSoC device firmware must respond with a write to the I2C_SCR register to continue the transfer (or terminate the transfer). The interrupt occurs two clocks after the rising edge of SCL_IN (see "Status Timing" on page 305). As illustrated in Figure 28-14, firmware has until one clock after the falling edge of SCL_IN to write to the I2C_SCR register; otherwise, a stall occurs. Once stalled, the IO write releases the stall. The set up time between data output and the next rising edge of SCL is always N-1 clocks.

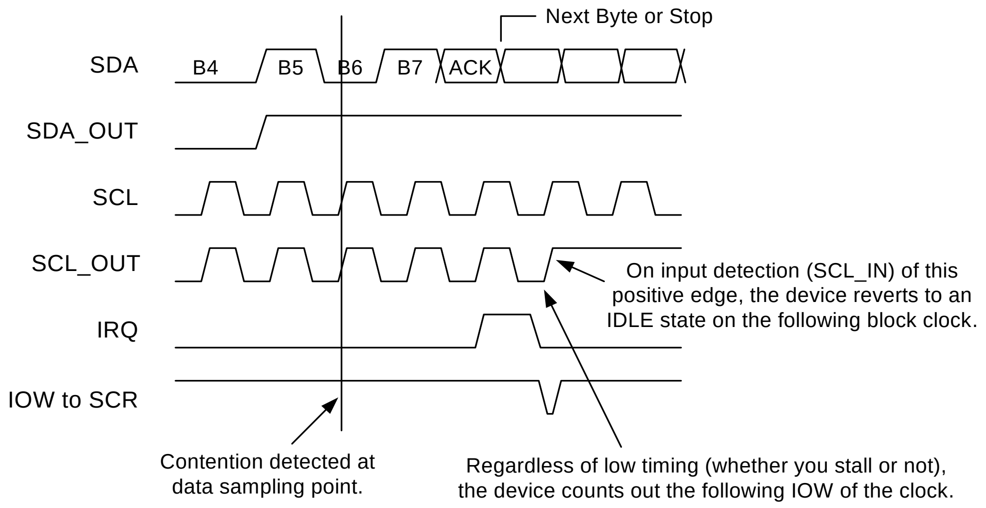
Figure 28-14. Master/Slave Stall Timing



28.4.8 Master Lost Arbitration Timing

Figure 28-15 shows a Lost Arbitration sequence. When contention is detected at the input (SDA_IN) sampling point, the SDA output is immediately released to an IDLE state. However, the master continues clocking until the Byte Complete interrupt, which is processed in the usual way. Any write to the I2C_SCR register results in the master reverting to an IDLE state, one clock after the next positive edge of the SCL_IN clock.

Figure 28-15. Lost Arbitration Timing (Transmitting Address or Data)

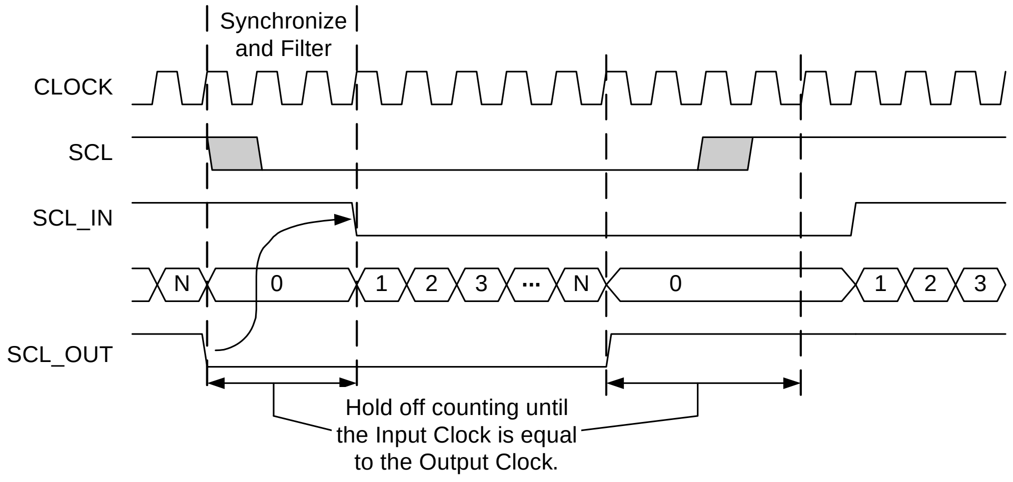

Next Byte or Stop
SDA
B4
B5
B6
B7
ACK
SDA_OUT
SCL
SCL_OUT
IRQ
IOW to SCR
On input detection (SCL_IN) of this positive edge, the device reverts to an IDLE state on the following block clock.
Contention detected at data sampling point.
Regardless of low timing (whether you stall or not), the device counts out the following IOW of the clock.

28.4.9 Master Clock Synchronization

Figure 28-16 shows the timing associated with Master Clock Synchronization. Clock synchronization is always operational, even if it is the only master on the bus. In which case, it is synchronizing to its own clock. In the wired AND bus, an SCL output of ‘0’ is seen by all masters. When the hardware asserts a ‘0’ to the output, it is immediately fed back from the PSoC device pin to the input synchronizer for the SCL input. The counter value (depending on the sampling rate) takes into account the worst case latency for input synchronization of three clocks, giving a net period of 8/16 clocks for both high and low time. This results in an overall clocking rate of 16/32 clocks per bit.

In multi-master environments when the hardware outputs a ‘1’ on the SCL output, if any other master is still asserting a ‘0’, the clock counter holds until the SCL input line matches the ‘1’ on the SCL output line. When matched, the remainder of the high time is counted down. In this way, the master with the fastest frequency determines the high time of the clock and the master with the lowest frequency determines the low time of the clock.

Figure 28-16. Master Clock Synchronization


Synchronize and Filter
CLOCK
SCL
SCL_IN
N
0
1
2
3
...
N
0
1
2
3
SCL_OUT
Hold off counting until the Input Clock is equal to the Output Clock.

29. Internal Voltage Reference

This chapter discusses the Internal Voltage Reference and its associated register. The internal voltage reference provides an absolute value of 1.3V to a variety of subsystems in the PSoC device. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

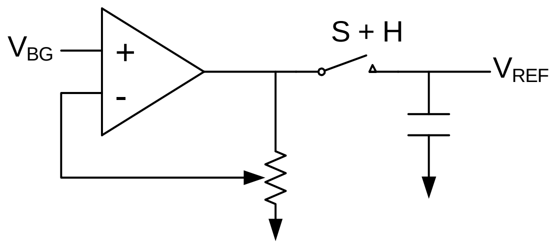
29.1 Architectural Description

The Internal Voltage Reference is made up of two blocks: a bandgap voltage generator and a buffer with sample and hold. The bandgap voltage generator is a typical (V_{BE} + K V_T) design.

The buffer circuit provides gain to the 1.20V bandgap voltage, to produce a 1.30V reference. A simplified ***schematic*** is illustrated in Figure 29-1. The connection between amplifier and capacitor is made through a CMOS switch, allowing the reference voltage to be used by the system while the reference circuit is powered down. The voltage reference is trimmed to 1.30V at room temperature.

A temperature proportional voltage is also produced in this block for use in temperature sensing.

Figure 29-1. Voltage Reference Schematic



29.2 Register Definitions

The following register is associated with the Internal Voltage Reference. The Internal Voltage Reference is trimmed for gain and temperature coefficient using the BDG_TR register. The register description below has an associated register table showing the bit structure.

29.2.1 BDG_TR Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,EAh	BDG_TR		AGNDBYP	TC[1:0]		V[3:0]				RW : 00

The Bandgap Trim Register (BDG_TR) is used to adjust the bandgap and add an RC filter to AGND.

Bit 6: AGNDBYP. When set, this bit adds an RC filter to AGND. (R is an internal 8.1K resistor and C is external to the PSoC device on P2[4].)

Bits 5 and 4: TC[1:0]. These bits are for setting the temperature coefficient inside the bandgap voltage generator. 10b is the design center for ‘0’ TC.

It is strongly recommended that the user not alter the value of these bits.

Bits 3 to 0: V[3:0]. These bits are for setting the gain in the reference buffer. Sixteen steps of 4 mV are available. 1000b is the design center for 1.30V.

It is strongly recommended that the user not alter the value of these bits.

For additional information, refer to the BDG_TR register on page 158.

30. System Resets

This chapter discusses the System Resets and their associated registers. PSoC devices support several types of resets. The various resets are designed to provide error-free operation during power up for any voltage ramping profile, to allow for user-supplied external reset and to provide recovery from errant code operation. For a complete table of the System Reset registers, refer to the "Summary Table of the System Resource Registers" on page 272. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

30.1 Architectural Description

When reset is initiated, all registers are restored to their default states. In the Register Details chapter on page 47, this is indicated by the POR column in the register tables and elsewhere it is indicated in the Access column, values on the right side of the colon, in the register tables. Minor exceptions are explained below.

The following types of resets occur in the PSoC device:

- Power on Reset (POR). This occurs at low supply voltage and is comprised of multiple sources.
- External Reset (XRES). This active high reset is driven into the PSoC device, on parts that contain an XRES pin.
- Watchdog Reset (WDR). This optional reset occurs when the watchdog timer expires, before being cleared by user firmware. Watchdog reset defaults to off.

The occurrence of a reset is recorded in the Status and Control registers (CPU_SCR0 for POR, XRES, and WDR) or in the System Status and Control Register 1. Firmware can interrogate these registers to determine the cause of a reset.

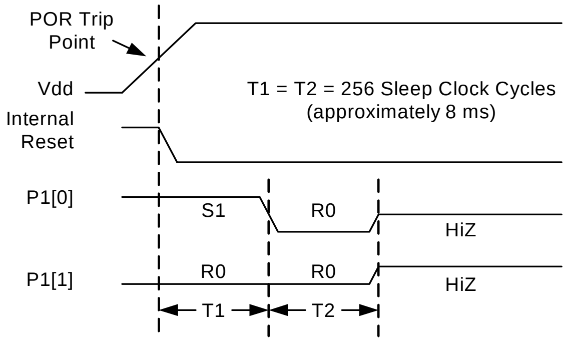
30.2 Pin Behavior During Reset

Power on Reset and External Reset cause toggling on two GPIO pins, P1[0] and P1[1], as described below and illustrated in Figure 30-1. This allows programmers to synchronize with the PSoC device. All other GPIO pins are placed in a high impedance state during and immediately following reset.

30.2.1 GPIO Behavior on Power Up

At power up, the internal POR causes P1[0] to initially drive a strong high (1) while P1[1] drives a resistive low (0). After 256 sleep oscillator cycles (approximately 8 ms), the P1[0] signal transitions to a resistive low state. After additional 256 sleep oscillator clocks, both pins transition to a high impedance state and normal CPU operation begins. This is illustrated in Figure 30-1.

Figure 30-1. P1[1:0] Behavior on Power Up



30.3 Register Definitions

The following registers are associated with the PSoC System Resets and are listed in address order. Each register description has an associated register table showing the bit structure for that register. The bits in the tables that are grayed out are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of ‘0’. For a complete table of system reset registers, refer to the “Summary Table of the System Resource Registers” on page 272.

30.3.1 CPU_SCR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW *	ECO EX *		IRAMDIS	# : 00

LEGEND
x An “x” before the comma in the address field indicates that this register can be read or written to no matter what bank is used.
# Access is bit specific. Refer to the Register Details chapter on page 47 for additional information.
* Bits 3 and 2 (ECO EXW and ECO EX, respectively) cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23 and CY8C22x13 PSoC devices.

The System Status and Control Register 1 (CPU_SCR1) is used to convey the status and control of events related to internal resets and watchdog reset.

Bit 7: IRESS. The Internal Reset Status bit is a read only bit used to determine if the booting process occurred more than once.

When this bit is set, it indicates that the SROM SWBootReset code executed more than once. If this bit is not set, the SWBootReset executed only once. In either case, the SWBootReset code does not allow execution from code stored in Flash until the M8C Core is in a safe operating mode with respect to supply voltage and Flash operation. There is no need for concern when this bit is set. It is provided for systems which may be sensitive to boot time, so that they can determine if the normal one-pass boot time was exceeded. For more information on the SWBootReest code see the Supervisory ROM (SROM) chapter on page 45.

Bit 4: SLIMO. When set, the Slow IMO bit allows the active power dissipation of the PSoC device to be reduced by slowing down the IMO from 24 MHz to 6 MHz. The IMO trim value must also be changed when SLIMO is set. When not in external clocking mode, the IMO is the source for SYSCLK; therefore, when the speed of the IMO changes, so does SYSCLK.

Bit 3: ECO EXW. The ECO Exists Written bit is used as a status bit to indicate that the ECO EX bit has been previously written to. It is read only. When this bit is a ‘1’, this indicates that the CPU_SCR1 register has been written to and is now locked. When this bit is a ‘0’, the register has not been written to since the last reset event. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23, and CY8C22x13 PSoC devices.

Bit 2: ECO EX. The ECO Exists bit serves as a flag to the hardware, to indicate that an external crystal ***oscillator*** exists in the system. Just after boot, it may be written *only once* to a value of ‘1’ (crystal exists) or ‘0’ (crystal does not exist). If the bit is ‘0’, a switch-over to the ECO is locked out by hardware. If the bit is ‘1’, hardware allows the firmware to freely switch between the ECO and ILO. It should be written as early as possible after a ***Power On Reset (POR)*** or ***External Reset (XRES)*** event, where it is assumed that program execution integrity is high. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24533, CY8C23533, CY8C23433, CY8C24633, CY8C24x23 and CY8C22x13 PSoC devices.

Bit 0: IRAMDIS. The Initialize RAM Disable bit is a control bit that is readable and writeable. The ***default value*** for this bit is ‘0’, which indicates that the maximum amount of SRAM should be initialized on watchdog reset to a value of 00h. When the bit is ‘1’, the minimum amount of SRAM is initialized after a watchdog reset. For more information on this bit, see the “SROM Function Descriptions” on page 46.

For additional information, refer to the CPU_SCR1 register on page 121.

30.3.2 CPU_SCR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	# : XX

LEGEND
# Access is bit specific. Refer to register detail for additional information.
XX The reset value is 10h after POR and 20h after a watchdog reset.

The System Status and Control Register 0 (CPU_SCR0) is used to convey the status and control of events for various functions of a PSoC device.

Bit 7: GIES. The Global Interrupt Enable Status bit is a read only status bit and its use is discouraged. The GIES bit is a legacy bit which was used to provide the ability to read the GIE bit of the CPU_F register. However, the CPU_F register is now readable. When this bit is set, it indicates that the GIE bit in the CPU_F register is also set which, in turn, indicates that the microprocessor services interrupts.

Bit 5: WDRS. The WatchDog Reset Status bit may not be set. It is normally '0' and automatically set whenever a watchdog reset occurs. The bit is readable and clearable by writing a zero to its bit position in the CPU_SCR0 register.

Bit 4: PORS. The Power On Reset Status (PORS) bit, which is the watchdog enable bit, is set automatically by a POR or External Reset (XRES). If the bit is cleared by user code, the watchdog timer is enabled. Once cleared, the only way to reset the PORS bit is to go through a POR or XRES. Thus, there is no way to disable the watchdog timer, other than to go through a POR or XRES.

Bit 3: Sleep. The Sleep bit is used to enter Low Power Sleep mode when set. To wake up the system, this register bit is cleared asynchronously by any enabled interrupt. There are two special features of this register bit that ensures proper sleep operation. First, the write to set the register bit is blocked, if an interrupt is about to be taken on that instruction boundary (immediately after the write). Second, there is a hardware interlock to ensure that, once set, the Sleep bit may not be cleared by an incoming interrupt until the sleep circuit has finished performing the sleep sequence and the system-wide power down signal has been asserted. This prevents the sleep circuit from being interrupted in the middle of the process of system power down, possibly leaving the system in an indeterminate state.

Bit 0: STOP. The STOP bit is readable and writeable. When set, the PSoC M8C stops executing code until a reset event occurs. This can be either a POR or WDR, or XRES. If an application wants to stop code execution until a reset, the preferred method is to use the HALT instruction rather than a register write to this bit.

For additional information, refer to the CPU_SCR0 register on page 122.

30.4 Timing Diagrams

30.4.1 Power On Reset

A Power on Reset (POR) is triggered whenever the supply voltage is below the POR trip point. POR ends once the supply voltage rises above this voltage. Refer to the POR and LVD chapter on page 319 for more information on the operation of the POR block.

POR consists of two pieces: an imprecise POR (IPOR) and a Precision POR (PPOR). “POR” refers to the OR of these two functions. IPOR has coarser accuracy and its trip point is typically lower than PPOR’s trip point. PPOR is derived from a circuit that is calibrated (during boot), for a very accurate location of the POR trip point.

During POR (POR=1), the IMO is powered off for low power during start up. Once POR de-asserts, the IMO is started (see Figure 30-3).

POR configures register reset status bits as shown in Table 30-1. PPOR does not affect the BandGap Trim register (BDG_TR), but IPOR does reset this register.

30.4.2 Watchdog Timer Reset

The user has the option to enable the Watchdog Timer Reset (WDR), by clearing the PORS bit in the CPU_SCR0 register. Once the PORS bit is cleared, the watchdog timer cannot be disabled. The only exception to this is if a POR/XRES event takes place, which disables the WDR. Note that a WDR does not clear the Watchdog timer. See “Watchdog Timer” on page 40 for details of the Watchdog operation.

When the watchdog timer expires, a watchdog event occurs resulting in the reset sequence. Some characteristics unique to the WDR are as follows.

- PSoC device reset asserts for one cycle of the CLK32K clock (at its reset state).
- The IMO is not halted during or after WDR (that is, the part does not go through a low power phase).
- CPU operation re-starts one CLK32K cycle after the internal reset de-asserts (see Figure 30-2).

How the WDR configures register reset status bits is shown in Table 30-1.

Figure 30-2. Key Signals During WDR

WDR: Reset 1 cycle, then one additional cycle before the CPU reset is released.

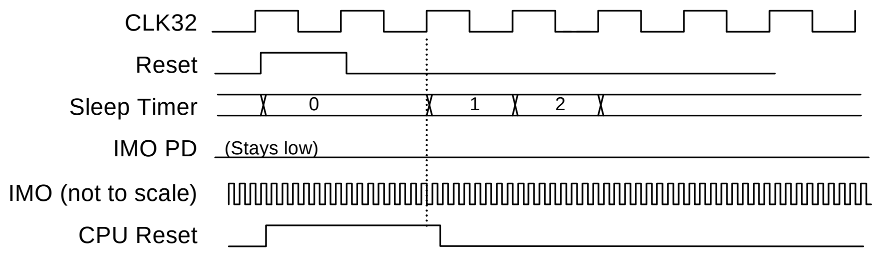
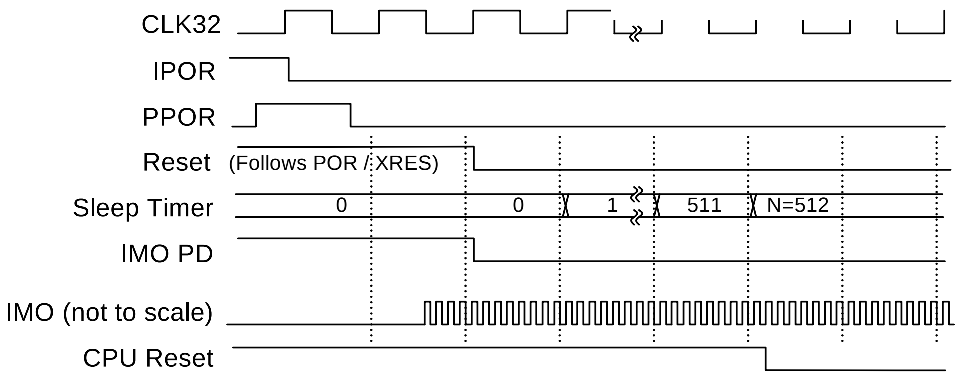
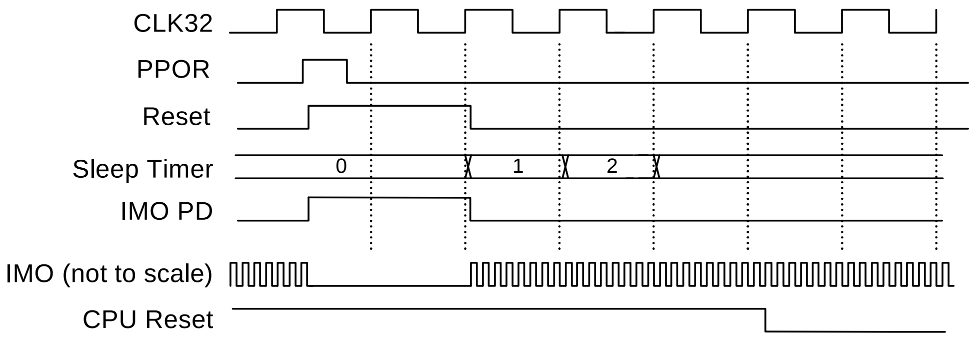


Figure 30-3. Key Signals During POR and XRES

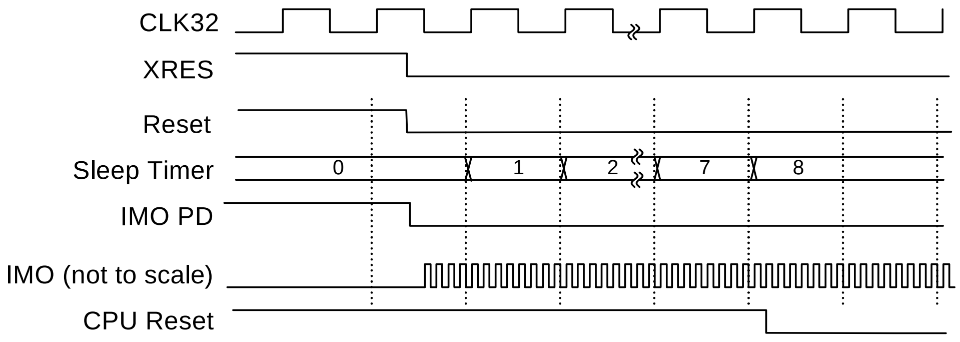
POR (IPOR followed by PPOR): Reset while POR is high (IMO off), then 511(+) cycles (IMO on), and then the CPU reset is released. **XRES** is the same, with N=8.



PPOR (with no IPOR): Reset while PPOR is high and to the end of the next 32K cycle (IMO off); 1 cycle IMO on before the CPU reset is released. Note that at the 5V level, PPOR tends to be brief, because the reset clears the POR range register (VLT_CR) back to the default 3V setting.



XRES: Reset while XRES is high (IMO off), then 7(+) cycles (IMO on), and then the CPU reset is released.



30.4.3 Reset Details

Timing and functionality details are summarized in Table 30-1. Figure 30-3 shows some of the relevant signals for IPOR and PPOR, and XRES, while Figure 30-2 shows signaling for WDR .

Table 30-1. Details of Functionality for Various Resets

Item	IPOR (Part of POR)	PPOR (Part of POR)	XRES	WDR
Reset Length	While POR=1	While PPOR=1, plus 30-60 µs (1-2 clocks)	While XRES=1	30 µs (1 clock)
Low Power (IMO Off) During Reset	Yes	Yes	Yes	No
Low Power Wait Following Reset	No	No	No	No
CLK32K Cycles from End of Reset to CPU Reset De-asserts[a]	512	1	8	1
Register Reset (See next line for CPU_SCR0, CPU_SCR1)	All	All, except PPOR does not reset Bandgap Trim register	All	All
Reset Status Bits in CPU_SCR0, CPU_SCR1	Set PORS, Clear WDRS, Clear IRAMDIS	Set PORS, Clear WDRS, Clear IRAMDIS	Set PORS, Clear WDRS, Clear IRAMDIS	Clear PORS, Set WDRS, IRAMDIS unchanged
Bandgap Power	On	On	On	On
Boot Time[b]	2.2 ms	2.2 ms	2.2 ms	2.2 ms

a. CPU reset is released after synchronization with the CPU Clock.

b. Measured from CPU reset release to execution of the code at Flash address 0x0000.

30.5 Power Consumption

The ILO block drives the CLK32K clock used to time most events during the reset sequence. This clock is powered down by IPOR, but not by any other reset. The sleep timer provides interval timing.

While POR or XRES asserts, the IMO is powered off to reduce start up power consumption.

During and after POR or XRES, the bandgap circuit is powered up.

The IMO is always on for at least one CLK32K cycle, before CPU reset is de-asserted.

31. POR and LVD

This chapter briefly discusses the POR and LVD circuits and their associated registers. For a complete table of the POR and LVD registers, refer to the "Summary Table of the System Resource Registers" on page 272. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 47.

31.1 Architectural Description

The Power on Reset (POR) and Low Voltage Detect (LVD) circuits provide protection against low voltage conditions. The POR function senses Vdd and holds the system in reset until the magnitude of Vdd supports operation to specification.

The LVD function senses Vdd and provides an interrupt to the system when Vdd falls below a selected ***threshold***. Other outputs and status bits are provided to indicate important voltage trip levels.

Refer to Section 30.2 Pin Behavior During Reset for a description of GPIO pin behavior during power up.

31.2 Register Definitions

The following registers are associated with the POR and LVD, and are listed in address order. The register descriptions below have an associated register table showing the bit structure.

The bits that are grayed out in the register tables are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of '0'. For a complete table of the POR and LVD registers, refer to the "Summary Table of the System Resource Registers" on page 272.

31.2.1 VLT_CR Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E3h	VLT_CR			PORLEV[1:0]		LVDTBEN	VM[2:0]			RW : 00

The Voltage Monitor Control Register (VLT_CR) is used to set the trip points for POR, LVD, and the supply pump.

The VLT_CR register is cleared by all resets, which can cause reset cycling during very slow supply ramps to 5V when the POR range is set for the 5V range. This is because the reset clears the POR range setting back to 3V and a new boot/start up occurs (possibly many times). The user can manage this with sleep mode and/or reading voltage status bits, if such cycling is an issue.

Bits 5 and 4: PORLEV[1:0]. These bits set the Vdd level at which PPOR switches to one of three valid values. Note that 11b is a reserved value and therefore should not be used.

The three valid settings for these bits are:
- ❒ 00b (3V or 2.4V operation
- ❒ 01b (4.5V or 3.0V operation
- ❒ 10b (4.75V operation)

See the "DC POR and LVD Specifications" table in the Electrical Specifications section of the PSoC device data sheet for voltage tolerances for each setting.

Bit 3: LVDTBEN. This bit is AND'ed with LVD to produce a throttle-back signal that reduces CPU clock speed when low voltage conditions are detected. When the throttle-back signal is asserted, the CPU speed bits in the OSC_CR0 register are reset, forcing the CPU speed to 3 MHz or EXTCLK / 8.

Bits 2 to 0: VM[2:0]. These bits set the Vdd level at which LVD and the Pump Comparator switches.

For additional information, refer to the VLT_CR register on page 154.

31.2.2 VLT_CMP Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E4h	VLT_CMP						PUMP	LVD	PPOR	R : 00

The Voltage Monitor Comparators Register (VLT_CMP) is used to read the state of internal supply voltage monitors.

Bit 3: NoWrite. This bit is only used in PSoC devices with a 2.4V minimum POR. It reads the state of the Flash write voltage monitor.

Bit 2: PUMP. This bit reads the state of the Switch Mode Pump Vdd comparator. The trip points for both LVD and PUMP are set by VM[2:0] in the VLT_CR register.

Bit 1: LVD. This bit reads the state of the low voltage detect comparator. The trip points for both LVD and PUMP are set by VM[2:0] in the VLT_CR register.

Bit 0: PPOR. This bit reads back the state of the PPOR output. This can only be meaningfully read with PORLEV[1:0] set to disable PPOR. In that case, the PPOR status bit shows the comparator state directly.

For additional information, refer to the VLT_CMP register on page 155.

Section G: Glossary

The Glossary section explains the terminology used in this technical reference manual. Glossary terms are characterized in ***bold, italic font*** throughout the text of this manual.

A

accumulator	In a CPU, a register in which intermediate results are stored. Without an accumulator, it would be necessary to write the result of each calculation (addition, subtraction, shift, and so on.) to main memory and read them back. Access to main memory is slower than access to the accumulator, which usually has direct paths to and from the arithmetic and logic unit (ALU).
active high	1. A logic signal having its asserted state as the logic 1 state. 2. A logic signal having the logic 1 state as the higher voltage of the two states.
active low	1. A logic signal having its asserted state as the logic 0 state. 2. A logic signal having its logic 1 state as the lower voltage of the two states: inverted logic.
address	The label or number identifying the memory location (RAM, ROM, or register) where a unit of information is stored.
algorithm	A procedure for solving a mathematical problem in a finite number of steps that frequently involve repetition of an operation.
ambient temperature	The temperature of the air in a designated area, particularly the area surrounding the PSoC device.
analog	See ***analog signals***.
analog blocks	The basic programmable opamp circuits. These are SC (switched capacitor) and CT (continuous time) blocks. These blocks can be interconnected to provide ADCs, DACs, multi-pole filters, gain stages, and much more.
analog output	An output that is capable of driving any voltage between the supply rails, instead of just a logic 1 or logic 0.
analog signals	A signal represented in a continuous form with respect to continuous times, as contrasted with a digital signal represented in a discrete (discontinuous) form in a sequence of time.
analog-to-digital (ADC)	A device that changes an analog signal to a digital signal of corresponding magnitude. Typically, an ADC converts a voltage to a digital number. The ***digital-to-analog (DAC)*** converter performs the reverse operation.

AND	See ***Boolean Algebra***.
API (Application Programming Interface)	A series of software routines that comprise an interface between a computer application and lower-level services and functions (for example, user modules and libraries). APIs serve as building blocks for programmers that create software applications.
array	An array, also known as a vector or list, is one of the simplest data structures in computer programming. Arrays hold a fixed number of equally-sized data elements, generally of the same data type. Individual elements are accessed by index using a consecutive range of integers, as opposed to an associative array. Most high level programming languages have arrays as a built-in data type. Some arrays are multi-dimensional, meaning they are indexed by a fixed number of integers; for example, by a group of two integers. One- and two-dimensional arrays are the most common. Also, an array can be a group of capacitors or resistors connected in some common form.
assembly	A symbolic representation of the machine language of a specific processor. Assembly language is converted to machine code by an assembler. Usually, each line of assembly code produces one machine instruction, though the use of macros is common. Assembly languages are considered low level languages; where as C is considered a high level language.
asynchronous	A signal whose data is acknowledged or acted upon immediately, irrespective of any clock signal.
attenuation	The decrease in intensity of a signal as a result of absorption of energy and of scattering out of the path to the detector, but not including the reduction due to geometric spreading. Attenuation is usually expressed in dB.

B

bandgap reference	A stable voltage reference design that matches the positive temperature coefficient of V_T with the negative temperature coefficient of V_{BE}, to produce a zero temperature coefficient (ideally) reference.
bandwidth	1. The frequency range of a message or information processing system measured in hertz. 2. The width of the spectral region over which an amplifier (or absorber) has substantial gain (or loss); it is sometimes represented more specifically as, for example, full width at half maximum.
bias	1. A systematic deviation of a value from a reference value. 2. The amount by which the average of a set of values departs from a reference value. 3. The electrical, mechanical, magnetic, or other force (field) applied to a device to establish a reference level to operate the device.
bias current	The constant low level DC current that is used to produce a stable operation in amplifiers. This current can sometimes be changed to alter the bandwidth of an amplifier.
binary	The name for the base 2 numbering system. The most common numbering system is the base 10 numbering system. The base of a numbering system indicates the number of values that may exist for a particular positioning within a number for that system. For example, in base 2, binary, each position may have one of two values (0 or 1). In the base 10, decimal, numbering system, each position may have one of ten values (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9).

bit	A single digit of a binary number. Therefore, a bit may only have a value of ‘0’ or ‘1’. A group of 8 bits is called a byte. Because the PSoC's M8C is an 8-bit microcontroller, the PSoC's native data chunk size is a byte.
bit rate (BR)	The number of bits occurring per unit of time in a bit stream, usually expressed in bits per second (bps).
block	1. A functional unit that performs a single function, such as an oscillator. 2. A functional unit that may be configured to perform one of several functions, such as a digital PSoC block or an analog PSoC block.
Boolean Algebra	In mathematics and computer science, Boolean algebras or Boolean lattices, are algebraic structures which "capture the essence" of the logical operations AND, OR and NOT as well as the set theoretic operations union, intersection, and complement. Boolean algebra also defines a set of theorems that describe how Boolean equations can be manipulated. For example, these theorems are used to simplify Boolean equations, which reduce the number of logic elements needed to implement the equation. The operators of Boolean algebra may be represented in various ways. Often they are simply written as AND, OR, and NOT. In describing circuits, NAND (NOT AND), NOR (NOT OR), XNOR (exclusive NOT OR), and XOR (exclusive OR) may also be used. Mathematicians often use + (for example, A+B) for OR and • for AND (for example, A*B) (since in some ways those operations are analogous to addition and multiplication in other algebraic structures) and represent NOT by a line drawn above the expression being negated (for example, ~A, A_, !A).
break-before-make	The elements involved go through a disconnected state entering (‘break”) before the new connected state (“make”).
broadcast net	A signal that is routed throughout the microcontroller and is accessible by many blocks or systems.
buffer	1. A storage area for data that is used to compensate for a speed difference, when transferring data from one device to another. Usually refers to an area reserved for IO operations, into which data is read, or from which data is written. 2. A portion of memory set aside to store data, often before it is sent to an external device or as it is received from an external device. 3. An amplifier used to lower the output impedance of a system.
bus	1. A named connection of nets. Bundling nets together in a bus makes it easier to route nets with similar routing patterns. 2. A set of signals performing a common function and carrying similar data. Typically represented using vector notation; for example, address[7:0]. 3. One or more conductors that serve as a common connection for a group of related devices.
byte	A digital storage unit consisting of 8 bits.

C

C	A high level programming language.
capacitance	A measure of the ability of two adjacent conductors, separated by an insulator, to hold a charge when a voltage differential is applied between them. Capacitance is measured in units of Farads.

capture	To extract information automatically through the use of software or hardware, as opposed to hand-entering of data into a computer file.
chaining	Connecting two or more 8-bit digital blocks to form 16-, 24-, and even 32-bit functions. Chaining allows certain signals such as Compare, Carry, Enable, Capture, and Gate to be produced from one block to another.
checksum	The checksum of a set of data is generated by adding the value of each data word to a sum. The actual checksum can simply be the result sum or a value that must be added to the sum to generate a pre-determined value.
clear	To force a bit/register to a value of logic ‘0’.
clock	The device that generates a periodic signal with a fixed frequency and duty cycle. A clock is sometimes used to synchronize different logic blocks.
clock generator	A circuit that is used to generate a clock signal.
CMOS	The logic gates constructed using ***MOS*** transistors connected in a complementary manner. CMOS is an acronym for complementary metal-oxide semiconductor.
comparator	An electronic circuit that produces an output voltage or current whenever two input levels simultaneously satisfy pre-determined amplitude requirements.
compiler	A program that translates a high level language, such as C, into machine language.
configuration	In a computer system, an arrangement of functional units according to their nature, number, and chief characteristics. Configuration pertains to hardware, software, firmware, and documentation. The configuration affects system performance.
configuration space	In PSoC devices, the register space accessed when the XIO bit, in the CPU_F register, is set to ‘1’.
crowbar	A type of over-voltage protection that rapidly places a low resistance shunt (typically an SCR) from the signal to one of the power supply rails, when the output voltage exceeds a pre-determined value.
crystal oscillator	An oscillator in which the frequency is controlled by a piezoelectric crystal. Typically a piezoelectric crystal is less sensitive to ambient temperature than other circuit components.
cyclic redundancy check (CRC)	A calculation used to detect errors in data communications, typically performed using a linear feedback shift register. Similar calculations may be used for a variety of other purposes such as data compression.

D

data bus	A bi-directional set of signals used by a computer to convey information from a memory location to the central processing unit and vice versa. More generally, a set of signals used to convey data between digital functions.
data stream	A sequence of digitally encoded signals used to represent information in transmission.
data transmission	The sending of data from one place to another by means of signals over a channel.

debugger	A hardware and software system that allows the user to analyze the operation of the system under development. A debugger usually allows the developer to step through the firmware one step at a time, set break points, and analyze memory.
dead band	A period of time when neither of two or more signals are in their active state or in transition.
decimal	A base-10 numbering system, which uses the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 (called digits) together with the decimal point and the sign symbols + (plus) and - (minus) to represent numbers.
default value	Pertaining to the pre-defined initial, original, or specific setting, condition, value, or action a system assumes, uses, or takes in the absence of instructions from the user.
device	The device referred to in this manual is the PSoC chip, unless otherwise specified.
die	An unpackaged integrated circuit (IC), normally cut from a wafer.
digital	A signal or function, the amplitude of which is characterized by one of two discrete values: ‘0’ or ‘1’.
digital blocks	The 8-bit logic blocks that can act as a counter, timer, serial receiver, serial transmitter, CRC generator, pseudo-random number generator, or SPI.
digital logic	A methodology for dealing with expressions containing two-state variables that describe the behavior of a circuit or system.
digital-to-analog (DAC)	A device that changes a digital signal to an analog signal of corresponding magnitude. The ***analog-to-digital (ADC)*** converter performs the reverse operation.
direct access	The capability to obtain data from a storage device, or to enter data into a storage device, in a sequence independent of their relative positions by means of addresses that indicate the physical location of the data.
duty cycle	The relationship of a clock period ***high time*** to its ***low time***, expressed as a percent.

E

emulator	Duplicates (provides an emulation of) the functions of one system with a different system, so that the second system appears to behave like the first system.
External Reset (XRES)	An active high signal that is driven into the PSoC device. It causes all operation of the CPU and blocks to stop and return to a pre-defined state.

F

falling edge	A transition from a logic 1 to a logic 0. Also known as a negative edge.
feedback	The return of a portion of the output, or processed portion of the output, of a (usually active) device to the input.
filter	A device or process by which certain frequency components of a signal are attenuated.

firmware	The software that is embedded in a hardware device and executed by the CPU. The software may be executed by the end user, but it may not be modified.
flag	Any of various types of indicators used for identification of a condition or event (for example, a character that signals the termination of a transmission).
Flash	An electrically programmable and erasable, non-***volatile*** technology that provides users with the programmability and data storage of EPROMs, plus in-system erasability. Non-volatile means that the data is retained when power is off.
Flash bank	A group of Flash ROM blocks where Flash block numbers always begin with '0' in an individual Flash bank. A Flash bank also has its own block level protection information.
Flash block	The smallest amount of Flash ROM space that may be programmed at one time and the smallest amount of Flash space that may be protected. A Flash block holds 64 bytes.
flip-flop	A device having two stable states and two input terminals (or types of input signals) each of which corresponds with one of the two states. The circuit remains in either state until it is made to change to the other state by application of the corresponding signal.
frequency	The number of cycles or events per unit of time, for a periodic function.

G

gain	The ratio of output current, voltage, or power to input current, voltage, or power, respectively. Gain is usually expressed in dB.
gate	1. A device having one output channel and one or more input channels, such that the output channel state is completely determined by the input channel states, except during switching transients. 2. One of many types of combinational logic elements having at least two inputs (for example, AND, OR, NAND, and NOR (also see ***Boolean Algebra***)).
ground	1. The electrical neutral line having the same potential as the surrounding earth. 2. The negative side of DC power supply. 3. The reference point for an electrical system. 4. The conducting paths between an electric circuit or equipment and the earth, or some conducting body serving in place of the earth.

H

hardware	A comprehensive term for all of the physical parts of a computer or embedded system, as distinguished from the data it contains or operates on, and the software that provides instructions for the hardware to accomplish tasks.
hardware reset	A reset that is caused by a circuit, such as a POR, watchdog reset, or external reset. A hardware reset restores the state of the device as it was when it was first powered up. Therefore, all registers are set to the POR value as indicated in register tables throughout this document.

hexidecimal	A base 16 numeral system (often abbreviated and called hex), usually written using the symbols 0-9 and A-F. It is a useful system in computers because there is an easy mapping from four bits to a single hex digit. Thus, one can represent every byte as two consecutive hexadecimal digits. Compare the binary, hex, and decimal representations:

bin	**=**	**hex**	**=**	**dec**
0000b	=	0x0	=	0
0001b	=	0x1	=	1
0010b	=	0x2	=	2
...				
1001b	=	0x9	=	9
1010b	=	0xA	=	10
1011b	=	0xB	=	11
...				
1111b	=	0xF	=	15

So the decimal numeral 79 whose binary representation is 0100 1111b can be written as 4Fh in hexadecimal (0x4F).

high time	The amount of time the signal has a value of ‘1’ in one period, for a periodic digital signal.

I

I2C	A two-wire serial computer bus by Phillips Semiconductors. I2C is an Inter-Integrated Circuit. It is used to connect low-speed peripherals in an embedded system. The original system was created in the early 1980s as a battery control interface, but it was later used as a simple internal bus system for building control electronics. I2C uses only two bi-directional pins, clock and data, both running at +5V and pulled high with resistors. The bus operates at 100 kbits/second in standard mode and 400 kbits/second in fast mode. I^{2}C™ is a trademark of the Philips Semiconductors.
ICE	The in-circuit emulator that allows users to test the project in a hardware environment, while viewing the debugging device activity in a software environment (PSoC Designer).
idle state	A condition that exists whenever user messages are not being transmitted, but the service is immediately available for use.
impedance	1. The resistance to the flow of current caused by resistive, capacitive, or inductive devices in a circuit. 2. The total passive opposition offered to the flow of electric current. Note the impedance is determined by the particular combination of resistance, inductive reactance, and capacitive reactance in a given circuit.
input	A point that accepts data, in a device, process, or channel.
input/output (IO)	A device that introduces data into or extracts data from a system.
instruction	An expression that specifies one operation and identifies its operands, if any, in a programming language such as C or assembly.
integrated circuit (IC)	A device in which components such as resistors, capacitors, diodes, and ***transistors*** are formed on the surface of a single piece of semiconductor.
interface	The means by which two systems or devices are connected and interact with each other.

interrupt	A suspension of a process, such as the execution of a computer program, caused by an event external to that process, and performed in such a way that the process can be resumed.
interrupt service routine (ISR)	A block of code that normal code execution is diverted to when the M8C receives a hardware interrupt. Many interrupt sources may each exist with its own priority and individual ISR code block. Each ISR code block ends with the RETI instruction, returning the device to the point in the program where it left normal program execution.

J

jitter	1. A misplacement of the timing of a transition from its ideal position. A typical form of corruption that occurs on serial data streams. 2. The abrupt and unwanted variations of one or more signal characteristics, such as the interval between successive pulses, the amplitude of successive cycles, or the frequency or phase of successive cycles.

K

keeper	A circuit that holds a signal to the last driven value, even when the signal becomes un-driven.

L

latency	The time or delay that it takes for a signal to pass through a given circuit or network.
least significant bit (LSb)	The binary digit, or bit, in a binary number that represents the least significant value (typically the right-hand bit). The bit versus byte distinction is made by using a lower case “b” for bit in LSb.
least significant byte (LSB)	The byte in a multi-byte word that represents the least significant values (typically the right-hand byte). The byte versus bit distinction is made by using an upper case “B” for byte in LSB.
Linear Feedback Shift Register (LFSR)	A shift register whose data input is generated as an ***XOR*** of two or more elements in the register chain.
load	The electrical demand of a process expressed as power (watts), current (amps), or resistance (ohms).
logic function	A mathematical function that performs a digital operation on digital data and returns a digital value.
look-up table (LUT)	A logic block that implements several logic functions. The logic function is selected by means of select lines and is applied to the inputs of the block. For example: A 2 input LUT with 4 select lines can be used to perform any one of 16 logic functions on the two inputs resulting in a single logic output. The LUT is a combinational device; therefore, the input/output relationship is continuous, that is, not sampled.
low time	The amount of time the signal has a value of ‘0’ in one period, for a periodic digital signal.
low voltage detect (LVD)	A circuit that senses Vdd and provides an interrupt to the system when Vdd falls below a selected threshold.

M

M8C	An 8-bit Harvard Architecture microprocessor. The microprocessor coordinates all activity inside a PSoC by interfacing to the Flash, SRAM, and register space.
macro	A programming language macro is an abstraction, whereby a certain textual pattern is replaced according to a defined set of rules. The interpreter or compiler automatically replaces the macro instance with the macro contents when an instance of the macro is encountered. Therefore, if a macro is used 5 times and the macro definition required 10 bytes of code space, 50 bytes of code space is needed in total.
mask	1. To obscure, hide, or otherwise prevent information from being derived from a signal. It is usually the result of interaction with another signal, such as noise, static, jamming, or other forms of interference. 2. A pattern of bits that can be used to retain or suppress segments of another pattern of bits, in computing and data processing systems.
master device	A device that controls the timing for data exchanges between two devices. Or when devices are cascaded in width, the master device is the one that controls the timing for data exchanges between the cascaded devices and an external interface. The controlled device is called the ***slave device***.
microcontroller	An integrated circuit chip that is designed primarily for control systems and products. In addition to a CPU, a microcontroller typically includes memory, timing circuits, and IO circuitry. The reason for this is to permit the realization of a controller with a minimal quantity of chips, thus achieving maximal possible miniaturization. This in turn, reduces the volume and the cost of the controller. The microcontroller is normally not used for general-purpose computation as is a microprocessor.
mnemonic	A tool intended to assist the memory. Mnemonics rely on not only repetition to remember facts, but also on creating associations between easy-to-remember constructs and lists of data. A two to four character string representing a microprocessor instruction.
mode	A distinct method of operation for software or hardware. For example, the Digital PSoC block may be in either counter mode or timer mode.
modulation	A range of techniques for encoding information on a carrier signal, typically a sine-wave signal. A device that performs modulation is known as a modulator.
Modulator	A device that imposes a signal on a carrier.
MOS	An acronym for metal-oxide semiconductor.
most significant bit (MSb)	The binary digit, or bit, in a binary number that represents the most significant value (typically the left-hand bit). The bit versus byte distinction is made by using a lower case “b” for bit in MSb.
most significant byte (MSB)	The byte in a multi-byte word that represents the most significant values (typically the left-hand byte). The byte versus bit distinction is made by using an upper case “B” for byte in MSB.
multiplexer (mux)	1. A logic function that uses a binary value, or address, to select between a number of inputs and conveys the data from the selected input to the output. 2. A technique which allows different input (or output) signals to use the same lines at different times, controlled by an external signal. Multiplexing is used to save on wiring and IO ports.

N

NAND	See ***Boolean Algebra***.
negative edge	A transition from a logic 1 to a logic 0. Also known as a falling edge.
net	The routing between devices.
nibble	A group of four bits, which is one-half of a byte.
noise	1. A disturbance that affects a signal and that may distort the information carried by the signal. 2. The random variations of one or more characteristics of any entity such as voltage, current, or data.
NOR	See ***Boolean Algebra***.
NOT	See ***Boolean Algebra***.

O

OR	See ***Boolean Algebra***.
oscillator	A circuit that may be crystal controlled and is used to generate a clock frequency.
output	The electrical signal or signals which are produced by an analog or digital block.

P

parallel	The means of communication in which digital data is sent multiple bits at a time, with each simultaneous bit being sent over a separate line.
parameter	Characteristics for a given block that have either been characterized or may be defined by the designer.
parameter block	A location in memory where parameters for the SSC instruction are placed prior to execution.
parity	A technique for testing transmitting data. Typically, a binary digit is added to the data to make the sum of all the digits of the binary data either always even (even parity) or always odd (odd parity).
path	1. The logical sequence of instructions executed by a computer. 2. The flow of an electrical signal through a circuit.
pending interrupts	An interrupt that has been triggered but has not been serviced, either because the processor is busy servicing another interrupt or global interrupts are disabled.
phase	The relationship between two signals, usually the same frequency, that determines the delay between them. This delay between signals is either measured by time or angle (degrees).
Phase-Locked Loop (PLL)	An electronic circuit that controls an ***oscillator*** so that it maintains a constant phase angle relative to a reference signal.

pin	A terminal on a hardware component. Also called lead.
pinouts	The pin number assignment: the relation between the logical inputs and outputs of the PSoC device and their physical counterparts in the printed circuit board (PCB) package. Pinouts involve pin numbers as a link between schematic and PCB design (both being computer generated files) and may also involve pin names.
port	A group of pins, usually eight.
positive edge	A transition from a logic 0 to a logic 1. Also known as a rising edge.
posted interrupts	An interrupt that has been detected by the hardware but may or may not be enabled by its mask bit. Posted interrupts that are not masked become pending interrupts.
Power On Reset (POR)	A circuit that forces the PSoC device to reset when the voltage is below a pre-set level. This is one type of ***hardware reset***.
program counter	The instruction pointer (also called the program counter) is a register in a computer processor that indicates where in memory the CPU is executing instructions. Depending on the details of the particular machine, it holds either the address of the instruction being executed, or the address of the next instruction to be executed.
protocol	A set of rules. Particularly the rules that govern networked communications.
PSoC	Cypress Semiconductor's Programmable System-on-Chip (PSoC). PSoC® is a registered trademark and Programmable System-on-Chip™ is a trademark of Cypress.
PSoC blocks	*See* ***analog blocks*** and ***digital blocks***.
PSoC Designer	The software for Cypress' Programmable System-on-Chip technology.
pulse	A rapid change in some characteristic of a signal (for example, phase or frequency), from a baseline value to a higher or lower value, followed by a rapid return to the baseline value.
pulse width modulator (PWM)	An output in the form of duty cycle which varies as a function of the applied measurand.

R

RAM	An acronym for random access memory. A data-storage device from which data can be read out and new data can be written in.
register	A storage device with a specific capacity, such as a bit or byte.
reset	A means of bringing a system back to a know state. See ***hardware reset*** and ***software reset***.
resistance	The resistance to the flow of electric current measured in ohms for a conductor.
revision ID	A unique identifier of the PSoC device.
ripple divider	An asynchronous ripple counter constructed of flip-flops. The clock is fed to the first stage of the counter. An n-bit binary counter consisting of n flip-flops that can count in binary from 0 to 2^n - 1.

rising edge	*See **positive edge***.
ROM	An acronym for read only memory. A data-storage device from which data can be read out, but new data cannot be written in.
routine	A block of code, called by another block of code, that may have some general or frequent use.
routing	Physically connecting objects in a design according to design rules set in the reference library.
runt pulses	In digital circuits, narrow pulses that, due to non-zero rise and fall times of the signal, do not reach a valid high or low level. For example, a runt pulse may occur when switching between asynchronous clocks or as the result of a race condition in which a signal takes two separate paths through a circuit. These race conditions may have different delays and are then recombined to form a glitch or when the output of a flip-flop becomes metastable.

S

sampling	The process of converting an analog signal into a series of digital values or reversed.
schematic	A diagram, drawing, or sketch that details the elements of a system, such as the elements of an electrical circuit or the elements of a logic diagram for a computer.
seed value	An initial value loaded into a linear feedback shift register or random number generator.
serial	1. Pertaining to a process in which all events occur one after the other. 2. Pertaining to the sequential or consecutive occurrence of two or more related activities in a single device or channel.
set	To force a bit/register to a value of logic 1.
settling time	The time it takes for an output signal or value to stabilize after the input has changed from one value to another.
shift	The movement of each bit in a word one position to either the left or right. For example, if the hex value 0x24 is shifted one place to the left, it becomes 0x48. If the hex value 0x24 is shifted one place to the right, it becomes 0x12.
shift register	A memory storage device that sequentially shifts a word either left or right to output a stream of serial data.
sign bit	The most significant binary digit, or bit, of a signed binary number. If set to a logic 1, this bit represents a negative quantity.
signal	A detectable transmitted energy that can be used to carry information. As applied to electronics, any transmitted electrical impulse.
silicon ID	A unique identifier of the PSoC silicon.
skew	The difference in arrival time of bits transmitted at the same time, in parallel transmission.

slave device	A device that allows another device to control the timing for data exchanges between two devices. Or when devices are cascaded in width, the slave device is the one that allows another device to control the timing of data exchanges between the cascaded devices and an external interface. The controlling device is called the master device.
software	A set of computer programs, procedures, and associated documentation concerned with the operation of a data processing system (for example, compilers, library routines, manuals, and circuit diagrams). Software is often written first as source code, and then converted to a binary format that is specific to the device on which the code executes.
software reset	A partial reset executed by software to bring part of the system back to a known state. A software reset restores the M8C to a known state but not PSoC blocks, systems, peripherals, or registers. For a software reset, the CPU registers (CPU_A, CPU_F, CPU_PC, CPU_SP, and CPU_X) are set to 0x00. Therefore, code execution begins at Flash address 0x0000.
SRAM	An acronym for static random access memory. A memory device allowing users to store and retrieve data at a high rate of speed. The term static is used because, once a value has been loaded into an SRAM cell, it remains unchanged until it is explicitly altered or until power is removed from the device.
SROM	An acronym for supervisory read only memory. The SROM holds code that is used to boot the device, calibrate circuitry, and perform Flash operations. The functions of the SROM may be accessed in normal user code, operating from Flash.
stack	A stack is a data structure that works on the principle of Last In First Out (LIFO). This means that the last item put on the stack is the first item that can be taken off.
stack pointer	A stack may be represented in a computer's inside blocks of memory cells, with the bottom at a fixed location and a variable stack pointer to the current top cell.
state machine	The actual implementation (in hardware or software) of a function that can be considered to consist of a set of states through which it sequences.
sticky	A bit in a register that maintains its value past the time of the event that caused its transition, has passed.
stop bit	A signal following a character or block that prepares the receiving device to receive the next character or block.
switching	The controlling or routing of signals in circuits to execute logical or arithmetic operations, or to transmit data between specific points in a network.
Switch phasing	The clock that controls a given switch, PHI1 or PHI2, in respect to the switched capacitor (SC) blocks. The PSoC SC blocks have two groups of switches. One group of these switches is normally closed during PHI1 and open during PHI2. The other group is open during PHI1 and closed during PHI2. These switches can be controlled in the normal operation, or in reverse mode if the PHI1 and PHI2 clocks are reversed.
synchronous	1. A signal whose data is not acknowledged or acted upon until the next active edge of a clock signal. 2. A system whose operation is synchronized by a clock signal.

T

tap	The connection between two blocks of a device created by connecting several blocks/components in a series, such as a shift register or resistive voltage divider.
terminal count	The state at which a counter is counted down to zero.
threshold	The minimum value of a signal that can be detected by the system or sensor under consideration.
transistors	The transistor is a solid-state semiconductor device used for amplification and switching, and has three terminals: a small current or voltage applied to one terminal controls the current through the other two. It is the key component in all modern electronics. In digital circuits, transistors are used as very fast electrical switches, and arrangements of transistors can function as logic gates, RAM-type memory, and other devices. In analog circuits, transistors are essentially used as amplifiers.
tri-state	A function whose output can adopt three states: 0, 1, and Z (high-impedance). The function does not drive any value in the Z state and, in many respects, may be considered to be disconnected from the rest of the circuit, allowing another output to drive the same ***net***.

U

UART	A UART or universal asynchronous receiver-transmitter translates between parallel bits of data and serial bits.
user	The person using the PSoC device and reading this manual.
user modules	Pre-built, pre-tested hardware/firmware peripheral functions that take care of managing and configuring the lower level Analog and Digital PSoC Blocks. User Modules also provide high level ***API (Application Programming Interface)*** for the peripheral function.
user space	The bank 0 space of the register map. The registers in this bank are more likely to be modified during normal program execution and not just during initialization. Registers in bank 1 are most likely to be modified only during the initialization phase of the program.

V

Vdd	A name for a power net meaning "voltage drain." The most positive power supply signal. Usually 5 or 3.3 volts.
volatile	Not guaranteed to stay the same value or level when not in scope.
Vss	A name for a power net meaning "voltage source." The most negative power supply signal.

W

watchdog timer	A timer that must be serviced periodically. If it is not serviced, the CPU resets after a specified period of time.
waveform	The representation of a signal as a plot of amplitude versus time.

X

XOR	See ***Boolean Algebra***.

Index