CY8C22113, CY8C22213

PSoC™ Mixed Signal Array
Preliminary Data Sheet
For Silicon Revision A

December 22, 2003

Cypress MicroSystems
2700 162nd Street SW
Building D
Lynnwood, WA 98037
Phone: 800.669.0557
FAX: 425.787.4641
http://www.cypress.com

Document No. 38-12009 Rev. *D

Contents

SECTION A OVERVIEW

The PSoC™ family consists of many *Mixed Signal Array with On-Chip Controller* devices. These devices are designed to replace multiple traditional MCU-based system components with one, low cost single-chip programmable component. A PSoC device includes configurable blocks of analog 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 IO are included in a range of convenient pin-outs.

The Overview section discusses the Features, Getting Started, Top-Level Architecture, Development Tools, User Modules and Development Process, along with Ordering Information. It also lists the Conventions used in this document. This section encompasses the following chapters:

Features

- Powerful Harvard Architecture Processor
  - –M8C Processor Speeds to 24 MHz
  - –Low Power at High Speed
  - –3.0 to 5.25 V Operating Voltage
  - –Industrial Temperature Range: –40°C to +85°C
- Advanced Peripherals (PSoC Blocks)
  - –3 Rail-to-Rail Analog PSoC Blocks Provide:
    - Up to 14-Bit ADCs
    - Up to 8-Bit DACs
    - Programmable Gain Amplifiers
    - Programmable Filters and Comparators
  - –4 Digital PSoC Blocks Provide:
    - 8- to 32-Bit Timers, Counters and PWMs
    - CRC and PRS Modules
    - Full-Duplex UART
    - SPI™ Masters or Slaves
    - Connectable to all GPIO Pins
  - –Complex Peripherals by Combining Blocks
- Flexible On-Chip Memory
  - –2K Bytes Flash Program Storage
    - 50,000 Erase/Write Cycles
  - –256 Bytes SRAM Data Storage
  - –In-System Serial Programming (ISSP™)
  - –Partial Flash Updates
  - –Flexible Protection Modes
  - –EEPROM Emulation in Flash
- Precision, Programmable Clocking
  - –Internal ±2.5% 24/48 MHz Oscillator
  - –High Accuracy 24 MHz with Optional 32 kHz Crystal and PLL
  - –Optional External Oscillator, up to 24 MHz
  - –Internal Oscillator for Watchdog and Sleep
- Programmable Pin Configurations
  - –25 mA Drive on all GPIO
  - –Pull up, Pull down, High Z, Strong, or Open Drain Drive Modes on all GPIO
  - –Up to 8 Analog Inputs on GPIO
  - –One 30 mA Analog Output on GPIO
  - –Configurable Interrupt on all GPIO
- Additional System Resources
  - –I^2C Slave, Master, and Multi-Master to 400 kHz
  - –Watchdog and Sleep Timers
  - –User-Configurable Low Voltage Detection
  - –Integrated Supervisory Circuit
  - –On-Chip Precision Voltage Reference
- Complete Development Tools
  - –Free Development Software (PSoC Designer)
  - –Full-Featured, In-Circuit Emulator and Programmer:
    - Full-Speed Emulation
    - Complex Breakpoint Structure
    - 128K Bytes Trace Memory

Getting Started

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

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

Development Kits

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

Tele-Training

PSoC "Tele-training" is available for beginners and is taught by a live marketing or application engineer over the phone. Please see http://www.cypress.com/support/training.cfm for more details. Five training classes are available to accelerate the learning curve including introduction, designing, debugging, advanced design, advanced analog, as well as application-specific classes covering topics like PSoC and the LIN bus.

Consultants

Certified PSoC Consultants offer everything from technical assistance to completed PSoC designs. To contact or become a PSoC Consultant go to the following web site, http://www.cypress.com/support/cypros.cfm.

Technical Support

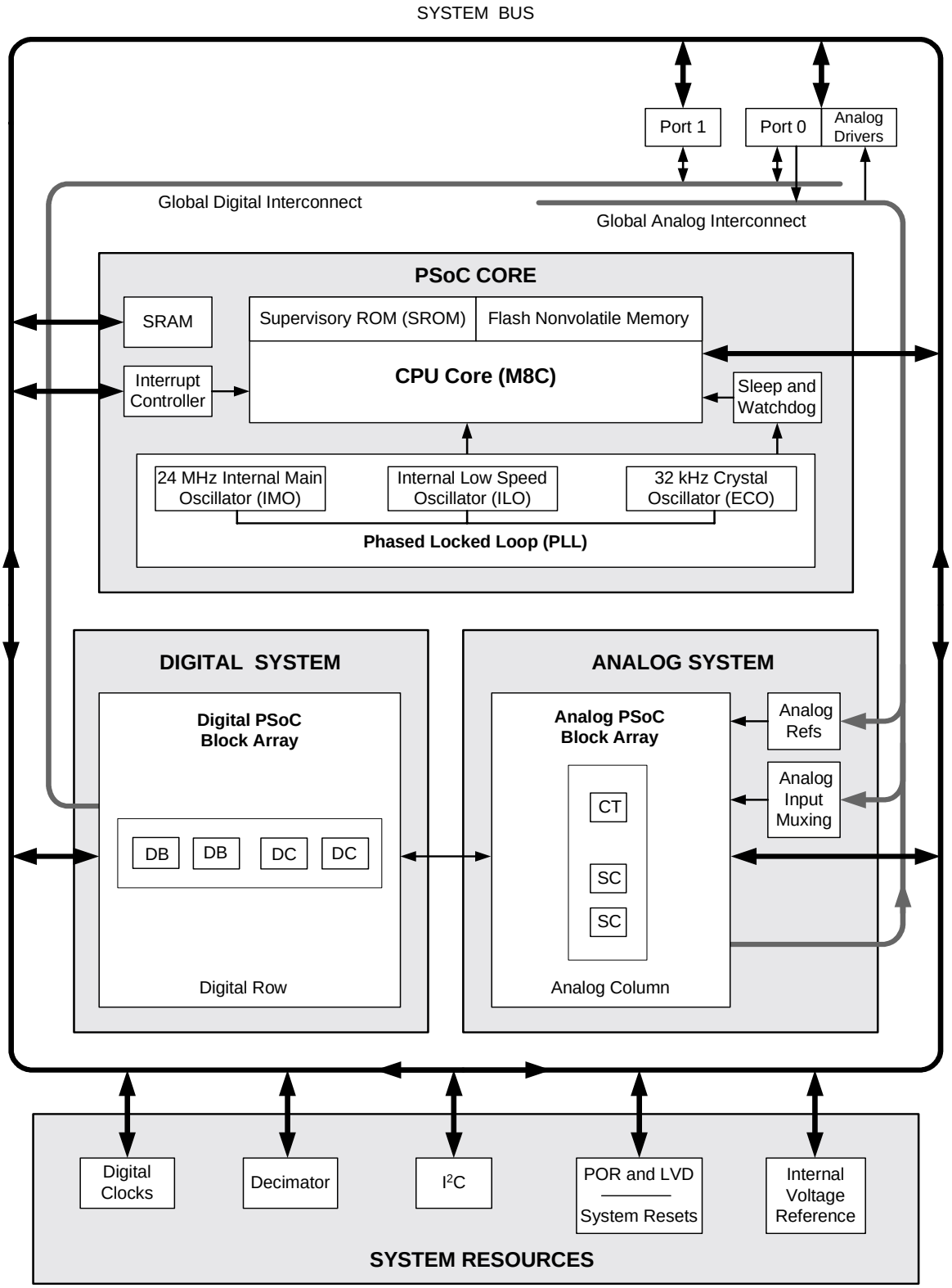
PSoC application engineers take pride in fast and accurate response. They can be reached with a 4-hour guaranteed response at http://www.cypress.com/support/login.cfm.

Application Notes

A long list of application notes will assist you in every aspect of your design effort. Go to http://www.cypress.com/design/results.cfm to locate the PSoC application notes.

Top-Level Architecture

The figure below illustrates the top-level architecture of the PSoC CY8C22xxx.



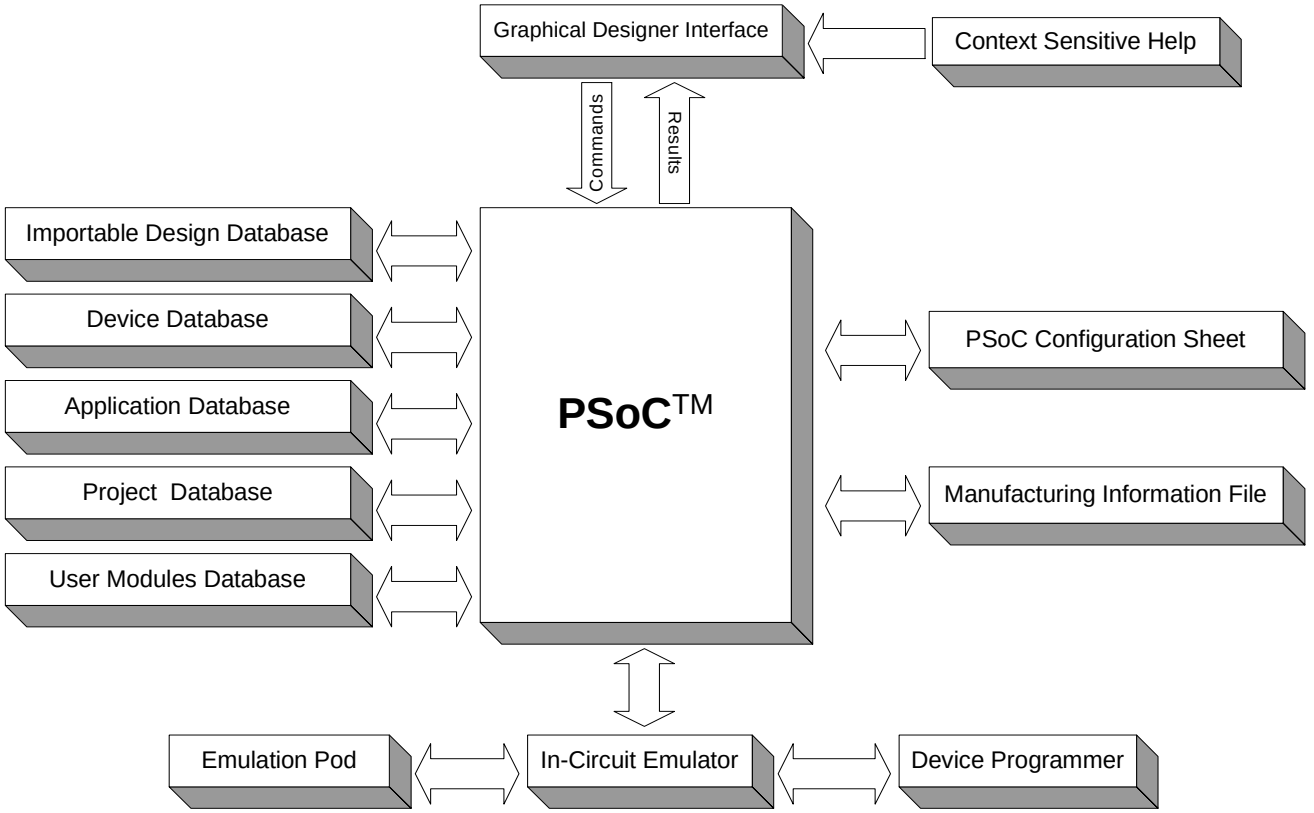
PSoC CY8C22xxx Top-Level Block Diagram

Development Tools

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

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

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



PSoC Designer Subsystems

PSoC Designer Software Subsystems

Device Editor

PSoC Designer has several main functions. In the Design Editor you can easily configure a design and APIs are automatically generated for the user modules. The Device Editor subsystem allows the user to select different onboard analog and digital components called user modules using the PSoC blocks. Examples of user modules are ADCs, DACs, Amplifiers, and Filters.

The device editor also supports easy development of multiple configurations and dynamic reconfiguration. Dynamic configuration allows for changing configuration at run time.

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

Design Browser

The Design Browser allows users to select and import preconfigured designs into the user's project. User's can easily browse a catalog of preconfigured designs to facilitate time-to-design. Recent examples provided in the tools include a 300-baud modem, Lin Bus master and slave, fan controller, and magnetic card reader.

Application Editor

In the Application Editor you can edit your C language and Assembly language source code. You can also assemble, compile, link, and build.

Assembler. The macro assembler allows the assembly code to be merged seamlessly with C code. The link libraries automatically use absolute addressing or can be compiled in relative mode, and linked with other software modules to get absolute addressing.

C Language Compiler. An ANSI C language compiler supports Cypress MicroSystems' PSoC family devices (except for 64-bit doubles). Even if you have never worked in the C language before, the product quickly allows you to create complete C programs for the PSoC family devices.

The embedded, optimizing C compiler provides all the features of C tailored to the PSoC architecture. It comes complete with embedded libraries providing port and bus operations, standard keypad and display support, and extended math functionality.

Debugger

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

Online Help System

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

Hardware Tools

In-Circuit Emulator

A low cost, high functionality ICE (In-Circuit Emulator) is available for development support. This hardware has the capability to program single devices.

The emulation consists of a base unit that connects to the PC by way of the parallel port. The base unit is universal and will operate with all PSoC devices. Emulation pods for each device family are available separately. The emulation pod takes the place of the PSoC device in the target board and performs full speed (24 MHz) operation.

PSoC Development Tool Kit

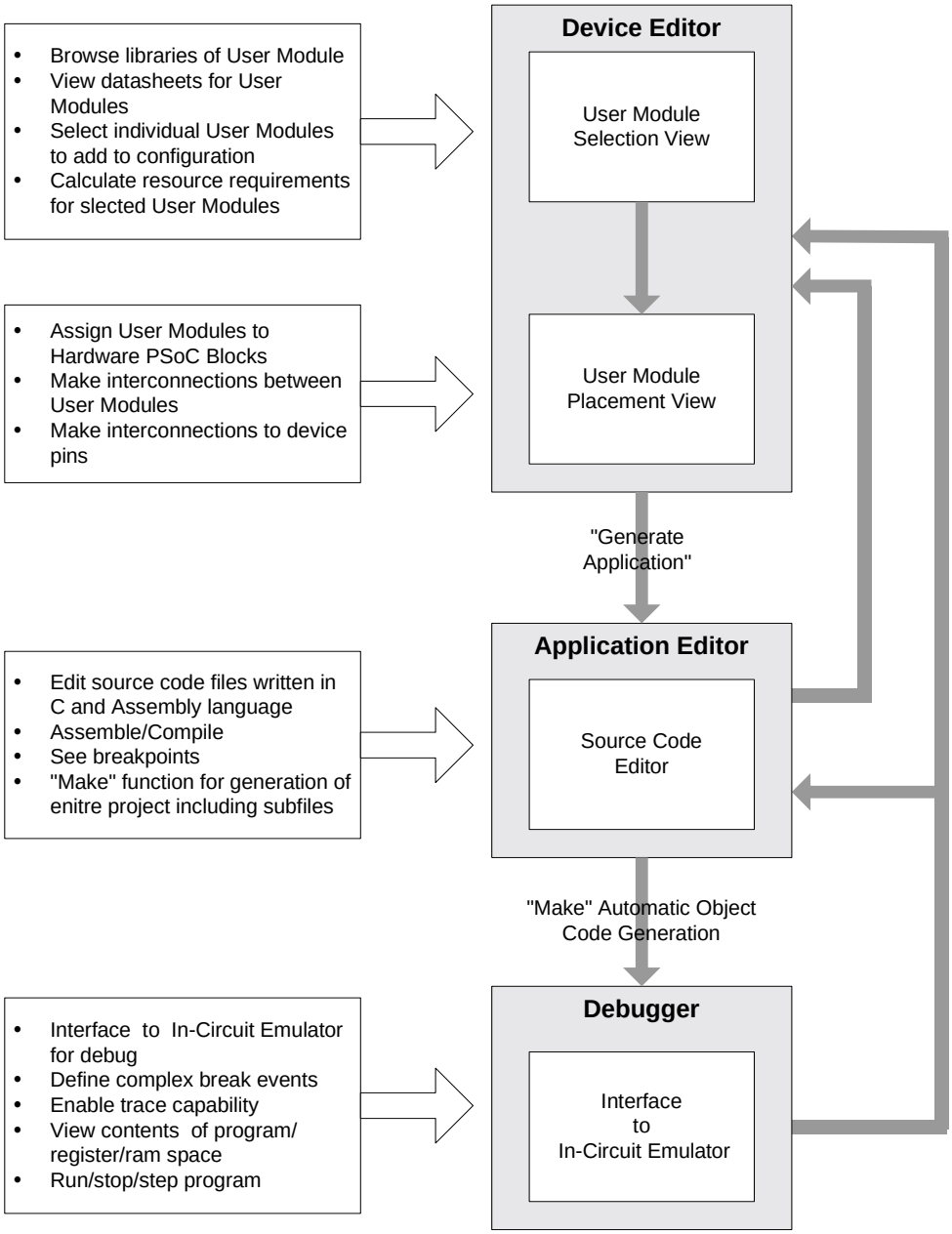
User Modules and Development Process

The development process for the PSoC is different than a traditional fixed function microcontroller. The flexibility of the PSoC architecture comes from configurable analog and digital hardware blocks called PSoC Blocks. These blocks have the capability to implement a wide variety of user selectable functions. Each block has several registers that allow you to select the function. These registers also determine the interconnections between this block and other blocks, as well as the connection to the I/O pins (reference the figure below).

To make the entire development process of your project easier, the PSoC Designer Integrated Development Environment (IDE) has libraries of open source code software modules, called "User Modules," that simplify the configuration process. These user modules have been created to make selecting and implementing peripheral functions very easy. User modules come in analog, digital, and mixed signal varieties. Each user module contains all the register settings to implement the selected function and also contains Application Programmer Interface (API) software to make the interface to your source code simple.

The development process starts when you open a new project. You than pick a set of user modules, as the basis of the custom configuration for that project. You can view the details of all the available user modules inside the development software and pick the user modules that are perfect for your application. You then must assign each of these user

modules to hardware resources. You also make the interconnections between the user modules, and between the user modules and the I/O pins. This process step takes place in the Device Editor subsystem within PSoC Designer. There are two views inside this step: one for selecting user modules and one for assigning them to the hardware blocks and interconnecting them. The last action in this step is to "Generate Application," which causes the development software to automatically generate the required files for the selected configuration.



User Modules and Development Process Flow Chart

The next step in the process is to write your main program, and any other sub-routines required by your application. This step takes place in the Application Editor subsystem. You will have all the subroutines automatically generated for the user modules you have chosen and the source code for these routines can be viewed in this step as well. The different files created for the project are all contained in a tree structure for easy reference. The development software has a handy "Make" function, which assembles and compiles all source files, and links them into an object file ready for the debugging process.

The last step in development takes place in the Debugger subsystem. This is where the object code is downloaded into the In-Circuit Emulator and run. The Debugger is both the interface to the ICE and also contains an advanced set

of tools for finding and removing bugs from your software. Some of the capabilities of the tools are full-speed emulation, defining complex breakpoint events, and a large trace memory.

Ordering Information

The following table lists the PSoC Device family's key features and ordering codes.

PSoC Device Family Key Features

Package	Ordering Code	Flash (Kbytes)	RAM (Bytes)	Switch Mode Pump	Temperature Range	Digital PSoC Blocks (Rows of 4)	Analog PSoC Blocks (Columns of 3)	Digital IO Pins	Analog Inputs	Analog Outputs	XRES Pin
8 Pin (300 Mil) DIP	CY8C22113-24PI	2	256	No	-40C to +85C	4	3	6	4	1	No
8 Pin (150 Mil) SOIC	CY8C22113-24SI	2	256	No	-40C to +85C	4	3	6	4	1	No
20 Pin (300 Mil) DIP	CY8C22213-24PI	2	256	No	-40C to +85C	4	3	16	8	1	Yes
20 Pin (210 Mil) SSOP	CY8C22213-24PVI	2	256	No	-40C to +85C	4	3	16	8	1	Yes
20 Pin (210 Mil) SSOP (Tape and Reel)	CY8C22213-24PVIT	2	256	No	-40C to +85C	4	3	16	8	1	Yes
20 Pin (300 Mil) SOIC	CY8C22213-24SI	2	256	No	-40C to +85C	4	3	16	8	1	Yes
20 Pin (300 Mil) SOIC (Tape and Reel)	CY8C22213-24SIT	2	256	No	-40C to +85C	4	3	16	8	1	Yes
32 Pin (5x5 mm) MLF	CY8C22213-24LFI	2	256	No	-40C to +85C	4	3	16	8	1	Yes

Organization and Conventions

Document Organization

This document is organized into the following sections:

- Overview
- Architecture
- Registers
- Digital System
- Analog System
- System Resources
- Electrical Specifications
- Revision History

Each section and its associated chapters is organized according to PSoC functionality. If applicable, all chapters have a brief introduction, an architectural/application description, register definitions, and timing diagrams. The last section, Electrical Specifications, has no chapters associated with it and presents the PSoC device's electrical specifications. The Revision History section chronologically lists the document's history.

Document Conventions

Register Conventions

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

Convention	Example	Description
'x' in a register name	ACBxxCR1	Multiple instances/address ranges of the same register.
RW	RW:00	Read and write register or bit(s)
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
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'). 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 document.

Symbol	Units of Measure
°C	degree Celsius
AC	alternating current
dB	decibels
DC	direct current
fF	femto Farad
Hz	hertz
k	kilo, 1000
K	2^{10}, 1024
KB	1024 bytes
Kbit	1024 bits
kHz	kilohertz
kΩ	kilohm
MHz	megahertz
MΩ	megaohm
µA	microampere
µs	microsecond
µV	microvolts
µVrms	microvolts root-mean-square
mA	milliampere
ms	millisecond
mV	millivolts
nA	nanoamphere
ns	nanosecond
nV	nanovolts
Ω	ohm
pF	pico Farad
pp	peak-to-peak
ppm	parts per million
sps	samples per second
σ	sigma: one standard deviation
V	volts

Acronyms Used

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

Acronym	Description
AC	alternating current
AI	analog input
API	application programming interface
APOR	analog power on reset
BC	broadcast clock
CMRR	common mode rejection ratio
CPU	central processing unit
CRC	cyclic redundancy check
CT	continuous time
DAC	digital-to-analog converter
DC	direct current
DNL	differential nonlinearity
DO	digital or data output
ECO	external crystal oscillator
EEPROM	electrically erasable programmable read-only memory
FB	feedback
FSR	full scale range
GIE	global interrupt enable
GPIO	general purpose IO
ICE	in-circuit emulator
IDE	integrated development environment
ILO	internal low speed oscillator
INL	integral nonlinearity
IO	input/output
IOW	IO write
IPOR	imprecise power on reset
IRA	interrupt request acknowledge
IRQ	interrupt request
ISR	interrupt service routine
ISSP	in-circuit system serial programming
IVR	interrupt vector read
LFSR	linear feedback shift register
LPF	low pass filter
LSB	least-significant bit
LUT	lookup table
MISO	master-in-slave-out
MOSI	master-out-slave-in
MSB	most-significant bit
PC	program counter
PD	power down
PDDSC	power system sleep duty cycle
PGA	programmable gain amplifier
POR	power on reset
PPOR	precision power on reset
PRS	pseudo random sequence
PSoC™	Programmable System-on-Chip
PSRR	power supply rejection ratio
PVT	process voltage temperature
PWM	pulse width modulator
RAM	random access memory
RAS	ROM access strobe
RETI	return from interrupt
RI	row input
RO	row output
ROM	read only memory
SAR	successive approximation register
SC	switched capacitor
SNR	signal-to-noise ratio
SOI	start of instruction
SP	stack pointer
SPD	sequential phase detector
SPI	serial peripheral interconnect
TC	terminal count
VCO	voltage controlled oscillator
WDT	watchdog timer
WDR	watchdog reset

1. Pin Information

This chapter lists, describes, and illustrates the PSoC device pins and pinouts. Table 1-1 presents a summary of the device pins, and the following tables and illustrations detail a representation of the device's pinouts.

1.1 Pin Summary

Table 1-1. PSoC Device Pin Descriptions

Pin Name	Description	Input/Output
Vdd	Supply Voltage	Power
Vss	Ground	Power
XRES	External Reset (Active High)	Input
P0[0] – P0[4]	Port 0[0], 0[1], 0[2], 0[3], 0[4], Analog Input	Input/Output
P0[5]	Port 0[5], Analog Input/Output	Input/Output
P0[6] – P0[7]	Port 0[6], 0[7], Analog Input	Input/Output
P1[0]	Port 1[0], XTALOut/SDATA / I^2C SDA	Input/Output
P1[1]	Port 1[1], XTALIn/SCLK / I^2C SCL	Input/Output
P1[2]	Port 1[2]	Input/Output
P1[3]	Port 1[3]	Input/Output
P1[4]	Port 1[4], EXTCLK	Input/Output
P1[5]	Port 1[5], I^2C SDA	Input/Output
P1[6]	Port 1[6]	Input/Output
P1[7]	Port 1[7], I^2C SCL	Input/Output

1.2 Pinouts

The PSoC devices are available in a variety of packages. Refer to the following information for details on individual devices. Note that every port pin (labeled with a “P”), except for Vss, Vdd, and XRES in the following tables and illustrations, is capable of Digital IO.

Table 1-2. 8-Pin Part Pinout (PDIP, SOIC)

Pin No.	Description
1	P0[5], A in, out
2	P0[3], A in
3	P1[1], XTALin, I^2C SCL

Pin No.	Description
4	Vss
5	P1[0], XTALout, I^2C SDA
6	P0[2], A in

Pin No.	Description
7	P0[4], A in
8	Vdd

LEGEND A: analog, D: digital, IO: input or output.

Table 1-3. 20-Pin Part Pinout (PDIP, SSOP, SOIC)

Pin No.	Description
1	P0[7], A in
2	P0[5], A in, out
3	P0[3], A in
4	P0[1], A in
5	Vss
6	P1[7], I^2C SCL
7	P1[5], I^2C SDA

Pin No.	Description
8	P1[3]
9	P1[1], XTALin, I^2C SCL
10	Vss
11	P1[0], XTALout, I^2C SDA
12	P1[2]
13	P1[4], EXTCLK
14	P1[6]

Pin No.	Description
15	XRES
16	P0[0], A in
17	P0[2], A in
18	P0[4], A in
19	P0[6], A in
20	Vdd

LEGEND A: analog, D: digital, IO: input or output.

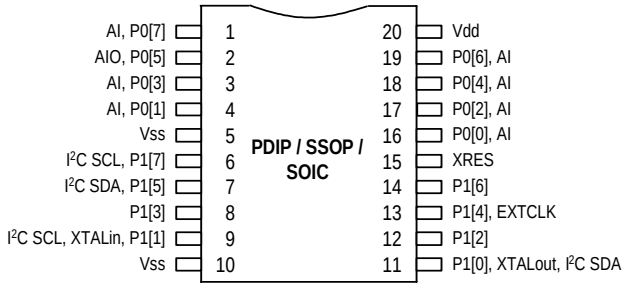


Table 1-4. 32-Pin Part Pinout (MLF)

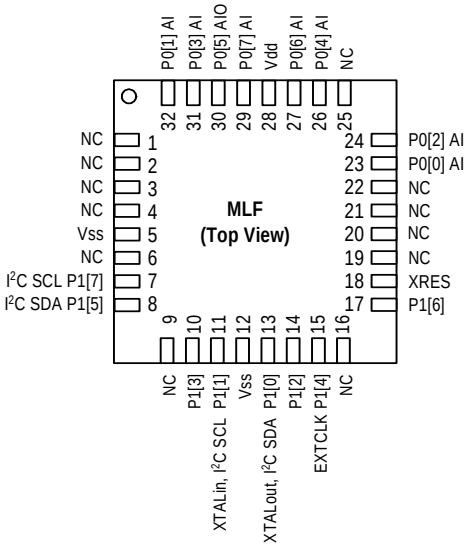
Pin No.	Description
1	NC
2	NC
3	NC
4	NC
5	Vss
6	NC
7	P1[7], I^2C SCL
8	P1[5], I^2C SDA
9	NC
10	P1[3]
11	P1[1], XTALin, I^2C SCL

Pin No.	Description
12	Vss
13	P1[0], XTALout, I^2C SDA
14	P1[2]
15	P1[4], EXTCLK
16	NC
17	P1[6]
18	XRES
19	NC
20	NC
21	NC
22	NC

Pin No.	Description
23	P0[0], A in
24	P0[2], A in
25	NC
26	P0[4], A in
27	P0[6], A in
28	Vdd
29	P0[7], A in
30	P0[5], A in, out
31	P0[3], A in
32	P0[1], A in

LEGEND A: analog, D: digital, IO: input or output, NC: no connection.

Note The MLF package has a center pad that must be connected to the same ground as the Vss pin.



2. Packaging Information

This chapter presents and illustrates the packaging specifications for the PSoC device, along with the thermal impedances for each package.

2.1 Packaging Dimensions

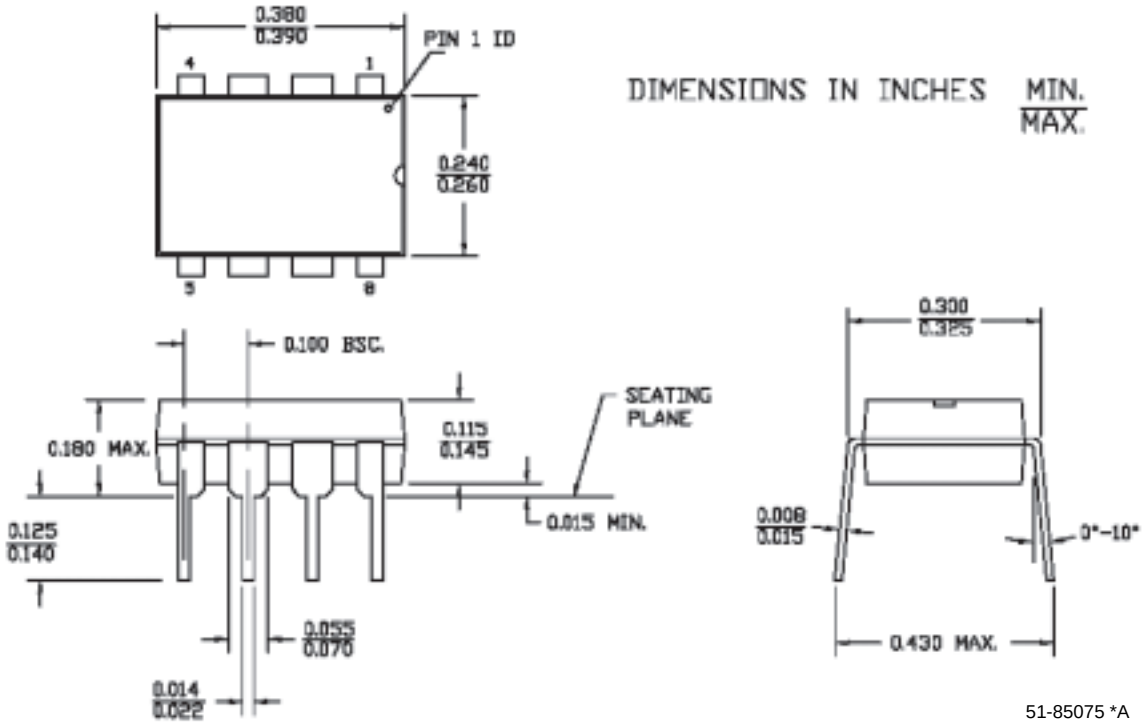


Figure 2-1. 8-Lead (300-Mil) PDIP

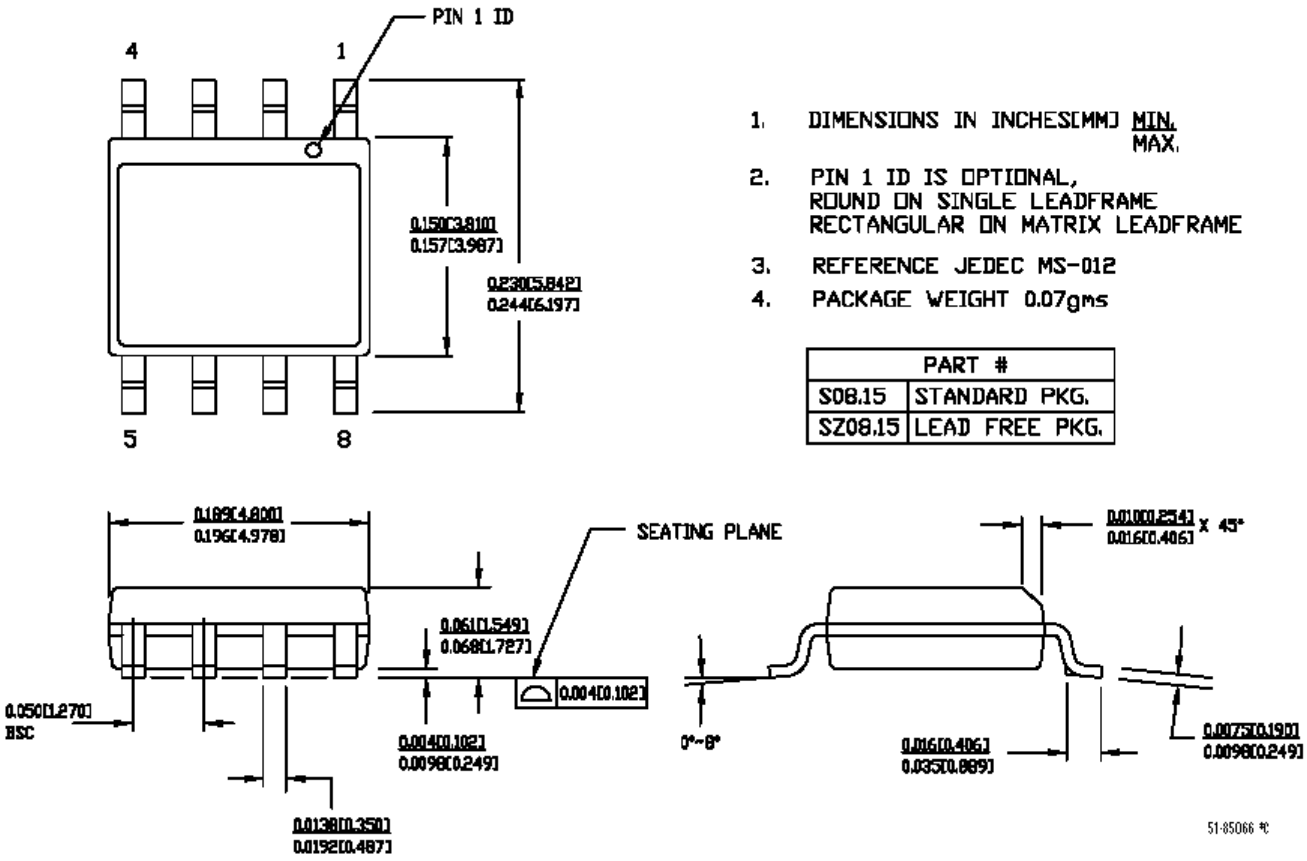


Figure 2-2. 8-Lead (150-Mil) SOIC

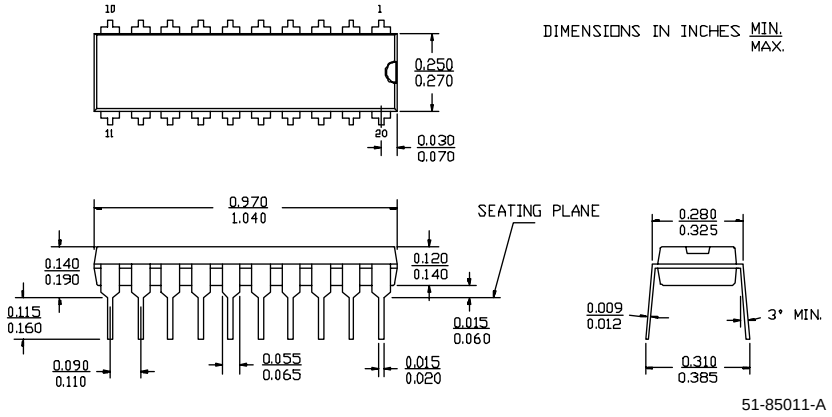


Figure 2-3. 20-Lead (300-Mil) Molded DIP

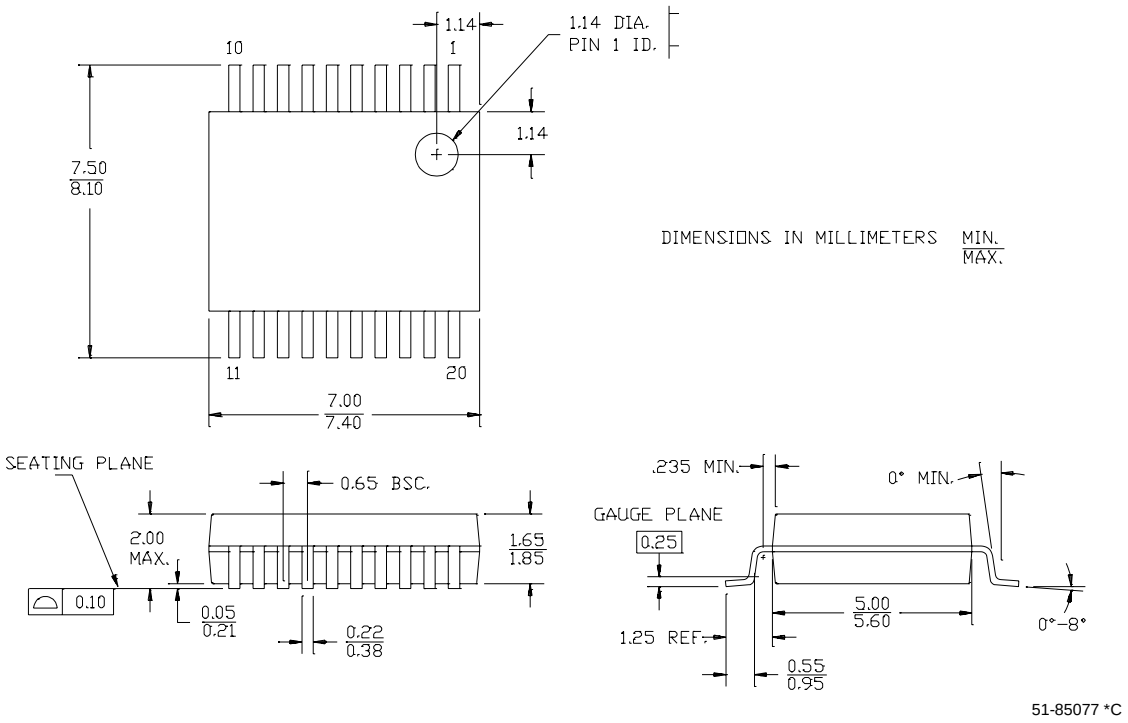


Figure 2-4. 20-Lead (210-Mil) SSOP

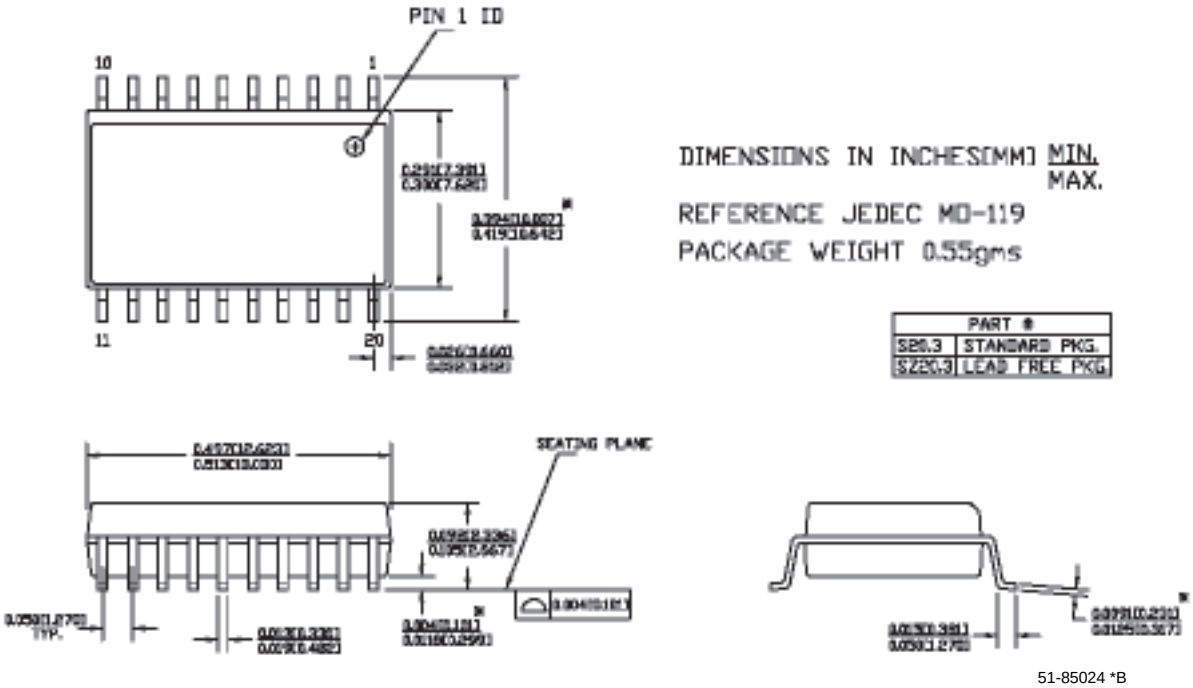


Figure 2-5. 20-Lead (300-Mil) Molded SOIC

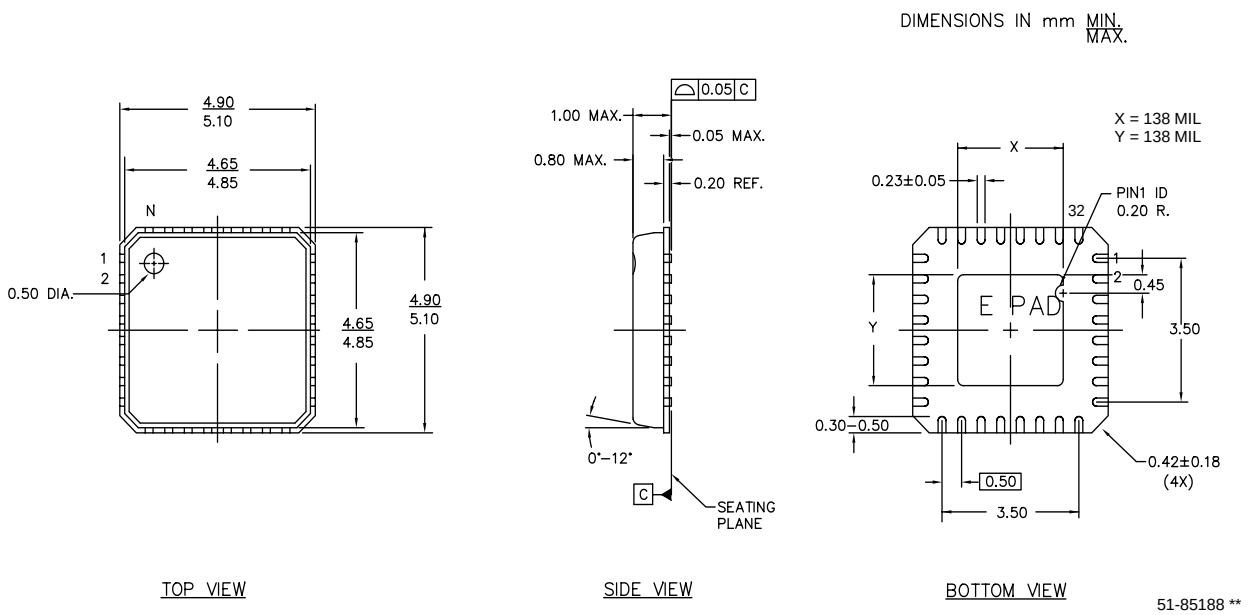


Figure 2-6. 32-Lead (5x5 mm) MLF

2.2 Thermal Impedances

Table 2-1. Thermal Impedances per Package

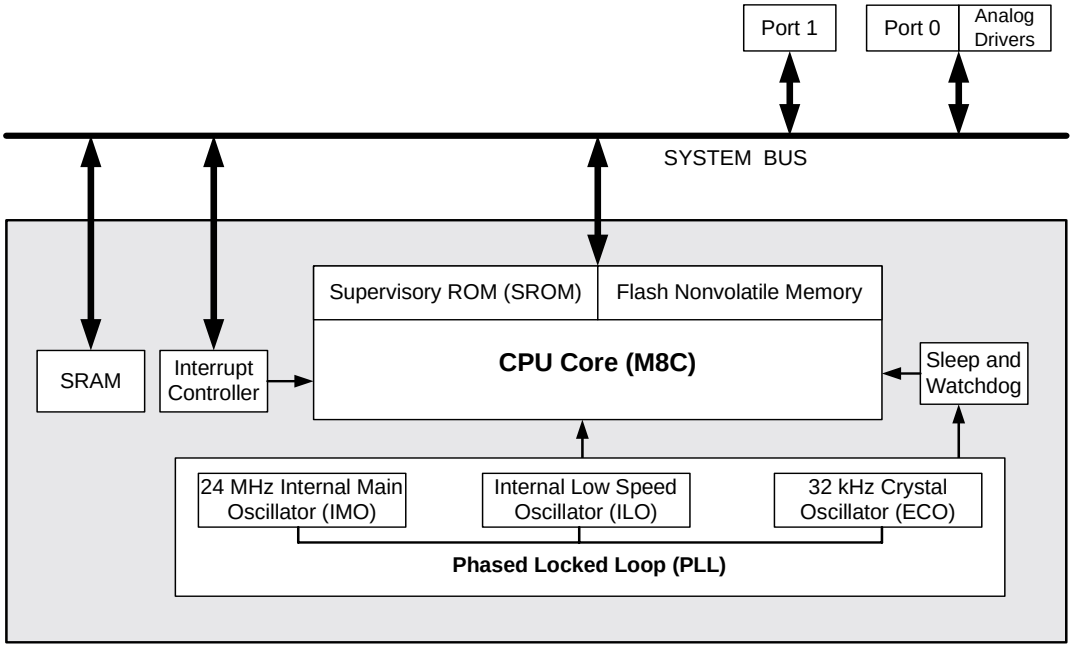
Package	Typical Θ_{JA}
8 PDIP	123 °C/W
8 SOIC	185°C/W
20 PDIP	109 °C/W
20 SSOP	117 °C/W
20 SOIC	81 °C/W
32 MLF	22 °C/W

SECTION B CORE ARCHITECTURE

The Architecture section discusses the core components of the PSoC device 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

Core Register Summary

The table below lists all the PSoC registers that the core of the device uses.

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 REGISTERS										
M8C Register										
x,F7h	CPU_F				XOI		Carry	Zero	GIE	RL : 00
Related Registers										
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00
x,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	RW : 17
SUPERVISORY ROM (SROM) REGISTER										
x,FEh	CPU_SCR1								IRAMDIS	RW : 00
INTERRUPT CONTROLLER REGISTERS										
0,DAh	INT_CLR0	VC3	Sleep	GPIO			Analog 1		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			Analog 1		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				XOI		Carry	Zero	GIE	RL : 00
GENERAL PURPOSE IO (GPIO) REGISTERS										
0,00h	PRT0DR	Data Input[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
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 Input[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
ANALOG OUTPUT DRIVER REGISTER										
1,62h	ABF_CR0	ACol1Mux		ABUF1EN0				Bypass	PWR	RW : 00
INTERNAL MAIN OSCILLATOR (IMO) REGISTER										
1,E8h	IMO_TR	Trim[7:0]								W : 00
INTERNAL LOW SPEED OSCILLATOR (ILO) REGISTER										
1,E9h	ILO_TR			Bias Trim[1:0]		Freq Trim[3:0]				W : 00
32 kHz CRYSTAL OSCILLATOR (ECO) REGISTER										
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
x,FEh	CPU_SCR1								IRAMDIS	RW : 00
PHASE LOCKED LOOP (PLL) REGISTERS										
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	IMODIS	SYSCLKX2 DIS	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
SLEEP AND WATCHDOG REGISTERS										
0,E0h	INT_MSK0	VC3	Sleep	GPIO			Analog 1		V Monitor	RW : 00
0,E3h	RES_WDT	WDSL_Clear								W : 00
x,FEh	CPU_SCR1					ECO EXW	ECO EX		IRAMDIS	RW : 00
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
x,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	W : XX

LEGEND
L: The AND, OR, and XOR flag instructions can be used to modify this register.
#: Access is bit specific. Refer to register detail 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.

3. CPU Core (M8C)

This chapter explains the CPU Core, called M8C, and its associated registers. It covers the internal M8C registers, address spaces, instruction formats, and addressing modes. For additional information concerning the M8C instruction set, reference the *Assembly Language User Guide* available at the Cypress.com web site.

Table 3-1. M8C Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
M8C Register										
x,F7h	CPU_F				XOI		Carry	Zero	GIE	RL : 00
Related Registers										
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00
x,FF	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	RW : 17

LEGEND
L: The AND, OR, and XOR 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 accessed or written to no matter what bank is used.

The M8C is a four MIPS 8-bit Harvard architecture microprocessor. Code 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.

3.1 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) – internal use only
- 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 will wrap 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 instructions such as:

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

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

3.2 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. Figure 3-1 illustrates the arrangement of the PSoC microcontroller address spaces.

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.

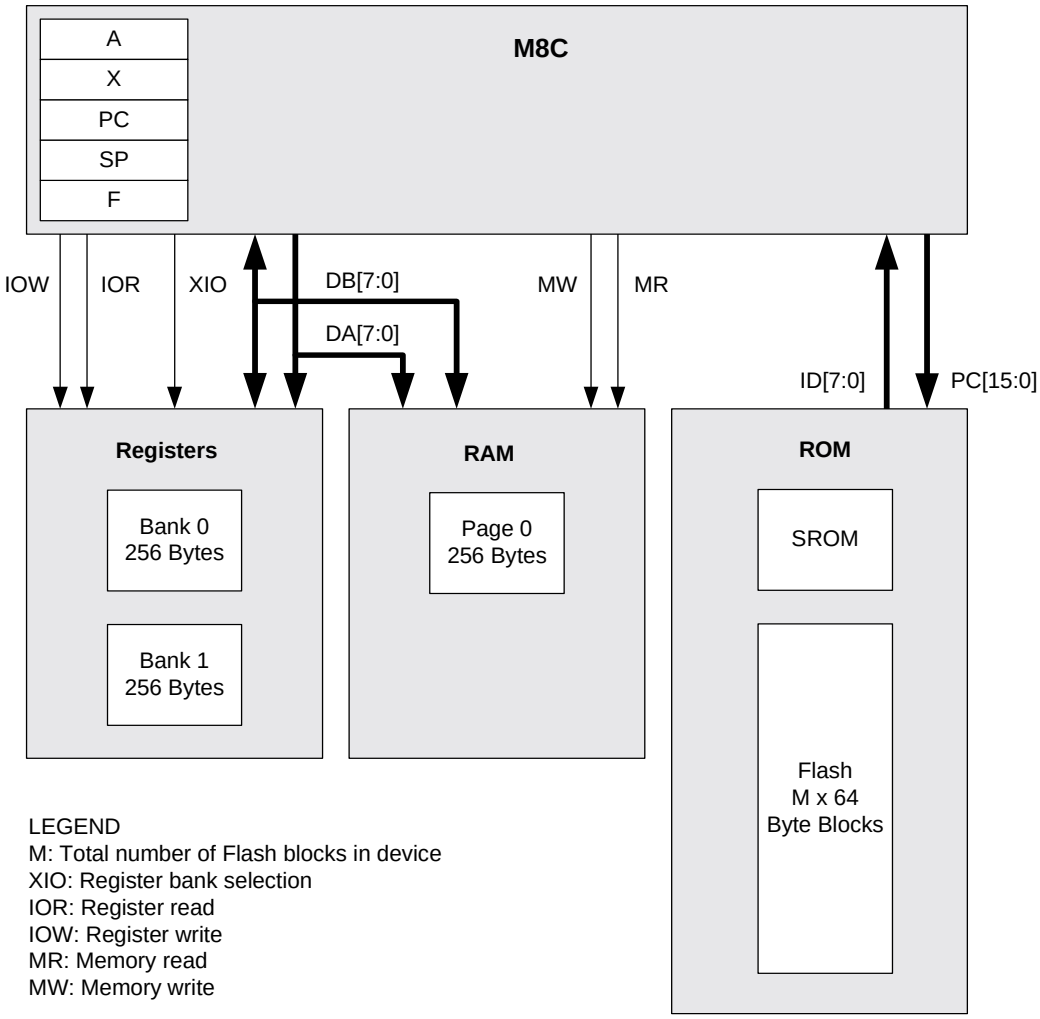


Figure 3-1. M8C Microcontroller Address Spaces

3.3 Instruction Set Summary

The instruction set is summarized below in Table 3-2. It is described in detail in the *PSoC Designer Assembly Language User Guide* (reference the Cypress.com web site).

Table 3-2. Instruction Set Summary

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

Note 1 Interrupt routines take 13 cycles before execution resumes at Interrupt Vector table.

Note 2 The number of cycles required by an instruction is increased by 1 for instructions that span 256 byte boundaries in the Flash memory space.

3.4 Instruction Format

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 will be given in this section, refer to the PSoC Designer Assembly Language User Guide for detailed information on individual instructions.

3.4.1 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 3-3, 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 3-3. 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 will cause SROM code to run which will modify RAM and 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 single-byte instruction that causes a user register to be modified. The HALT instruction modifies user register space address FFh (CPU_SCR).

The final category for single-byte instructions are those that cause internal M8C registers to be updated. 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 be updated.

3.4.2 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 3-4. 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 Table 3-4 is used by short jumps and calls: CALL, JMP, JACC, INDEX, JC, JNC, JNZ, JZ. This instruction format uses only 4-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 (Table 3-4) is used by instructions that employ the Source Immediate addressing mode ("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 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.

3.4.3 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 3-5. 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 Table 3-5 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 Table 3-5 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 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 source address in RAM. The following is an example of this instruction: `MOV [7], [5]`.

3.5 Addressing Modes

The M8C has ten addressing modes:

- Source Immediate
- Source Direct
- Source Indexed
- Destination Direct
- Destination Indexed
- Destination Direct Source Immediate
- Destination Indexed Source Immediate
- Destination Direct Source Direct
- Source Indirect Post Increment
- Destination Indirect Post Increment

3.5.1 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 3-6. 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 into the X register.
`AND F, 9`	`70 09`	The immediate value of 9 is logically AND'ed with the F register and the result is placed in the F register.

3.5.2 Source Direct

For these instructions the source address is stored in operand 1 of the instruction. During instruction execution the address will be 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 3-7. 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 into the Accumulator.
MOV	A, REG[8]	5D 08	The value in the register space at address 8 is moved into the Accumulator.

3.5.3 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 3-8. 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 into the X register.

3.5.4 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 3-9. 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.

3.5.5 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 3-10. 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.

3.5.6 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 3-11. 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 into register space at address 8.

3.5.7 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 3-12. 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 into the register space at address X+8.

3.5.8 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. All instructions using the Destination Direct Source Direct addressing mode are three bytes in length.

Table 3-13. 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.

3.5.9 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 will be 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 (DPR_DR) register is used to determine which RAM page to use with the source address. Therefore, values from pages other than the current page may be retrieved without changing the Current Page Pointer (CPP_DR). The pointer is always read from the current RAM page. For information on the DPR_DR and CPP_DR registers, see the device data sheet.

Table 3-14. 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 into the Accumulator. The indirect address, at address 8 in memory, is then incremented.

3.5.10 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 will be 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 (DPW_DR) register is used to determine which RAM page to use with the destination address. Therefore, values may be stored in pages other than the current page without changing the Current Page Pointer (CPP_DR). The pointer is always read from the current RAM page. For information on the DPR_DR and CPP_DR registers, see the device data sheet.

Table 3-15. 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 into the memory location pointed to by the indirect address. The indirect address in memory, at address 8, is then incremented.

3.6 Register Definitions

3.6.1 CPU_F (Flag) Register

The Flag register has four chip dependent bits (FL[7:4]) and four dedicated bits (FL[3:0]), as shown in Table 3-1.

3.6.1.1 Chip-Dependent Flag Bits

The chip-dependent Flag bits have no effect internally on the M8C. These bits are manipulated by the user with the Flag-Logic opcodes (for example, XOR F, 80h). Bit Definitions for the PSoC Mixed Signal Array family are as follows.

Bits 7, 6, and 5: Reserved.

Bit 4: XOI. IO Bank Select. This bit is used to select between register banks, in order to support more than 256 registers.

3.6.1.2 Dedicated Flag Bits

The dedicated Flag bits are described as follows.

Bit 3: Reserved.

Bit 2: Carry. Carry Flag. This bit is set or cleared in response to the result of several instructions. It may 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. Zero Flag. This bit is set or cleared in response to the result of several instructions. It may 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. Global Interrupt Enable. The state of this bit determines whether interrupts (by way of the IRQ) will be recognized by the M8C. This bit is set or cleared by the user, using the Flag-Logic opcodes (e.g., OR F, 1). GIE is also cleared automatically by the interrupt routine, after the flag byte has been stored on the stack.

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

For additional information, reference the CPU_F register on page 142.

4. Supervisory ROM (SROM)

This chapter discusses the Supervisory ROM (SROM) and its associated register. It covers both the physical SROM block and the code stored in the SROM for the PSoC devices.

Table 4-1. SROM Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,FEh	CPU_SCR1								IRAMDIS	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.

The SROM holds code that is used to boot the part, calibrate circuitry, and perform Flash operations. The functions of the SROM may be accessed in normal user code, operating from Flash.

4.1 Architectural Description

The SROM is used to boot the part and provide interface functions to the Flash macros. (Table 4-2 lists the SROM functions.) The SROM functions are accessed by executing the Supervisory System Call instruction (SSC) which has an opcode of 00h. Prior to executing the SSC the M8C’s accumulator needs to be loaded with the desired SROM function code from Table 4-2. Undefined functions will cause a HALT if called from user code. The SROM functions are executing 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 must be configured before executing the SSC. Table 4-3 lists all possible parameter block variables. The meaning of each parameter, with regards to a specific SROM function, is described later in this chapter.

Table 4-2. List of SROM Functions

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

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 when the SROM function begins execution. This would be the SP value when the SSC opcode is executed, plus three. If either of the keys do not match the expected values, the M8C will halt (with the exception of the SWBootReset function). The following code puts the correct value in KEY1 and KEY2. The code starts with a halt, to force the program to jump directly into the setup code and not run into it.

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

Table 4-3. SROM Function Variables

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

4.1.1 Additional SROM Feature

The SROM has the following additional SROM 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 4-4. SROM Return Code Meanings

Return Code Value	Description
00h	Success
01h	Function not allowed due to level of protection on 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.

4.1.2 SROM Function Descriptions

4.1.2.1 SWBootReset Function

The SROM function SWBootReset is the function that is responsible for transitioning the device from a reset state to running user code. See the Types of Resets chapter for more information on what events will 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 will happen, 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.

The SWBootReset's calibration function, Calibrate1, transfers the calibration data one byte at a time from Flash to SRAM. As the bytes are transferred, the sum of the bytes, plus a hard coded offset value of EBh, is calculated in a 2-byte SRAM variable (CHECKSUM). If at the end of the transfer the value of CHECKSUM (plus the offset value of EBh) is zero, the SWBootReset function uses the values stored in SRAM to calibrate the registers in the PSoC device. If CHECKSUM has a non-zero value, the IRES bit in CPU_SCR1 is set, which causes a hardware reset similar to a POR event. For more information on this condition, see "System Resets" on page 281.

If the checksum of the calibration data is zero, the SWBootReset function ends by setting the M8C registers (CPU_SP, CPU_PC, CPU_X, CPU_F, CPU_A) to 00h, after writing 00h to most SRAM addresses, and then begins to execute user code at address 0000h.

Table 4-5 documents the value of all the SRAM addresses in page zero, after a successful SWBootReset. A cell in the table with "xx" in it, 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 in CPU_SCR1. If IRAMDIS is not set, these addresses will be initialized to 00h. If IRAMDIS is set, these addresses will not be modified by a SWBootReset. The IRAMDIS bit allows variables to be preserved even if a watchdog reset 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 4-5. SRAM Map Post SWBootReset

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, for this function, the only acceptable values are 00h and 02h. Address FCh is the fail count variable. After POR, WDR, or XRES, 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.

4.1.2.2 Read Block Function

The ReadBlock function is used to read 64 contiguous bytes from Flash: a block. The number of blocks in a device is simply the total number of bytes divided by 64. For the CY8C22xxx, the Flash contains 32 blocks of 64 bytes.

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

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

Table 4-6. ReadBlock Parameters (01h)

Name	Address	Description
KEY1	0,F8h	3Ah
KEY2	0,F9h	Stack Pointer value, when SSC is executed.
BLOCKID	0,FAh	Flash block number
POINTER	0,FBh	First of 64 addresses in SRAM where returned data should be stored.

4.1.2.3 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.

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 will exit setting the accumulator and KEY2 back to 00h. KEY1 will have a value of 01h, indicating a write failure.

The configuration of the WriteBlock function is straight forward. The BLOCKID of the Flash block, where the data is stored, must be determined and stored at SRAM address FAh. Valid BLOCKID values are between 00h and.

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 will be 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 49 for additional information.

Table 4-7. WriteBlock Parameters (02h)

Name	Address	Description
KEY1	0,F8h	3Ah
KEY2	0,F9h	Stack Pointer value, when SSC is executed.
BLOCKID	0,FAh	Flash block number (00h – 3Fh).
POINTER	0,FBh	First of 64 addresses in SRAM, where the data to be stored in Flash is located prior to calling WriteBlock.
CLOCK	0,FCh	Clock divider used to set the write pulse width.
DELAY	0,FEh	For a CPU speed of 12 MHz set to 56h.

4.1.2.4 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 will exit setting the accumulator and KEY2 back to 00h. KEY1 will have 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 49.

Table 4-8. EraseBlock Parameters (03h)

Name	Address	Description
KEY1	0,F8h	3Ah
KEY2	0,F9h	Stack Pointer value, when SSC is executed.
BLOCKID	0,FAh	Flash block number (00h – 3Fh).
CLOCK	0,FCh	Clock divider used to set the erase pulse width.
DELAY	0,FEh	For a CPU speed of 12 MHz set to 56h.

4.1.2.5 TableRead Function

The TableRead function gives the user access to part-specific data stored in the Flash during manufacturing. It also returns a Revision ID for the die (not to be confused with the Silicon ID stored in Table 0).

Table 4-9. TableRead Parameters (06h)

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

Table 4-10. Table with Assigned Values in Flash Macro 0

	F8h	F9h	FAh	FBh	FCh	FDh	FEh	FFh
Table 0	Silicon ID		(May be used for serialization in the future.)					
Table 1	Voltage Reference trim for 3.3 V reg[1,EA]	Main Oscillator trim for 3.3 V reg[1,E8]	Room Temperature Calibration for 3.3V	Hot Temperature Calibration for 3.3V	Voltage Reference trim for 5 V reg[1,EA]	Main Oscillator trim for 5 V reg[1,E8]	Room Temperature Calibration for 5V	Hot Temperature Calibration for 5V
Table 2								
Table 3	M	B	Mult	M	B	Mult	00h	01h

4.1.2.6 Checksum Function

The Checksum function calculates a 16-bit checksum over a user specifiable number of blocks, within a single Flash macro (Bank) starting from block zero. The BLOCKID parameter is used to pass in the number of blocks to calculate the checksum over. A BLOCKID value of 1 will calculate the checksum of only block 0, while a BLOCKID value of 0 will calculate the checksum of all 256 user blocks.

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.

The checksum algorithm executes the following sequence of three instructions over the number of blocks times 64 to be checksumed.

```
romx
add [KEY1], A
adc [KEY2], 0
```

Table 4-11. Checksum Parameters (07h)

Name	Address	Description
KEY1	0,F8h	3Ah
KEY2	0,F9h	Stack Pointer value, when SSC is executed.
BLOCKID	0,FAh	Number of Flash blocks to calculate checksum on.

4.1.2.7 Calibrate0 Function

The Calibrate0 function transfers the calibration values stored in a special area of the Flash to their appropriate registers.

Table 4-12. Calibrate0 Parameters (08h)

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

4.1.2.8 Calibrate1 Function

While Calibrate1 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 will cause a hardware reset by setting the IRES bit of CPU_SCR1.

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 will be set to 00h.

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

Table 4-13. Calibrate1 Parameters (09h)

Name	Address	Description
KEY1	0,F8h	3Ah
KEY2	0,F9h	Stack Pointer value, when SSC is executed.
POINTER	0,FBh	First of 32 SRAM addresses used by this function.

4.2 Register Definitions

4.2.1 CPU_SCR1 Register

The CPU_SCR1 register is used to convey status and control of events related to internal resets and watchdog reset.

Bits 7 to 1: Reserved.

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 reset to a value of 00h. When the bit is set, 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, reference the CPU_SCR1 register on page 143.

4.3 Clocking

The values M, B, and Mult are found in Flash Table 3. The user must supply a value for T, which is the ambient temperate in degrees Celsius. The calculated value $CLOCK_W$ is used for write operations, while $CLOCK_E$ is used for erase operations.

$$CLOCK_W = \frac{CLOCK_E \times Mult}{64}$$ **Equation 1**

Equation 2

$$CLOCK_E = \frac{CPU}{5648} \cdot \left[\frac{B}{12 \times 10^6} - \frac{M \times T}{1536 \times 10^6} \right] - 89$$

Equation 2 is valid for CPU speeds from 3 MHz up to 12 Mhz. The clock and delay parameters support Flash operations.

The other clocking related parameter is “DELAY.” For 12 MHz operation, the value is 56h. For other CPU speeds, the following equation may be used.

$$DELAY = \frac{(100 \times 10^{-6}) \times CPU - 84}{13}$$ **Equation 3**

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 Mixed Signal Array devices to change program execution to a new address, without regard to the current task being performed by the code being executed.

Table 5-1. Interrupt Controller 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			Analog 1		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			Analog 1		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				XOI		Carry	Zero	GIE	RL : 00

LEGEND
L: The AND, OR, and XOR flag instructions can be used to modify this register.
C: Cearable register or bits.
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.

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 may 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 all interrupts and the priorities that are available in the PSoC device.

Table 5-2. CY8C22xxx Interrupt Table

Interrupt Priority	Interrupt Address	Interrupt Name
0 (highest)	0000h	Reset
1	0004h	Supply voltage monitor
3	000Ch	Analog column 1
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

5.1 Architectural Description

A block diagram of the PSoC Interrupt Controller is shown in Figure 5-1. It illustrates the notion of posted and pending interrupts.

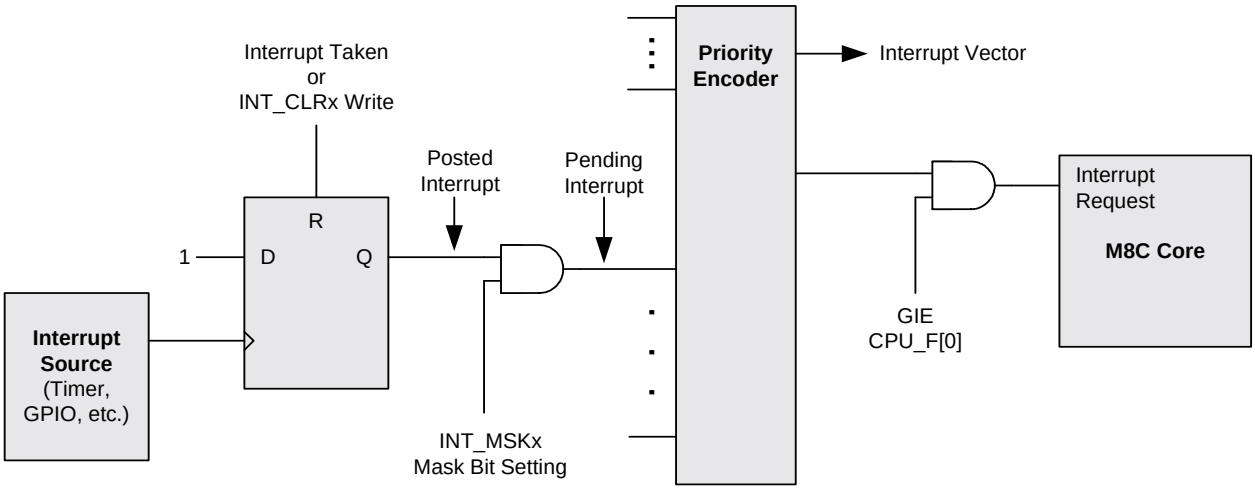


Figure 5-1. Interrupt Controller Block Diagram

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

1. An interrupt becomes active, either because (a) the interrupt condition occurs (e.g., 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 placed into PCL (PC[7:0]). This sets the program counter to point to the appropriate address in the interrupt table (e.g., 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.
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 will be 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 from the following equation.

Latency =
Time for current instruction to finish +
Time for internal interrupt routine to execute +
Time for LJMP instruction in interrupt table to execute.

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

(1 to 5 cycles for JMP to finish) +
(13 cycles for interrupt routine) +
(7 cycles for LJMP) = 21 to 25 cycles.

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

5.2 Register Definitions

Table 5-1 gives an overview of all registers related to interrupt controller operation. The following text presents details on the use of each register.

5.2.1 INT_CLRx Register

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. However, 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 way an individual bit value written to an INT_CLRx register is interpreted is determined by the Enable Software Interrupt (ENSWINT) bit in INT_MSK3[7]. 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, will cause the corresponding interrupt to be cleared. If the ENSWINT bit is set, any 0's written to the INT_CLRx registers will be ignored. However, 1's written to an INT_CLRx register, while ENSWINT is set, will cause an interrupt to be posted for the corresponding interrupt. Enabling software interrupts allows a user's code to create software interrupts that can aid in debugging interrupt service routines, by eliminating the need to create the system level interactions that may be necessary to create a hardware interrupt.

For additional information, reference the INT_CLR0 register on page 129, the INT_CLR1 register on page 131, and the INT_CLR3 register on page 132.

5.2.2 INT_MSKx Register

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 an interrupt from becoming a pending interrupt (input to the priority encoder). However, an interrupt may 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 will become 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 will allow 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 will be pending and not the GPIO interrupt. INT_MSK3[7] (ENSWINT) 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.

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

For additional information, reference the INT_MSK0 register on page 134, the INT_MSK1 register on page 135, and the INT_MSK3 register on page 133.

5.2.3 INT_VC Register

The interrupt vector clear register (INT_VC) performs two different functions. When the register is read, the least significant byte of the highest priority pending interrupt is returned. For example, if the GPIO and I^2C interrupts were pending and the INT_VC register was read, the value 1Ch would be read. However, if no interrupt were pending, the value 00h would be returned. This is the reset vector in the interrupt table; however, reading 00h from the INT_VC register should not be considered to be 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 performs an Interrupt Vector Read (IVR). The clear of the highest priority pending interrupt occurs asynchronously.

Reading the INT_VC has limited usefulness. If interrupts are enabled, a read to the INT_VC register would not be able to 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 will be. 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, reference the INT_VC register on page 136.

5.2.4 CPU_F Register

Only the GIE bit in the CPU_F register is related to the interrupt controller. This bit is the Global Interrupt Enable. When this bit is set, the M8C will take a pending interrupt. When the GIE bit is cleared, the M8C will not take any interrupts. By default, this bit is cleared. To set or clear this bit, the *AND F, expr,* or *OR F, expr,* or *XOR F, expr* instructions must be used. (Written another way: *AND*/*OR*/*XOR F, expr* instructions must be used.) The GIE flag bit is covered in more detail in the chapter titled "CPU Core (M8C)" on page 35.

For additional information, reference the CPU_F register on page 142.

6. General Purpose IO (GPIO)

This chapter discusses the General Purpose IO (GPIO) and its associated registers. 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 operations for both digital and analog systems.

Table 6-1. GPIO Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	PRTxDR	Data Register								RW : 00
0,xxh	PRTxIE	Bit Interrupt Enables								RW : 00
0,xxh	PRTxGS	Global Select								RW : 00
0,xxh	PRTxDM2	Drive Mode 2								RW : FF
1,xxh	PRTxDM0	Drive Mode 0								RW : 00
1,xxh	PRTxDM1	Drive Mode 1								RW : FF
1,xxh	PRTxIC0	Interrupt Control 0								RW : 00
1,xxh	PRTxIC1	Interrupt Control 1								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 “Core Register Summary” on page 32.

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. Therefore, the registers shown in Table 6-1 are actually for a GPIO port (eight GPIO blocks), where the bit position indicates which of the eight GPIO bit blocks is controlled in the GPIO port.

Each GPIO block can be used for the following types of IO:

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

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

The main block diagram for the GPIO block is illustrated 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 chip and get information into the M8C from outside the chip; this is accomplished by way of the port data register (PRTxDR). Writes from the M8C to the PRTxDR 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 below). The actual voltage on the pin depends on the drive mode and the external load.

The M8C may read the value of a port by reading the PRTxDR. When the M8C reads the PRTxDR, the current value of the pin voltage is translated into a logic value and returned to the M8C. These operations read the pin voltage, not the data drive state stored in the local PRTxDR register bit latch.

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. Also, the drive mode for the GPIO must be set to the digital Hi-Z state. (Refer to "PRTxDMx Registers" on page 58 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-Hi-Z states.

6.1.3 Analog IO

Analog signals can pass between the chip core and chip pins through the block's AOUT pin. This provides a resistive path (~300 ohms) directly through the block. For analog modes, the GPIO block is typically configured into a High Impedance Analog Drive mode (Hi-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 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 use some user designed mechanism, to determine which pin was the source of the interrupt.

Using a GPIO interrupt requires the following steps:

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

These last two steps are common to all interrupts and are described in "Interrupt Controller" on page 51.

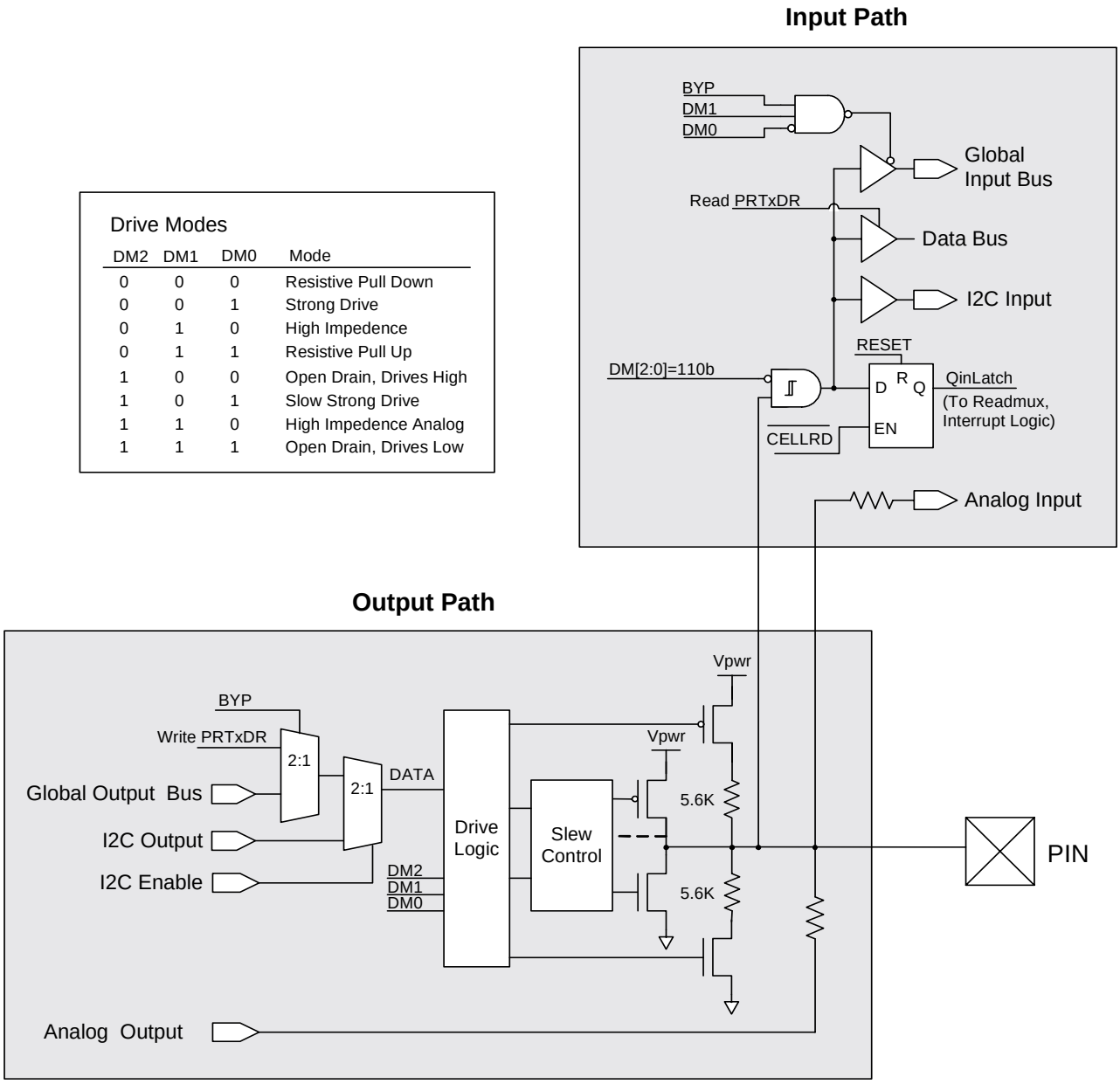
The first two steps, bit interrupt enable and interrupt mode, are set at the GPIO block level (i.e., at each port pin), by way of the block's configuration registers.

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 chip level, due to their wire-OR nature, the GPIO interrupts are neither true edge-sensitive interrupts nor true level-sensitive interrupts. They could be 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 will occur 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 will pull 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.) Note that 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 will continue to hold INTO low until one of these conditions changes: 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 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 will be seen, i.e., no new interrupt would be asserted on the GPIO interrupt. Care must be taken, using polling and/or the states of the GPIO pin and global Interrupt Enables, to catch all interrupts among a set of wire-OR GPIO blocks.



Drive Modes

DM2	DM1	DM0	Mode
0	0	0	Resistive Pull Down
0	0	1	Strong Drive
0	1	0	High Impedence
0	1	1	Resistive Pull Up
1	0	0	Open Drain, Drives High
1	0	1	Slow Strong Drive
1	1	0	High Impedence Analog
1	1	1	Open Drain, Drives Low

Figure 6-1. GPIO Block Diagram

The interrupt logic portion of the block is shown in Figure 6-2.

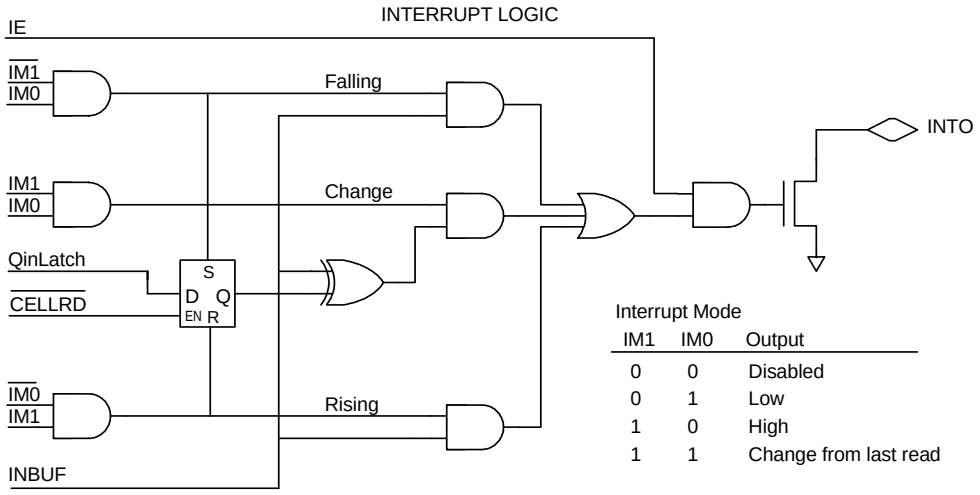


Figure 6-2. GPIO Interrupt Mode Block Diagram

6.2 Register Definitions

For a selected GPIO block, the individual registers are addressed as shown in Table 6-2. In the register names, the 'x' is the port number, configured at the chip level (x = 0 to 7 typically). DA[1:0] refers to the two LSB of the register address.

All register values are readable, except for the PRTxDR register; reads of this register return the pin state instead of the register bit state.

Table 6-2. Internal Register Bit Addressing

XOI	DA[1:0]	Register	Resets to:	(Name)	Function
0	00b	PRTxDR	0	DIN	Data
0	01b	PRTxIE	0	IE	Interrupt Enable
0	10b	PRTxGS	0	BYP	Global Select
0	11b	PRTxDM2	**1**	DM2	Drive Mode, Bit 2
1	00b	PRTxDM0	0	DM0	Drive Mode, Bit 0
1	01b	PRTxDM1	1	DM1	Drive Mode, BIt 1
1	10b	PRTxIC0	0	IM0	Intrpt. Mask, Bit 0
1	11b	PRTxIC1	0	IM1	Intrpt. Mask, Bit 1

6.2.1 PRTxDR Registers

Writing the PRTxDR register bit 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 PRTxDR 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.

For additional information, reference the PRTxDR register on page 86.

group. These are referred to as DM2, DM1, and DM0, or

6.2.2 PRTxIE Registers

The PRTxIE register is used to enable/disable the interrupt enable internal to the GPIO block. A '1' enables the INTO output at the block, a '0' disables INTO so it can only be Hi-Z.

For additional information, reference the PRTxIE register on page 87.

6.2.3 PRTxGS Registers

The PRTxGS register is used to select the block for connection to global inputs or outputs. Writing this register high enables the global bypass (BYP=1 in Figure 6-1). If the drive mode is set to digital Hi-Z (DM[2:0] = 010b), then the pin is selected for global input (PIN drives to the Global Input Bus). In non-Hi-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.

For additional information, reference the PRTxGS register on page 88.

6.2.4 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 (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 (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

together as DM[2:0]. Drive modes are shown in Table 6-3.

Table 6-3. Pin Drive Modes

Drive Mode DM[2:0]	Pin State	Description
000b	Resistive pull down	Strong high, resistive low
001b	Strong drive	Strong high, strong low
010b	High impedance	Hi-Z high and low, digital input enabled
011b	Resistive pull up	Resistive high, strong low
100b	Open drain high	Slow strong high, Hi-Z low
101b	Slow strong drive	Slow strong high, slow strong low
110b	High impedance, analog (**reset state**)	Hi-Z high and low, digital input disabled (for zero power) (**reset state**)
111b	Open drain low	Slow strong low, Hi-Z high

For analog IO, the drive mode should be set to one of the Hi-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 will always be read as a zero by the CPU and the pin will not be able to generate a useful interrupt. (It is not strictly required that a Hi-Z mode be selected for analog operation).

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

This GPIO provides a default drive mode of high impedance (Hi-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 I^2C mode 111b (although the slow edge rate is not slow enough to meet the I^2C fast mode specification).

For additional information, reference the PRTxDM2 register on page 89, the PRTxDM0 register on page 145, and the PRTxDM1 register on page 146.

6.2.5 PRTxICx Registers

The interrupt mode for the pin is determined by bits in two registers: PRTxIC1 and PRTxIC0. These are referred to as IM1 and IM0, or together as IM[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 (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 (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 6-4, in order to get an interrupt from the pin.

Table 6-4. GPIO Interrupt Modes

Interrupt Mode IM[1:0]	Description
00b	Bit interrupt disabled, INTO de-asserted
01b	Assert INTO when PIN = low
10b	Assert INTO when PIN = high
11b	Assert INTO when PIN = change from last read

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 will assert 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 will assert 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 will assert the interrupt line (INTO) when the pin voltage is the opposite of the last state read from the pin (again 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 that was read from the port during reads of the data register (PRTxDR). If the last value read from the GPIO was 0, the GPIO will subsequently be in interrupt high mode. If the last value read from the GPIO was 1, the GPIO will then be in interrupt low mode.

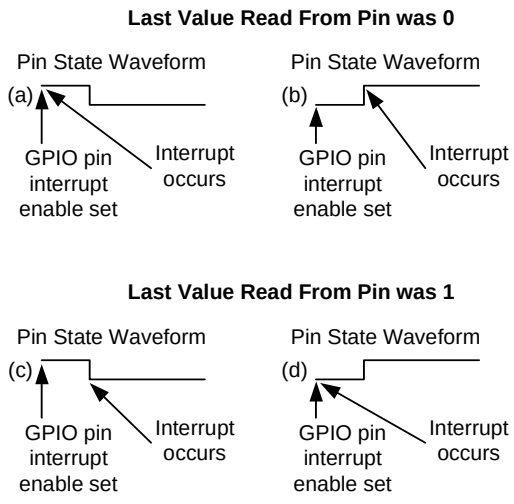


Figure 6-3. GPIO Interrupt Mode 11b

Figure 6-3 assumes that the GIE is set, GPIO interrupt mask is set, and that the GPIO interrupt mode has been 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 will act 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, reference the PRTxIC0 register on page 147 and the PRTxIC1 register on page 148.

7. Analog Output Drivers

This chapter presents the Analog Output Drivers and its associated register. The analog output drivers provide a means for driving analog signals off-chip.

Table 7-1. Analog Output Driver Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,62h	ABF_CR0	ACol1Mux		ABUF1EN0				Bypass	PWR	RW : 00

7.1 Architectural Description

The PSoC device has one analog driver used to output analog values. For a detailed drawing of the analog output drivers in relation to the analog system, reference the Analog Input Configuration chapter on page 235.

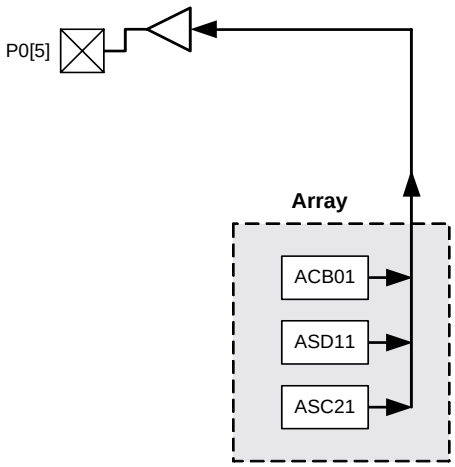


Figure 7-1. Analog Output Drivers

The PSoC device has one analog driver used to output analog values on port pins. This driver is a resource available to all the analog blocks in the column. The user can select 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 the column can be enabled and disabled using the Analog Output Driver register ABF_CR0.

7.2 Register Definitions

Table 7-1 presents an overview of all registers related to the analog output drivers. The following section presents a detail on the use of the register's bits.

7.2.1 ABF_CR0 Register

This register 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 6: Reserved.

Bit 5: ABUF1EN0. This bit enables or disables the column output amplifiers.

Bits 4, 3, and 2: Reserved.

Bit 1: Bypass. Bypass mode connects the amplifier input directly to the output. When this bit is set, all amplifiers controlled by the register will be in bypass mode.

Bit 0: PWR. This bit is used to set the power level of the amplifiers. When this bit is set, all amplifiers controlled by the register will be in a high power.

For additional information, reference the ABF_CR0 register on page 156.

8. Internal Main Oscillator (IMO)

This chapter briefly presents the Internal Main Oscillator (IMO) and its associated register. The IMO produces clock signals of 24 MHz and 48 MHz.

Table 8-1. Internal Main Oscillator 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

8.1 Architectural Description

The Internal Main Oscillator outputs two clocks: a SYSCLK, which can be the internal 24 MHz clock or an external clock, and a SYSCLK2X that is always twice the SYSCLK frequency. In the absence of a high-precision input source from the 32 kHz crystal oscillator, the accuracy of the internal 24 MHz/48 MHz clocks will be +/-2.5% over temperature variation and two voltage ranges (3.3V +/-.3V and 5.0V +/-5%). 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. The choice of crystal and its inherent accuracy will determine the overall accuracy of the oscillator. The External Crystal Oscillator must be stable prior to locking the frequency of the Internal Main Oscillator to this reference source.

The IMO can be disabled when using an external clocking source. Also, the frequency doubler circuit, which produces SYSCLK2X, can be disabled to save power. Note that when using an external clock, if SYSCLK2X is needed, then the IMO can not be disabled. Registers for controlling these operations are found in the Digital Clocks chapter on page 253.

8.2 Register Definitions

8.2.1 IMO_TR Register

The device specific value for 5 volt operation is loaded into the Internal Main Oscillator Trim Register (IMO_TR) at boot time. The Internal Main oscillator will operate within specified tolerance over a voltage range of 4.75V to 5.25V, with no modification of this register. If the device is operated at a lower voltage, user code must modify the contents of this register. For operation in the voltage range of 3.3V +/-.3V, this can be accomplished with a Table Read command to the Supervisor ROM, which will supply 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.

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

For additional information, reference the IMO_TR register on page 170.

9. Internal Low Speed Oscillator (ILO)

This chapter briefly explains the Internal Low Speed Oscillator (ILO) and its associated register. The Internal Low Speed Oscillator produces a 32 kHz clock.

Table 9-1. Internal Low Speed Oscillator 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

9.1 Architectural Description

The Internal Low Speed Oscillator is an internal low speed oscillator of nominally 32 kHz. It is available 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 and can be selected during non-sleep, but provides less frequency accuracy.

9.2 Register Definitions

9.2.1 ILO_TR Register

This register sets the adjustment for the ILO. 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 7 and 6: Reserved.

Bits 5 and 4: Bias Trim. 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 9-2.

Table 9-2. Bias Current in PTAT

Bias Current	Bit 5	Bit 4
Medium Bias	0	0
Maximum Bias	0	1
Minimum Bias	1	0
Not needed *	1	1

* About 15% higher than the minimum bias.

Bits 3 to 0: Freq Trim. Four bits are used to trim the frequency. Bit 0 is the LSB, Bit 3 is the MSB. Bit 3 gets inverted inside the register; therefore, a code of 8h turns all current sources off (f=0 kHz), a code of 0h turns only the MSB current source on (f=mid-scale), and a code of 7h turns on all the current sources (f=max).

For additional information, reference the ILO_TR register on page 171.

10. 32 kHz Crystal Oscillator (ECO)

This chapter briefly explains the 32 kHz Crystal Oscillator (ECO) and its associated register. The 32 kHz 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.

Table 10-1. Crystal Oscillator 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
1,EBh	ECO_TR	PSSDC[1:0]								W : 00
x,FEh	CPU_SCR1								IRAMDIS	RW:00

10.1 Architectural Description

The crystal oscillator circuit uses an inexpensive watch crystal and two small valued load capacitors as external components. All other components are on the PSoC chip. The crystal oscillator may be configured to provide a reference to the internal main oscillator in PLL mode for generating a more accurate 24 MHz system clock.

The XTALIn and XTALOut pins support connection of a 32.768 kHz watch crystal. To run from the external crystal, Bit 7 of the Oscillator Control 0 Register (OSC_CR0) must be set (default is off). The only external components are the crystal and the two load capacitors that connect to Vdd. 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 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 ILO will again be activated and the switch is made back to the ILO immediately.

The firmware steps involved in switching between the internal low speed oscillator to the 32 kHz Crystal Oscillator are as follows.

1. At reset, the chip begins operation, using the internal low speed oscillator.
2. Select sleep interval of 1 second using bits[4:3] in the Oscillator Control 0 Register (OSC_CR0), as the oscillator stabilization interval.
3. Enable the 32 kHz Crystal Oscillator, by setting bit [7] in Oscillator Control 0 Register (OSC_CR0) to 1.
4. The 32 kHz Crystal Oscillator 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 internal low speed oscillator continues to run, until the oscillator is automatically switched over by the sleep timer interrupt.
5. It is strongly advised to wait the one-second stabilization period prior to engaging the PLL mode to lock the Internal Main Oscillator frequency to the 32 kHz Crystal Oscillator frequency.

Note 1 The internal low speed oscillator switches back instantaneously by writing the 32K Select control bit to zero.

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 32K clock bus. Functions that require accuracy on the 32K clock should be enabled after the transition in oscillator domains.

10.1.1 ECO External Components

The External Crystal Oscillator component connections and selections are illustrated in Figure 10-1.

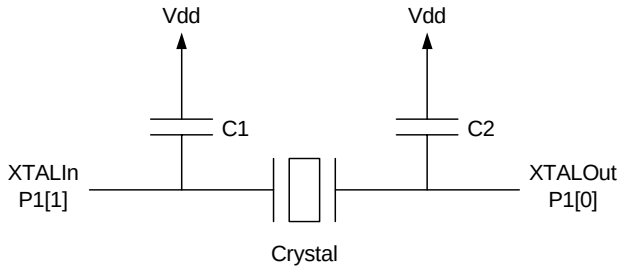


Figure 10-1. External Crystal Oscillator Connections

- Crystal – 32.768 kHz watch crystal such as Edson 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 Cap) - (Board Parasitic Cap)

If you do employ PLL with the External Crystal Oscillator, see *Application Note AN2027* under Support at http://www.Cypress.com/ for equation and details. An error of 1 pF in C1 and C2 gives about a 3 ppm error in frequency.

Table 10-2: Typical Package Capacitance on Crystal Pins

Package	Package Capacitance
8 PDIP	2.8 pF
8 SOIC	2.0 pF
20 PDIP	3.0 pF
20 SSOP	2.6 pF
20 SOIC	2.5 pF
32 MLF	2.0 pF

10.2 Register Definitions

10.2.1 OSC_CR0 Register

Bit 7: 32k Select. By default, the 32 kHz clock source is the Internal Low-Speed Oscillator (ILO). Optionally, the 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 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 chip systems are powered down, including the Band Gap reference. However, to facilitate the detection of POR and LVD events at a rate higher than the Sleep Interval, the Band Gap circuit is powered up periodically for about 60 us 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 Band Gap 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 10-3. It must be remembered that when the ILO is the selected 32 kHz clock source, sleep intervals are approximate.

Table 10-3. 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, 1 and 0: CPU Speed[2:0]. The PSoC M8C may operate over a range of CPU clock speeds (Table 10-4), 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.

The CPU frequency is changed with a write to the OSC_CR0 register. There are eight frequencies generated from a power-of-2 divide circuit, which are selected by a 3-bit code. At any given time, the CPU 8:1 clock multiplexer is selecting one of the available frequencies, which is re-synchronized 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 will be 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.

Table 10-4. 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, reference the OSC_CR0 register on page 165.

10.2.2 ECO_TR Register

The External Crystal Oscillator Trim register (ECO_TR) sets the adjustment for the 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 recommended that the user not alter the bits in this register.

Bits 7 and 6: PSSDC[1:0]. These bits are used to set the sleep duty cycle.

Bits 5 to 0: Reserved.

For additional information, reference the ECO_TR register on page 173.

10.2.3 CPU_SCR1 Register

The CPU_SCR1 register is used to convey status and control of events related to internal resets and watchdog reset.

Bits 7 to 1: Reserved.

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 reset to a value of 00h. When the bit is set, 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, reference the CPU_SCR1 register on page 143.

11. Phase Locked Loop (PLL)

This chapter briefly presents the Phase Locked Loop (PLL) and its associated registers.

Table 11-1. Phase Locked Loop Registers

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
1,E2h	OSC_CR2	PLLGAIN					EXTCLKEN	IMODIS	SYSCLKX2 DIS	RW : 00

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 will be less than 10 ms. If this bit is held high, the lock time will be 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 External Crystal Oscillator 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 the OSC_CR2 register bit 7.
- Set CPU to a faster frequency, if desired. To do this, write the bits CPU Speed[2:0] in the OSC_CR0 register. The CPU frequency will immediately change 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

11.2.1 OSC_CR0 Register

Bit 7: 32k Select. By default, the 32 kHz clock source is the Internal Low-Speed Oscillator (ILO). Optionally, the 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 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 chip systems are powered down, including the Band Gap reference. However, to facilitate the detection of POR and LVD events at a rate higher than the Sleep Interval, the Band Gap circuit is powered up periodically for about 60 us 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 Band Gap 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 11-2. It must be remembered that when the ILO is the selected 32 kHz clock source, sleep intervals are approximate.

Table 11-2. 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, 1, and 0: CPU Speed[2:0]. The PSoC M8C may operate over a range of CPU clock speeds (Table 11-3), 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.

The CPU frequency is changed with a write to the OSC_CR0 register. There are eight frequencies generated from a power-of-2 divide circuit, which are selected by a 3-bit code. At any given time, the CPU 8:1 clock multiplexer is selecting one of the available frequencies, which is re-synchronized 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 will be 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.

Table 11-3. 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, reference the OSC_CR0 register on page 165.

11.2.2 OSC_CR2 Register

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.

Bits 6 to 3: Reserved.

Bit 2: EXTCLKEN. When the EXTCLKEN bit is set, the external clock becomes the source for the internal clock tree, SYSCLK, which drives most chip 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.

Bit 1: IMODIS. When set, the Internal Main Oscillator is disabled. If the doubler is enabled (SYSCLKX2DIS=0), the Internal Main oscillator will be forced on.

Bit 0: SYSCLKX2DIS. When set, the Internal Main Oscillator's doubler is disabled. This will result 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, reference the OSC_CR2 register on page 167.

12. Sleep and Watchdog

This chapter discusses the Sleep and Watchdog operations and its associated registers.

Table 12-1. Sleep and Watchdog Registers

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			Analog 1		V Monitor	RW : 00
0,E3h	RES_WDT	WDSL_Clear								W : 00
x,FEh	CPU_SCR1					ECO EXW	ECO EX		IRAMDIS	RW : 00
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
x,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	W : 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 accessed or written to no matter what bank is used.
*: This bit is read only.

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, the Flash memory module, and band-gap-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, however, switched capacitor analog blocks will 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. GPIO (pin) interrupts, 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. This functionality serves to reboot the system in the event of a CPU crash. It can also restart the system from the CPU halt state.

Once the WDT is enabled, it can only be disabled by an external reset (XRES) or a power on reset (POR). A WDT reset will leave 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 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 band gap refresh circuit (to periodically refresh the reference voltage during sleep), and the Watchdog timer.

12.1.1 32 kHz Clock Selection

By default, the 32 kHz clock source is the Internal Low-Speed Oscillator (ILO). Optionally, the External Crystal Oscillator (ECO) may be activated. This selection is made in bit 7 of the OSC_CR0 register. Selecting the ECO as the active source for the 32 kHz clock allows the Sleep timer and sleep interrupt to be used in real-time 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 wakeup. It is also used to periodically refresh the bandgap voltage during sleep.

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 the Stop bit in the CPU_SCR0 is set; the Sleep timer is disabled, so that the user will 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 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, i.e., power-on reset (POR), external reset (XRES) or Watchdog reset (WDR) will reset 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 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, etc., may be significant. In addition, a significant amount of program initialization may be required. However, the Sleep timer starts counting immediately after POR and will be 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 will be a full interval.

12.1.3 Sleep Bit

Sleep is initiated in firmware by setting the SLEEP bit (bit 3) in the System Control register (CPU_SCR0). To wake up the system, this register bit is cleared asynchronously by any enabled interrupt. However, 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 that 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.

12.2 Application Description

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

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 will not go to sleep. The instruction will still execute, but it will not be able to set the SLEEP bit in the CPU_SCR0 register. Instead, the interrupt will be taken and the effect of the sleep instruction is ignored.
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 will not service the ISR associated with that interrupt. However, the system will wake up and continue 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 will be continuously asserted, and although the sleep bit may be written and the sleep sequence executed, as soon as the device enters Sleep mode, the sleep bit will be cleared by the pending interrupt and Sleep mode will be exited.
3. On wake up, the instruction immediately after the sleep instruction will be 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 will just continue where it left off before sleep).
4. If PLL mode is enabled, CPU frequency must be reduced to 3 MHz before going to sleep. Since the PLL will overshoot as it attempts to re-lock after wakeup, the CPU frequency must be relatively low. It is recommended to wait 10 ms after wakeup, before normal CPU operating frequency may be restored.
5. Analog power must be turned off by firmware, before going to sleep. 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.
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 will be executed, possibly bypassing any firmware preparations that need to be 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 could occur is after the next instruction (write to the sleep bit) has been executed. Therefore, if an interrupt is pending, the sleep instruction will be executed; but as described in #1, the sleep instruction will be ignored. The first instruction executed after the ISR will be the instruction after sleep.

12.3 Register Definitions

12.3.1 INT_MSK0 Register

The INT_MSK0 register holds bits that are used by several different resources. The digital clocks only use bit 7 of the INT_MSK0 register for the VC3 clock and bits zero through six are used by other resources. The Sleep bit (bit 6) controls whether the Sleep timer may be used as an interrupt source. For a full discussion of the INT_MSK0 register, see the Interrupt Controller chapter on page 51.

For additional information, reference the INT_MSK0 register on page 134.

12.3.2 RES_WDT Register

This write-only register has two functions. A write of any value will clear the Watchdog Timer. A write of 38h will clear both the Watchdog Timer (WDT) and the Sleep Timer. It is important to recall that the WDT is designed to timeout at 3 rollover events of the Sleep Timer. Therefore, if only the WDT is cleared, the next Watchdog Reset will occur anywhere from two to three times the current Sleep Interval setting. If the Sleep Timer is near the beginning of its count, the WD timeout will be closer to three times. However, if the Sleep Timer is very close to its terminal count, the WD timeout will be 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 of the Sleep Interval.

For additional information, reference the RES_WDT register on page 137.

12.3.3 OSC_CR0 Register

Bit 7: 32k Select. By default, the 32 kHz clock source is the Internal Low-Speed Oscillator (ILO). Optionally, the 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 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 chip systems are powered down, including the Band Gap reference. However, to facilitate the detection of POR and LVD events at a rate higher than the Sleep Interval, the Band Gap circuit is powered up periodically for about 60 us at the Sleep System Duty cycle (set in ECO_TR), which is independent of the Sleep Interval and typically more frequent. When the No Buzz bit is set, the Sleep System Duty Cycle value is overridden, and the Band Gap 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 12-2. It must be remembered that when the ILO is the selected 32 kHz clock source, sleep intervals are approximate.

Table 12-2. 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, 1, and 0: CPU Speed[2:0]. The PSoC M8C may operate over a range of CPU clock speeds (Table 12-3), 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 254 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-2 divide circuit, which are selected by a 3-bit code. At any given time, the CPU 8:1 clock multiplexer is selecting one of the available frequencies, which is re-synchronized 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 will be 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.

Table 12-3. 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, reference the OSC_CR0 register on page 165.

12.3.4 CPU_SCR1 Register

This register contains bits (3 and 2) that ensure high Watchdog Reset integrity. Since the 32 kHz oscillator source may be programmatically switched between the ECO and ILO, the switch-over must only be allowed to occur if the external crystal system actually exists. These bits are not reset by a Watchdog Reset event.

Bits 7 to 1: Reserved.

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 reset to a value of 00h. When the bit is set, 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, reference the CPU_SCR1 register on page 143.

12.3.5 ILO_TR Register

This register sets the adjustment for the ILO. 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 7 and 6: Reserved.

Bits 5 and 4: Bias Trim.

Bits 3 to 0: Freq Trim. Four bits are used to trim the frequency. The value is set in the factory and should not be changed.

For additional information, reference the ILO_TR register on page 171.

12.3.6 ECO_TR Register

The External Crystal Oscillator Trim register (ECO_TR) sets the adjustment for the 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 recommended that the user does not alter the bits in this register.

Bits 7 and 6: PSSDC[1:0]. These bits are used to set the sleep duty cycle.

Bits 5 to 0: Reserved.

For additional information, reference the ECO_TR register on page 173.

12.3.7 CPU_SCR0 Register

The bits of the CPU_SCR0 register are used to convey 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 will service interrupts.

Bit 6: Reserved.

Bit 5: WDRS. The WatchDog Reset Status bit is normally zero, but 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. This bit may not be set.

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

Bit 3: Sleep. The Sleep bit is used to enter low power Sleep mode when set, as described in this chapter.

Bits 2 and 1: Reserved.

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

For additional information, reference the CPU_SCR0 register on page 144.

12.4 Timing Diagrams

12.4.1 Sleep Sequence

The SLEEP bit 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. Due to the specific timing of the register write, 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 (power down) signal controls three major circuit blocks: The Flash memory module, the internal main oscillator (24/48 MHz oscillator, or IMO), and the bandgap voltage reference. These circuits transition into a zero power state. The only operational circuits on chip 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.

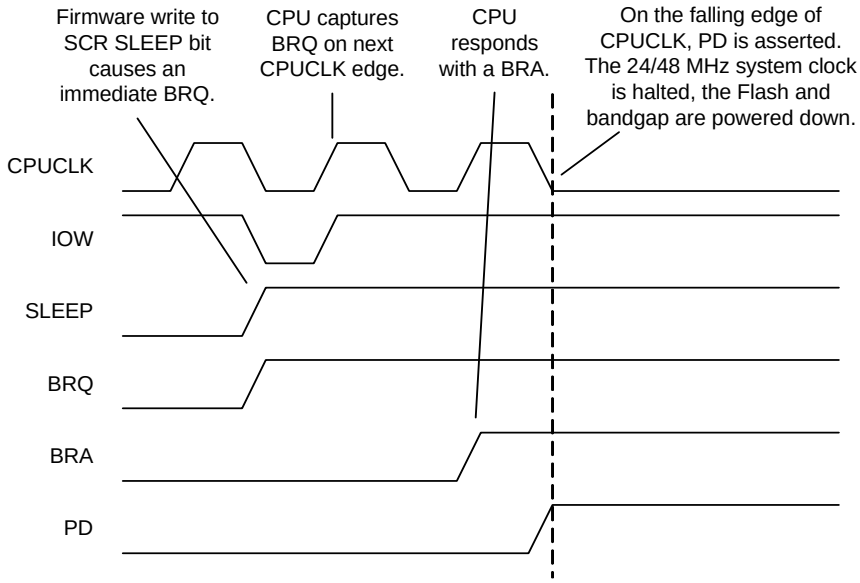


Figure 12-1. Sleep Sequence

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 will wake the system up. It is optional for the CPU to actually take the interrupt after the wakeup 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 circuit are all powered up to a normal operating state.
3. At the following positive edge of the 32 kHz clock, the current values for the precision POR and LVD have settled and are sampled.
4. At the following negative edge of the 32 kHz clock (after about 15 µs, nominal), the BRQ signal is negated by the sleep logic circuit. On the following CPUCLK, BRA is negated by the CPU and instruction execution resumes. Note that in Figure 12-2 fixed function clocks, such as Flash, IMO, and bandgap, have about 15 µs to start up.

The wake-up times (interrupt to CPU operational) will range from 75 to 105 µs.

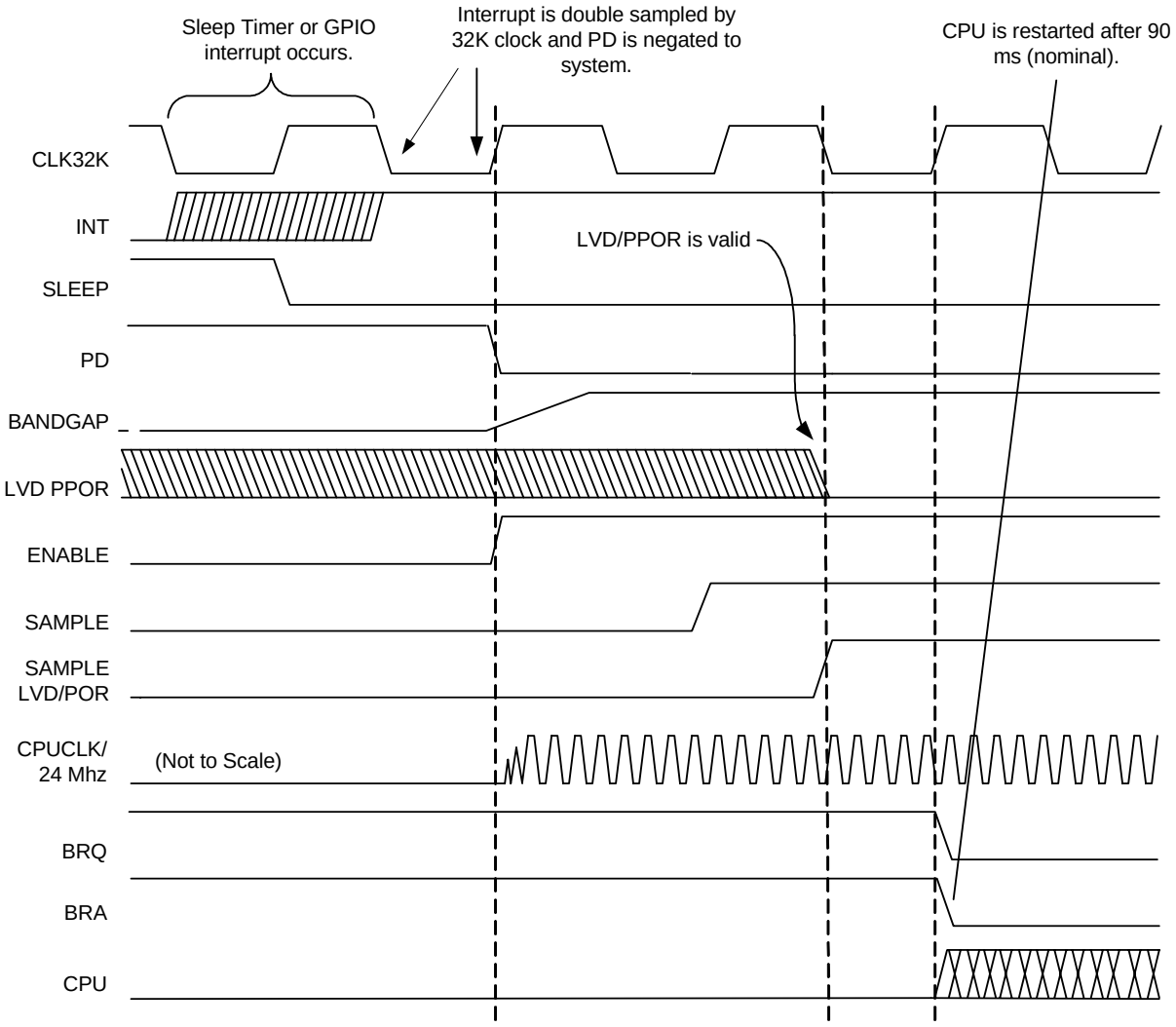


Figure 12-2. Wakeup Sequence

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 32K clock period and connecting it to VREF to refresh the reference voltage for the following 32K clock period as shown in Figure 12-3.

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

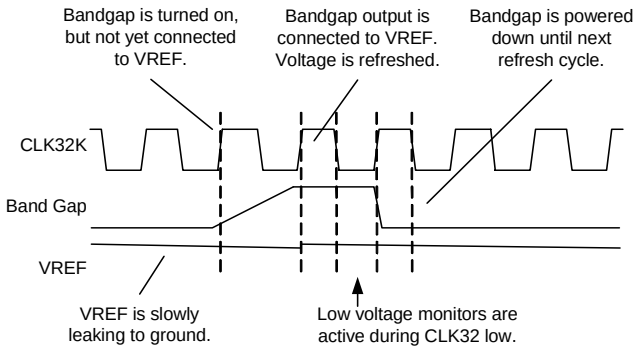


Figure 12-3. Bandgap Refresh Operation

The rate at which the refresh occurs is related to the 32 kHz clock and controlled by the Power System Sleep Duty Cycle (PDDSC, bits [7:6] of the ECO_TR register). Table 12-4 enumerates the available selections. The default setting (128 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 us

12.4.4 Watchdog Timer (WDT)

On device boot up, the 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 (i.e., 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 cognizant of this.

The Watchdog timer is three counts of the Sleep Timer interrupt output and therefore, the watchdog interval is three times the selected Sleep Timer interval. The available selections for the Watchdog interval are shown in Table 12-2. 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 will go high after the following edge of the 32 kHz clock and be held asserted for 1 cycle (30us 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.

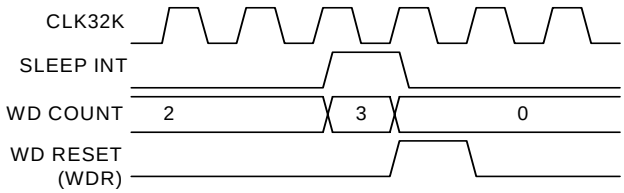


Figure 12-4. Watchdog Reset

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 will clear the Watchdog timer. (Note: A write of 38h will also clear the Sleep timer). If for any reason the firmware fails to clear the WDT within the selected interval, the circuit will assert WDR to the device. WDR is equivalent in effect to any other reset. All internal registers are set to their reset state. An important aspect to remember about WDT resets is that RAM initialization can be disabled (IRAMDIS in CPU_SCR1). 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 3X 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 will be just over 2X of the Sleep Timer interval.

12.5 Power Consumption

Sleep mode power consumption consists of the following items. 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

IPOR	1 uA
ICLK32 (ILO/ECO)	1 uA

While the CLK32 can be turned off in sleep, this mode is not useful since it makes it impossible to restart unless an IPOR reset occurs. (The sleep bit can't be cleared without CLK32.) During the sleep mode buzz, the band-gap 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

IBG (Band-gap)	(2/N) * 60 uA
ILVD (LVD comparators)	(2/N) * 50 uA

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
CLK32	1	1
IBG	0.46	0.12
ILVD	0.4	0.1
Total	2.9 uA	2.2 uA

SECTION C REGISTER REFERENCE

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

- Register Details on page 85

Register Conventions

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

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 register.
RW	Read and write register or bit(s)
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 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 XOI bit in the Flag register determines which bank the user is currently in. When the XOI bit is set, the user is said to be in the "extended" address space or the "configuration" registers.

Register Map 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	86		40				80				C0		
PRT0IE	01	RW	87		41				81				C1		
PRT0GS	02	RW	88		42				82				C2		
PRT0DM2	03	RW	89		43				83				C3		
PRT1DR	04	RW	86		44			ASD11CR0	84	RW	114		C4		
PRT1IE	05	RW	87		45			ASD11CR1	85	RW	115		C5		
PRT1GS	06	RW	88		46			ASD11CR2	86	RW	116		C6		
PRT1DM2	07	RW	89		47			ASD11CR3	87	RW	117		C7		
	08				48				88				C8		
	09				49				89				C9		
	0A				4A				8A				CA		
	0B				4B				8B				CB		
	0C				4C				8C				CC		
	0D				4D				8D				CD		
	0E				4E				8E				CE		
	0F				4F				8F				CF		
	10				50				90				D0		
	11				51				91				D1		
	12				52				92				D2		
	13				53				93				D3		
	14				54			ASC21CR0	94	RW	110		D4		
	15				55			ASC21CR1	95	RW	111		D5		
	16				56			ASC21CR2	96	RW	112	I2C_CFG	D6	RW	125
	17				57			ASC21CR3	97	RW	113	I2C_SCR	D7	#	126
	18				58				98			I2C_DR	D8	RW	127
	19				59				99			I2C_MSCR	D9	#	128
	1A				5A				9A			INT_CLR0	DA	RW	129
	1B				5B				9B			INT_CLR1	DB	RW	131
	1C				5C				9C				DC		
	1D				5D				9D			INT_CLR3	DD	RW	132
	1E				5E				9E			INT_MSK3	DE	RW	133
	1F				5F				9F				DF		
DBB00DR0	20	#	90	AMX_IN	60	RW	101		A0			INT_MSK0	E0	RW	134
DBB00DR1	21	W	91		61				A1			INT_MSK1	E1	RW	135
DBB00DR2	22	RW	92		62				A2			INT_VC	E2	RC	136
DBB00CR0	23	#	93	ARF_CR	63	RW	102		A3			RES_WDT	E3	W	137
DBB01DR0	24	#	90	CMP_CR0	64	#	103		A4			DEC_DH	E4	RC	138
DBB01DR1	25	W	91	ASY_CR	65	#	104		A5			DEC_DL	E5	RC	139
DBB01DR2	26	RW	92	CMP_CR1	66	RW	105		A6			DEC_CR0	E6	RW	140
DBB01CR0	27	#	93		67				A7			DEC_CR1	E7	RW	141
DCB02DR0	28	#	90		68				A8				E8		
DCB02DR1	29	W	91		69				A9				E9		
DCB02DR2	2A	RW	92		6A				AA				EA		
DCB02CR0	2B	#	93		6B				AB				EB		
DCB03DR0	2C	#	90		6C				AC				EC		
DCB03DR1	2D	W	91		6D				AD				ED		
DCB03DR2	2E	RW	92		6E				AE				EE		
DCB03CR0	2F	#	93		6F				AF				EF		
	30				70			RDI0RI	B0	RW	118		F0		
	31				71			RDI0SYN	B1	RW	119		F1		
	32				72			RDI0IS	B2	RW	120		F2		
	33				73			RDI0LT0	B3	RW	121		F3		
	34			ACB01CR3	74	RW	106	RDIOLT1	B4	RW	122		F4		
	35			ACB01CR0	75	RW	107	RDI0RO0	B5	RW	123		F5		
	36			ACB01CR1	76	RW	108	RDI0RO1	B6	RW	124		F6		
	37			ACB01CR2	77	RW	109		B7			CPU_F	F7	RL	142
	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	#	143
	3F				7F				BF			CPU_SCR0	FF	#	144

Blank fields are Reserved and should not be accessed. # Access is bit specific. Refer to indicated page for details.

Register Map 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	145		40				80				C0		
PRT0DM1	01	RW	146		41				81				C1		
PRT0IC0	02	RW	147		42				82				C2		
PRT0IC1	03	RW	148		43				83				C3		
PRT1DM0	04	RW	145		44			ASD11CR0	84	RW	114		C4		
PRT1DM1	05	RW	146		45			ASD11CR1	85	RW	115		C5		
PRT1IC0	06	RW	147		46			ASD11CR2	86	RW	116		C6		
PRT1IC1	07	RW	148		47			ASD11CR3	87	RW	117		C7		
	08				48				88				C8		
	09				49				89				C9		
	0A				4A				8A				CA		
	0B				4B				8B				CB		
	0C				4C				8C				CC		
	0D				4D				8D				CD		
	0E				4E				8E				CE		
	0F				4F				8F				CF		
	10				50				90			GDI_O_IN	D0	RW	159
	11				51				91			GDI_E_IN	D1	RW	160
	12				52				92			GDI_O_OU	D2	RW	161
	13				53				93			GDI_E_OU	D3	RW	162
	14				54			ASC21CR0	94	RW	110		D4		
	15				55			ASC21CR1	95	RW	111		D5		
	16				56			ASC21CR2	96	RW	112		D6		
	17				57			ASC21CR3	97	RW	113		D7		
	18				58				98				D8		
	19				59				99				D9		
	1A				5A				9A				DA		
	1B				5B				9B				DB		
	1C				5C				9C				DC		
	1D				5D				9D				DD		
	1E				5E				9E			OSC_CR4	DE	RW	163
	1F				5F				9F			OSC_CR3	DF	RW	164
DBB00FN	20	RW	149	CLK_CR0	60	RW	154		A0			OSC_CR0	E0	RW	165
DBB00IN	21	RW	151	CLK_CR1	61	RW	155		A1			OSC_CR1	E1	RW	166
DBB00OU	22	RW	152	ABF_CR0	62	RW	156		A2			OSC_CR2	E2	RW	167
	23				63				A3			VLT_CR	E3	RW	168
DBB01FN	24	RW	149		64				A4			VLT_CMP	E4	R	169
DBB01IN	25	RW	151		65				A5				E5		
DBB01OU	26	RW	152	AMD_CR1	66	RW	157		A6				E6		
	27			ALT_CR0	67	RW	158		A7				E7		
DCB02FN	28	RW	149		68				A8			IMO_TR	E8	W	170
DCB02IN	29	RW	151		69				A9			ILO_TR	E9	W	171
DCB02OU	2A	RW	152		6A				AA			BDG_TR	EA	RW	172
	2B				6B				AB			ECO_TR	EB	W	173
DCB03FN	2C	RW	149		6C				AC				EC		
DCB03IN	2D	RW	151		6D				AD				ED		
DCB03OU	2E	RW	152		6E				AE				EE		
	2F				6F				AF				EF		
	30				70			RDI0RI	B0	RW	118		F0		
	31				71			RDI0SYN	B1	RW	119		F1		
	32				72			RDI0IS	B2	RW	120		F2		
	33				73			RDI0LT0	B3	RW	121		F3		
	34			ACB01CR3	74	RW	106	RDIOLT1	B4	RW	122		F4		
	35			ACB01CR0	75	RW	107	RDI0RO0	B5	RW	123		F5		
	36			ACB01CR1	76	RW	108	RDI0RO1	B6	RW	124		F6		
	37			ACB01CR2	77	RW	109		B7			CPU_F	F7	RL	142
	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	#	143
	3F				7F				BF			CPU_SCR0	FF	#	144

Blank fields are Reserved and should not be accessed. # Access is bit specific. Refer to indicated page for details.

13. Register Details

This chapter details all the PSoC device registers in offset order for Bank 0 and Bank 1. The registers that are in both banks are incorporated with the Bank 0 registers, designated with at least one ‘x’ in the register name and offset.

Bank 0 registers are listed first and begin on page 86. Bank 1 registers are listed second and begin on page 145. If you need a condensed view of all the registers, reference the “Register Mapping Tables” on page 81. The conventions specific to the registers in this chapter are listed below.

Table 13-1: Register Conventions

Convention	Example	Description
‘x’ in a register name	ACBxxCR1	Multiple instances/address ranges of the same register.
RW	RW : 00	Read and write register or bit(s)
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

13.1 Bank 0 Registers

The following registers are all in bank 0 and are listed in offset order.

13.1.1 PRTxDR

Port Data Register

Individual Register Names and Addresses

PRT0DR : 0,00h PRT1DR : 0,04h

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

For additional information, reference the “Register Definitions” on page 58 in the GPIO chapter.

Bit	Name	Description
[7:0]	**Data Input[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.

13.1.2 PRTxIE

Port Interrupt Enable Register

Individual Register Names and Addresses

PRT0IE : 0,01h PRT1IE : 0,05h

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

For additional information, reference the “Register Definitions” on page 58 in the GPIO chapter.

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

13.1.3 PRTxGS
Port Global Select Register

Individual Register Names and Addresses

PRT0GS : 0,02h PRT1GS : 0,06h

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

For additional information, reference the “Register Definitions” on page 58 in the GPIO chapter.

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

13.1.4 PRTxDM2

Port Drive Mode Bit 2 Register

Individual Register Names and Addresses

PRT0DM2 : 0,03h PRT1DM2 : 0,07h

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

In register PRTxDM2 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 145, "PRTxDM1" on page 146, and PRTxDM2). The bit position of the effected port pin (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 (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**10] refers to the combination (in order) of bits in a given bit position); however, this register only controls the most significant bit of the drive mode. For additional information, reference the "Register Definitions" on page 58 in the GPIO chapter.

Bit	Name	Description			
[7:0]	**Drive Mode 2[7:0]**	Bit 2 of the drive mode, for each of an 8-port pin, for a GPIO port.			
		[210**]**	**Pin Output High**	**Pin Output Low**	**Notes**
		000b	Strong	Resistive	
		001b	Strong	Strong	
		010b	Hi-z	Hi-z	Digital input enabled.
		011b	Resistive	Strong	
		100b	Slow + strong	Hi-z	
		101b	Slow + strong	Slow + strong	
		110b	Hi-z	Hi-z	Digital input disabled for zero power. Reset state.
		111b	Hi-z	Slow + strong	I^2C Compatible mode.

13.1.5 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]							

The function of this register is dependant on the function its block has been configured for (selected in the FN[2:0] bits of the DxBxxFN register on page 149). For the Timer, Counter, Dead Band, and CRCPRS functions, a read of the DxBxxDR0 register returns 00h and transfers DxBxxDR0 to DxBxxDR2.

For additional information, reference the "Register Definitions" on page 198 in the Digital Blocks chapter.

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

Block Function	Register Function
Timer	Count Value
Counter	Count Value
Dead Band	Count Value
CRCPRS	Linear Feedback Shift Register (LFSR)
SPIM	Shifter
SPIS	Shifter
TXUART	Shifter
RXUART	Shifter

13.1.6 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]							

The function of this register is dependant on the function its block has been configured for (selected in the FN[2:0] bits of the DxBxxFN register on page 149). For additional information, reference the "Register Definitions" on page 198 in the Digital Blocks chapter.

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

Block Function	Register Function
Timer	Period
Counter	Period
Dead Band	Period
CRCPRS	Polynomial
SPIM	TX Buffer
SPIS	TX Buffer
TXUART	TX Buffer
RXUART	Not applicable

13.1.7 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]							

The function of this register is dependant on the function its block has been configured for (selected in the FN[2:0] bits of the DxBxxFN register on page 149. For additional information, reference the "Register Definitions" on page 198 in the Digital Blocks chapter.

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

Block Function	Register Function
Timer	Capture/Compare
Counter	Compare
Dead Band	Buffer
CRCPRS	Seed/Residue
SPIM	RX Buffer
SPIS	RX Buffer
TXUART	Not applicable
RXUART	RX Buffer

13.1.8 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

For additional information, reference the “Register Definitions” on page 198 in the Digital Blocks chapter.

Bit	Name	Description
[7:3]	**Reserved**	
[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.

13.1.9 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

For additional information, reference the “Register Definitions” on page 198 in the Digital Blocks chapter.

Bit	Name	Description	
[7:1]	**Reserved**		
[0]	**Enable**	0	Counter is not enabled.
		1	Counter is enabled.

13.1.10 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

For additional information, reference the "Register Definitions" on page 198 in the Digital Blocks chapter.

Bit	Name	Description
[7:3]	**Reserved**	
[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.

13.1.11 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

For additional information, reference the “Register Definitions” on page 198 in the Digital Blocks chapter.

Bit	Name	Description
[7:2]	**Reserved**	
[1]	**Pass Mode**	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.

13.1.12 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

The LSB First, Clock Phase, and Clock Polarity bit 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. For additional information, reference the "Register Definitions" on page 198 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 has been received and loaded into the RX Buffer before the previous one could be read. Cleared on 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 has been shifted out and all associated clocks have been generated. Cleared on read of this (CR0) register. Optional interrupt.
[4]	**TX Reg Empty**	The 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 can be written to the TX register. Cleared on write of the TX Buffer (DR1) register. Default interrupt. This status will initially be asserted on block enable; however, the TX Reg Empty interrupt will occur 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 has been received and loaded into the RX register. Cleared on read of the RX Buffer (DR2) register.
[2]	**Clock Phase**	0 Data is latched on the leading edge of the clock. Data changes on the trailing edge (Modes 0,1). 1 Data changes on the leading edge of the clock. 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.

13.1.13 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

The LSB First, Clock Phase, and Clock Polarity bit 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. For additional information, reference the "Register Definitions" on page 198 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 has been received and loaded into the RX Buffer before the previous one could be read. Cleared on 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 has been shifted out and all associated clocks have been generated. Cleared on read of this (CR0) register. Optional interrupt.
[4]	**TX Reg Empty**	The 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 can be written to the TX register. Cleared on write of the TX Buffer (DR1) register. Default interrupt. This status will initially be asserted on block enable; however, the TX Reg Empty interrupt will occur 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 has been received and loaded into the RX register. Cleared on read of the RX Buffer (DR2) register.
[2]	**Clock Phase**	0 Data is latched on the leading edge of the clock. Data changes on the trailing edge. 1 Data changes on the leading edge of the clock. Data is latched on the trailing edge.
[1]	**Clock Polarity**	0 Non-inverted, clock idles low. 1 Inverted, clock idles high.
[0]	**Enable**	0 SPI Slave is not enabled. 1 SPI Slave is enabled.

13.1.14 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

For additional information, reference the "Register Definitions" on page 198 in the Digital Blocks chapter. For the receive mode definition, refer to section 13.1.15 on page 100.

Bit	Name	Description
[7:6]	**Reserved**	
[5]	**TX Complete**	0 Indicates that a byte may still be in the process of shifting out. 1 Indicates that a byte has been shifted out and all associated framing bits have been generated. Optional interrupt. Cleared on read of this (CR0) register.
[4]	**TX Reg Empty**	The 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 can be written to the TX register. Cleared on write of the TX Buffer register. Default interrupt. TX Reg Empty interrupt will occur only after the first data byte is written and transferred into the shifter.
[3]	**Reserved**	
[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.

13.1.15 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

For additional information, reference the "Register Definitions" on page 198 in the Digital Blocks chapter. For the transmit mode definition, refer to section 13.1.14 on page 99.

Bit	Name	Description	
[7]	**Parity Error**	0 1	Indicates that no parity error has occurred. Valid when RX Reg Full is set, indicating that a parity error has occurred in the received byte. Cleared on read of this (CR0) register.
[6]	**Overrun**	0 1	Indicates that no overrun has occurred. 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. Cleared on read of this (CR0) register.
[5]	**Framing Error**	0 1	Indicates no framing error has occurred. 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'). Cleared on read of this (CR0) register.
[4]	**RX Active**	0 1	Indicates that no reception is in progress. Indicates that a reception is in progress. Set by the detection of a START bit and cleared at the sampling of the STOP bit.
[3]	**RX Reg Full**	0 1	Indicates that the RX Buffer register is empty. Indicates that a byte has been 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 1	Even parity Odd parity
[1]	**Parity Enable**	0 1	Parity is not enabled. Parity is enabled, frame includes parity bit.
[0]	**Enable**	0 1	Serial Receiver is not enabled. Serial Receiver is enabled.

13.1.16 AMX_IN

Analog Input Select Register

Individual Register Names and Addresses

AMX_IN: 0,60h

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

For additional information, reference the “Register Definitions” on page 235 in the Analog Input Configuration chapter.

Bit	Name	Description
[7:4]	**Reserved**	
[3:2]	**ACI1[1:0]**	Selects the Analog Column Mux 1, even inputs. 00b ACM1 P0[0] 01b ACM1 P0[2] 10b ACM1 P0[4] 11b ACM1 P0[6] **Note** ACol1Mux (ABF_CR, Address = Bank1, 62h) 0 AC1 = ACM1 1 AC1 = ACM0
[1:0]	**ACI0[1:0]**	Selects the Analog Column Mux 1, odd inputs. 00b ACM0 P0[1] 01b ACM0 P0[3] 10b ACM0 P0[5] 11b ACM0 P0[7]

13.1.17 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]		

For additional information, reference the “Register Definitions” on page 238 in the Analog Reference chapter.

Bit **Name** **Description**

[7] **Reserved**

[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 (RefHigh), and the low reference (RefLow).
000b Invalid Reference
001b Invalid Reference
010b Valid Reference: AGND = Vdd/2, RefHigh = Vdd, RefLow = Vss.
011b Invalid Reference
100b Invalid Reference
101b Invalid Reference
110b Invalid Reference
111b Invalid Reference

[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

13.1.18 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]				AINT[1]	

For additional information, reference the “Register Definitions” on page 228 in the Analog Interface chapter.

Bit	Name	Description
[7:6]	**Reserved**	
[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. If the comparator latch disable bits are set, then this bit is transparent to the comparator bus in the analog array.
[4:2]	**Reserved**	
[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]	**Reserved**	

13.1.19 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]		SYNCEN

For additional information, reference the “Register Definitions” on page 228 in the Analog Interface chapter.

Bit	Name	Description
[7]	**Reserved**	
[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, will result 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 should be set to ‘0’ when processing the most significant block, and ‘1’ when processing the least significant block. This is because the least significant block is an inverting input to the most significant block.
[2]	**SARCOL[1]**	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
[1]	**Reserved**	
[0]	**SYNCEN**	Set to ‘1’, will stall the CPU until the rising edge of PHI1, if a write to a register within an analog Switch Cap block takes place. 0 CPU stalling disabled. 1 CPU stalling enabled.

13.1.20 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					
Bit Name			CLDIS[1]					

For additional information, reference the “Register Definitions” on page 228 in the Analog Interface chapter.

Bit	Name	Description
[7:6]	**Reserved**	
[5]	**CLDIS[1]**	Controls the comparator output latch, column 1. 0 The comparator data bits are updated from the analog comparator bus, on the rising edge of PHI2. 1 The comparator data bits are connected transparently to the analog comparator bus. This mode can be used in power down operation, to wake the part out of sleep as a result of an analog column interrupt.
[4:0]	**Reserved**	

13.1.21 ACBxxCR3

Analog Continuous Time Type B Block Control Register 3

Individual Register Names and Addresses

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

For additional information, reference the “Register Definitions” on page 247 in the Continuous Time Block chapter.

Bit	Name	Description	
[7:4]	**Reserved**		
[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 ACBxxCR0)

13.1.22 ACBxxCR0

Analog Continuous Time Type B Block Control Register 0

Individual Register Names and Addresses

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]	

For additional information, reference the “Register Definitions” on page 247 in the Continuous Time Block chapter.

Bit **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

[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

[1:0] RbotMux[1:0] Encoding for feedback resistor select. Bits [1:0] are overridden if bit 1 of ACBxxCR3 is set. In that case, the bottom of the resistor string is connected cross columns. Note that available mux inputs vary by individual PSoC block.

	ACB01
00b	Reserved
01b	AGND
10b	Vss
11b	ASD11

13.1.23 ACBxxCR1

Analog Continuous Time Type B Block Control Register 1

Individual Register Names and Addresses

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]		

For additional information, reference the “Register Definitions” on page 247 in the Continuous Time Block chapter.

Bit	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. **ACB01** 000b Reserved 001b AGND 010b Vss 011b Vdd 100b FB# 101b ASD11 110b Reserved 111b Port Inputs # Feedback point from tap of the feedback resistor as defined by corresponding CR0 bits [7:4] and CR3 bit 0.
[2:0]	**PMux[2:0]**	Encoding for positive input select. Note that available mux inputs vary by individual PSoC block. **ACB01** 000b Vss 001b Port Inputs 010b Reserved 011b AGND 100b ASD11 101b Reserved 110b ABUS1 111b FB# # Feedback point from tap of the feedback resistor as defined by corresponding CR0 bits [7:4] and CR3 bit 0.

13.1.24 ACBxxCR2

Analog Continuous Time Type B Block Control Register 2

Individual Register Names and Addresses

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]	

For additional information, reference the “Register Definitions” on page 247 in the Continuous Time Block chapter.

Bit	Name	Description
[7]	**CPhase**	0 Comparator Control latch transparent on PHI1. 1 Comparator Control latch 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. **ACB01** 00b Positive Input to ABUS1 01b AGND to ABUS1 10b REFLO to ABUS1 11b REFHI to ABUS1
[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 102. 00b Off 01b Low 10b Medium 11b High

13.1.25 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]				

For additional information, reference the “Register Definitions” on page 241 in the Switched Capacitor Block chapter.

Bit	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 cap 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.

13.1.26 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]				

For additional information, reference the “Register Definitions” on page 241 in the Switched Capacitor Block chapter.

Bit	Name	Description
[7:5]	**ACMux[2:0]**	Encoding for selecting A and C inputs. (Note that available mux inputs vary by individual PSoC block.)

ASC21

	A Inputs	*C Inputs*
000b	ASD11	ASD11
001b	Reserved	Reserved
010b	Vdd	ASD11
011b	Vtemp	ASD11
100b	Reserved	Reserved
101b	Reserved	Reserved
110b	ABUS1	ASD11
111b	Reserved	Reserved

[4:0]	**BCap[4:0]**	Binary encoding for 32 possible capacitor sizes of the capacitor BCap.

13.1.27 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]				

For additional information, reference the “Register Definitions” on page 241 in the Switched Capacitor Block chapter.

Bit	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]	**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.

13.1.28 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]	

For additional information, reference the “Register Definitions” on page 241 in the Switched Capacitor Block chapter.

Bit	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.
[3:2]	**BMuxSC[1:0]**	Encoding for selecting B inputs. Note that the available mux inputs vary by individual PSoC block. **ASC21** 00b ASD11 01b Reserved 10b Reserved 11b TrefGND
[1:0]	**PWR[1:0]**	Encoding for selecting one of four power levels. 00b Off 01b Low 10b Medium 11b High

13.1.29 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]				

For additional information, reference the “Register Definitions” on page 241 in the Switched Capacitor Block chapter.

Bit	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 cap 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.

13.1.30 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]				

For additional information, reference the “Register Definitions” on page 241 in the Switched Capacitor Block chapter.

Bit	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.) **ASD11** 000b ACB01 001b Reserved 010b Reserved 011b ASC21 100b Vdd 101b Reserved 110b Reserved 111b Reserved
[4:0]	**BCap[4:0]**	Binary encoding for 32 possible capacitor sizes for capacitor BCap.

13.1.31 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]				

For additional information, reference the “Register Definitions” on page 241 in the Switched Capacitor Block chapter.

Bit	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]	**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 for capacitor CCap.

13.1.32 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]	

For additional information, reference the “Register Definitions” on page 241 in the Switched Capacitor Block chapter.

Bit	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.) **ASD11** 0 Reserved 1 ACB01
[1:0]	**PWR[1:0]**	Encoding for selecting one of four power levels. 00b Off 01b 10 µA, typical 10b 50 µA, typical 11b 200 µA, typical

13.1.33 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		RW : 0		RW : 0		RW : 0	
Bit Name	RI3[2:0]		RI2[2:0]		RI1[2:0]		RI0[2:0]	

For additional information, reference the “Register Definitions” on page 186 in the Row Digital Interconnect chapter.

Bit	Name	Description
[7:6]	**RI3[2:0]**	Select source for row input 3. 00b GIE[3] 01b GIE[7] 10b GIO[3] 11b GIO[7]
[5:4]	**RI2[2:0]**	Select source for row input 2. 00b GIE[2] 01b GIE[6] 10b GIO[2] 11b GIO[6]
[3:2]	**RI1[2:0]**	Select source for row input 1. 00b GIE[1] 01b GIE[5] 10b GIO[1] 11b GIO[5]
[1:0]	**RI0[2:0]**	Select source for row input 0. 00b GIE[0] 01b GIE[4] 10b GIO[0] 11b GIO[4]

13.1.34 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	RW : 0	RW : 0	RW : 0
Bit Name					RI3SYN	RI2SYN	RI1SYN	RI0SYN

For additional information, reference the “Register Definitions” on page 186 in the Row Digital Interconnect chapter.

Bit	Name		Description
[7:4]	**Reserved**		
[3]	**RI3SYN**	0	Row input 3 is synchronized to SYSCLK system clock.
		1	Row input 3 is passed without synchronization.
[2]	**RI2SYN**	0	Row input 2 is synchronized to SYSCLK system clock.
		1	Row input 2 is passed without synchronization.
[1]	**RI1SYN**	0	Row input 1 is synchronized to SYSCLK system clock.
		1	Row input 1 is passed without synchronization.
[0]	**RI0SYN**	0	Row input 0 is synchronized to SYSCLK system clock.
		1	Row input 0 is passed without synchronization.

13.1.35 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

For additional information, reference the “Register Definitions” on page 186 in the Row Digital Interconnect chapter.

Bit	Name	Description
[7:6]	**Reserved**	
[5:4]	**BCSEL[1:0]**	When the BCSELL 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 Reserved 10b Reserved 11b Reserved
[3]	**IS3**	0 The ‘A’ input of LUT 3 is RO[3]. 1 The ‘A’ input of LUT 3 is RI[3].
[2]	**IS2**	0 The ‘A’ input of LUT 2 is RO[2]. 1 The ‘A’ input of LUT 2 is RI[2].
[1]	**IS1**	0 The ‘A’ input of LUT 1 is RO[1]. 1 The ‘A’ input of LUT 1 is RI[1].
[0]	**IS0**	0 The ‘A’ input of LUT 0 is RO[0]. 1 The ‘A’ input of LUT 0 is RI[0].

13.1.36 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]			

For additional information, reference the “Register Definitions” on page 186 in the Row Digital Interconnect chapter.

Bit	Name	Description
[7:4]	**LUT1[3:0]**	Select logic function for LUT 1.

	Function
0h	FALSE
1h	A AND $\overline{B}$
2h	A AND $\overline{B}$
3h	$\overline{A}$
4h	$\overline{A}$ AND B
5h	B
6h	A XOR B
7h	A OR B
8h	A NOR B
9h	$\overline{A}$ XNOR B
Ah	$\overline{B}$
Bh	$\overline{A}$ 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 LUT 0.

	Function
0h	FALSE
1h	A AND $\overline{B}$
2h	A AND $\overline{B}$
3h	$\overline{A}$
4h	$\overline{A}$ AND B
5h	B
6h	A XOR B
7h	A OR B
8h	A NOR B
9h	$\overline{A}$ XNOR B
Ah	$\overline{B}$
Bh	$\overline{A}$ OR $\overline{B}$
Ch	$\overline{A}$
Dh	$\overline{A}$ OR B
Eh	A NAND B
Fh	TRUE

13.1.37 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]			

For additional information, reference the “Register Definitions” on page 186 in the Row Digital Interconnect chapter.

Bit	Name	Description
[7:4]	LUT3[3:0]	Select logic function for LUT 3.

	Function
0h	FALSE
1h	A AND $\overline{B}$
2h	A AND $\overline{B}$
3h	$\overline{A}$
4h	$\overline{A}$ AND B
5h	B
6h	A XOR B
7h	A OR B
8h	A NOR B
9h	$\overline{A}$ XNOR B
Ah	$\overline{B}$
Bh	$\overline{A}$ OR $\overline{B}$
Ch	$\overline{A}$
Dh	$\overline{A}$ OR B
Eh	A NAND B
Fh	TRUE

[3:0]	LUT2[3:0]	Select logic function for LUT 2.

	Function
0h	FALSE
1h	A AND $\overline{B}$
2h	A AND $\overline{B}$
3h	$\overline{A}$
4h	$\overline{A}$ AND B
5h	B
6h	A XOR B
7h	A OR B
8h	A NOR B
9h	$\overline{A}$ XNOR B
Ah	$\overline{B}$
Bh	$\overline{A}$ OR $\overline{B}$
Ch	$\overline{A}$
Dh	$\overline{A}$ OR B
Eh	A NAND B
Fh	TRUE

13.1.38 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

For additional information, reference the “Register Definitions” on page 186 in the Row Digital Interconnect chapter.

Bit	Name	Description	
[7]	**GOO5EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOO[5].
[6]	**GOO1EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOO[1].
[5]	**GOE5EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOE[5].
[4]	**GOE1EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOE[1].
[3]	**GOO4EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOO[4].
[2]	**GOO0EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOO[0].
[1]	**GOE4EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOE[4].
[0]	**GOE0EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOE[0].

13.1.39 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

For additional information, reference the “Register Definitions” on page 186 in the Row Digital Interconnect chapter.

Bit	Name	Description	
[7]	**GOO7EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOO[7].
[6]	**GOO3EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOO[3].
[5]	**GOE7EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOE[7].
[4]	**GOE3EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOE[3].
[3]	**GOO6EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOO[6].
[2]	**GOO2EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOO[2].
[1]	**GOE6EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOE[6].
[0]	**GOE2EN**	0	Disable LUT output to global output.
		1	Enable LUT output to GOE[2].

13.1.40 I2C_CFG

I2C 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		Enable Master	Enable Slave

For additional information, reference the "Register Definitions" on page 265 in the I^2C chapter.

Bit	Name	Description
[7]	**Reserved**	
[6]	**PSelect**	I^2C Pin Select 0 P1[5] and P1[7] 1 P1[0] and P1[1] **Note** Read the I^2C chapter for a discussion of the side effects of choosing this pair of pins: P1[0] and P1[1].
[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**	00b 100K Standard Mode 01b 400K Fast Mode 10b 50K Standard Mode 11b Reserved
[1]	**Enable Master**	0 Disabled 1 Enabled
[0]	**Enable Slave**	0 Disabled 1 Enabled

13.1.41 I2C_SCR

I2C 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

Bits in this register are held in reset until one of the enable bits in I2C_CFG is set. For additional information, reference the "Register Definitions" on page 265 in the I^{2}C 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 was 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, will also clear 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 was 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**	This 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, will also clear 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, will also clear the bit. 0 Last transmitted byte was ACKed by the receiver. 1 Last transmitted byte was NACKed by the receiver.
[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, will also clear 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.

13.1.42 I2C_DR

I2C 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 is read only for received data and write only for transmitted data. For additional information, reference the “Register Definitions” on page 265 in the I^2C chapter.

Bit	Name	Description
[7:0]	**Data**	Read received data or write data to transmit.

13.1.43 I2C_MSCR

I2C 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

Bits in this register are held in reset until one of the enable bits in I2C_CFG is set. For additional information, reference the "Register Definitions" on page 265 in the I^2C chapter.

Bit	Name	Description
[7:4]	**Reserved**	
[3]	**Bus Busy**	This bit is set to: 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.

13.1.44 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
Bit Name	VC3	Sleep	GPIO			Analog 1		V Monitor

When bits in this register are read, a '1' will be returned for every bit position that has a corresponding posted interrupt. When bits in this register are written with a zero (0) and ENSWINT is not set, posted interrupts will be 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 one (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 133. For additional information, reference the "Register Definitions" on page 53 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).
[4:3]	**Reserved**	
[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]	**Reserved**	

(continued on next page)

13.1.44 INT_CLR0 *(continued)*

[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.

13.1.45 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 : 00	RW : 0
Bit Name					DCB03	DCB02	DBB01	DBB00

When bits in this register are read, a ‘1’ will be returned for every bit position that has a corresponding posted interrupt. When bits in this register are written with a zero (0) and ENSWINT is not set, posted interrupts will be 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 one (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 133. For additional information, reference the “Register Definitions” on page 53 in the Interrupt Controller chapter.

Bit	Name	Description
[7:4]	**Reserved**	
[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.

13.1.46 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

When bits in this register are read, a ‘1’ will be returned for every bit position that has a corresponding posted interrupt. When bits in this register are written with a zero (0) and ENSWINT is cleared, any posted interrupt will be cleared. If there was not a posted interrupt, there is no effect. When bits in this register are written with a one (1) and ENSWINT is set, an interrupt is posted in the interrupt controller. For additional information, reference the “Register Definitions” on page 53 in the Interrupt Controller chapter.

Bit	Name	Description
[7:1]	**Reserved**	
[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.

13.1.47 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

Note that when an interrupt is masked off, the mask bit is ‘0’. The interrupt will still post in the interrupt controller. Therefore, clearing the mask bit only prevents a posted interrupt from becoming a pending interrupt. For additional information, reference the “Register Definitions” on page 53 in the Interrupt Controller chapter.

Bit	Name	Description	
[7]	**ENSWINT**	0	Disable software interrupts.
		1	Enable software interrupts.
[6:1]	**Reserved**		
[0]	**I2C**	0	Mask I2C interrupt
		1	Unmask I2C interrupt

13.1.48 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
Bit Name	VC3	Sleep	GPIO			Analog 1		V Monitor

Note that when an interrupt is masked off, the mask bit is ‘0’. The interrupt will still post in the interrupt controller. Therefore, clearing the mask bit only prevents a posted interrupt from becoming a pending interrupt. For additional information, reference the “Register Definitions” on page 53 in the Interrupt Controller chapter.

Bit	Name	Description	
[7]	**VC3**	0 1	Mask VC3 interrupt. Unmask VC3 interrupt.
[6]	**Sleep**	0 1	Mask sleep interrupt. Unmask sleep interrupt.
[5]	**GPIO**	0 1	Mask GPIO interrupt. Unmask GPIO interrupt.
[4:3]	**Reserved**		
[2]	**Analog 1**	0 1	Mask analog interrupt, column 1. Unmask analog interrupt.
[1]	**Reserved**		
[0]	**V Monitor**	0 1	Mask voltage monitor interrupt. Unmask voltage monitor interrupt.

13.1.49 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	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name					DCB03	DCB02	DBB01	DBB00

Note that when an interrupt is masked off, the mask bit is '0'. The interrupt will still post in the interrupt controller. Therefore, clearing the mask bit only prevents a posted interrupt from becoming a pending interrupt. For additional information, reference the "Register Definitions" on page 53 in the Interrupt Controller chapter.

Bit	Name	Description	
[7:4]	**Reserved**		
[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.

13.1.50 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]							

For additional information, reference the “Register Definitions” on page 53 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.

13.1.51 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]							

For additional information, reference the “Register Definitions” on page 75 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.

13.1.52 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]							

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, reference the “Register Definitions” on page 259 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 (i.e., it is not necessary to write both).

13.1.53 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]							

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, reference the “Register Definitions” on page 259 in the Decimator chapter.

Bit	Name	Description	
[7:0]	**Data Low 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 (i.e., it is not necessary to write both).

13.1.54 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 : 00				RW : 0	RW : 0		RW : 0
Bit Name	IGEN[3:0]				ICLKS0	DCOL[1:0]		DCLKS0

For additional information, reference the “Register Definitions” on page 259 in the Decimator chapter.

Bit	Name	Description
[7:4]	**IGEN[3:0]**	Incremental Gate Enable. Selects on a column basis which comparator outputs will be gated with the incremental gating function. 1h Reserved 2h Analog Column 1 4h Reserved 8h Reserved
[3]	**ICLKS0**	Incremental Gate Source. Along with ICLKS1 in DEC_CR1, selects one of the possible digital blocks, depending on your chip resources, to control the incremental gating function. **ICLKS1 (see the DEC_CR1 register), ICLKS0** 000b Digital block 02 010b Digital block 01
[2:1]	**DCOL[1:0]**	Decimator Column Source. Selects the analog comparator column as a data source for the decimator. 00b Reserved 01b Analog Column 1 10b Reserved 11b Reserved
[0]	**DCLKS0**	Decimator Latch Select. Along with DCLKS1 in DEC_CR1, selects one of the possible digital blocks, depending on your chip resources, to control the decimator output latch. **DCLKS1 (see the DEC_CR1 register), DCLKS0** 000b Digital block 02 010b Digital block 01

13.1.55 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
Bit Name	ECNT	IDEC			ICLKS1			DCLKS1

For additional information, reference the “Register Definitions” on page 259 in the Decimator chapter.

Bit	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 {DCLKS[1:0], DCLKS0}). 0 Non-inverted 1 Inverted
[5:4]	**Reserved**	
[3]	**ICLKS1**	Incremental Gate Source. Along with ICLKS0 in DEC_CR0, selects one of the possible digital blocks, depending on your chip resources, to control the incremental gating function. **ICLKS1, ICLKS0 (see the DEC_CR0 register)** 000b Digital block 02 010b Digital block 01
[2:1]	**Reserved**	
[0]	**DCLKS1**	Decimator Latch Select. Along with DCLKS0 in DEC_CR0, selects one of the possible digital blocks, depending on your chip resources, to control the decimator output latch. **DCLKS1, DCLKS0 (see the DEC_CR0 register)** 000b Digital block 02 010b Digital block 01

13.1.56 CPU_F

M8C Flags 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
Bit Name				XOI		Carry	Zero	GIE

The AND, OR, and XOR flag instructions can be used to modify this register. For additional information, reference the "Register Definitions" on page 43 in the M8C chapter and the "Register Definitions" on page 53 in the Interrupt Controller chapter.

Bit	Name	Description
[7:5]	**Reserved**	
[4]	**XOI**	0 Normal register address space 1 Extended register address space. Primarily used for configuration.
[3]	**Reserved**	
[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 will not process any interrupts. 1 Interrupt processing enabled.

13.1.57 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								RW : 0
Bit Name								IRAMDIS

For additional information, reference the “Register Definitions” on page 49 in the SROM chapter or “Register Definitions” on page 68 of the 32 kHz Crystal Oscillator chapter.

Bit	Name	Description	
[7:1]	**Reserved**		
[0]	**IRAMDIS**	0	SRAM is initialized to 00h after POR, XRES, and WDR.
		1	Address 03h - D7h are not modified by WDR.

13.1.58 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

For additional information, reference the “Register Definitions” on page 75 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 142. This bit is Read Only for GIES. Its use is discouraged, as the Flag register is now readable at address x,F7h (read only).
[6]	**Reserved**	
[5]	**WDRS**	Watchdog Reset Status. This bit may not be set by user code; however, it may be cleared by writing it with a zero (0). 0 No Watchdog Reset has occurred. 1 Watchdog Reset has 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 zero (0). 0 Power On Reset has not occurred and Watchdog Timer is enabled. 1 Will be set after external reset or Power On Reset.
[3]	**Sleep**	Set by the user to enable the CPU sleep state. CPU will remain in Sleep mode until any interrupt is pending. 0 Normal operation 1 Sleep
[2:1]	**Reserved**	
[0]	**STOP**	0 M8C is free to execute code. 1 M8C is halted. Can only be cleared by POR, XRES, or WDR.

13.2 Bank 1 Registers

The following registers are all in bank 1 and are listed in offset order.

13.2.1 PRTxDM0

Port Drive Mode Bit Register 0

Individual Register Names and Addresses

PRT0DM0 : 1,00h PRT1DM0 : 1,04h

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

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 146, and "PRTxDM2" on page 89). The bit position of the effected port pin (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 (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 of the drive mode. For additional information, reference the "Register Definitions" on page 58 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	Hi-z	Hi-z	Digital input enabled.
		01**1**b	Resistive	Strong	
		10**0**b	Slow + strong	Hi-z	
		10**1**b	Slow + strong	Slow + strong	
		11**0**b	Hi-z	Hi-z	Digital input disabled for zero power. Reset state.
		11**1**b	Hi-z	Slow + strong	I^2C Compatible mode.
		Note A bold digit, in the table above, signifies that the digit is used in this register.			

13.2.2 PRTxDM1

Port Drive Mode Bit Register 1

Individual Register Names and Addresses

PRT0DM1 : 1,01h PRT1DM1 : 1,05h

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

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 145, PRTxDM1, and "PRTxDM2" on page 89). The bit position of the effected port pin (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 (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, reference the "Register Definitions" on page 58 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.

[21**0]**	**Pin Output High**	**Pin Output Low**	**Notes**
0**0**0b	Strong	Resistive	
0**0**1b	Strong	Strong	
0**1**0b	Hi-z	Hi-z	Digital input enabled.
0**1**1b	Resistive	Strong	
1**0**0b	Slow + strong	Hi-z	
1**0**1b	Slow + strong	Slow + strong	
1**1**0b	Hi-z	Hi-z	Digital input disabled for zero power. Reset state.
1**1**1b	Hi-z	Slow + strong	I^2C Compatible mode.

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

13.2.3 PRTxIC0

Port Interrupt Control Register 0

Individual Register Names and Addresses

PRT0IC0 : 1,02h PRT1IC0 : 1,06h

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

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 148). The bit position of the effected port pin (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 (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, reference the “Register Definitions” on page 58 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.	

13.2.4 PRTxIC1

Port Interrupt Control Register 1

Individual Register Names and Addresses

PRT0IC1 : 1,03h PRT1IC1 : 1,07h

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

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 147 and PRTxIC1). The bit position of the effected port pin (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 (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, reference the “Register Definitions” on page 58 in the GPIO chapter.

Bit	Name	Description	
[7:0]	**Interrupt Control 1[7:0]**	[1**0**]	**Interrupt Type**
		00b	Disabled
		01b	Low
		10b	High
	‘	**1**1b	Change from last read
		Note A bold digit, in the table above, signifies that the digit is used in this register.	

13.2.5 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]		

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'.

For additional information, reference the "Register Definitions" on page 198 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]**	(Function Dependent)
	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
	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

(continued on next page)

13.2.5 DxBxxFN *(continued)*

	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

13.2.6 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]			

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’.

For additional information, reference the “Register Definitions” on page 198 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
		4h	Analog column comparator 0
		5h	Analog column comparator 1
		6h	Reserved
		7h	Reserved
		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
[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

13.2.7 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]t		OUTEN	Output Select[1:0]	

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'.

For additional information, reference the "Register Definitions" on page 198 in the Digital Blocks chapter.

Bit **Name** **Description**

[7:6] **AUXCLK**

00b	No sync	16 to 1 clock mux output
01b	Synchronize to SYSCLK	Output of 16 to 1 clock mux to SYSCLK
10b	Synchronize to SYSCLKX2	Output of 16 to 1 clock mux to SYSCLKX2
11b	SYSCLK	Directly connect SYSCLK to block clock input

[5] **AUXEN**

Aux IO Enable (function dependent)
All Functions except SPI Slave: Enable Auxiliary Output Driver
0 Disabled
1 Enabled

SPI Slave: Slave Select Input

Aux IO Enable, Aux IO Select[1:0] (function dependent, SPIS only)
Source for SS_ input
000b AUXDATA[0] (Row Input 0)
001b AUXDATA[1] (Row Input 1)
010b AUXDATA[2] (Row input 2)
011b AUXDATA[3] (Row input 3)
100b Force SS_ active
101b Reserved
110b Reserved
111b Reserved

[4:3] **AUX IO Select[1:0]**

All Functions except SPI Slave: Row Output Select for Auxiliary Function Output (function dependent)
00b Row Output 0
01b Row Output 1
10b Row Output 2
11b Row Output 3

[2] **OUTEN**

Enable Primary Function Output Driver
0 Disabled.
1 Enabled.

(continued on next page)

13.2.7 DxBxxOU *(continued)*

[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

13.2.8 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			
Bit Name					Acolumn1[1:0]			

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. For additional information, reference the “Register Definitions” on page 228 in the Analog Interface chapter.

Bit	Name	Description
[7:4]	**Reserved**	
[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]	**Reserved**	

13.2.9 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]		

For additional information, reference the “Register Definitions” on page 228 in the Analog Interface chapter.

Bit	Name	Description
[7]	**Reserved**	
[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 PSoC Block 00 001b Digital PSoC Block 01 010b Digital PSoC Block 02 011b Digital PSoC Block 03 100b Reserved 101b Reserved 110b Reserved 111b Reserved
[2:0]	**ACLK0[2:0]**	Select the clocking source for Analog Clock 0. 000b Digital PSoC Block 00 001b Digital PSoC Block 01 010b Digital PSoC Block 02 011b Digital PSoC Block 03 100b Reserved 101b Reserved 110b Reserved 111b Reserved

13.2.10 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
Bit Name	ACol1Mux		ABUF1EN0				Bypass	PWR

For additional information, reference the “Register Definitions” on page 61 in the Analog Output Drivers chapter or the “Register Definitions” on page 235 in the Analog Input Configuration chapter.

Bit	Name	Description
[7]	**ACol1Mux**	0 Set column 1 input to column 1 input mux output. (Selects among P0[6,4,2,0]) 1 Set column 1 input to column 0 input mux output. (Selects among P0[7,5,3,1])
[6]	**Reserved**	
[5]	**ABUF1EN0**	Enables the analog output buffer for Analog Column 1 (Pin P0[5]). 0 Disable analog output buffer. 1 Enable analog output buffer.
[4:2]	**Reserved**	
[1]	**Bypass**	Connects the positive input of the amplifier directly to its output. Amplifiers need to 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

13.2.11 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]		

For additional information, reference the “Register Definitions” on page 228 in the Analog Interface chapter.

Bit	Name	Description
[7:3]	**Reserved**	
[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 Reserved 101b Analog Column Comparator 1 110b Reserved 111b Reserved

13.2.12 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							
Bit Name	LUT1[3:0]							

For additional information, reference the “Register Definitions” on page 228 in the Analog Interface chapter.

Bit	Name	Description
[7:4]	**LUT1[3:0]**	Select one of 16 logic functions for the output of comparator bus 1. Note that B=0. 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
[3:0]	**Reserved**	

13.2.13 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

For additional information, reference the “Register Definitions” on page 179 in the Global Digital Interconnect chapter.

Bit	Name	Description	
[7]	**GIONOUT7**	0	GIO[7] does not drive GOO[7].
		1	GIO[7] drives its value on to GOO[7].
[6]	**GIONOUT6**	0	GIO[6] does not drive GOO[6].
		1	GIO[6] drives its value on to GOO[6].
[5]	**GIONOUT5**	0	GIO[5] does not drive GOO[5].
		1	GIO[5] drives its value on to GOO[5].
[4]	**GIONOUT4**	0	GIO[4] does not drive GOO[4].
		1	GIO[4] drives its value on to GOO[4].
[3]	**GIONOUT3**	0	GIO[3] does not drive GOO[3].
		1	GIO[3] drives its value on to GOO[3].
[2]	**GIONOUT2**	0	GIO[2] does not drive GOO[2].
		1	GIO[2] drives its value on to GOO[2].
[1]	**GIONOUT1**	0	GIO[1] does not drive GOO[1].
		1	GIO[1] drives its value on to GOO[1].
[0]	**GIONOUT0**	0	GIO[0] does not drive GOO[0].
		1	GIO[0] drives its value on to GOO[0].

13.2.14 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

For additional information, reference the “Register Definitions” on page 179 in the Global Digital Interconnect chapter.

Bit	Name	Description	
[7]	**GIENOUT7**	0	GIE[7] does not drive GOE[7].
		1	GIE[7] drives its value on to GOE [7].
[6]	**GIENOUT6**	0	GIE[6] does not drive GOE[6].
		1	GIE[6] drives its value on to GOE [6].
[5]	**GIENOUT5**	0	GIE[5] does not drive GOE[5].
		1	GIE[5] drives its value on to GOE [5].
[4]	**GIENOUT4**	0	GIE[4] does not drive GOE[4].
		1	GIE[4] drives its value on to GOE [4].
[3]	**GIENOUT3**	0	GIE[3] does not drive GOE[3].
		1	GIE[3] drives its value on to GOE [3].
[2]	**GIENOUT2**	0	GIE[2] does not drive GOE[2].
		1	GIE[2] drives its value on to GOE [2].
[1]	**GIENOUT1**	0	GIE[1] does not drive GOE[1].
		1	GIE[1] drives its value on to GOE [1].
[0]	**GIENOUT0**	0	GIE[0] does not drive GOE[0].
		1	GIE[0] drives its value on to GOE [0].

13.2.15 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

For additional information, reference the “Register Definitions” on page 179 in the Global Digital Interconnect chapter.

Bit	Name	Description	
[7]	**GOOUTIN7**	0	GOO[7] does not drive GIO[7].
		1	GOO[7] drives its value on to GIO[7].
[6]	**GOOUTIN6**	0	GOO[6] does not drive GIO[6].
		1	GOO[6] drives its value on to GIO[6].
[5]	**GOOUTIN5**	0	GOO[5] does not drive GIO[5].
		1	GOO[5] drives its value on to GIO[5].
[4]	**GOOUTIN4**	0	GOO[4] does not drive GIO[4].
		1	GOO[4] drives its value on to GIO[4].
[3]	**GOOUTIN3**	0	GOO[3] does not drive GIO[3].
		1	GOO[3] drives its value on to GIO[3].
[2]	**GOOUTIN2**	0	GOO[2] does not drive GIO[2].
		1	GOO[2] drives its value on to GIO[2].
[1]	**GOOUTIN1**	0	GOO[1] does not drive GIO[1].
		1	GOO[1] drives its value on to GIO[1].
[0]	**GOOUTIN0**	0	GOO[0] does not drive GIO[0].
		1	GOO[0] drives its value on to GIO[0].

13.2.16 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

For additional information, reference the “Register Definitions” on page 179 in the Global Digital Interconnect chapter.

Bit	Name	Description	
[7]	**GOEUTIN7**	0	GOE[7] does not drive GIE[7].
		1	GOE[7] drives its value on to GIE[7].
[6]	**GOEUTIN6**	0	GOE[6] does not drive GIE[6].
		1	GOE[6] drives its value on to GIE[6].
[5]	**GOEUTIN5**	0	GOE[5] does not drive GIE[5].
		1	GOE[5] drives its value on to GIE[5].
[4]	**GOEUTIN4**	0	GOE[4] does not drive GIE[4].
		1	GOE[4] drives its value on to GIE[4].
[3]	**GOEUTIN3**	0	GOE[3] does not drive GIE[3].
		1	GOE[3] drives its value on to GIE[3].
[2]	**GOEUTIN2**	0	GOE[2] does not drive GIE[2].
		1	GOE[2] drives its value on to GIE[2].
[1]	**GOEUTIN1**	0	GOE[1] does not drive GIE[1].
		1	GOE[1] drives its value on to GIE[1].
[0]	**GOEUTIN0**	0	GOE[0] does not drive GIE[0].
		1	GOE[0] drives its value on to GIE[0].

13.2.17 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]	

For additional information, reference the “Register Definitions” on page 256 in the Digital Clocks chapter.

Bit	Name	Description
[7:2]	**Reserved**	
[1:0]	**VC3 Input Select[1:0]**	Selects the clocking source for the VC3 Clock Divider. 00b SYSCLK 01b VC1 10b VC2 11b SYSCLKX2

13.2.18 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]							

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 will be one eighth the input frequency. For additional information, reference the “Register Definitions” on page 256 in the Digital Clocks chapter.

Bit	Name	Description
[7:0]	**VC3 Divider[7:0]**	Reference 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

13.2.19 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]		

For additional information, reference the “Register Definitions” on page 256 in the Digital Clocks chapter.

Bit	Name	Description		
[7]	**32k Select**	0	Internal low precision 32 kHz oscillator	
		1	External Crystal 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]**		**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

13.2.20 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]			

For additional information, reference the “Register Definitions” on page 256 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

13.2.21 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	IMODIS	SYSCLKX2DIS

Note that in OCD mode (OCDM=1), bits [1:0] have no effect. For additional information, reference the “Register Definitions” on page 256 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.
[6:3]	**Reserved**	
[2]	**EXTCLKEN**	External clock mode enable. 0 Disabled. Operate from internal main oscillator. 1 Enabled. Operate from clock supplied at port P1[4].
[1]	**IMODIS**	Internal oscillator disable. Can be set to save power when using an external clock on P1[4]. 0 Enabled. Internal oscillator enabled. 1 Disabled, if SYSCLKX2DIS is set (1).
[0]	**SYSCLKX2DIS**	48 MHz clock source disable. 0 Enabled. If enabled, the 48 MHz clock is forced on. 1 Disabled for power reduction.

13.2.22 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]		

For additional information, reference the “Register Definitions” on page 277 in the POR and LVD chapter.

Bit	Name	Description
[7:6]	**Reserved**	
[5:4]	**PORLEV[1:0]**	Sets the POR level. 00b POR level for 3V operation 01b POR level for 4.5V operation 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 LVDp levels per “DC POR and LVD Specifications” on page 294. 000b Typical LVD setting for operation above 3.0V. 001b 010b 011b 100b 101b Typical LVD setting for operation above 4.75V (no switch mode pump). 110b 111b Typical pump setting for 5V operation.

13.2.23 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
Bit Name							LVD	PPOR

For additional information, reference the “Register Definitions” on page 277 in the POR and LVD chapter.

Bit	Name	Description
[7:2]	**Reserved**	
[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.

13.2.24 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]							

For additional information, reference the “Register Definitions” on page 63 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 in this register should not be changed***. 00h Lowest frequency setting 01h ... ... 7Fh 80h Design center setting 81h ... ... FEh FFh Highest frequency setting

13.2.25 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 : 00			
Bit Name			Bias Trim[1:0]		Freq Trim[3:0]			

It is strongly recommended that the user not alter this register's value. The trim bits are set to factory specifications and should not be changed.

For additional information, reference the "Register Definitions" on page 65 in the Internal Low Speed Oscillator chapter.

Bit	Name	Description
[7:6]	**Reserved**	
[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 in this register should not be changed.*** 00b Medium bias 01b Maximum bias (recommended) 10b Minimum bias 11b Reserved
[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 in this register should not be changed.***

13.2.26 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 : 1		RW : 8			
Bit Name			TC[1:0]		V[3:0]			

For additional information, reference the “Register Definitions” on page 279 in the Internal Voltage Reference chapter.

Bit	Name	Description
[7:6]	**Reserved**	
[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 its should not be changed.***

13.2.27 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]							

The value of this register is used to trim the External Crystal Oscillator. Its value is set to the device specific, best value during boot. The value in this register should not be changed. For additional information, reference the “Register Definitions” on page 68 in the 32 kHz Crystal Oscillator chapter.

Bit	Name	Description
[7:6]	**PSSDC[1:0]**	Sleep duty cycle. Controls the ratios (in numbers of 32 kHz clock periods) of “on” time versus “off” time for PORLVD, Bandgap reference, and pspump. 00b 1 / 128 01b 1 / 512 10b 1 / 32 11b 1 / 8
[5:0]	**Reserved**	

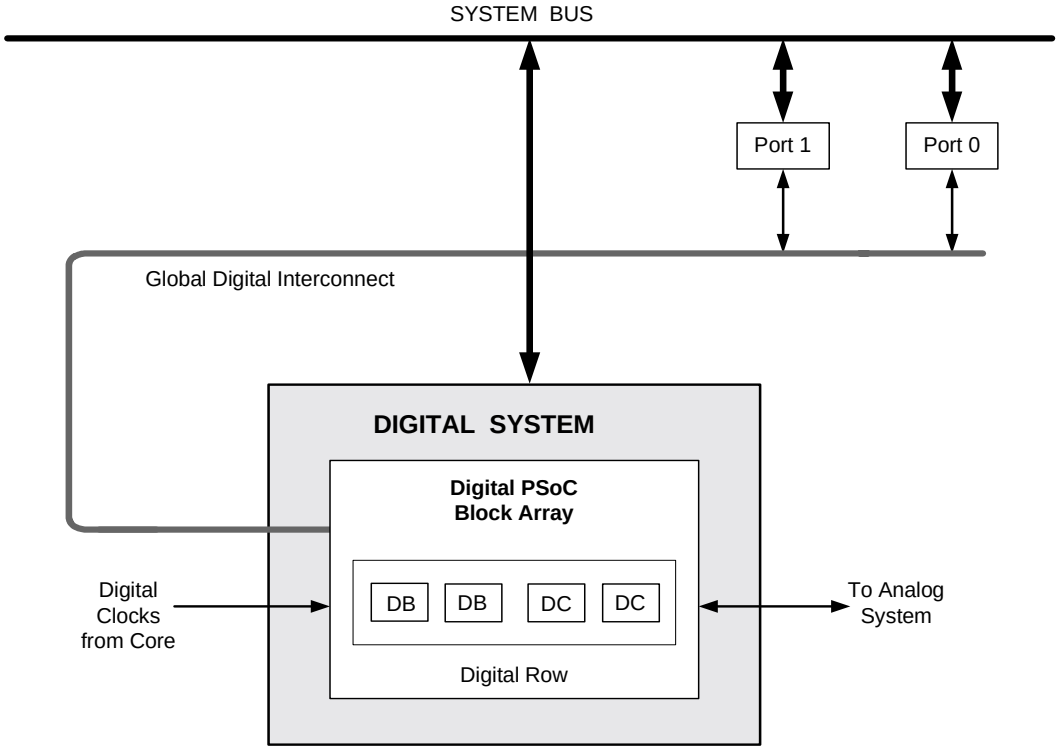
SECTION D DIGITAL SYSTEM

The Digital System section discusses the digital components of the PSoC device and the registers associated with those components. This section encompasses the following chapters:

- Global Digital Interconnect (GDI) on page 177
- Array Digital Interconnect (ADI) on page 181
- Row Digital Interconnect (RDI) on page 183
- Digital Blocks on page 189

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.



PSoC Digital System Block Diagram

Digital Register Summary

The table below lists all the PSoC registers in the digital system.

Summary Table of the Digital Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
GLOBAL DIGITAL INTERCONNECT (GDI) REGISTERS										
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										
x,B0h	RDI0RI	RI3[1:0]		RI2[1:0]		RI1[1:0]		RI0[1:0]		RW : 00
x,B1h	RDI0SYN					RI3SYN	RI2SYN	RI1SYN	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										
Data and Control Registers										
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
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

LEGEND
#: Access is bit specific. Refer to register detail for additional information.
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.

14. Global Digital Interconnect (GDI)

This chapter discusses the Global Digital Interconnect (GDI) and its associated registers. GDI is the most general level of interconnect configuration available in the PSoC Mixed Signal Arrays.

Table 14-1. Global Digital Interconnect (GDI) Registers

Address	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
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

Global Digital Interconnect (GDI) consists of four 8-bit busses. Two of the busses are input busses, which allow signals to pass from the device pins to the core of the chip. These busses are called Global Input Odd (GIO[7:0]) and Global Input Even (GIE[7:0]). The other two busses are output busses and allow signals to pass from the core of the chip 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. Busses with odd in their name connect to all odd numbered ports and busses with even in their name connect to all even numbered ports. Note that the word odd or even in the bus name refers to ports and not pins.

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 its output from a global output. Currently there are two types of core signals connected to the global busses. The digital blocks, which may be a source or a destination for a global net, and system clocks, which may only drive global nets.

14.1 Architectural Description

The primary goal, of the architectural block diagram that follows, is to communicate the relationship between global busses (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 figure. 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.

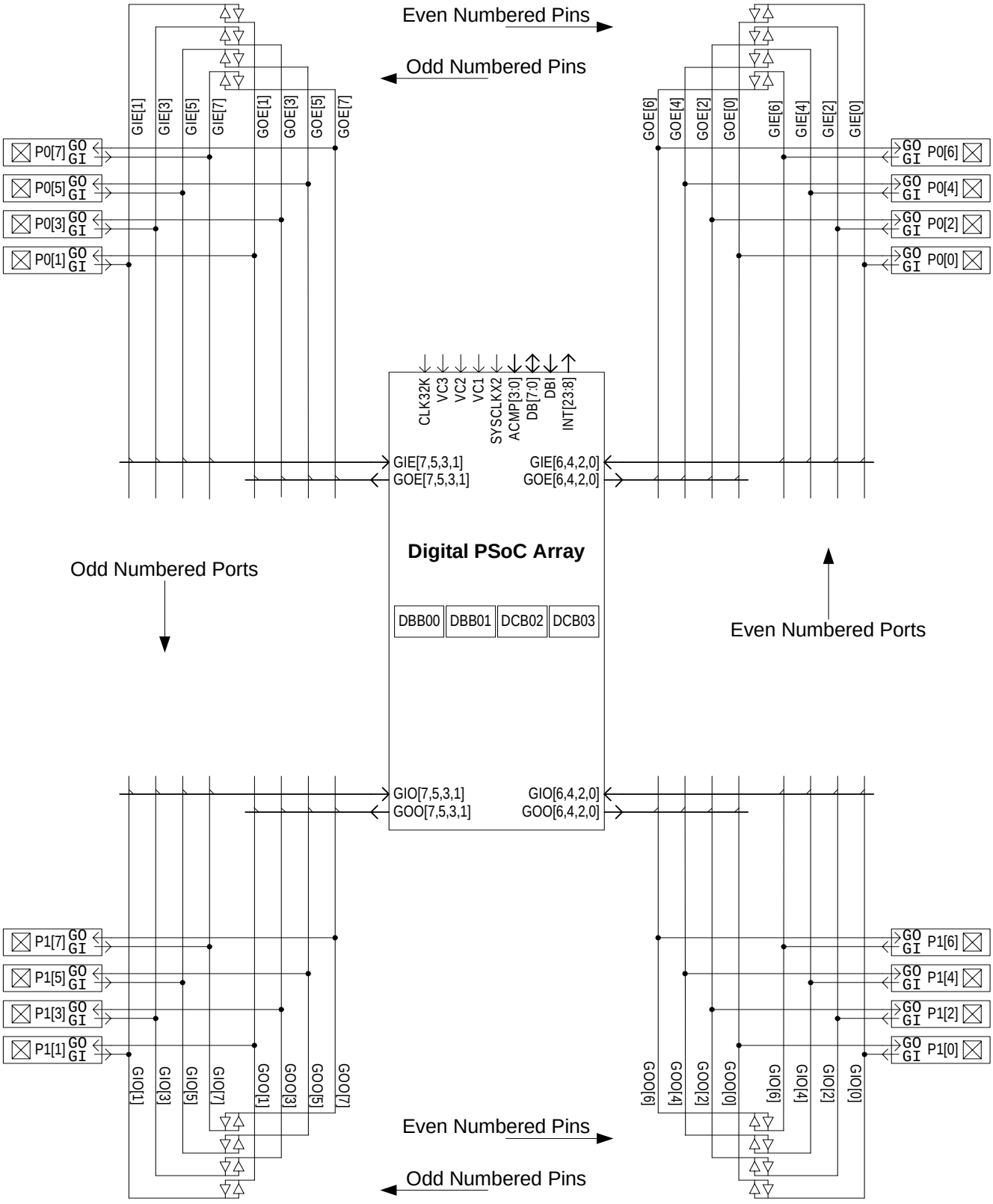


Figure 14-1. Global Interconnect Block Diagram

14.2 Register Definitions

14.2.1 GDI_O_IN and GDI_E_IN Registers

The PSoC device has a configurable Global Digital Interconnect (GDI). 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,

$$GIE[7] \rightarrow GOE[7]$$

There are a total of 16-bits that control the ability of global inputs to drive global outputs. These bits are in the GDI_O_IN and GDI_E_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, reference the GDI_O_IN register on page 159 and the GDI_E_IN register on page 160.

14.2.2 GDI_O_OU and GDI_E_OU Registers

Additional configuration bits are offered in the GDI_x_OU registers that allow a global output to drive its corresponding global input. For example,

$$GOE[7] \rightarrow GIE[7]$$

There are a total of 16 bits that control the ability of global outputs to drive global inputs. These bits are in the GDI_O_OU and GDI_E_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]

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.

$$GOE[7] \rightarrow GIO[7]$$
$$GOE[0] \rightarrow GIE[7]$$

For additional information, reference the GDI_O_OU register on page 161 and the GDI_E_OU register on page 162.

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 183. The digital PSoC array does not have any configurable interconnect; therefore, there are no associated registers.

15.1 Architectural Description

The Array Digital Interconnect (ADI) 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.

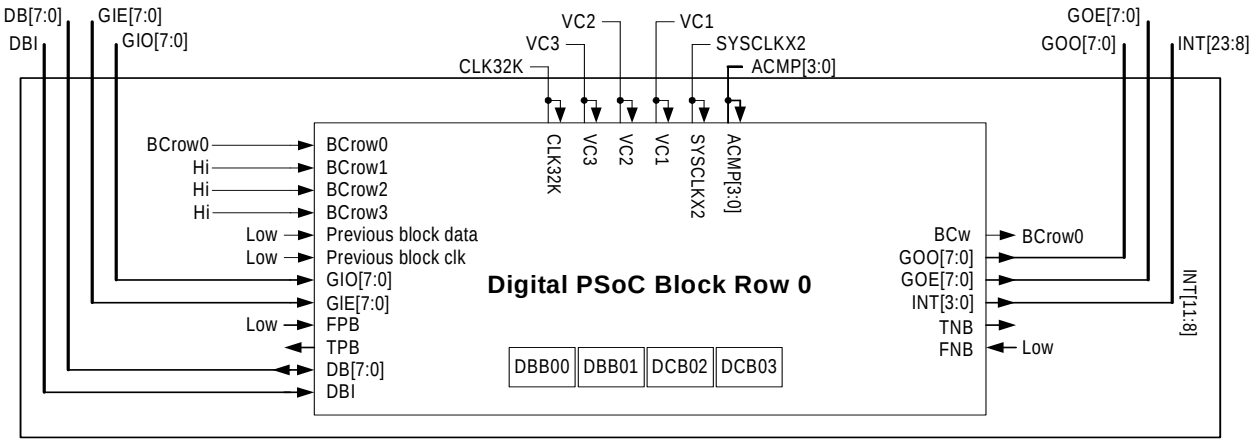


Figure 15-1. Digital PSoC Block Array Structure

The different members of the PSoC family have varying numbers of digital PSoC blocks in the digital array. These blocks are arranged into rows and the ADI provides a regular interconnect architecture between the Global Digital Interconnect (GDI) and the Row Digital Interconnect (RDI), regardless of the number of rows available in a particular device. 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 in the following bulleted list.

- Register Address: Clearly 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 will have 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 circle. The first row in the array will have its previous chaining inputs tied low. If there is a second row in the array, the next chaining outputs will be connected to the next row. For the last row in the array, the next inputs are tied low.

In Figure 15-1, the detailed view of a Digital PSoC block row has been replaced by a box labeled “Digital PSoC Block Row.” The rest of this figure illustrates how all rows are connected to the same globals, clocks, and so on. The figure also illustrates how the broadcast clock nets (BCxxxx) are connected between rows.

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. Information about individual digital PSoC blocks is covered in the Digital Blocks chapter on page 189.

Table 16-1. Digital PSoC Row Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,xxh	RDIxRI	RI3[1:0]		RI2[1:0]		RI1[1:0]		RI0[1:0]		RW : 00
x,xxh	RDIxSYN					RI3SYN	RI2SYN	RI1SYN	RI0SYN	RW : 00
x,xxh	RDIxIS			BCSEL[1:0]		IS3	IS2	IS1	IS0	RW : 00
x,xxh	RDIxLT0	LUT1[3:0]				LUT0[3:0]				RW : 00
x,xxh	RDIxLT1	LUT3[3:0]				LUT2[3:0]				RW : 00
x,xxh	RDIxRO0	GOO5EN	GOO1EN	GOE5EN	GOE1EN	GOO4EN	GOO0EN	GOE4EN	GOE0EN	RW : 00
x,xxh	RDIxRO1	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.
xx: An “x” after the comma in the address field indicates that there are multiple instances of the register. For a detailed address listing of these registers, refer to the “Digital Register Summary” on page 176.

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 busses. What follows is a discussion of the signals that are configurable by way of the registers listed in Table 16-1.

16.1 Architectural Description

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 through the bold line in Figure 16-1 are shown at the next level of hierarchy (Figure 16-2 on page 185).

In Figure 16-2, the detailed view of the four PSoC block grouping, as seen in Figure 16-1, 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. The next level of hierarchy is shown in Figure 16-1.

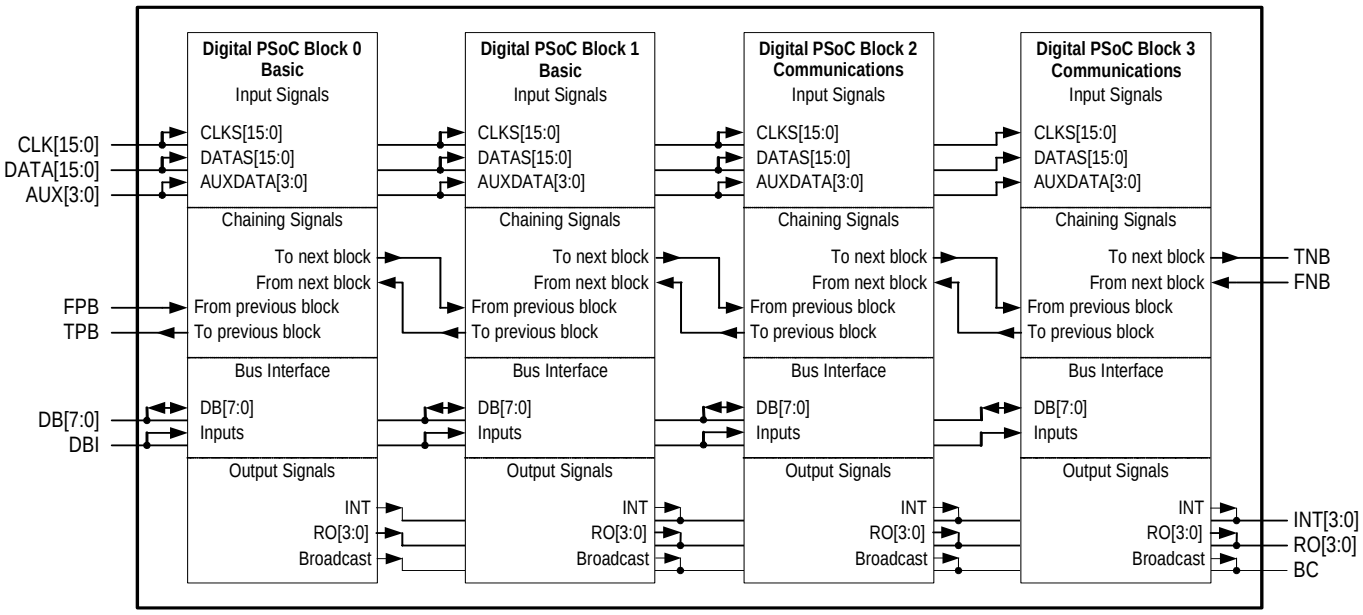


Figure 16-1. Detailed View of Four PSoC Block Groupings

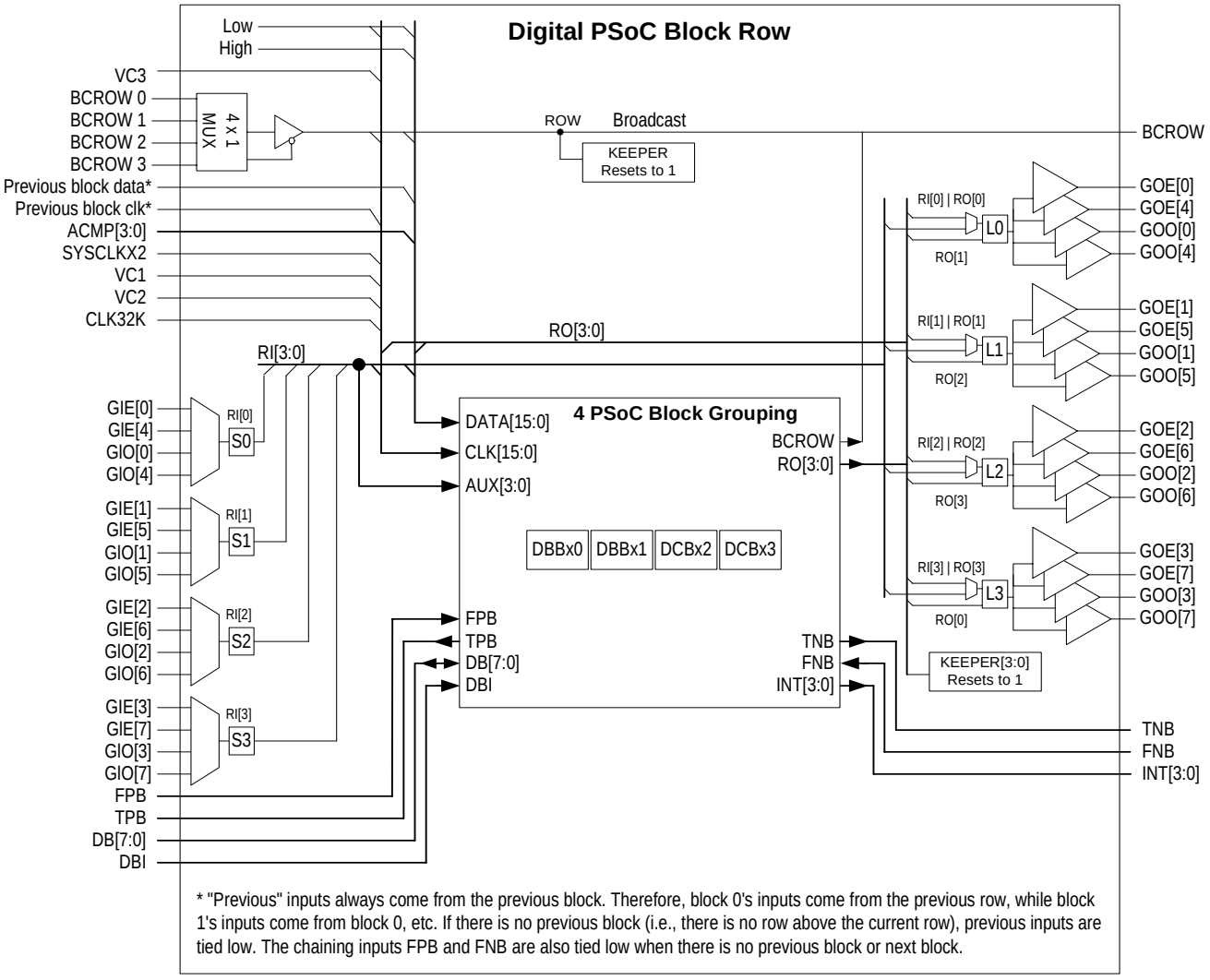


Figure 16-2. Digital PSoC Block Row Structure

16.2 Register Definitions

The only configurable inputs to a Digital PSoC Block Row are the Global Input Even and Global Input Odd 8-bit busses. The only configurable outputs from the Digital PSoC Block Row are the Global Output Even and Global Output Odd 8-bit busses. Figure 16-2 on page 185 illustrates the relationships between global signals and row signals.

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 multiplexers are Row Inputs (RI[x]). Because there are four 4-to-1 multiplexers, each with a unique set of inputs, a row has access to every global input line in a PSoC device.

For a complete list of the Digital Row registers showing their addresses and bit names, reference the "Digital Register Summary" on page 176.

16.2.1 RDIxRI Register

The select bits used to control the four multiplexers are located in the RDIxRI register, where "x" denotes a place holder for the row index. Table 16-2 lists the meaning for each multiplexer's four possible settings.

Table 16-2. RDIxRI Register

RI0[1:0]	0h: GIE[0] 1h: GIE[4] 2h: GIO[0] 3h: GIO[4]
RI1[1:0]	0h: GIE[1] 1h: GIE[5] 2h: GIO[1] 3h: GIO[5]
RI2[1:0]	0h: GIE[2] 1h: GIE[6] 2h: GIO[2] 3h: GIO[6]
RI3[1:0]	0h: GIE[3] 1h: GIE[7] 2h: GIO[3] 3h: GIO[7]

The RDIxRI and RDIxSYN registers are the only two registers that affect Digital PSoC Row input signals. All other registers are related to output signal configuration. The options, with respect to output signals, are discussed below.

For additional information, reference the RDIxRI register on page 118.

16.2.2 RDIxSYN Register

By default, each row input is double synchronized to the SYSCLK (system clock). However, a user may choose to disable this synchronization by setting the appropriate RIxSYN bit in the RDIxSYN register. Table 16-3 lists the bit meanings for each implemented bit of the RDIxSYN register.

Table 16-3. RDIxSYN Register

RI3SYN	0: Row input 3 in synchronized to 24 MHz system clock 1: Row input 3 is passed without synchronization
RI2SYN	0: Row input 2 in synchronized to 24 MHz system clock 1: Row input 2 is passed without synchronization
RI1SYN	0: Row input 1 in synchronized to 24 MHz system clock 1: Row input 1 is passed without synchronization
RI0SYN	0: Row input 0 in synchronized to 24 MHz system clock 1: Row input 0 is passed without synchronization

The RDIxRI and RDIxSYN registers are the only two registers that affect Digital PSoC Row input signals. All other registers are related to output signal configuration. The options, with respect to output signals, are discussed below.

For additional information, reference the RDIxSYN register on page 119.

16.2.3 RDIxIS Register

As mentioned previously, 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. The configurable LUT input (Input A) chooses between a single row output and a single row input. Table 16-4 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], i.e., LUT0's fixed input is RO[1], LUT1's fixed input is RO[2],..., and LUT3's fixed input is RO[0].

Table 16-4. RDIxIS Register Bits

BCSEL[1:0]	0: Row 0 driver local row broadcast net* 1: Row 1 driver local row broadcast net* 2: Row 2 driver local row broadcast net* 3: Row 3 driver local row broadcast net*
IS3	0: The 'A' input of LUT 3 is RO[3] 1: The 'A' input of LUT 3 is RI[3]
IS2	0: The 'A' input of LUT 2 is RO[2] 1: The 'A' input of LUT 2 is RI[2]
IS1	0:The 'A' input of LUT 1 is RO[1] 1: The 'A' input of LUT 1 is RI[1]
IS0	0: The 'A' input of LUT 0 is RO[0] 1: The 'A' input of LUT 0 is RI[0]
* When the BCSELL 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. * If the row is not present in the part, the selection provides a Logic 1 value.	

For additional information, reference the RDIxIS register on page 120.

16.2.4 RDIxLTx Registers

The outputs from a Digital PSoC Row are a bit more complicated than the inputs. Figure 16-2 on page 185 illustrates the output circuitry in a Digital PSoC Row. Notice in the figure a block labeled “Lx.” This block represents a 2-input lookup 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 will determine the value that may be driven on to the Global Output Even and Global Output Odd busses. Table 16-5 lists the relationship between a lookup 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 16-5 lists seven examples of the relationship between the LUT’s output column for a truth table and the LUTx[3:0] configuration bits.

Table 16-5. 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 16-6. RDIxLTx Register

LUTx[3:0]	0h: 0000: FALSE 1h: 0001: A .AND. $\underline{B}$ 2h: 0010: A .AND. $\overline{B}$ 3h: 0011: $\underline{A}$ 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: $\underline{A}$ .XNOR. B Ah: 1010: $\overline{B}$ Bh: 1011: $\underline{A}$ .OR. $\overline{B}$ Ch: 1100: $\underline{A}$ Dh: 1101: $\overline{A}$ .OR. B Eh: 1110: A. NAND. B Fh: 1111: TRUE

For additional information, reference the RDIxLT0 register on page 121 and the RDIxLT1 register on page 122.

16.2.5 RDIxROx Registers

The final configuration bits for outputs from Digital PSoC Rows are in the two RDIxROx registers. These registers hold the 16 bits and 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. This means that any row can drive any global output. Keep in mind that tri-state drivers are being used to drive the global output lines; therefore, it is possible for a part, with more than one Digital PSoC Row, to have multiple drivers on a single global output line. It is the user’s responsibility to ensure that the part is not configured with multiple drivers on any of the global output lines.

For additional information, reference the RDIxRO0 register on page 123 and the RDIxRO1 register on page 124.

16.3 Timing Diagram

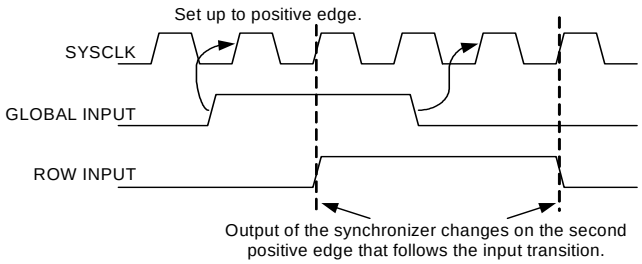


Figure 16-3. Optional Row Input Synchronization to SYSCLK

17. Digital Blocks

This chapter presents the Digital Blocks and their associated registers. It covers the configuration and use of the Digital PSoC blocks.

Table 17-1. Digital PSoC Block Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
Data and Control Registers										
0,xxh	DxBxxDR0	Data[7:0]								# : 00
0,xxh	DxBxxDR1	Data[7:0]								W : 00
0,xxh	DxBxxDR2	Data[7:0]								# : 00
0,xxh	DxBxxCR0	Function control/status bits for selected function[6:0]							Enable	# : 00
0,E1h	INT_MSK1					DCB03	DCB02	DBB01	DBB00	RW : 00
Configuration Registers										
1,xxh	DxBxxFN	Data Invert	BCEN	End/Single	Mode[1:0]		Function[2:0]			RW : 00
1,xxh	DxBxxIN	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,xxh	DxBxxOU	AUXCLK		AUXEN	AUX IO Select[1:0]		OUTEN	Output Select[1:0]		RW : 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 176.

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 block 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 summarized in Table 17-1. Digital PSoC block function registers end with FN. The individual bit settings for a blocks function register are listed in Table 17-15 on page 203. The input register’s name ends with IN and its bit meanings are listed in Table 17-15 on page 203. Finally the blocks outputs are controlled by the output register which always ends with 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 dependant and are discussed with each functions 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 can have its own interrupt service routine.

17.1 Architectural Description

At the top level, the main components of the digital block are the data path, input multiplexers, output de-multiplexers, CRCPRS tri-state buses, system bus interface, configuration registers, and chaining signals (see Figure 17-2).

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 16-1 input multiplexer.

In addition, there is a 4-1 multiplexer that 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 multiplexer are intended to be a selection of GPIO inputs (row inputs).

Note 1 If the input data source for a given block comes from another block, the destination block must be enabled prior to the source block being enabled.

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), pin 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 chip 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 (e.g., divided down) from SYSCLK, re-synchronize to SYSCLK at the digital block. Most chip clocks are in this category. For example, VC1 and VC2, and the output of other blocks clocked by VC1 and VC2 or SYSCLK (setting 01 in AUXCLK).
2. If the clock input is derived from SYSCLKX2, re-synchronize to SYSCLKX2. For example, VC3 clocked by SYSCLKX2, or other digital blocks clocked by SYSCLKX2 (setting 10 in AUXCLK).
3. Choose direct SYSCLK (setting 11 in AUXCLK).
4. Choose direct SYSCLKX2 (select SYSCLKX2 in the Clock Input field of the DxBxxIN register).
5. Bypass synchronization. This should be a very rare selection. Because if clocks are not synchronized, they may fail setup 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).

The following notes enumerate configurations that are not allowed, although the hardware does not prevent them. The summary of these notes is that 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 1 When VC1 is configured to divide by one, choosing an input clock of VC1 is not allowed. This configuration produces a clock frequency that is equal to SYSCLK; therefore, SYSCLK direct should be used by setting the AUXCLK bits in DxBxOU to 11b.

Note 2 When both VC2 and VC1 are configured to divide by one, choosing an input clock of VC2 is not allowed. This configuration produces a clock frequency that is equal to SYSCLK; therefore, SYSCLK direct should be used by setting the AUXCLK bits in DxBxOU to 11b.

Note 3 When VC3 is configured to divide by one with a source clock of SYSCLK, choosing an input clock of VC3 is not allowed. This configuration produces a clock frequency that is equal to SYSCLK. There are two other VC3 configurations to avoid that will result in an output frequency equal to SYSCLK. The first is when VC3 is configured to divide by one with a source clock of VC1 divide by one. The second is when VC3 is configured to divide by one with a source clock of VC2 divide by one and VC1 is also configured to divide by one. All of these configurations result in a VC3 frequency equal to SYSCLK and this is not allowed. When a frequency equal to SYSCLK is desired, SYSCLK direct should be used by setting the AUXCLK bits in DxBxOU to 11b.

Note 4 When VC3 is configured to divide by one with a source clock of SYSCLKX2, choosing an input clock of VC3 is not allowed. This configuration produces a clock frequency that is equal to SYSCLKX2. When a frequency equal to SYSCLKX2 is desired, SYSCLKX2 should be selected by setting the Clock Input bits of the DxBxxIN register to 4h and the AUXCLK bits of DxBxOU to 00b.

All of these issues have been addressed in the actual clock resynchronizer, illustrated in Figure 17-1.

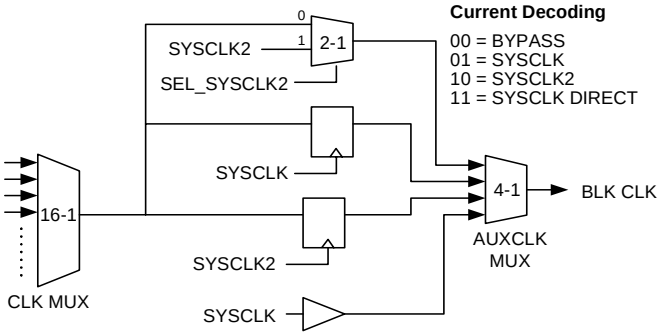


Figure 17-1. Input Clock Resynchronization

Table 17-2: AUXCLK Bit Selections

Code	Description	Usage
00	Bypass	Use this setting only for asynchronous inputs. Also used when SYSCLK2 (48 MHz) is selected.
01	Resync 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	Resync to SYSCLK2 (48 MHz)	Use this setting for any SYSCLK2 based clock. VC3 driven by SYSCLK2, digital blocks with SYSCLK2 based source clocks, broadcast bus with source based on SYSCLK2, row input and row outputs with source based on SYSCLK2.
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.

17.1.3 Output De-Multiplexers

Most functions have two outputs: a primary and an auxiliary output. Each of these outputs may be driven onto the row output bus. Each de-multiplexer is implemented with four tri-state drivers. There are two bits to select one of the four and an additional bit to enable the selected driver.

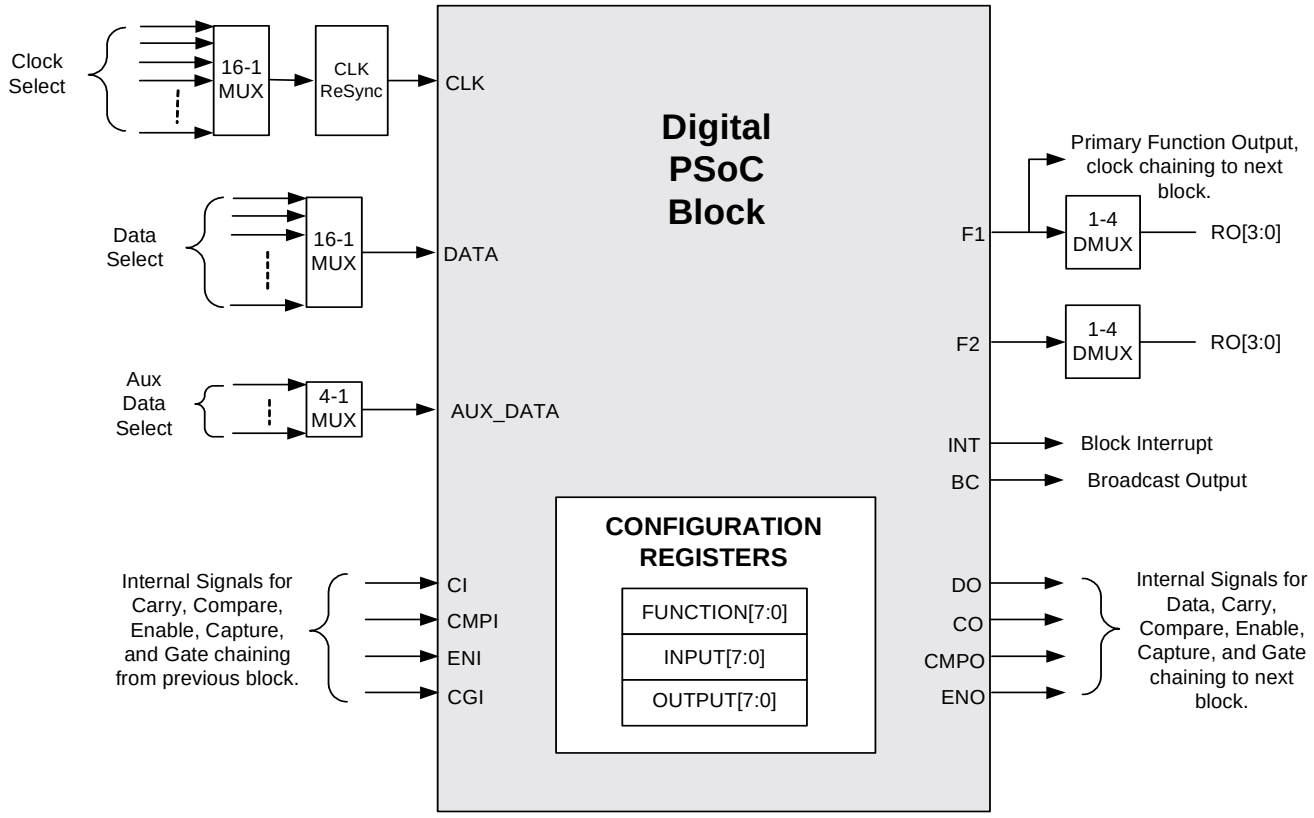


Figure 17-2. Digital Blocks Top-Level Block Diagram

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, will automatically make the proper selections of the chaining signals, to ensure the correct information flow between blocks.

17.1.5 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.

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 will 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 will overwrite 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.

17.1.5.1 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.

17.1.5.2 Block Interrupt

The Timer block has a selection of three interrupt sources. Interrupt on Terminal Count (TC) and Compare may be selected in Mode bit 0 of the Function register. The 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 closing of the DR2 latch on capture.

17.1.6 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 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.

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 will only count and reload 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.

17.1.6.1 Usability Exceptions

The following are usability exceptions for the Counter function.

1. DR0 may only be read (to transfer DR0 data to DR2) when the block is disabled.

17.1.6.2 Block Interrupt

The Counter block has a selection of three interrupt sources. Interrupt on Terminal Count and 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.

17.1.7 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 will make a two output PWM with guaranteed non-overlapping outputs. An active signal on the “Kill” input will disable 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 Timer, when the Dead Band is disabled and a period value is written into DR1, the period value is also loaded into DR0.

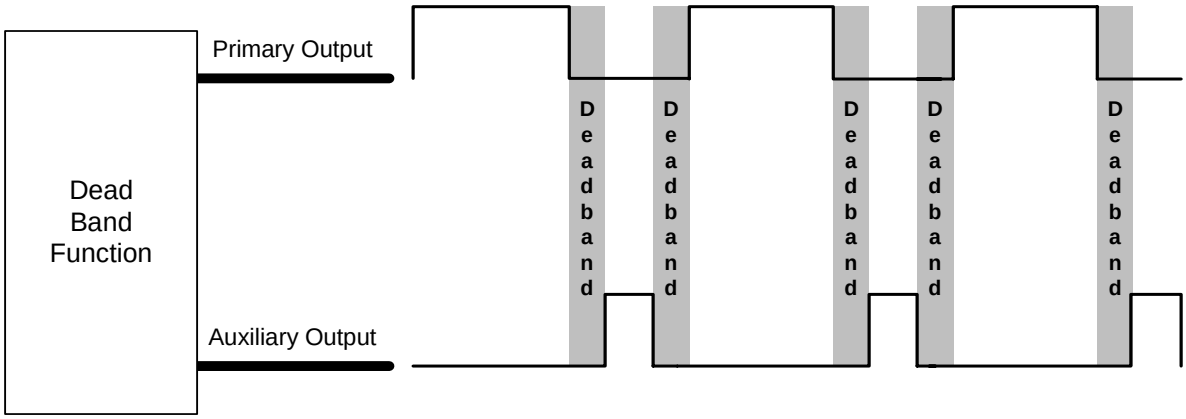


Figure 17-3. Dead Band Functional Overview

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 16-1 clock input multiplexer. In the Dead Band case, this signal (PREVF1) 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 stand-alone 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, the output is forced low immediately. Mode bits are encoded for Kill options and are detailed in the following table.

Table 17-3. 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'.

When asserted, the KILL signal functions as an immediate disable of the outputs (forced to logic '0'). There are three optional modes for resuming operation after the KILL. These are described in detail in the following section.

Note that the Dead Band function may not be chained.

17.1.7.1 Usability Exceptions

The following are usability exceptions for the Dead Band function.

1. Programming a dead band period value of 00h is not supported. The block output is undefined under this condition.
2. If the period, of either the high time or the low time of the reference input, is less than the programmed dead time that associated output phase will be held low.
3. DR0 may only be read (to transfer DR0 data to DR2) when the block is disabled.

17.1.7.2 Block Interrupt

The Dead Band 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.

17.1.8 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. 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 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 and the interrupt.

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 interrupt output. Similarly, the CLK input is passed transparently to the auxiliary output.

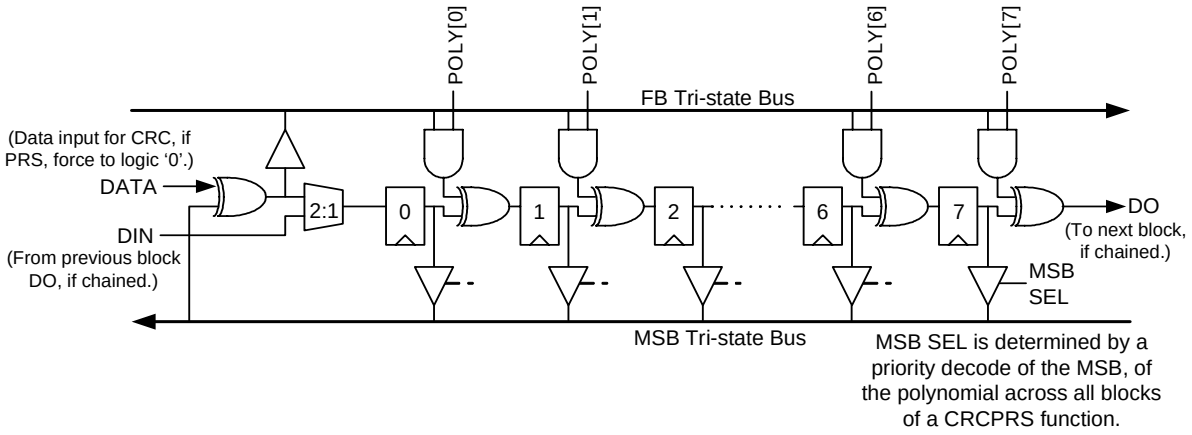


Figure 17-4. CRCPRS LFSR Structure

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

$$X_0 = 1$$ **Equation 1**

As an example, the CRC-CCIT 16-bit polynomial is:

$$CRC - CCIT = X_{16} + X_{12} + X_5 + 1$$ **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 (pseudo random sequence) 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.

Converting a Polynomial Spec to a Modular Spec

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 would be [7,7-2,7-4,7-6]m or [7,5,3,2]m. 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 [7,5,3,2] would correspond to 01010110 or 56h.

In both the CRC and PRS cases, an appropriate Seed value should be selected that is greater than or equal in bit length.

17.1.8.1 Usability Exceptions

The following are usability exceptions for the CRCPRS function.

1. The polynomial register must only be written when the block is disabled.

17.1.8.2 Block Interrupt

The CRCPRS has one fixed interrupt source, which is the compare auxiliary output.

17.1.9 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 setup and hold requirements.

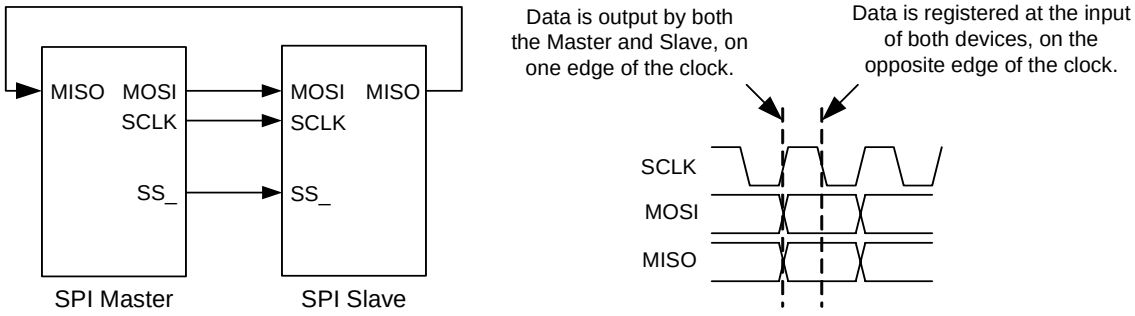


Figure 17-5. Basic SPI Configuration

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 is generating the clocking and initiating 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 are transmitting and receiving simultaneously. If the Master is only sending 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.

17.1.9.1 SPI Protocol Register Definitions

The SPI Protocol register definitions are located in Table 17-4. The use of the SS_ signal varies according to the capability of the Slave device.

Table 17-4. SPI Protocol Register Descriptions

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 will receive the input data and drive the MISO line.

17.1.10 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.

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 fifty percent duty cycle. This clock is gated, to provide the SCLK output on the auxiliary output, during byte transmissions.

There are four control bits and four status bits in the Control register that provide for host 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 chip dependent and, if required, must be implemented in firmware.

This SPIM function may not be chained.

17.1.10.1 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.

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.

17.1.11 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.

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, has been 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.

Since there are no internal clocks used in the SPIS, it may be clocked asynchronously (if input synchronization is turned off). In this case, synchronization between the CPU and the SPIS block can be accomplished with polling and/or interrupts.

There are four control bits and four status bits in the Control register that provide for host 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 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.

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 Hi-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.

17.1.11.1 Usability Exceptions

The following are usability exceptions for the SPI Slave function.

1. The SS_ input must be synchronized, but the MOSI and SCLK inputs may be synchronized or not. Unsynchronized data and clock inputs reduce the latency through the block and thus allow an SPI system to run at a slightly higher clock rate.

17.1.11.2 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.

17.1.12 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.

Unlike SPI, which has no output latency, the TXD output has one cycle of latency. This is because a multiplexer at the output must select which bits to shift out: the shift register data, framing bits, parity, or mark bits. The output of this multiplexer is registered to unglitch it. 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 setup 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.

The Transmitter function may not be chained.

17.1.12.1 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 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.

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 clock is generated, subsequently sampling the RXD input at the center of the bit time. Every succeeding 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 to be 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.

17.1.13.1 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, etc., so that RX Reg Full status bit is cleared; otherwise, no subsequent interrupts are generated.

17.2 Register Definitions

The Digital Block registers in this chapter are organized by function, as presented in Table 17-5. To reference timing diagrams associated with the digital block registers, see "Timing Diagrams" on page 204. For a complete list of the Digital Block registers showing their addresses and bit names, reference the "Digital Register Summary" on page 176.

Table 17-5. Digital Block Register Definitions

	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 access and status bits are Read-Only access.

Data and Control Registers

17.2.1 DxBxxDRx Registers

The Data and Control registers 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 tables by function type.

There are two banks of registers associated with the PSoC device. Bank 0 encompasses the user registers for the device and Bank 1 encompasses the configuration registers for the device. Both are defined below. Reference the "Bank 0 Registers" on page 86 and the "Bank 1 Registers" on page 145 for more information.

For additional information, reference the Register Details chapter for the following registers:

- DxBxxDR0 register on page 90.
- DxBxxDR1 register on page 91.
- DxBxxDR2 register on page 92.

17.2.1.1 Timer Register Definitions

Bank 0: There are three 8-bit data registers and a 3-bit control register. Table 17-6 explains the meaning of these registers in the context of timer operation.

Bank 1: The mode bits in the Function register are block type specific. Other bit fields in this register, as well as the definitions of the Input and Output registers, are common to all functions and are described in the "DxBxxIN Registers" on page 204 and the "DxBxxOU Registers" on page 204.

These mode bits are independent in the Timer block and control the Interrupt Type and the Compare Type. Timers have a special divide by one mode, when the period of the DR0 register is set to 00h. In this configuration, the primary output Terminal Count (TC) is the inverted input clock. The interrupt output is also the input clock inverted.

Table 17-6. 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. 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) will be masked and will not occur. The hardware capture input must be low for a software capture to occur.

Table 17-6. Timer Data Register Descriptions *(continued)*

Name	Function	Description
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, a period value of 00h results in the primary output to be the inversion of the input clock. In the optional full cycle terminal count 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 will only be reloaded into DR0 in the clock following a terminal count. 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 it 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 will transfer 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

Bank 0: There are three 8-bit data registers and a 2-bit control register. Table 17-7 explains the meaning of these registers in the context of the Counter operation. Note that the descriptions of the registers are dependant 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).

Bank 1: The mode bits in the Function register are block type specific. Other bit fields in this register, as well as the definitions of the Input and Output registers are common to all functions. These mode bits are independent in the Counter block and control the Interrupt Type and the Compare Type (same as the Timer function).

Table 17-7. 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. In the default one-half cycle terminal count mode, a period value of 00h will result in the auxiliary output to be the inversion of the input clock. In the optional full cycle terminal count mode, 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 will only be reloaded into DR0 in the clock following a terminal count. 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 will transfer 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

Bank 0: There are three 8-bit data registers and a 3-bit control register. Table 17-8 explains the meaning of these registers in the context of Dead Band operation.

Bank 1: The Mode bits in the Function register are block type specific. Other bit fields in this register, as well as the definitions of the Input and Output registers, are common to all functions.

Mode [1:0] is encoded as the Kill Type. In all cases, the output is forced low immediately. Mode bits are encoded for Kill options and are detailed in the following table.

Reference "Dead Band Timing" on page 206 for additional information on the Dead Band Kill options.

Table 17-8. 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 will only be reloaded into DR0 in the clock following a terminal count. 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 will transfer the contents of DR0 into DR2.

17.2.1.4 CRCPRS Register Definitions

Bank 0: There are three data registers and one control register. Table 17-9 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.

Bank 1: The mode bits in the Function register are block type specific. Other bit fields in this register, as well as the definitions of the Input and Output registers, are common to all functions and are described in the "DxBxxIN Registers" on page 204 and the "DxBxxOU Registers" on page 204.

The mode bits are encoded to determine the Compare type.

Table 17-9. 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. Value written will be 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 will transfer 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

Bank 0: There are three 8-bit data registers and one 8-bit control/status register. The following tables explain the meaning of these registers in the context of SPIM operation.

Bank 1: Mode bit 1 in the Function register is block type specific and selects Interrupt Type. Mode bit 0 selects Master or Slave (for SPIM, it is '0'). Other bit fields in this register, as well as the definitions of the Input and Output registers, are common to all functions.

Table 17-10. 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 this 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

Bank 0: There are three 8-bit data registers and one 8-bit control/status register. Figure 17-11 explains the meaning of these registers in the context of SPIS operation.

Bank 1: Mode bit 1 in the Function register is block type specific and selects Interrupt Type. Mode bit 0 selects Master or Slave (for SPIS, it is '1').

The SPIS has block-specific bits in the Output register, to select and control the Slave Select (SS_) input and behavior. Other Input and Output register bit field definitions are common to all functions and are described in “DxBxxIN Registers” on page 204 and “DxBxxOU Registers” on page 204.

The SPIS 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 multiplexer 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, a SPIS block has the auxiliary row output drivers forced off and therefore, the auxiliary output is not available in this block.

Table 17-11. 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

Bank 0: There are three 8-bit data registers and one 5-bit control/status register. Figure 17-12 explains the meaning of these registers in the context of Transmitter operation.

Bank 1: The mode bits in the Function register are block type specific. Other Input and Output registers bit field definitions are common to all functions.

In the Function register, the Mode bit 0 selects between Transmitter and Receiver (in this case Mode bit 0 is set to 1 for TX) and the Mode bit 1 selects the Interrupt Type.

Table 17-12. 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 setup 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 this 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

Bank 0: There are three 8-bit data registers and one 8-bit control/status register. The following table explains the meaning of these registers in the context of Receiver operation.

Bank 1: The mode bits in the Function register are block type specific. Other Input and Output registers bit field definitions are common to all functions.

In the Function register, the Mode bit 0 selects between Transmitter and Receiver (in this case Mode bit 0 is set to 0 for RX) and the Mode bit 1 selects the Interrupt Type.

Table 17-13. 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 eight 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

The DxBxxCR0 register is the digital blocks' control register. It is described by function in Table 17-14. For additional information, reference the DxBxxCR0 register on page 93.

Table 17-14. DxBxxCR0 Register Description

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 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	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. 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

17.2.3 INT_MSK1 Register

The INT_MSK1 register is described in the "Interrupt Controller" chapter on page 51. For additional information, reference the INT_MSK1 register on page 135.

Configuration Registers

The Configuration block contains 3 registers: Function (DxBxxFN), Input (DxBxxIN), and Output (DxBxxOU). The values in these registers should not be changed while the block is enabled.

17.2.4 DxBxxFN Registers

These registers contain the primary Function and Mode bits. The function bits configure the block into one of the available block functions (six for the Comm block, four for the Basic block). The mode bits select the options available for the selected function. These bits should only be changed when the block is disabled.

Three additional control bits are found in this register. The End/Single 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. The Data Invert bit optimally inverts the selected data input.

The BCEN bits enable the primary output of the block, to drive the row broadcast block. The BCEN bits are 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 185).

Table 17-15. DxBxxFN Function Registers

[2:0]: Function	000b: Timer 001b: Counter 010b: CRCPRS 011b: Reserved 100b: Dead band for PWM 101b: UART 110b: SPI 111b: Reserved
[4:3]: Mode	Function specific
[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
[6]: BCEN	1 == Disable 0 == Enable
[7]: Data Invert	1 == Invert block's data input 0 == Do not invert block's data input

For additional information, reference the DxBxxFN register on page 149.

17.2.5 DxBxxIN Registers

The Input registers are 8 bits and consist of two 4-bit fields to control each of the 16-1 Clock and Data input multiplexers. The meaning of these fields depends on the external clock and data connections, which is context specific.

Table 17-16. Digital Block Input Definitions

	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 multiplexer. 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, reference the DxBxxIN register on page 151.

17.2.6 DxBxxOU Registers

The Output registers contains two 3-bit fields: two bits to select and one bit to enable the tri-state drivers for the primary and auxiliary outputs, to drive onto the row output bus.

In one case, that of the SPI Slave, the meaning of the Auxiliary IO Select bits is different. The SPI Slave function is unique in that it has three function inputs and one function output defined. In this configuration, the auxiliary row output drivers are disabled and the bits are used to select one of four inputs from the auxiliary data input multiplexer (normally connected to row inputs), which is used as the SS_ (Slave Select) signal. The Aux IO Enable bit also has a different meaning in SPI Slave mode. If set, the SS_ signal is internally forced active and therefore, driving the SS_ from an auxiliary data input would not be required.

The Output register also contains the clock synchronization bits. These two bits are used to enable the synchronization, and select between SYSCLK and SYSCLKX2. When enabled, the input clock is resynchronized to the selected system clock, which occurs after the 16-1 multiplexing. This minimizes clock skew incurred in the generation of clocks, which can be derived from a wide variety of sources and paths. Under normal circumstances, synchronization should be enabled. Care should be taken to resynchronize SYSCLKX2 clock sources to the SYSCLKX2 system clock. The resynchronization should only be disabled in cases where asynchronous external inputs are used, such as in SPI Slave configurations.

Table 17-17. Digital Block Output Definitions

	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_.

For additional information, reference the DxBxxOU register on page 152.

17.3 Timing Diagrams

The timing diagrams in this section are presented according to their functionality and are in the following order.

- "Timer Timing" on page 205
- "Counter Timing" on page 206
- "Dead Band Timing" on page 206
- "CRCPRS Timing" on page 208
- "SPI Mode Timing" on page 208
- "SPIM Timing" on page 209
- "SPIS Timing" on page 212
- "Transmitter Timing" on page 215
- "Receiver Timing" on page 216

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 state is reset to its 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 is standalone, or if it 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 will be asserted in the cycle following the compare true, and will be 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 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 setup to the next positive edge of the clock to enable the next higher block to count once for every terminal count of all lower blocks.

The TCO of a given block becomes the TCI of the next least significant block in the chain. The TCO output indicates that that block and all higher blocks are at 00h count. The TCI/TCO chaining signals provide a way for the lower blocks to know when the upper blocks are at terminal count. Reload occurs when all blocks are at terminal count, which can be determined by CI, TCI and the block zero detect. Example timing for a three block Timer is shown in Figure 17-6.

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 1 cycle clock delay.

CMPO is generated from a combination of the internal compare condition and the CMPI input by 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 will 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).

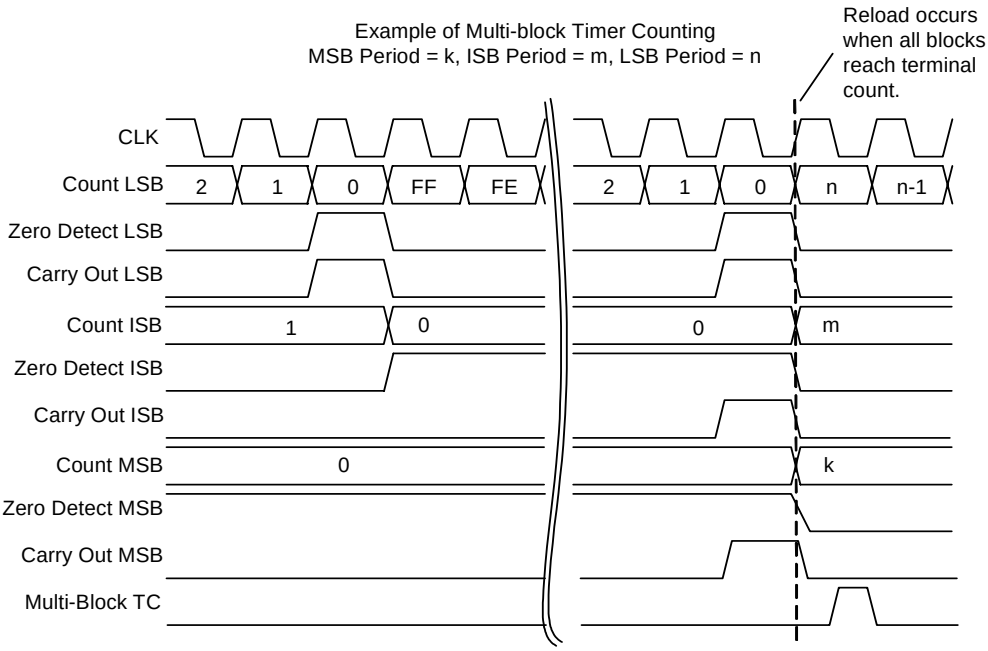


Figure 17-6. Multi-Block Timing

17.3.2 Counter Timing

Enable/Disable Operation. See Timer Enable/Disable Operation ("Timer Function" on page 192).

Terminal Count/Compare Operation. See Timer Terminal Count/Compare Operation ("Timer Function" on page 192).

Multi-Block Operation. See Timer Multi-Block Terminal Count/Compare Operation ("Timer Function" on page 192).

Gate (Enable) Operation. The data input controls the counter enable. The transition on this enable must have at least one 24 MHz cycle of setup time to the block clock. This will be ensured if internal or synchronized external inputs are used. For external unsynchronized signals, the user is responsible for this setup time.

As shown in Figure 17-7, when the data input is negated (counting is disabled) and the count is 00h, the TC output stays low until the clock following the assertion of the data input. When the block is disabled, the clock is immediately gated low. All internal state is reset, except for DR0, DR1, and DR2, which are unaffected.

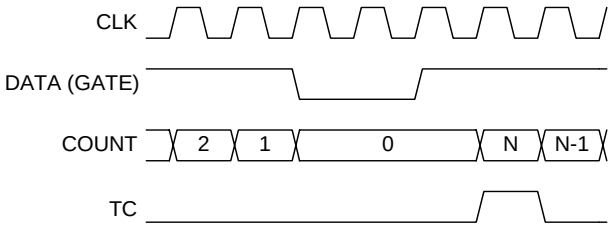


Figure 17-7. Counter Terminal Count Timing with Gate Disable

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 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 state is reset to its configuration specific reset state, except for DR0, DR1, and DR2, which are unaffected.

Normal Operation. Figure 17-8 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.

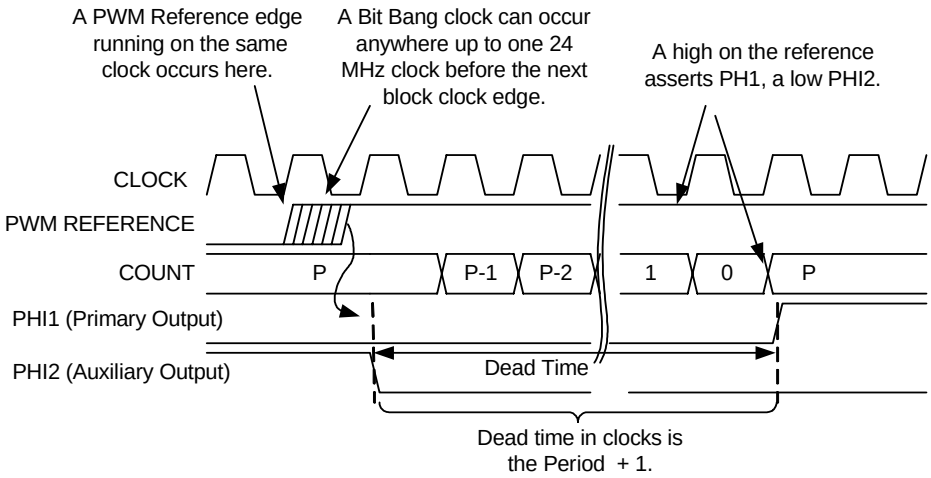


Figure 17-8. Basic Dead Band Timing

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 the following diagram where the low of the PWM is 4 clocks and the dead band period is 2 clocks and the high time of the PHI2 is 2 clocks.

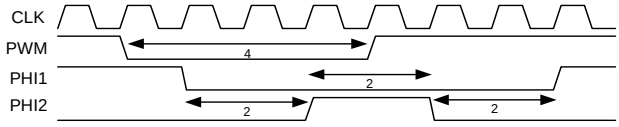


Figure 17-9. DB High Time is PWM Width Minus DB Period

In Figure 17-10, you reduce the width of the PWM low time by 1 clock (to 3 clocks). The dead band period remains the same, but the high time for PHI2 is reduced by 1 clock (to 1 clock). Of course the opposite phase, PHI1, increases in length by 1 clock.

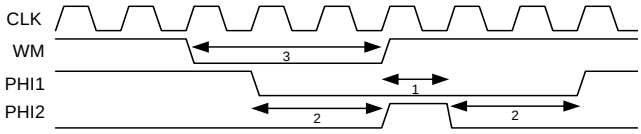


Figure 17-10. DB High Time is Reduced as PWM Width is Reduced

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 17-11 shows an example where the Dead Band period is 2 and the PWM width is 2. In this case, the high time of PHI2 is 0 clocks. Note that the Phase 1 dead band time is still 2 clocks.

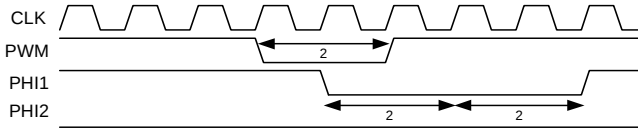


Figure 17-11. PWM Width Equal to Dead Band Period

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 will not be asserted high.

17.3.3.2 Kill Operation

It is assumed that the KILL input will not be synchronized at the row input. (This is not a requirement; however, if synchronized the KILL operation will have 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 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 17-12.
2. **Asynchronous Restart Mode**: When KILL is asserted high, internal state is not affected. When KILL is negated, outputs are restored, subject to a minimum disable time between one-half and one and one-half clock cycle. See Figure 17-13.
3. **Disable Mode**: There is no specific timing associated with Disable Mode. The block is disabled and the user must re-enable the function in firmware to continue processing.

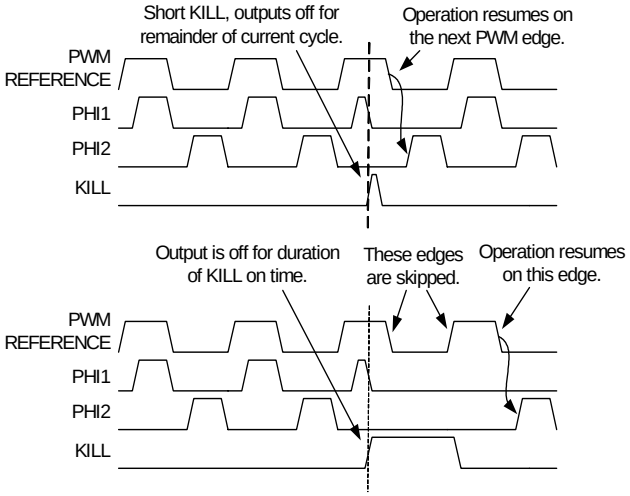


Figure 17-12. Synchronous Restart KILL Mode

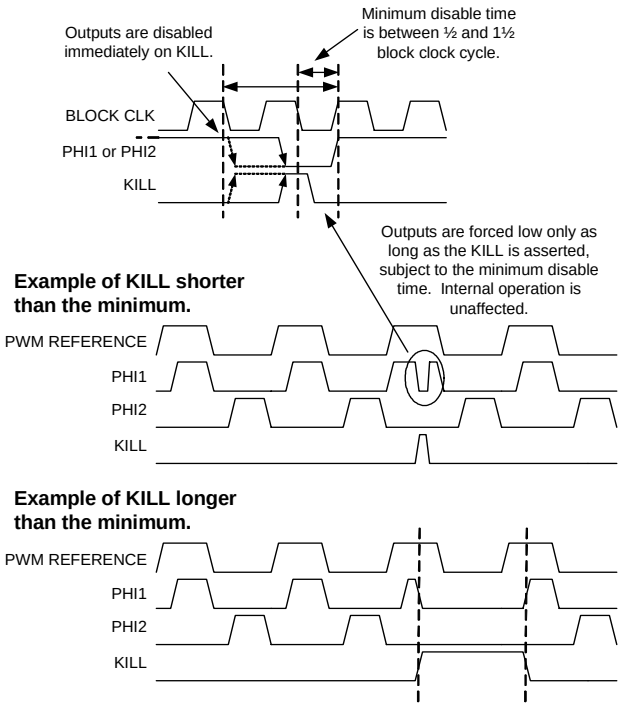


Figure 17-13. Asynchronous Restart Kill Mode

17.3.4 CRCPRS Timing

Enable/Disable Operation. Same as Timer Enable/Disable Operation ("Timer Timing" on page 205)

When the block is disabled, the clock is immediately gated low. All outputs are gated low, including the interrupt output. All internal state is reset to its configuration specific reset state, except for DR0, DR1, and DR2, which are unaffected.

17.3.5 SPI Mode Timing

Figure 17-14 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.

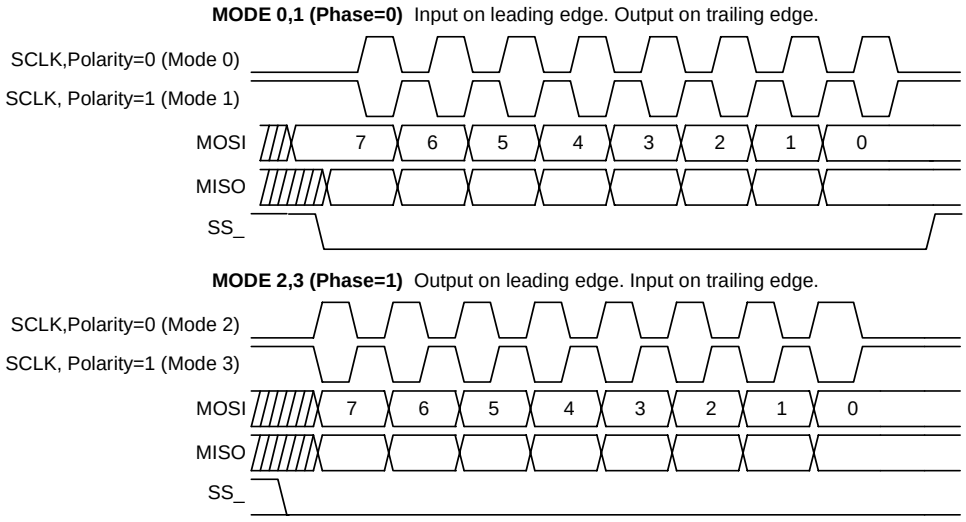


Figure 17-14. SPI Mode Timing

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 LSBF 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 state is reset (including CR0 status) to its configuration specific reset state, except for DR0, DR1, and DR2, which are unaffected.

Normal Operation. Typical timing for a SPIM transfer is shown in Figure 17-15 and Figure 17-16. 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 setup time to the next clock, a new byte transmission will be initiated. A 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.

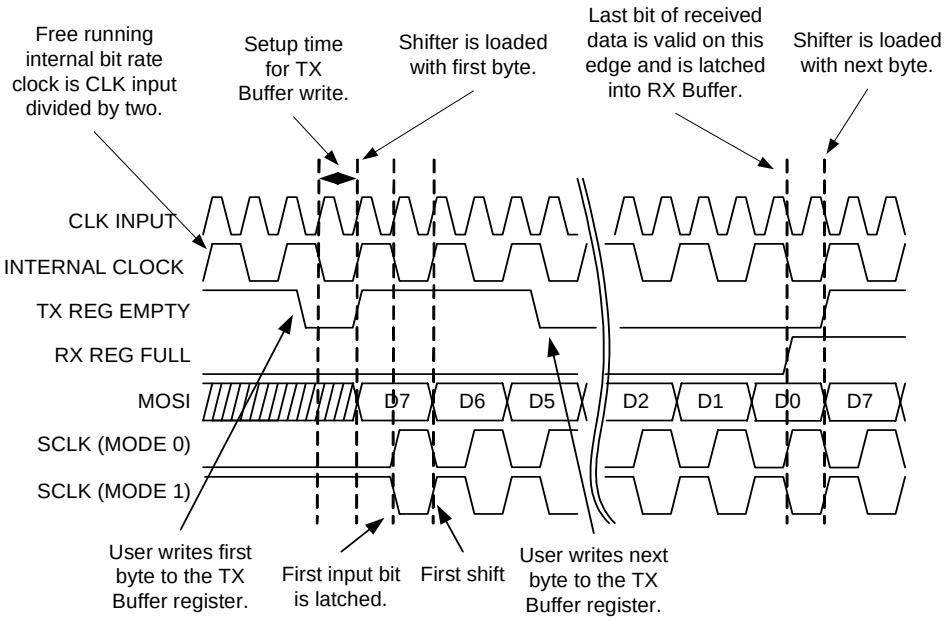


Figure 17-15. Typical SPIM Timing in Mode 0 and 1

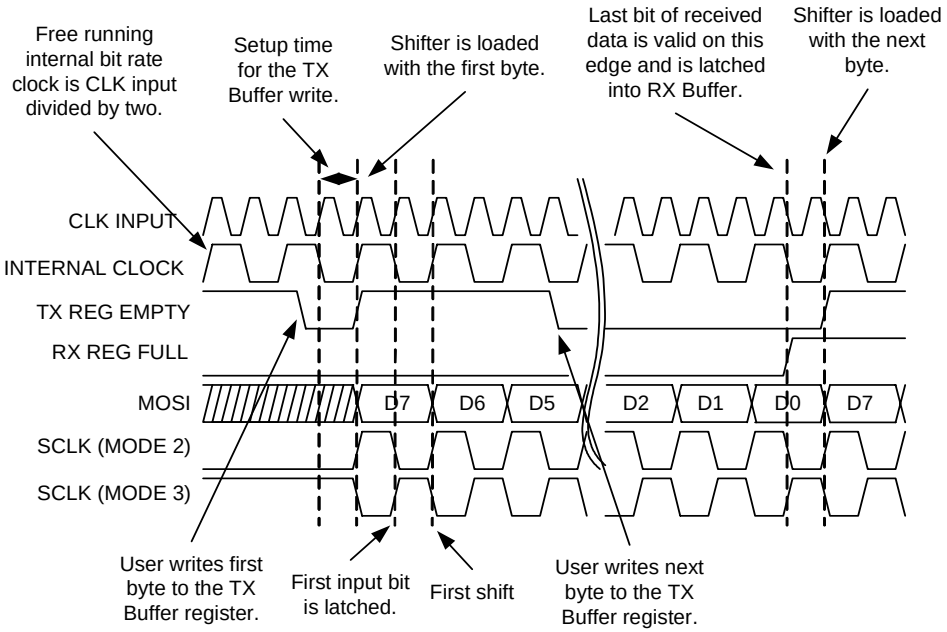


Figure 17-16. Typical SPIM Timing in Mode 2 and 3

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 that 8th bit of receive data. This status bit is cleared when the user reads the RX Buffer register (DR2).

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.

SPI Complete is an optional interrupt and is generated when 8 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.

See Figure 17-17 and Figure 17-18 for status timing relationships.

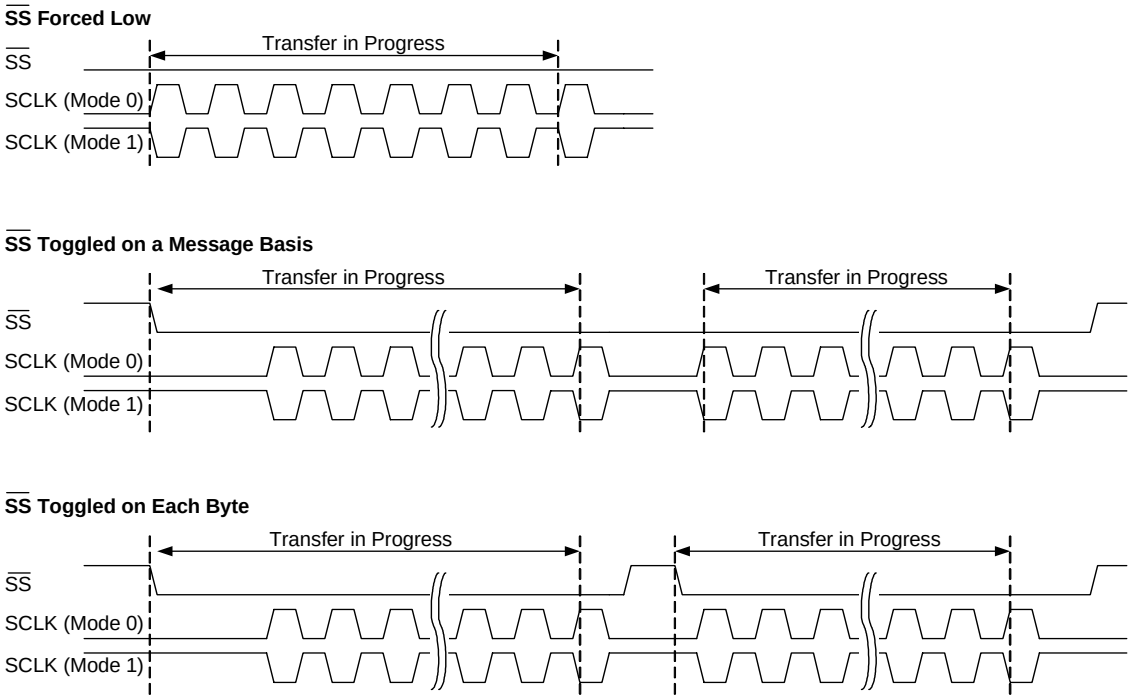


Figure 17-17. SPI Status Timing for Modes 0 and 1

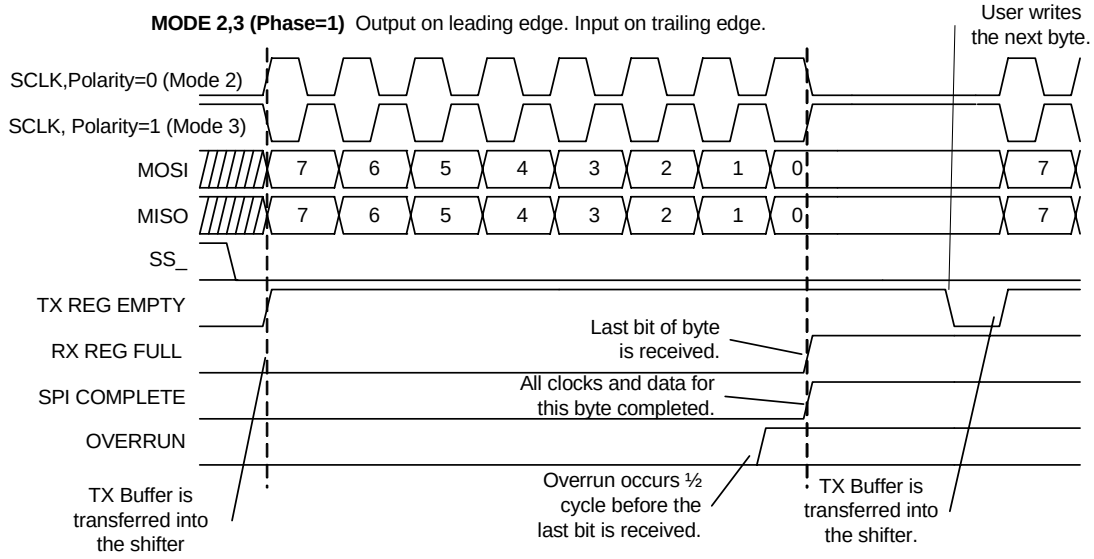


Figure 17-18. SPI Status Timing for Modes 2 and 3

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 LSBF 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 setup time to any external master supplying the clock. Setup 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 setup time must be assured through the protocol and an understanding of the timing between the master and slave in a system.

If SS_ is forced active (low) by configuration, before the block is enabled, no initial load from the TX Buffer register to the shifter will occur. TX loading only occurs on the falling edge of SS_ (modes 0 and 1 only).

When the block is disabled, the MISO output reverts to its idle '1' state. All internal state is reset (including CR0 status) to its configuration specific reset state, except for DR0, DR1, and DR2, which are unaffected.

Normal Operation. Typical timing for a SPIS transfer is shown in Figure 17-19 and Figure 17-20. 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 with 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).

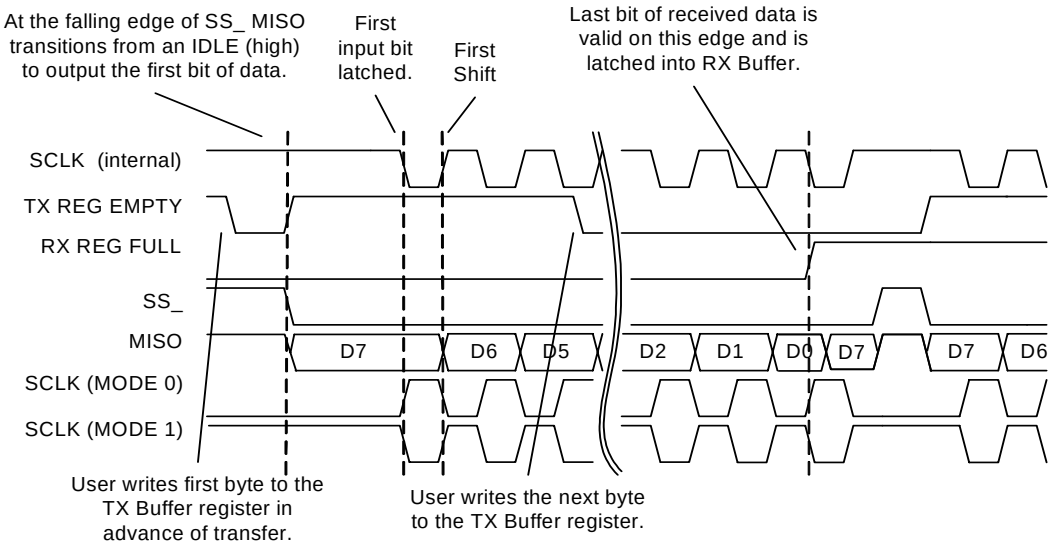


Figure 17-19. Typical SPIS Timing in Modes 0 and 1

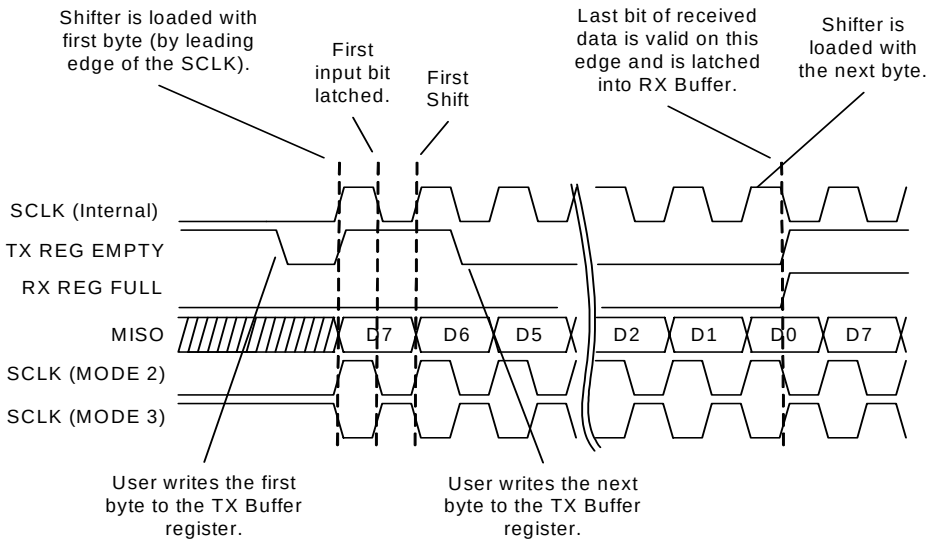


Figure 17-20. Typical SPIS Timing in Modes 2 and 3

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 will ignore 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 are 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 209.

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 17-21 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 setup 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 setup time (one-half clock minimum before the leading edge), with a knowledge of system latencies and response times.

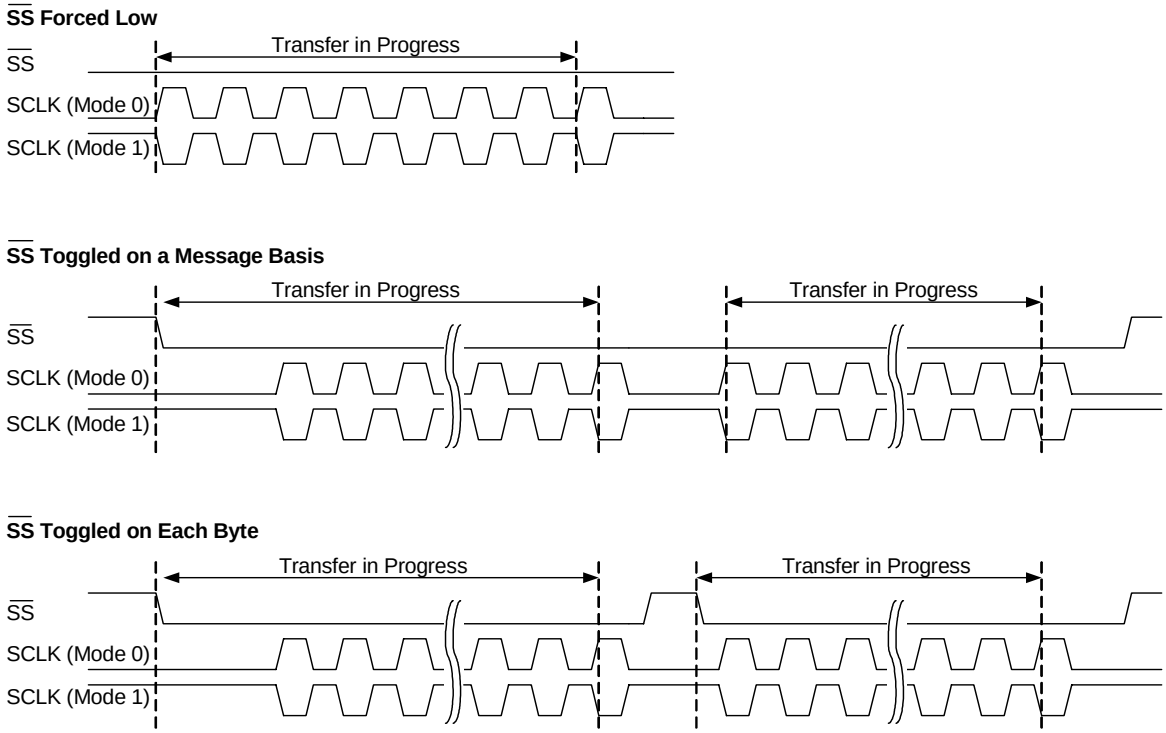


Figure 17-21. Mode 0 and 1 Transfer in Progress

Figure 17-22 illustrates TX data loading in modes 2 and 3. In this case, a transfer in progress is defined to be from the leading edge of the 1st 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 setup time, as the data is latched into the receiver on the trailing edge of the SCLK in these modes.

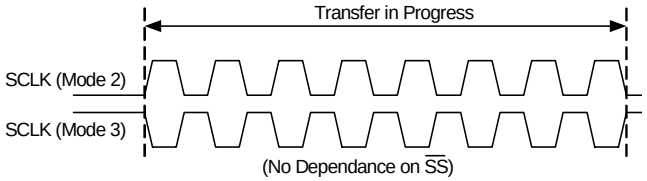


Figure 17-22. Mode 2 and 3 Transfer in Progress

17.3.8 Transmitter Timing

Enable/Disable Operation. As soon as the block is configured for Transmitter, and before enabling, the primary output is set to idle at logic '1' the mark state. The output will remain '1' until the block is enabled and a transmission is initiated. The auxiliary output will also idle 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 8 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 state is reset (including CR0 status) to its 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 setup time for the data to be recognized at the next positive internal bit clock edge. As shown in Figure 17-23, once the setup 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.

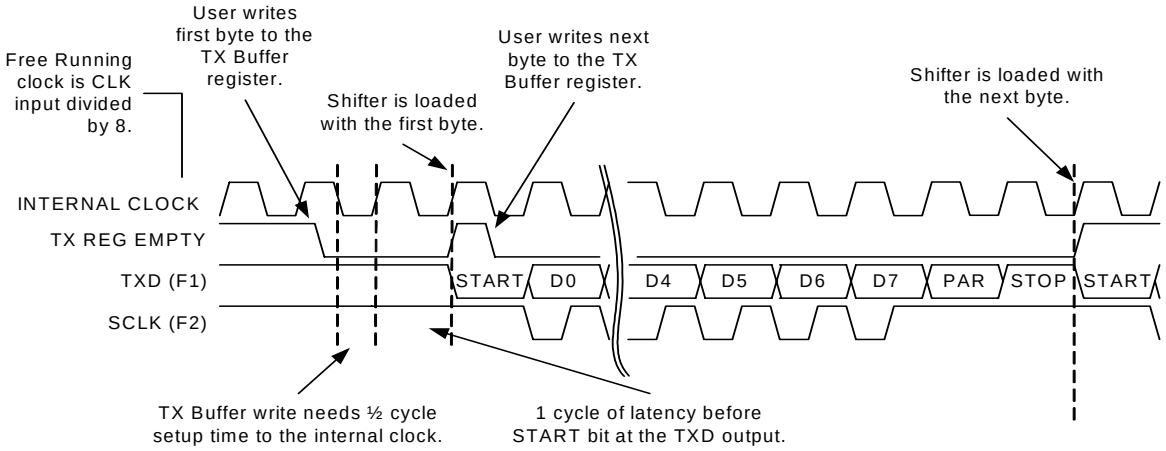


Figure 17-23. Typical Transmitter Timing

Figure 17-24 shows a detail of the Tx Buffer load timing. The data bits are shifted out on each of the subsequent clocks. Following the 8th 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 setup 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.

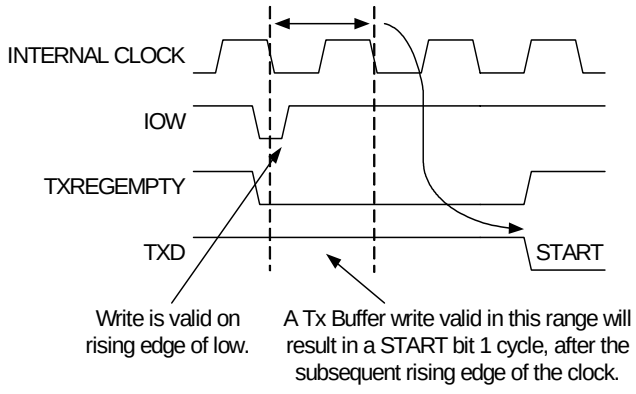


Figure 17-24. Tx Buffer Load Timing

The SCLK (auxiliary) output has an SPI mode 3 clock associated with the data bits (for the mode 3 timing see Figure 17-14). 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 will remain high and no subsequent interrupts will be logged. See Figure 17-25 for timing relationships.

Status Clear On Read. Refer to the SPIM subsection in "SPIM Timing" on page 209.

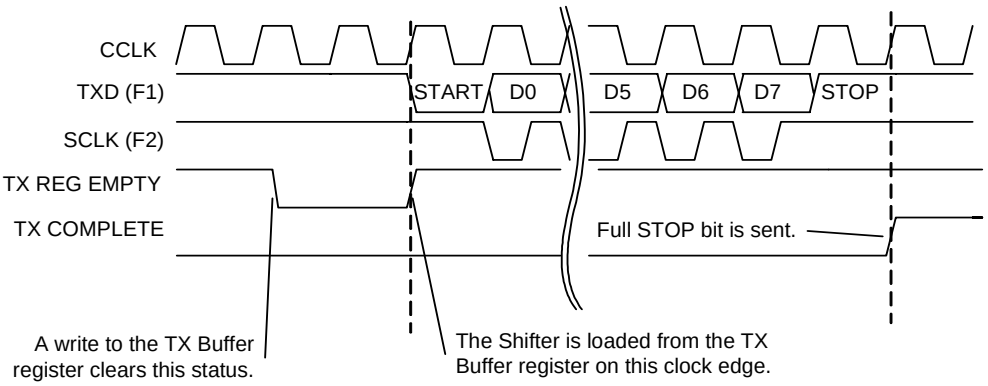


Figure 17-25. Status Timing for the Transmitter

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 will continue to follow the input, regardless of enable state. The auxiliary output will idle 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 setup time to the first START bit input.

When the block is disabled, the clock is immediately gated low. All internal state is reset (including CR0 status) to its configuration specific reset state, except for DR0, DR1, and DR2, which are unaffected.

Receive Operation. A clock, which must be 8X 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 17-26, 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 4), 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 1 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 re-synchronizing 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 17-26). During the mark (idle) and framing bits the SCLK output is high.

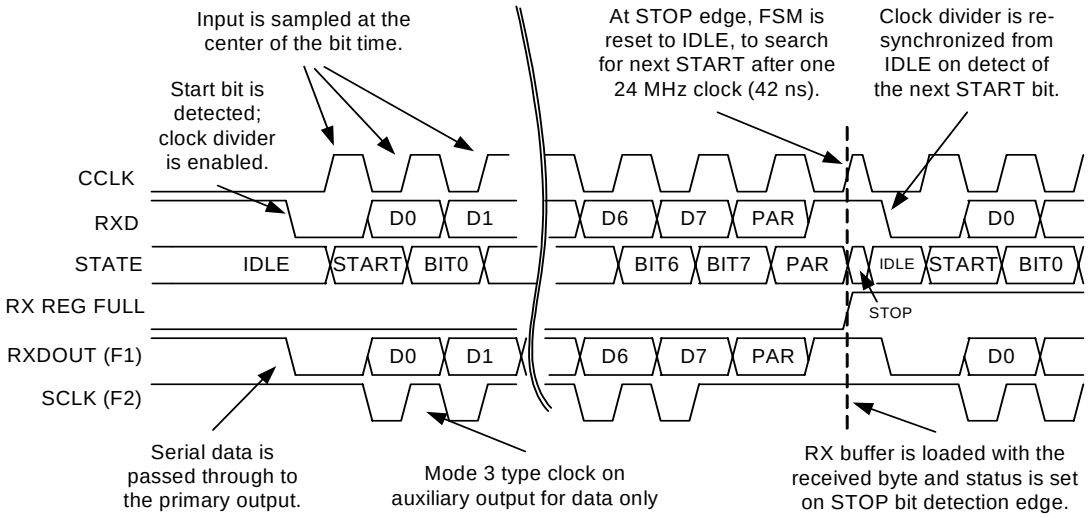


Figure 17-26. Receiver Operation

Clock Generation and Start Detection. The input clock selection is a free running 8X 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 17-27, 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 8X rate. If the RXD input is still logic '0' after 3 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 was assumed to be spurious, and the Receiver remains in the idle state.

As shown in Figure 17-27, the internal bit clock (CCLK) is running 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.

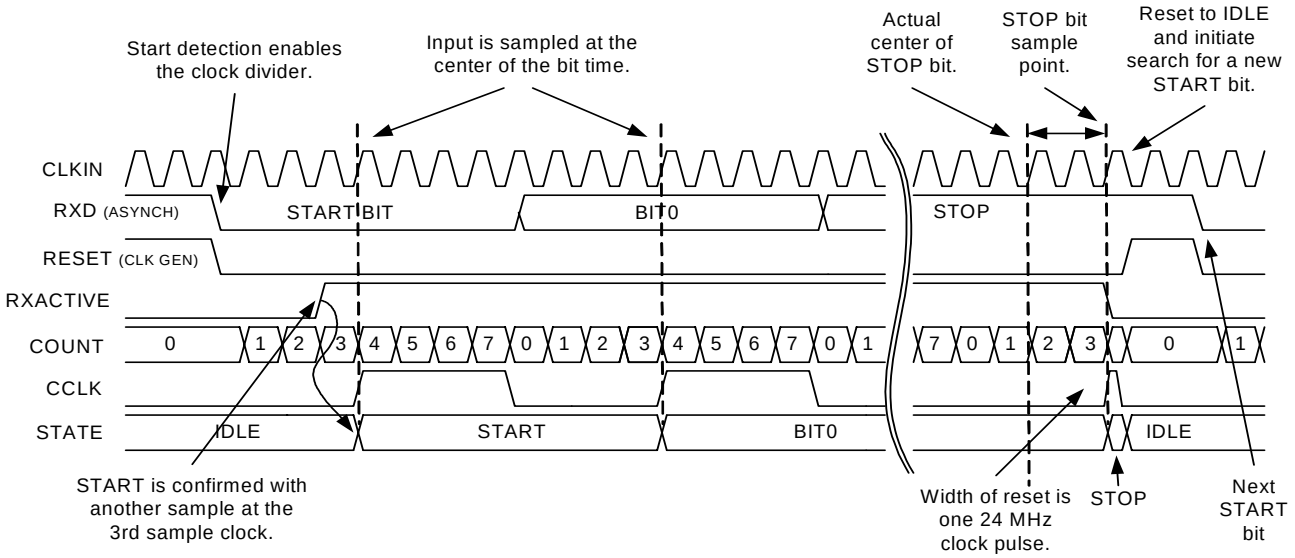


Figure 17-27. Clock Generation and Start Detection

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 17-28 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 has counted 9.5 bit times, while the transmitter has counted 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 5%. 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.

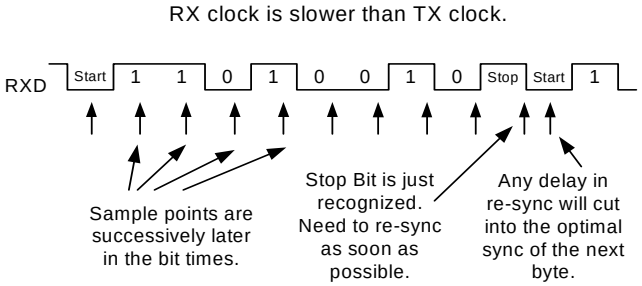


Figure 17-28. Example RX Re-Synchronization

This theoretical maximum will be 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 us. In this case the new maximum offset is:

$$((4.35\ \text{us} - 42\ \text{ns})/4.35\ \text{us}) \times 5\% \text{ or } 4.95\%$$

At slower baud rates, this value gets closer to the theoretical maximum of 5%.

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 got a '0'). This will typically occur 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 209.

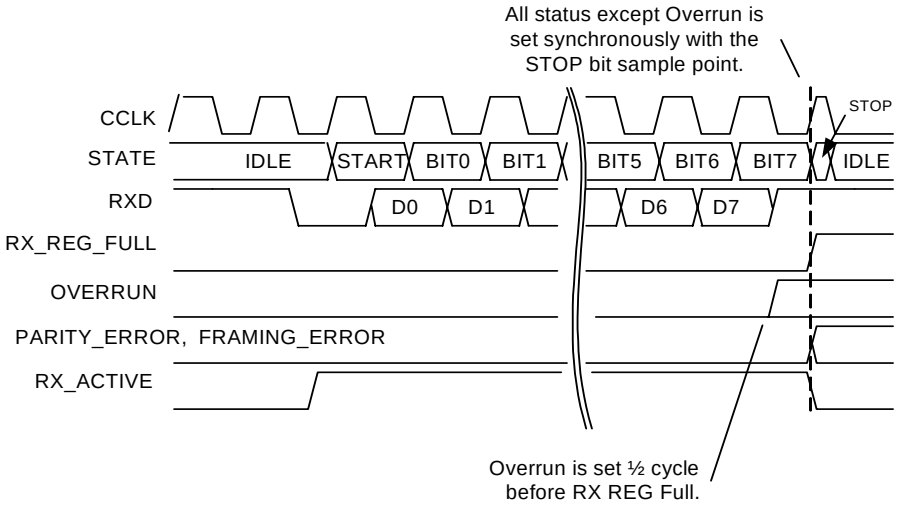


Figure 17-29. Status Timing for Receiver

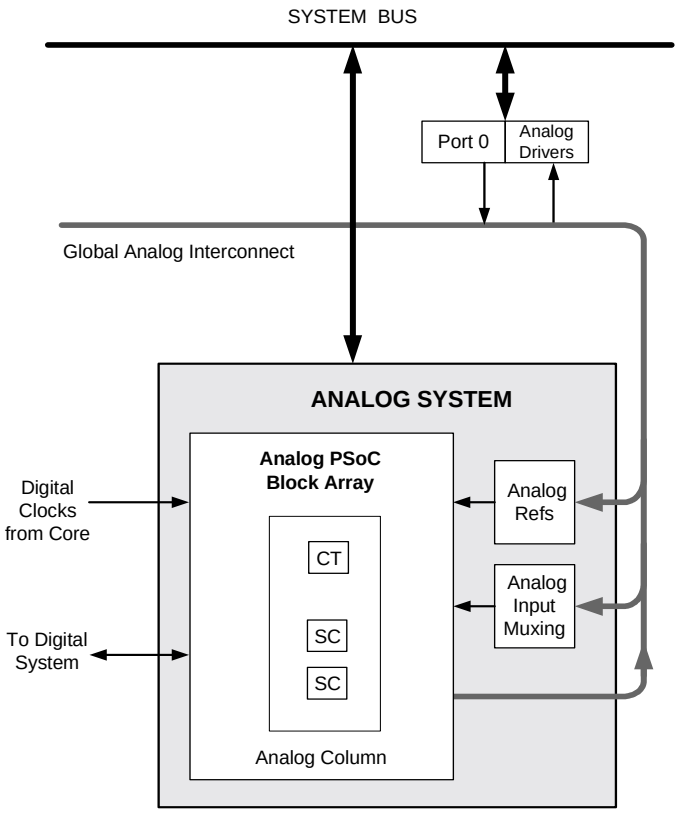
SECTION E ANALOG SYSTEM

The Analog System section discusses the analog components of the PSoC device and the registers associated with those components. This section encompasses the following chapters:

- Analog Interface on page 223
- Analog Array on page 231
- Analog Input Configuration on page 235
- Analog Reference on page 237
- Switched Capacitor Block on page 239
- Continuous Time Block on page 245

Top-Level Analog Architecture

The figure below displays the top-level architecture of the PSoC's analog system. With the exception of Analog Drivers, each component of the figure is discussed at length in this section. Analog drivers are discussed in detail in the Analog Output Drivers chapter on page 61.



PSoC Analog System Block Diagram

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. 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.

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.

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. Some available supported analog functions are 12-bit Incremental and 11-bit Delta-Sigma ADC, successive approximation ADCs up to 6 bits, DACs up to 8 bits, programmable gain stages, sample and hold circuits, programmable filters, comparators, and a temperature sensor.

The analog blocks are organized into columns. There is one analog column in the CY8C22xxx, which contains one Continuous Time Block, one Switch Capacitor (SC) Type C, and one Type D Switch Capacitor (SC). 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 223 for additional information.

There are three outputs from each analog block. (There are an 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 and the user has 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 and the user has control of this output through register settings.
3. The local outputs (OUT, plus GOUT, and LOUT in the Continuous Time blocks) are routed to neighbor blocks. The various input multiplexer connections (NMux, PMux, RBotMux, AMux, BMux, and CMux) all use the output bus from one block as their input.

Three analog PSoC blocks are available separately or combined with the digital PSoC blocks. 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.

The analog functionality provided is as follows.

- A/D and D/A converters, programmable gain blocks, comparators, and switched capacitor filters.
- Single ended configuration is cost effective for reasonable speed and accuracy, and provides a simple interface to most real-world analog inputs and outputs.
- Support is provided for sensor interfaces, audio codes, embedded modems, and general-purpose opamp circuits.
- Flexible, System on-a-Chip programmability, providing variations in functions.
- For a given function, easily selected trade-offs of accuracy and resolution with speed, resources (number of analog blocks), and power dissipated for that application.
- The analog section is an “Analog Computation Unit,” providing programmed steering of signal flow and selecting functionality through register-based control of analog switches. It also sets coefficients in Switched Capacitor Filters and noise shaping (Delta-Sigma) modulators, as well as program gains or attenuation settings in amplifier configurations.
- The architecture provides continuous time blocks and discrete time (Switched Capacitor) blocks. The continuous time blocks allow selection of precision amplifier or comparator circuitry, using programmable resistors as passive configuration and parameter setting elements. The Switched Capacitor (SC) blocks allow configuration of DACs, Delta Sigma, Incremental or Successive Approximation ADCs, or Switched Capacitor filters with programmable coefficients.

Analog Register Summary

The table below lists all the PSoC registers in the analog system.

Summary Table of the Analog Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
ANALOG INTERFACE REGISTERS										
0,64h	CMP_CR0			COMP[1]				AINT[1]	AINT[0]	RW : 00
0,66h	CMP_CR1			CLDIS[1]						RW : 00
0,65h	ASY_CR		SARCNT[2:0]			SARSIGN	SARCOL[1]		SYNCEN	RW : 00
0,E6h	DEC_CR0	IGEN[3:0]				ICLKS0	DCOL[1:0]		DCLKS0	RW : 00
0,E7h	DEC_CR1	ECNT	IDEC			ICLKS1			DCLKS1	RW : 00
1,60h	CLK_CR0					Acolumn1[1:0]				RW : 00
1,61h	CLK_CR1		SHDIS	ACLK1[2:0]			ACLK0[2:0]			RW : 00
1,66h	AMD_CR1						AMOD1[2:0]			RW : 00
1,67h	ALT_CR0	LUT1[3:0]								RW : 00
ANALOG INPUT CONFIGURATION REGISTERS										
0,60h	AMX_IN					ACI1[1:0]		ACI0[1:0]		RW : 00
1,62h	ABF_CR0	ACol1Mux		ABUF1EN0				Bypass	PWR	RW : 00
ANALOG REFERENCE REGISTER										
0,63h	ARF_CR		HBE	REF[2:0]			PWR[2:0]			RW : 00
SWITCHED CAPACITOR BLOCK REGISTERS										
Switched Capacitor Block Registers, Type C										
x,94h	ASC21CR0	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										
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
CONTINUOUS TIME BLOCK REGISTERS										
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

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.

18. Analog Interface

This chapter explains the Analog System Interface and its associated registers. The analog system interface is a collection of system level interfaces to the analog array and analog reference block.

Table 18-1. Analog Interface Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,64h	CMP_CR0			COMP[1]				AINT[1]		RW : 00
0,66h	CMP_CR1			CLDIS[1]						RW : 00
0,65h	ASY_CR		SARCNT[2:0]			SARSIGN	SARCOL[1]		SYNCEN	RW : 00
0,E6h	DEC_CR0	IGEN[3:0]				ICLKS0	DCOL[1:0]		DCLKS0	RW : 00
0,E7h	DEC_CR1	ECNT	IDEC			ICLKS1			DCLKS1	RW : 00
1,60h	CLK_CR0					Acolumn1[1:0]				RW : 00
1,61h	CLK_CR1		SHDIS	ACLK1[2:0]			ACLK0[2:0]			RW : 00
1,66h	AMD_CR1						AMOD1[2:0]			RW : 00
1,67h	ALT_CR0	LUT1[3:0]								RW : 00

18.1 Architectural Description

Figure 18-1 displays the top-level diagram of the PSoC device's analog system.

18.1.1 Analog Data Bus Interface

The Analog 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 can drive 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, Address = Bank 0,64H).

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 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 shown in Table 18-1. There is also an option to force the latch in each column into a transparent mode by setting bits in the CMP_CR1 register.

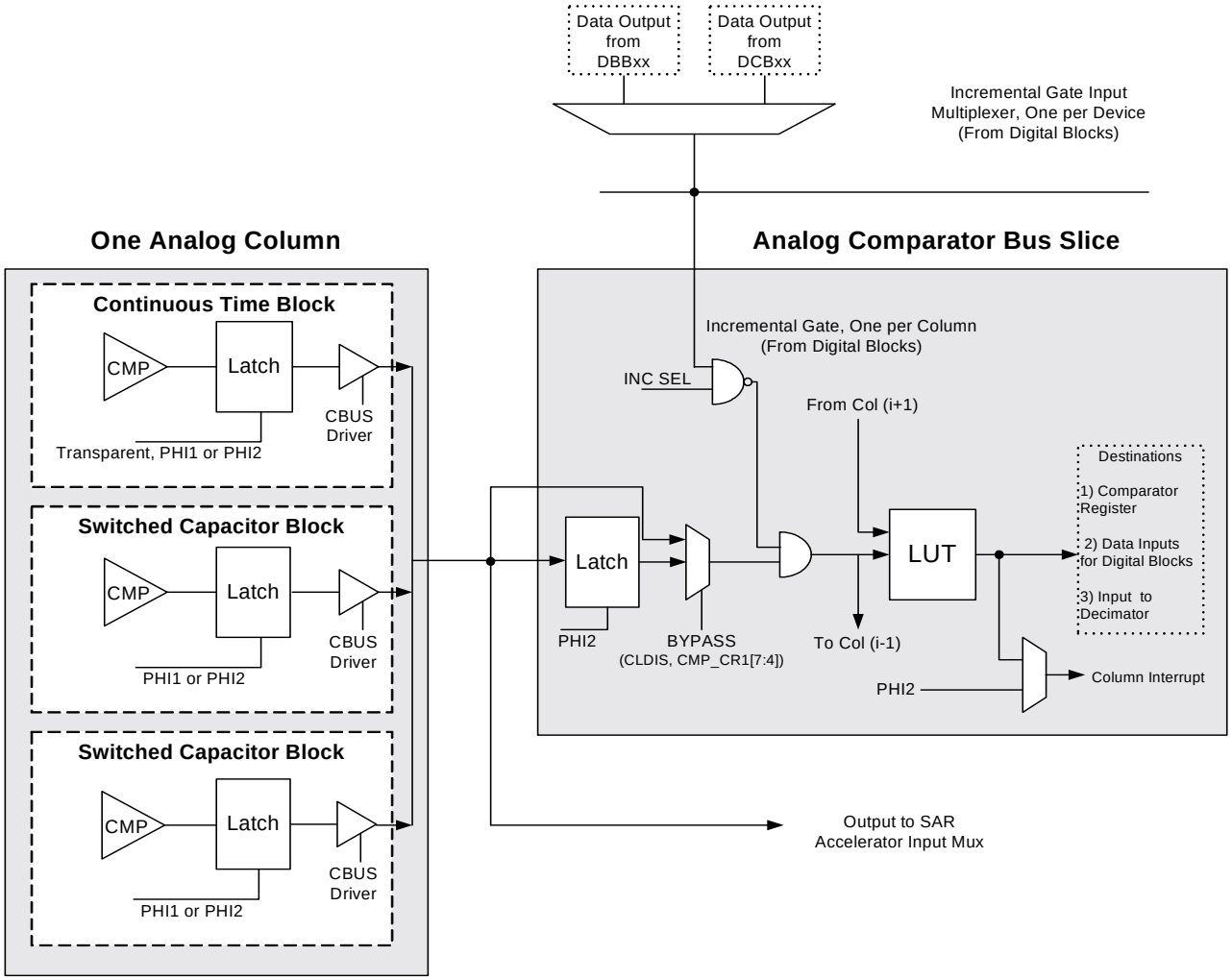


Figure 18-1. An Analog Comparator Bus Slice

As shown in Figure 18-1, the Comparator bus output is gated by a signal from the digital blocks. This feature is used to precisely control the integration period of an incremental ADC. There are two direct connect digital block output options per row for driving the gate signal: DBBx1 and DCBx2. This selection may be made with the ICCKSEL 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 function. The LUT takes two inputs, A and B, and provides a selection of 16 possible logic functions of those inputs. The LUT A and B inputs for each column comparator output is shown in the following table.

Table 18-2. A and B Inputs for Each Column Comparator Output

Comparator Output	A	B
Column 0	ACMP0	0
Column 1	0	0
Column 2	0	0
Column 3	0	ACMP0

The LUT configuration is set in two control registers, ALT_CR0 and ALT_CR1. 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 18-3. RDIxLTx Register

LUTx[3:0]	0h: 0000: FALSE 1h: 0001: A .AND. $\overline{B}$ 2h: 0010: A .AND. $\overline{B}$ 3h: 0011: $\overline{A}$ 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: $\overline{A}$ .XNOR. B Ah: 1010: $\overline{B}$ Bh: 1011: $\overline{A}$ .OR. $\overline{B}$ Ch: 1100: $\overline{A}$ Dh: 1101: $\overline{A}$ .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 cap blocks are arranged in four columns, 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 this clock 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, 24V1, 24V2, ACLK0, and ACLK1. The 24V1 and 24V2 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.

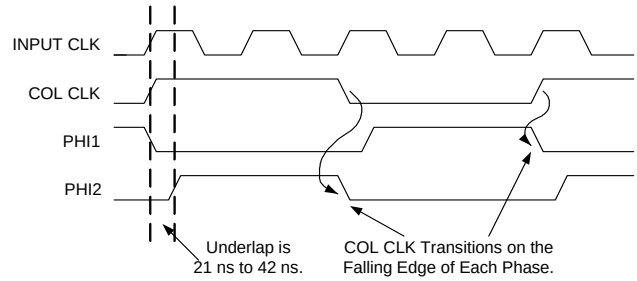


Figure 18-2. Two Phase Non-Overlapping Clock Generation

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 or more 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 will cause 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 reinitialized and thus a discontinuity in PHI1/PHI2 clocking will occur.

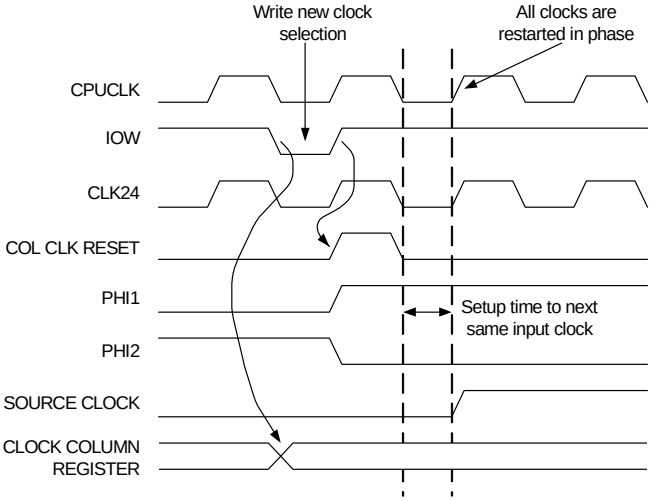


Figure 18-3. Column Clock Resynchronize on an IO Write

18.1.4 Decimator and Incremental ADC Interface

The Decimator and Incremental 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 is split between two registers: DEC_CR0 and DEC_CR1.

18.1.4.1 Decimator

The decimator is a hardware block that is 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 is running in conjunction with the analog processing. At the terminal count of this Timer, the primary output goes high for a one-half clock cycle. For purposes of Decimator operation, this signal is inverted and connected to the BW input. This becomes the 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 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 ICLKS0 and ICLKS1 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 Switch Cap 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 Switch Cap registers is blocked at the interface and the CPU will stall. 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 be performed at the optimum point in the analog cycle, at the expense of CPU bandwidth. Figure 18-4 shows the timing for this operation.

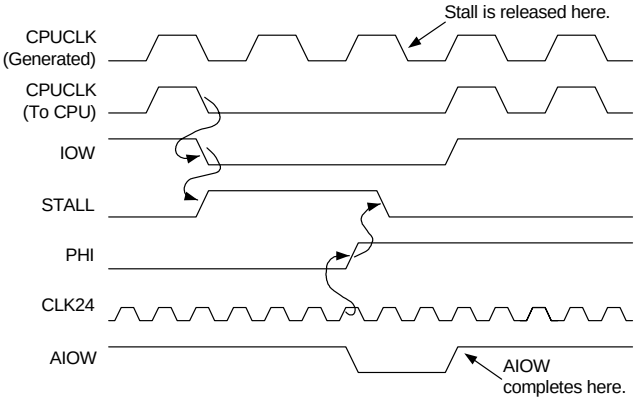


Figure 18-4. Synchronized Write to a DAC Register

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.

18.1.7 SAR Hardware Acceleration

The SAR algorithm is a binary search on the DAC code that best matches the input voltage that is 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 1 lsb of the input voltage.

The successive approximation A/D 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.

18.1.7.1 Architectural Description

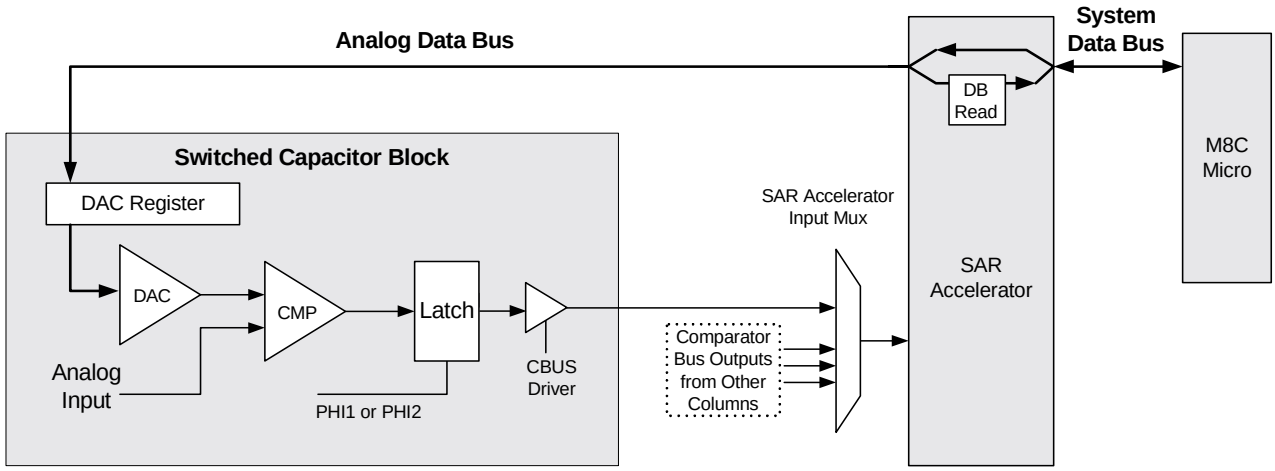


Figure 18-5. SAR Hardware Accelerator

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. However, 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 6 bits of a result in a given sequence. A higher resolution SAR is implemented with multiple passes.

18.1.7.2 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 clock cycles into the instruction, 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 one MHz, this allows over 300 ns for the DAC output and comparator to settle.

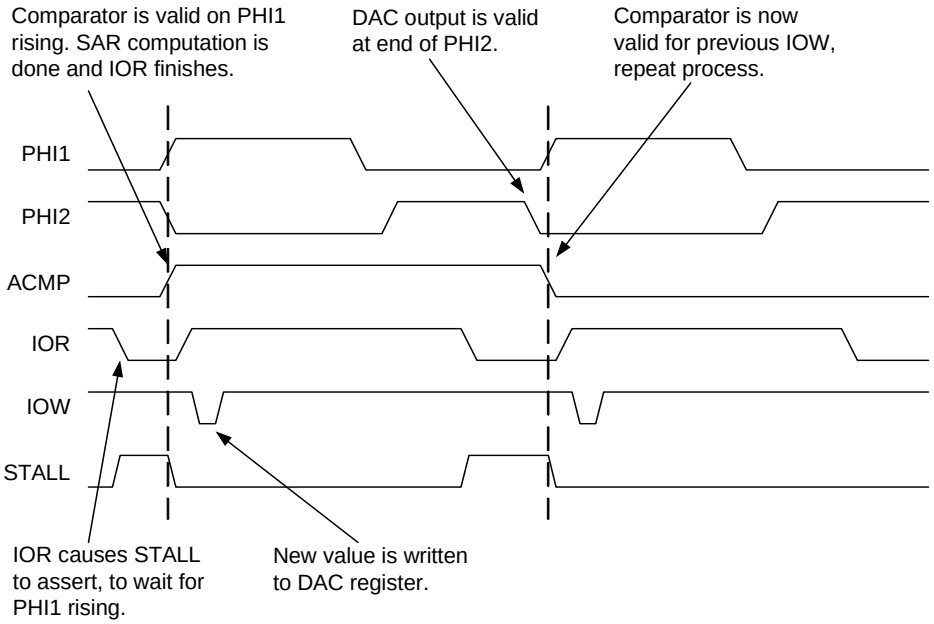


Figure 18-6. General SAR Timing

18.2 Register Definitions

18.2.1 CMP_CR0 Register

This register contains one field. Bit 5, COMP[1], is the read-only bit corresponding to the comparator bit in the analog column. This bit is synchronized to the column clock, and thus may be reliably polled by the CPU. Bit 1, AINT[1], selects the interrupt source for the column as the input to the interrupt controller:

By default, the interrupt is the comparator bit. However, if a bit in this field is set, the interrupt for that column will be derived from the falling edge of PHI2 clock for that column. Firmware can use this capability to synchronize to the current column clock.

For additional information, reference the CMP_CR0 register on page 103.

18.2.2 CMP_CR1 Register

The CLDIS bits in this register are used to override the analog column comparator synchronization. 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 32K clock) are turned off during Sleep, the synchronizer must be bypassed.

For additional information, reference the CMP_CR1 register on page 105.

18.2.3 ASY_CR Register

The SAR hardware control bits are located in the ASY_CR register. All bits are relevant to SAR operation except for bit 0, SYNCEN. SYNCEN is associated with analog register write stalling and is described in the Analog Interface Synchronization section.

The SAR hardware accelerator is a block of specialized hardware designed to sequence the SAR algorithm for efficient A/D 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 should initialize the ASY_CR register and set the sign bit of the DAC as the first guess in the algorithm. A sequence of OR instructions (Read, Modify, Write) to the DAC (CR0) 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 will complete the conversion of a 6-bit DAC. The resulting DAC code, which matches the input voltage to within 1 LSB, is then read back from the DAC CR0 register.

Bits 7 and 1: Reserved.

Bits 6, 5, and 4: SARCNT[2:0]. 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 would initialize 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 would be as follows.

```
mov  reg[ASY_CR], 60h  // SAR count value=6,
Sign=0, Col=0
or reg[ASC10CR0], 0  // Check sign, set bit 4
or reg[ASC10CR0], 0  // Check bit 4, set bit 3
or reg[ASC10CR0], 0  // Check bit 3, set bit 2
or reg[ASC10CR0], 0  // Check bit 2, set bit 1
or reg[ASC10CR0], 0  // Check bit 1, set bit 0
or reg[ASC10CR0], 0  // Check bit 0
```

Bit 3: SARSIGN. SAR sign selection. This bit optionally inverts the comparator input to the SAR accelerator and must be set based on the type of PSOC block configuration selected. Table 18-4 lists some typical examples.

Table 18-4. Typical PSOC Block Configurations

Configuration	Description	Sign
SAR6 – 2 blocks	1 DAC6, 1 COMP (could be CT)	0
SAR6 – 1 block	1 for both DAC6 and COMP	1
MS SAR10 –3 blocks	1 DAC10, 1 COMP (could be CT) (When processing MS DAC block)	0

Bits 2 and 1: SARCOL[1:0]. 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. The purpose of 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 will occur during the stall, including interrupt processing. Therefore, the effect of stalling on CPU throughput must be considered.

For additional information, reference the ASY_CR register on page 104.

18.2.4 DEC_CR0 Register

This register contains control bits to access hardware support for both the Incremental ADC and the DELISG ADC. For Incremental support, the upper four bits, IGEN[3:0], select which column comparator bit will be 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 ICLKS2 and ICLKS1 bits in DEC_CR1. Up to one of eight digital blocks may be selected, depending on the chip resources.

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).

In addition, 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 chip resources) with bit DCLKS0 and DCLKS2, DCLKS1 in DEC_CR1.

For additional information, reference the DEC_CR0 register on page 140.

18.2.5 DEC_CR1 Register

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.

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 will cause the decimator output to be sampled. However, when the IDEC bit is set, the negative edge of the selected digital block input will cause the decimator value to be sampled.

Bits 5 to 0: ICLKSx and DCLKSx. The ICLKS1 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, reference the DEC_CR1 register on page 141.

18.2.6 CLK_CR0 Register

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 up to one of four digital block outputs (functioning as clock generators) as selected by CLK_CR1.

For additional information, reference the CLK_CR0 register on page 154.

18.2.7 CLK_CR1 Register

Bit 7: Reserved.

Bit 6: SHDIS. The SHDIS bit in the CLK_CR1 register is described 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).

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 busses for the entire period of their respective PHI2s.

Bits 5 to 0: ACLKx. There are two 3-bit fields in this register that can select up to one of eight digital blocks (depending on chip 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).

For additional information, reference the CLK_CR1 register on page 155.

18.2.8 AMD_CR1 Register

This register controls the selection of the MODBIT for analog column 1. See the AMD_CR0 register. For additional information, reference the AMD_CR1 register on page 157.

18.2.9 ALT_CR0 Register

This register controls the selection of logic functions that may be selected for the analog comparator bits in column 1. 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 18-2 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. For CY8C22xxx parts, there is only one column and B=0.

For additional information, reference the ALT_CR0 register on page 158.

19. Analog Array

This chapter presents the Analog Array. It has no registers associated with it. The analog blocks can be used to implement a wide range of functions, limited only by the designer's imagination.

The following functions 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. Reference the *PSoC Designer User Modules Data Book* for additional information.

- Delta-Sigma A/D Converters
- Successive Approximation A/D Converters
- Incremental A/D Converters
- Programmable Gain/Loss Stage
- Analog Comparators
- Zero-Crossing Detectors
- Low-Pass Filter
- Band-Pass Filter
- Notch Filter
- Amplitude Modulators
- Amplitude Demodulators
- Sine-Wave Generators
- Sine-Wave Detectors
- Sideband Detection
- Sideband Stripping
- Audio Output Drive
- DTMF Generator
- FSK Modulator

By modifying registers, as described in this data sheet, users can configure PSoC blocks to perform these functions and more.

19.1 Architectural Description

The analog array is designed to allow moving between families without modifying projects, except for resource limitations.

The PSoC device has only one column (Column 1). Generally, nets that were inputs to the missing columns are left floating. The nets that were outputs from the missing columns are connected to Vss. See the following figures for specific details.

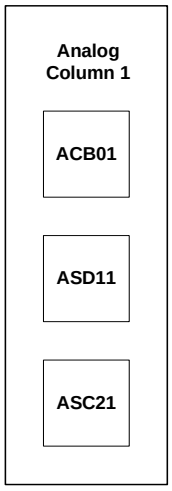


Figure 19-1. Array of Analog PSoC Blocks

The figures that follow illustrate the analog mux connections for the PSoC device.

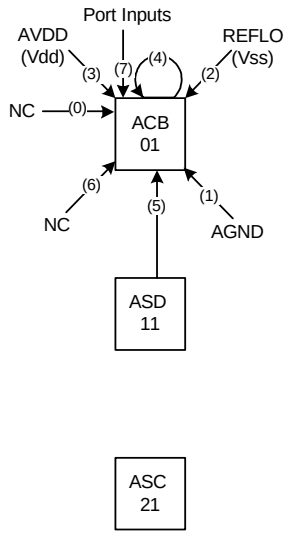


Figure 19-2. NMux Connections

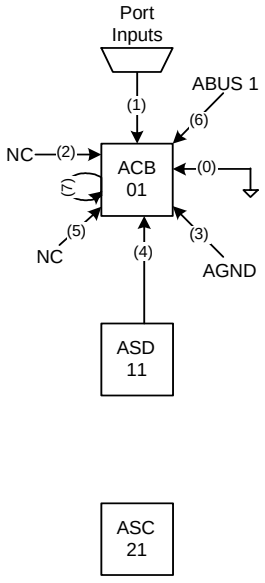


Figure 19-3. PMux Connections

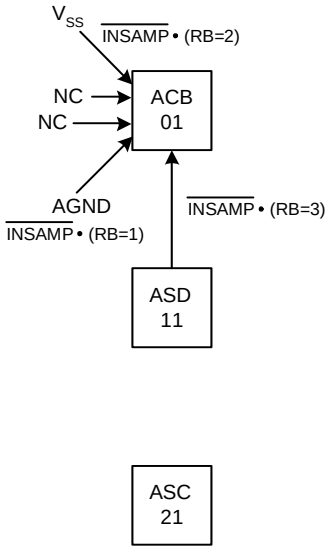


Figure 19-4. RBotMux Connections

The ACMux, as shown in the Analog Switch Cap Type C Block xx Control 1 register, controls the input muxing for both the A and C capacitor branches. The high order bit, ACMux[2], selects one of two inputs for the C branch. See the individual AMux and CMux diagrams.

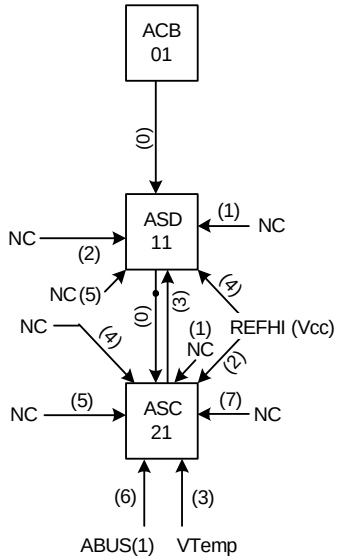


Figure 19-5. AMux Connections

Figure 19-6. CMux Connections

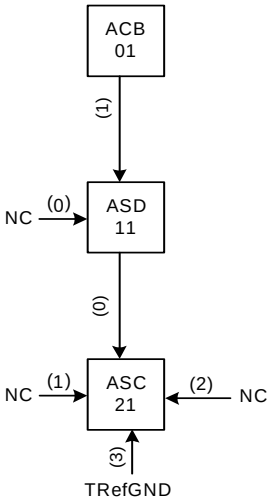


Figure 19-7. BMuxSC/SD Connections

19.1.1 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. The output on the comparator bus can drive into the digital blocks and is also available to be read in the CMP_CR register.

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

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

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

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

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), can give 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 briefly discusses the Analog Input Configuration and its associated registers.

Table 20-1. Analog Input Configuration Registers

Address	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
1,62h	ABF_CR0	ACol1Mux		ABUF1EN0				Bypass	PWR	RW : 00

The input multiplexor maps device inputs to analog array columns, based on bit values in the AMX_IN and ABF_CR0 registers. Edge columns are fed by one 4:1 mux; inner columns are fed by 1 of 2 4:1 muxes. The muxes are CMOS switches with typical resistances in the range of 2K ohms. Reference the analog block diagrams, on the following pages, to view the various analog input configurations.

The CY8C22xxx device uses only one “internal” column (column 1) and has unique analog mux connectivity from Port 0 (8:1 into CT block). This device contains a more limited reference block than larger family members.

20.1 Register Definitions

20.1.1 AMX_IN Register

Bits 7 to 4: Reserved.

Bits 3 to 0: ACI1[1:0] and ACI0[1:0]. These 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_CR).

For additional information, reference the AMX_IN register on page 101.

20.1.2 ABF_CR0 Register

This register 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 6: Reserved.

Bit 5: ABUF1EN0. This bit enables or disables the column output amplifiers.

Bits 4, 3, and 2: Reserved.

Bit 1: Bypass. Bypass mode connects the amplifier input directly to the output. When this bit is set, all amplifiers controlled by the register will be in bypass mode.

Bit 0: PWR. This bit is used to set the power level of the amplifiers. When this bit is set, all amplifiers controlled by the register will be in a high power state.

For additional information, reference the ABF_CR0 register on page 156.

20.2 Architectural Description

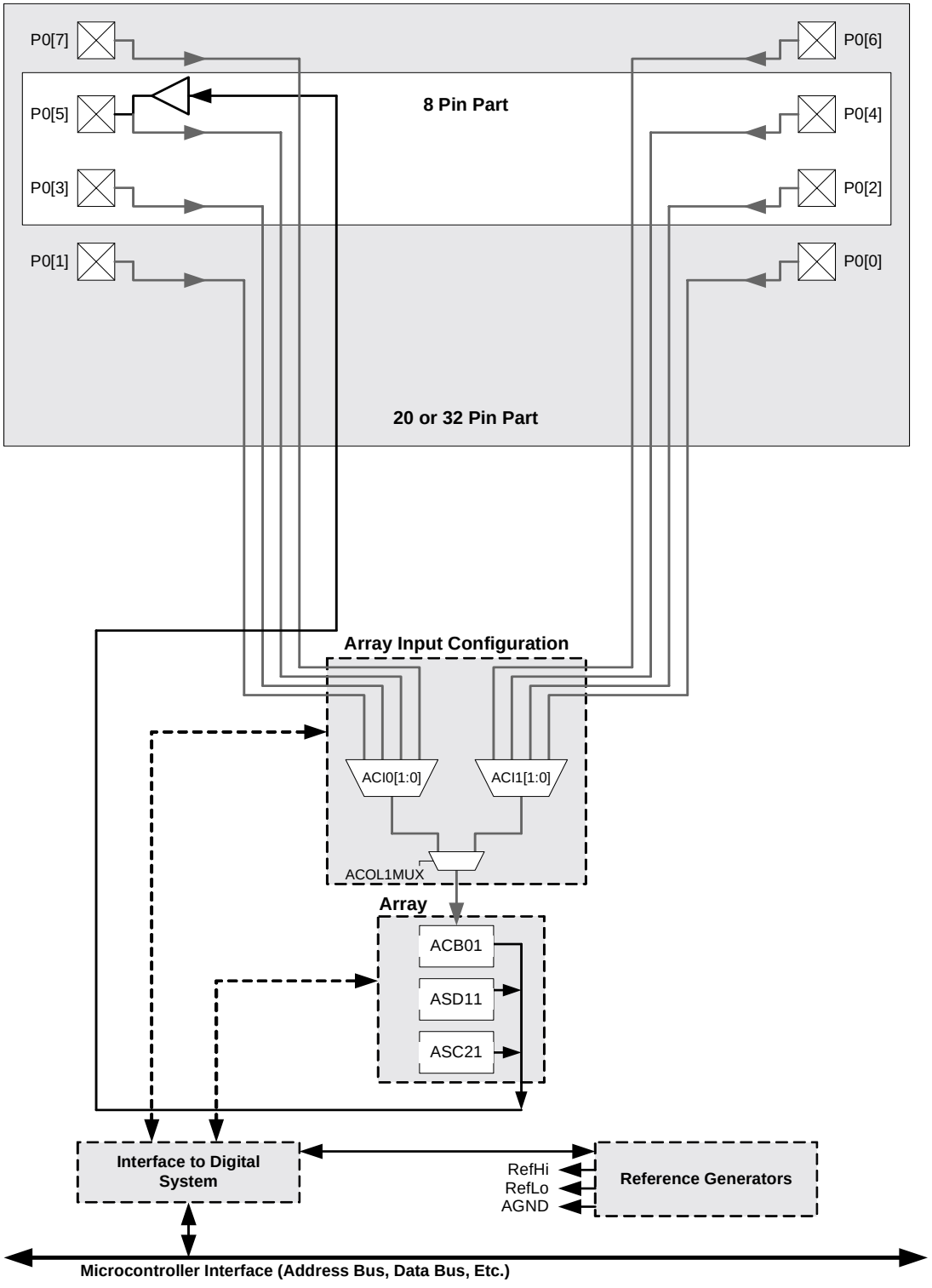


Figure 20-1. Analog Pin Block Diagram

21. Analog Reference

This chapter discusses the Analog Reference generator and its associated register. This device uses a fixed analog ground of Vdd/2.

Table 21-1. Analog Reference Register

Address	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

21.1 Architectural Description

The PSoC device is a single supply part, with no negative voltage available or applicable. 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 (i.e., span) of analog to digital (ADC) and digital to analog (DAC) converters.

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.

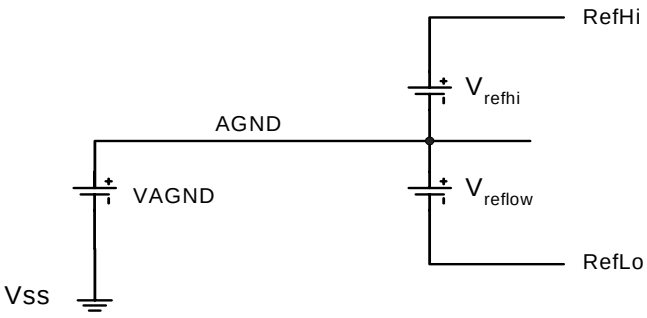


Figure 21-1. Reference Structure

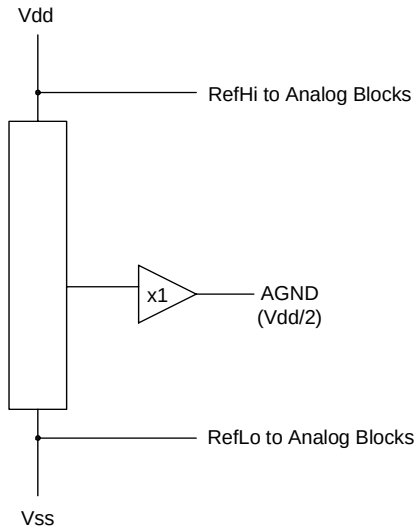


Figure 21-2. Analog Reference Control Schematic

21.2 Register Definitions

21.2.1 ARF_CR Register

Bit 7: Reserved.

Bit 6: HBE. Bias level (HBE) controls the bias level for all analog functions. It operates with the power setting in each block, to set the parameters of that block. Most applications will benefit most from the low bias level. At high bias, the analog block opamps have faster slew rate but slightly less voltage swing and higher power. **Bits 5 to 3: REF[2:0].**

Only the 010b setting is valid in the CY8C22xxx device. This sets AGND to Vdd/2, RefHi=Vdd, and RefLo=Vss.

Bits 2, 1, and 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. See Table 21-2.

Table 21-2: Analog Array Power Control Bits

PWR[2:0]	CT Row	Both SC Rows	REF Bias
000b	Off	Off	Off
001b	On	Off	Low Bias
010b	On	Off	Medium Bias
011b	On	Off	High Bias
100b	Off	Off	Off
101b	On	On	Low Bias
110b	On	On	Medium Bias
111b	On	On	High Bias

For additional information, reference the ARF_CR register on page 102.

22. Switched Capacitor 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.

Table 22-1. Analog Switched Capacitor Block Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
Analog Switch Cap Type C PSoC Block Control Registers										
x,xxh	ASCxxCR0	FCap	ClockPhase	ASign	ACap[4:0]					RW : 00
x,xxh	ASCxxCR1	ACMux[2:0]			BCap[4:0]					RW : 00
x,xxh	ASCxxCR2	AnalogBus	CompBus	AutoZero	CCap[4:0]					RW : 00
x,xxh	ASCxxCR3	ARefMux[1:0]		FSW1	FSW0	BMuxSC[1:0]		PWR[1:0]		RW : 00
Analog Switch Cap Type D PSoC Block Control Registers										
x,xxh	ASDxxCR0	FCap	ClockPhase	ASign	ACap[4:0]					RW : 00
x,xxh	ASDxxCR1	AMux[2:0]			BCap[4:0]					RW : 00
x,xxh	ASDxxCR2	AnalogBus	CompBus	AutoZero	CCap[4:0]					RW : 00
x,xxh	ASDxxCR3	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.
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 “Analog Register Summary” on page 221.

The Analog Switched Capacitor blocks are built around a rail-to-rail, input and output, low offset and low noise opamp. There are several analog 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.

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.

The four arrays are labeled A Cap Array, B Cap Array, C Cap Array, and F Cap Array, and 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 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 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 column. This signal may also be routed externally through the output buffer.
2. The comparator bus (CBUS), which is a digital bus resource shared by all of the analog blocks in the column.
3. The local output bus (OUT), which is an analog node which is routed to neighboring block inputs.

22.1 Architectural Description

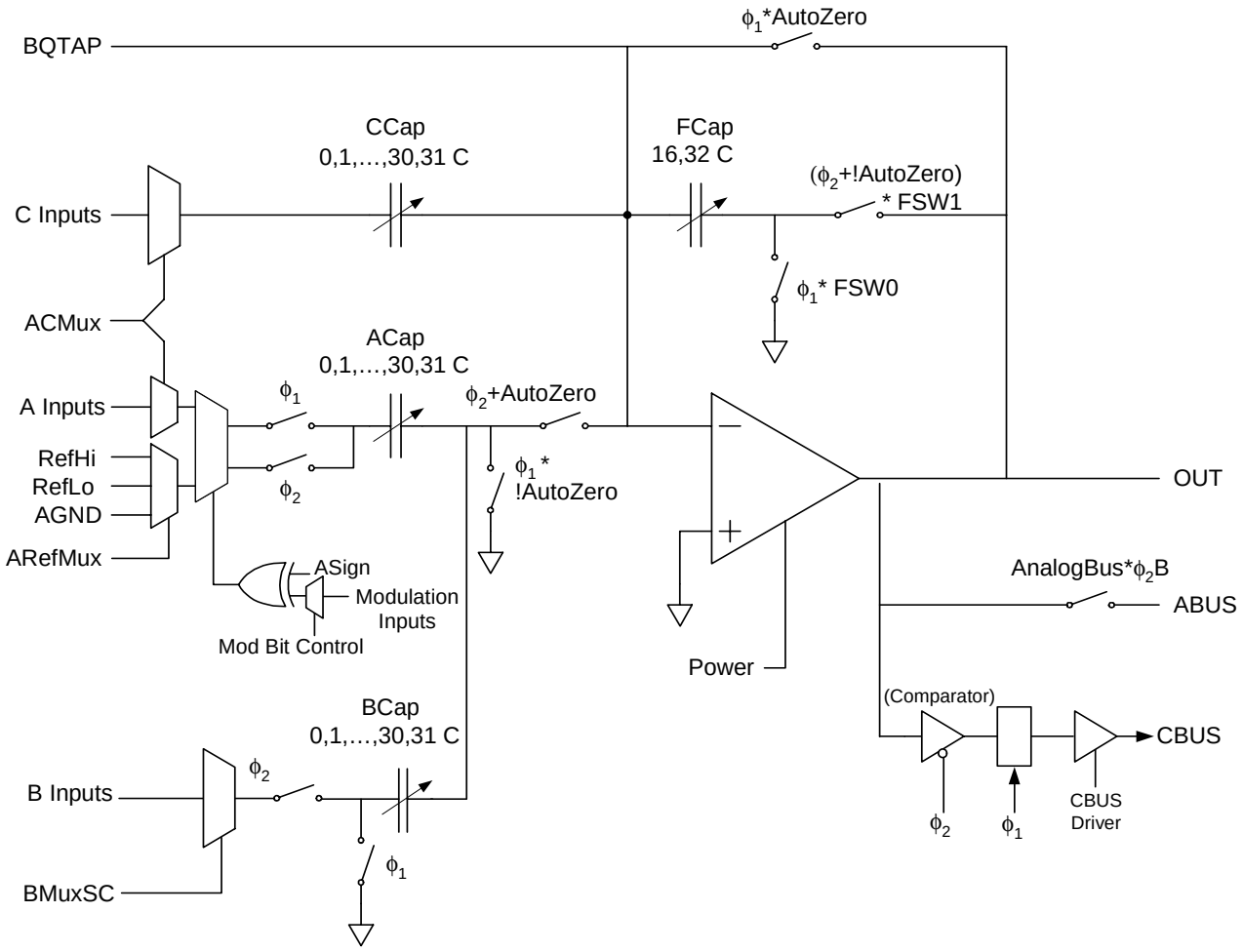


Figure 22-1. Analog Switch Cap Type C PSoC Blocks

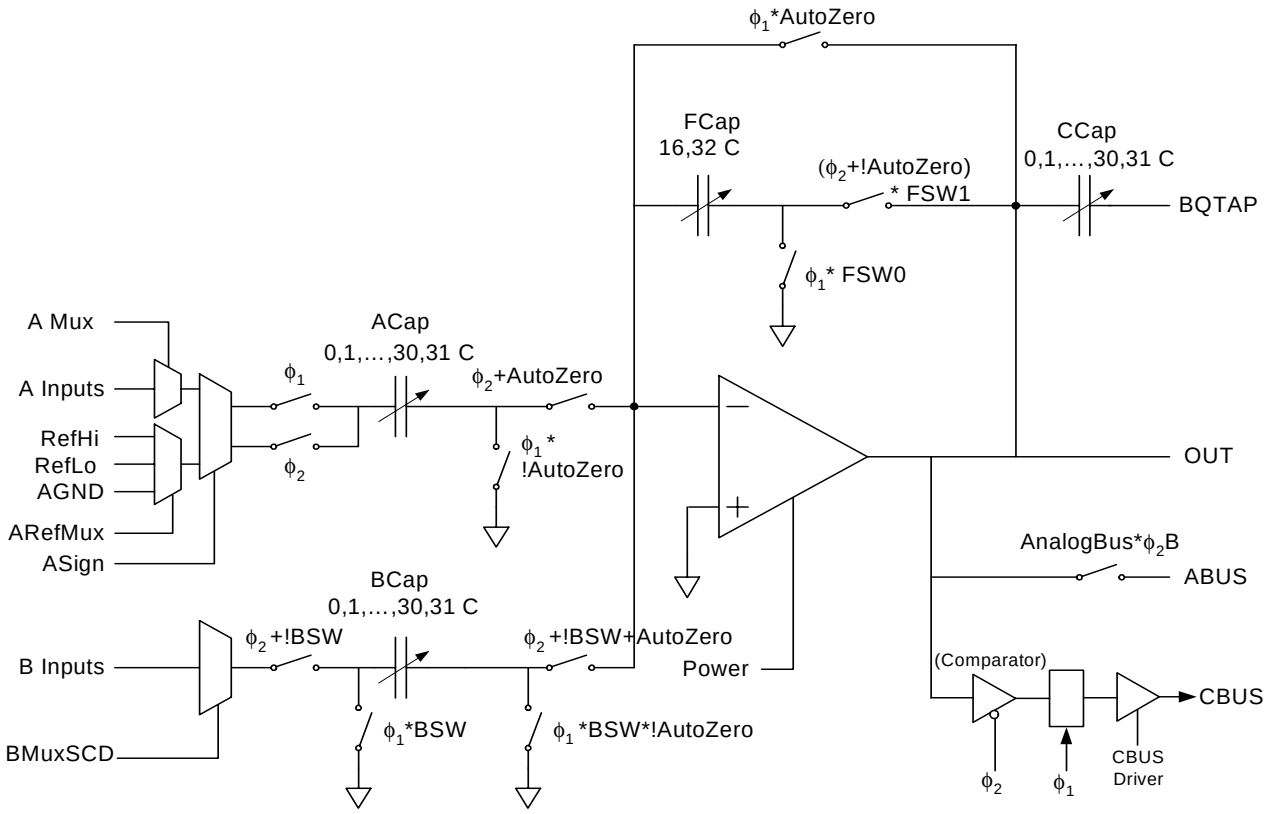


Figure 22-2. Analog Switch Cap Type D PSoC Blocks

22.2 Application Description

The Analog Switched Capacitor blocks support Delta-Sigma, Successive Approximation, and Incremental A/D 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 vertical between the SC blocks for the CY8C22xxx. Since the input of SC Block C has this additional switched capacitor, it is configured for the input stage of such a switched capacitor biquad filter. When followed by an SC Block D Integrator, this combination of blocks can be used to provide a full Switched Capacitor biquad.

22.3 Register Definitions

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 ASDxxCRx registers. Figure 22-3 illustrates the switch settings for the example ACap[4:0]=14h=10100b=20d.

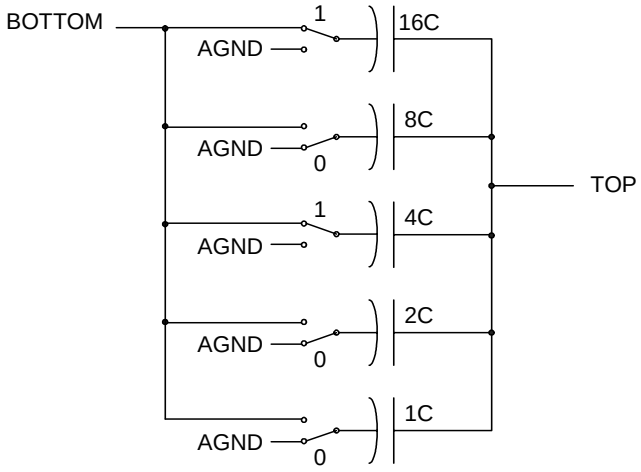


Figure 22-3. Example Switch Capacitor Settings

Analog Switch Cap Type C PSoC Block Control Registers

22.3.1 ASCxxCR0 Register

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 Control 2 Register (ASC21CR2).

Bit[6] 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).

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 busses 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, and the rising edge of the other clock phase will cause the value to come out (output point event). This bit determines which clock phase triggers the capture point event, and the other clock will trigger the output point event. The value output to the comparator bus will remain stable between output point events.

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, reference the ASCxxCR0 register on page 110.

22.3.2 ASCxxCR1 Register

Bits 7, 6, and 5: ACMUX[2:0]. ACMux controls the input muxing for both the A and C capacitor branches. The high order bit, ACMux[2], selects one of two inputs for the C branch.

Bits 4 to 0: BCap[4:0]. The BCap bits set the value of the capacitor in the B path.

For additional information, reference the ASCxxCR1 register on page 111.

22.3.3 ASCxxCR2 Register

Bit 7: AnalogBus. This bit gates the output to the analog column bus. The output on the analog column bus is affected by the state of the ClockPhase bit in Control 0 Register (ASC21CR0). 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 last part of PHI2. If the ClockPhase bit is 1, the block output continuously drives the analog column bus (ABUS).

Bit 6: CompBus. This bit controls the output to the column comparator bus (CBUS). Note that if the comparator bus is not driven by anything in the column, it is pulled low. The comparator output is evaluated on the rising edge of internal PHI1 and is latched so it is available during internal PHI2.

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 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, reference the ASCxxCR2 register on page 112.

22.3.4 ASCxxCR3 Register

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 Control 2 Register (ASC21CR2). 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 bit 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, reference the ASCxxCR3 register on page 113.

Analog Switch Cap Type D PSoC Block Control Registers

22.3.5 ASDxxCR0 Register

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 Control 2 Register (ASD11CR2).

Bit[6] 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).

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

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

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, reference the ASDxxCR0 register on page 114.

22.3.6 ASDxxCR1 Register

Bits 7, 6, and 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, reference the ASDxxCR1 register on page 115.

22.3.7 ASDxxCR2 Register

Bit 7: AnalogBus. This bit gates the output to the analog column bus. The output on the analog column bus is affected by the state of the ClockPhase bit in Control 0 Register (ASD11CR0,). If AnalogBus is set to 0, the output to the analog column bus is tri-stated. If AnalogBus is set to 1, the ClockPhase bit selects the signal that is output to the analog-column bus. If the ClockPhase bit is 0, the block output is gated by sampling clock on last part of PHI2. If the ClockPhase bit is 1, the block ClockPhase continuously drives the analog column bus (ABUS).

Bit 6: CompBus. This bit controls the output to the column comparator bus (CBUS). Note that if the comparator bus is not driven by anything in the column, it is pulled low. The comparator output is evaluated on the rising edge of internal PHI1 and is latched so it is available during internal PHI2.

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 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, reference the ASDxxCR2 register on page 116.

22.3.8 ASDxxCR3 Register

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 Control 2 Register (ASD11CR2,). 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, reference the ASDxxCR3 register on page 117.

23. Continuous Time Block

This chapter discusses the Analog Continuous Time Block and its associated registers. This block supports Programmable Gain or attenuation Opamp Circuits; Instrumentation Amplifiers, using two CT Blocks (Differential Gain); Continuous Time high-frequency, anti-aliasing filters; and modest response-time analog comparators.

Table 23-1. Analog Continuous Time Block Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,7xh	ACBxxCR3					LPCMPEN	CMOUT	INSAMP	EXGAIN	RW : 00
x,7xh	ACBxxCR0	RTapMux[3:0]				Gain	RTopMux	RBotMux[1:0]		RW : 00
x,7xh	ACBxxCR1	AnalogBus	CompBus	NMux[2:0]			PMux[2:0]			RW : 00
x,7xh	ACBxxCR2	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.
x: 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 “Analog Register Summary” on page 221.

23.1 Architectural Description

The Analog Continuous Time blocks are built around a rail-to-rail, input and output, low offset and low noise opamp. There are several analog muxes controlled by register-bit settings in the control registers that determine the signal topology inside the block. There is also a precision resistor matrix located in the feedback path of the opamp and 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 that is shared by all of the analog blocks in the analog column for that block. This signal may also be routed externally through an output buffer.
2. The comparator bus (CBUS), which is a digital bus that is a resource that is shared by all of the analog blocks in a column for that block.
3. The local output bus (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.

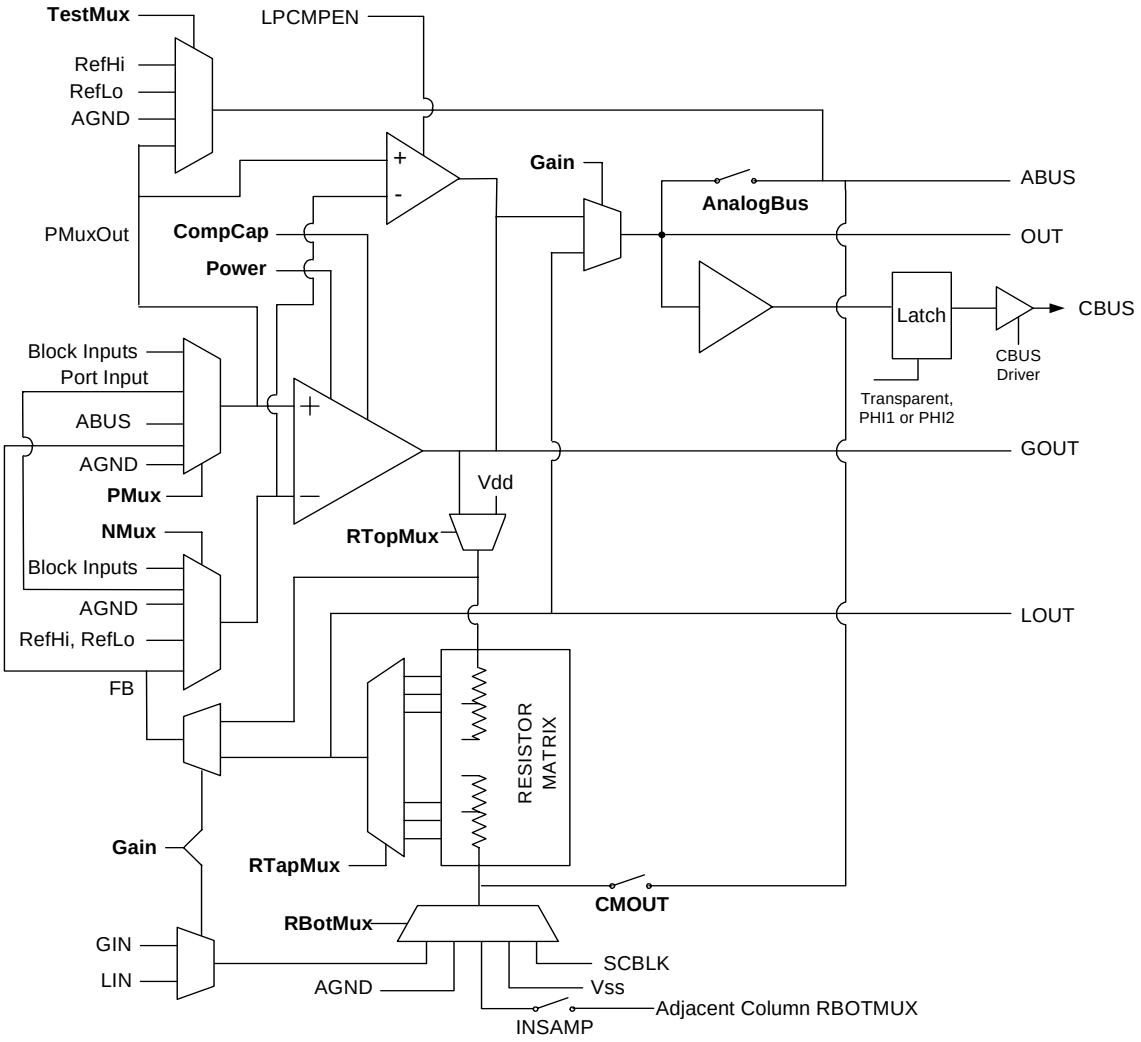


Figure 23-1. Analog Continuous Time Block Diagram

23.2 Register Definitions

23.2.1 ACBxxCR0 Register

Bits 7 to 4: RTapMux[3:0]. These bits, in combination with the EXGAIN bit, B0, of the CR3 register, control the center tap of the resistor string.

Bit 3: Gain. This bit controls whether the resistor string is connected around the opamp as for gain (center tap to inverting opamp input) or for loss (center tap to output of the block). Note that setting Gain alone does not guarantee a gain or loss block. Routing of the other ends of the resistor 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: RBotBux[1:0]. These bits, in combination with the INSAMP bit, B1, of the CR3 register, control the connection of the bottom end of the resistor string.

For additional information, reference the ACBxxCR0 register on page 107.

23.2.2 ACBxxCR1 Register

Bit 7: AnalogBus. This bit controls the analog output bus. 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 will be driven low externally to the blocks.

Bits 5, 4, and 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 an internal feedback selection.

Bits 2, 1, and 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 an internal feedback selection.

For additional information, reference the ACBxxCR1 register on page 108.

23.2.3 ACBxxCR2 Register

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 the compensation capacitor is switched in or not in the opamp. By not switching in the compensation capacitance, a much faster response can be obtained, if the amplifier is being used as a comparator.

Bit 4: TMUXEN. If the TMUXEN bit is high, then the value of TestMux[1:0] determines which testmux input is connected to the ABus for that particular continuous time block. If the TMUXEN bit is low, then none of the testmux inputs are connected to the ABus regardless of the value of TestMux[1:0].

Bits 3 and 2: TextMux[1:0]. TestMux selects block bypass mode.

Bits 1 and 0: PWR[1:0]. Power is encoded to select 1 of 3 power levels or power down (Off). The blocks power up in the off state. Combined with the Turbo mode, this provides 6 power levels. Turbo mode is controlled by the HBE bit of the Analog Reference Control Register.

For additional information, reference the ACBxxCR2 register on page 109.

23.2.4 ACBxxCR3 Register

Bits 7 to 4: Reserved.

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 situations where low power is more important than low noise, low offset, and high speed. 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 ACBxxCR2 and ARF_CR.

Bit 2: CMOUT. The analog array may be used to build two different forms of instrumentation amplifiers. Two continuous time blocks combine to make a 2-opamp instrumentation amplifier (see Figure 23-2).

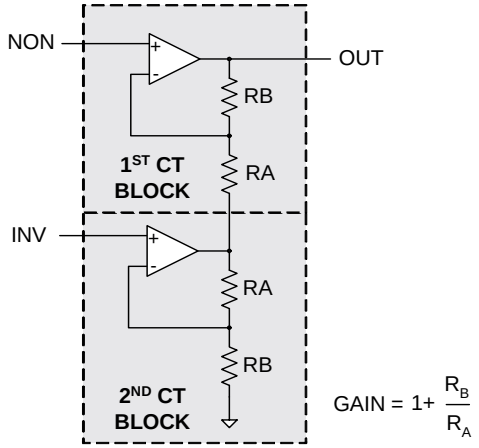


Figure 23-2. 2-Opamp Instrumentation Amplifier

Two continuous time blocks and one switched capacitor block combine to make a 3-opamp instrumentation amplifier (see Figure 23-3).

The 3-opamp instrumentation amplifier takes more resources, but handles a larger common mode input range. Bit2 (CMOUT) and bit1 (INSAMP) control switches are involved in the 3-opamp instrumentation amplifier. If bit2 (CMOUT) 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 3-opamp instrumentation amplifier.

Bit 1: INSAMP. This bit is used to connect the resistors of two continuous time blocks as part of a 3-opamp instrumentation amplifier. The INSAMP bit must be high for the 3-opamp instrumentation amplifier (see Figure 23-3).

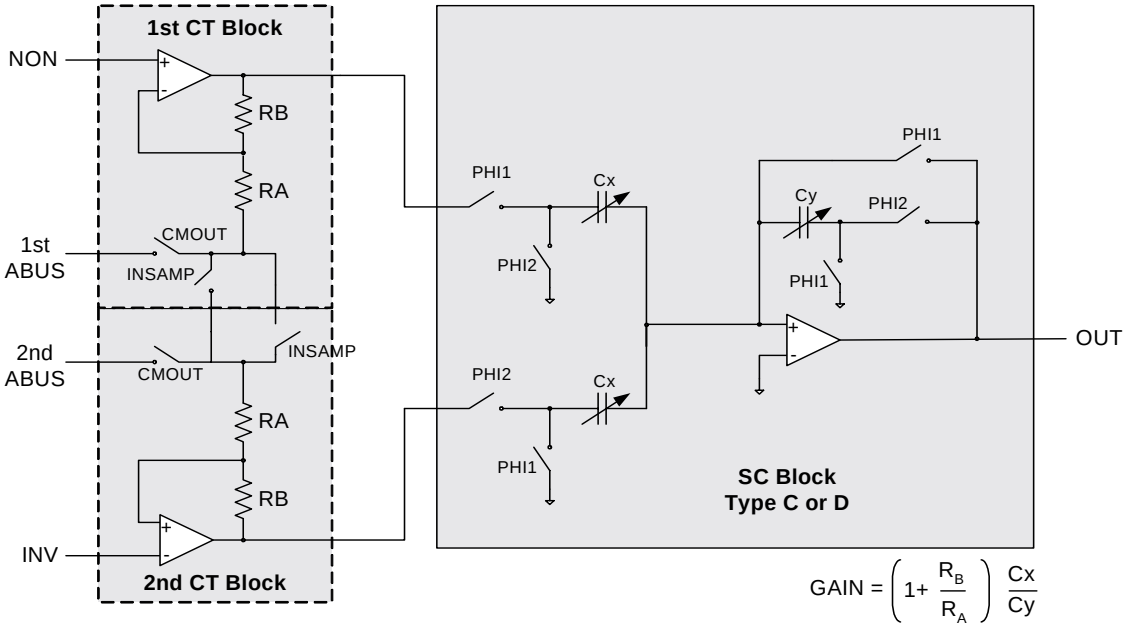


Figure 23-3. 3-Opamp Instrumentation Amplifier

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 details. The EXGAIN bit allows one additional tap selection for RtapMux = 01h and a second additional tap selection for RtapMux = 00h (see Figure 23-4).

For additional information, reference the ACBxxCR3 register on page 106.

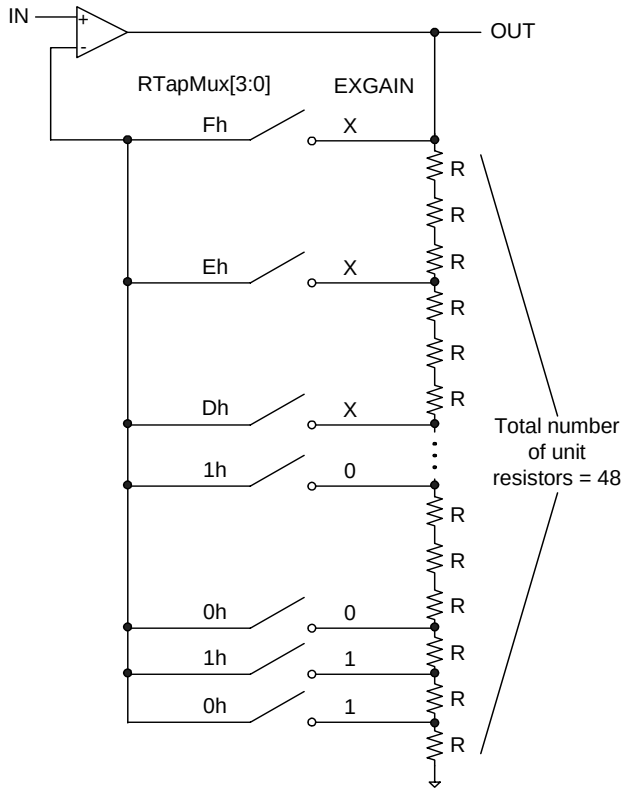


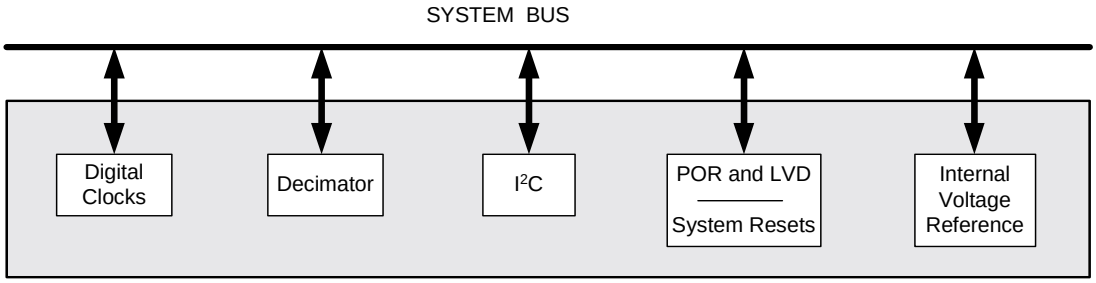
Figure 23-4. Continuous Time Block in Gain Configuration

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.



PSoC System Resources Block Diagram

System Resources Register Summary

The table below lists all the PSoC registers that the system resources of the device and its individual blocks use.

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										
0,DAh	INT_CLR0	VC3	Sleep	GPIO			Analog 1		V Monitor	RW : 00
0,E0h	INT_MSK0	VC3	Sleep	GPIO			Analog 1		V Monitor	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	IMODIS	SYSCLKX2 DIS	RW : 00
DECIMATOR REGISTERS										
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			ICLKS1			DCLKS1	RW : 00
I2C REGISTERS										
0,D6h	I2C_CFG		PSelect	Bus Error IE	Stop IE	Clock Rate		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
POR AND LVD REGISTERS										
1,E3h	VLT_CR			PORLEV[1:0]		LVDTBEN	VM[2:0]			RW : 00
1,E4h	VLT_CMP							LVD	PPOR	R : 00
INTERNAL VOLTAGE REFERENCE REGISTER										
1,EAh	BDG_TR			TC[1:0]		V[3:0]				RW:00
SYSTEM RESET REGISTERS										
0,FEh	CPU_SCR1								IRAMDIS	RW : 00
0,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	RW : XX

LEGEND
C: Cearable register or bits.
X: The value for power on reset is unknown.

24. Digital Clocks

This chapter discusses the Digital Clocks and its 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 chapters in the section called “CORE ARCHITECTURE” on page 31.

Table 24-1. Digital Clocking 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			Analog 1		V Monitor	RW : 00
0,E0h	INT_MSK0	VC3	Sleep	GPIO			Analog 1		V Monitor	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	IMODIS	SYSCLKX2 DIS	RW : 00

24.1 Architectural Description

The PSoC M8C core has a large number of clock sources that increase the flexibility of the PSoC mixed signal arrays, as illustrated in Figure 24-1.

24.1.1 Internal Main Oscillator

The Internal Main Oscillator (IMO) is the foundation upon which almost all other clock sources in the PSoC mixed signal arrays are based. The default mode of the IMO creates a 24 MHz reference clock that is used by many other circuits in the chip. 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 will become the base for all of the clocks the IMO normally serves.

Whether the external clock or the internal main oscillator is selected, all chip functions are clocked from a derivative of SYSCLK or are re-synchronized to SYSCLK. All external asynchronous signals (through row inputs), as well as the selected 32K oscillator, are resynchronized to SYSCLK for use in the digital blocks.

The IMO is discussed in detail in the chapter “Internal Main Oscillator (IMO)” on page 63.

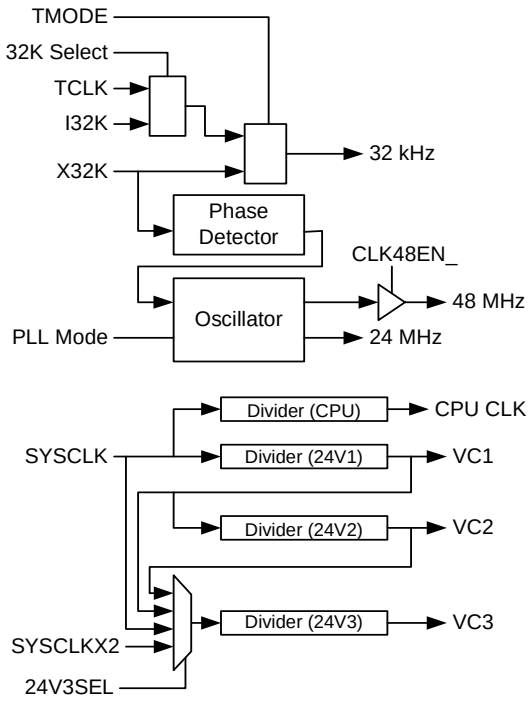


Figure 24-1. Overview of PSoC Clock Sources

24.1.2 Internal Low Speed Oscillator

The internal low speed oscillator (ILO), or sometimes referred to as the Sleep Oscillator, is always on unless the device is operating off a 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 65.

24.1.3 32 kHz Crystal Oscillator

The PSoC may be configured to use an external crystal. When configured in this way the internal low speed oscillator is turned off and the crystal becomes the clock source for all 32 kHz clocks.

The Crystal Oscillator is discussed in detail in the chapter "32 kHz Crystal Oscillator (ECO)" on page 67.

24.1.4 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 mixed signal arrays.

Pin P1[4] has been chosen as 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), and it is also not physically close to either the P1[0] and P1[1] crystal pins.

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 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 chip 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.

24.1.4.1 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, may be used as a clock source for the digital 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 spec for the external clock input ensures that the internal circuitry of the digital blocks, which is clocked by SYSCLKX2, will meet timing. However, since the doubled clock is generated from both edges of the input clock, clock jitter will be introduced if the duty cycle deviates greatly from fifty percent. Also, the high time of the clock out of the doubler is fixed at 21 ns, so the duty cycle of SYSCLKX2 will be proportional to the inverse of the frequency, as shown in Figure 24-2. Regardless of the input frequency, the high period of SYSCLKX2 is 21 ns nominal.

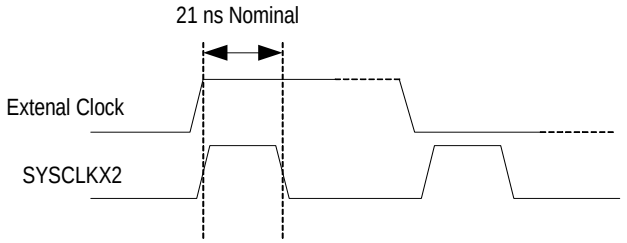


Figure 24-2. Operation of the Clock Doubler

24.1.4.2 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 chip resources run on clocks derived from or synchronized to SYSCLK, when the switch is made, analog and digital functions may be momentarily interrupted.

When a switch is made from the IMO to the external clock, the IMO may be turned off to save power. This can be done by setting the IMODIS bit and may be done immediately after the instruction that sets the EXTCLKEN bit. However, when switching back from an external clock to the IMO, the IMODIS bit must be cleared, and a firmware delay implemented. This gives the IMO sufficient start-up time before the EXTCLKEN bit may be cleared.

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 24-3, 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 setup 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 24-4, 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 and therefore, the setup time from SYSCLK to disable is one-half SYSCLK.

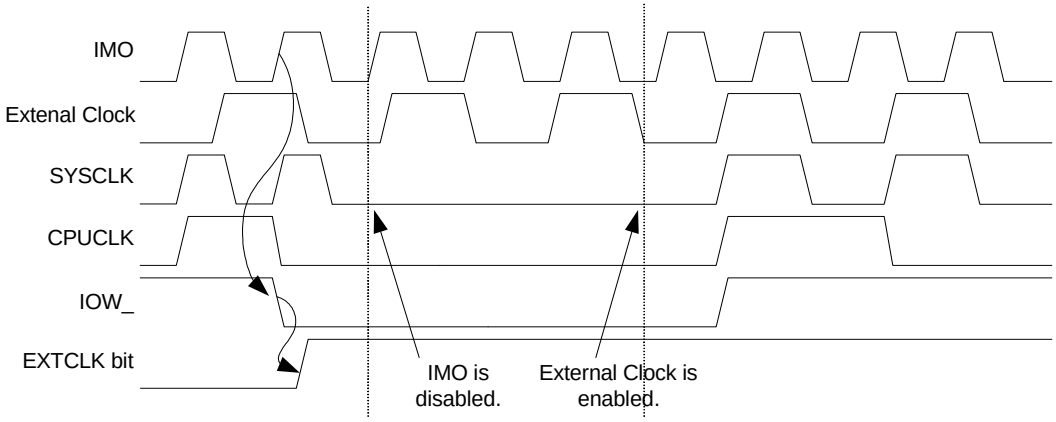


Figure 24-3. Switch from IMO to the External Clock with a CPU Clock Divider of Two or Greater

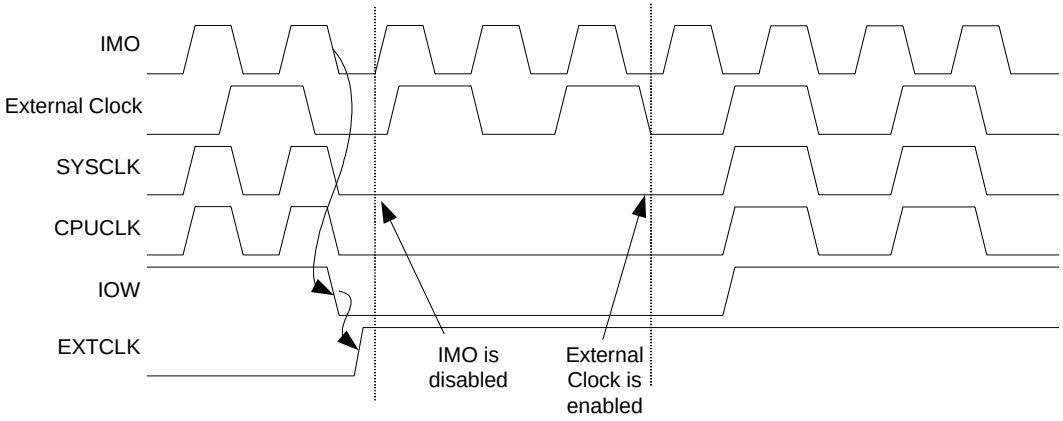


Figure 24-4. Switch from IMO to External Clock with the CPU Running with a CPU Clock Divider of One

24.2 Register Definitions

24.2.1 INT_CLR0 Register

The INT_CLR0 register holds bits that are used by several different resources. The digital clocks only use bit 7 of the INT_CLR0 register for the VC3 clock and bits zero through six are used by other resources. For a full discussion of the INT_CLR0 register, see the INT_CLRx Register in the Interrupt Controller chapter.

For additional information, reference the INT_CLR0 register on page 129.

24.2.2 INT_MSK0 Register

The INT_MSK0 register holds bits that are used by several different resources. The digital clocks only use bit 7 of the INT_MSK0 register for the VC3 clock and bits zero through six are used by other resources. The Sleep bit (bit 6) controls whether the Sleep timer may be used as an interrupt source. For a full discussion of the INT_MSK0 register, see the INT_MSKx Register in the Interrupt Controller chapter.

For additional information, reference the INT_MSK0 register on page 134.

24.2.3 OSC_CR0 Register

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

Bit 6: PLL Mode. This bit, in the OSC_CR0 register, is the only bit that directly influences the PLL. When set, this bit 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 chip systems are powered down, including the Band Gap reference. However, to facilitate the detection of POR and LVD events at a rate higher than the Sleep Interval, the Band Gap circuit is powered up periodically for about 60 us 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 Band Gap 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 24-2. It must be remembered that when the ILO is the selected 32 kHz clock source, sleep intervals are approximate.

Table 24-2. 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, 1, and 0: CPU Speed[2:0]. The PSoC M8C may operate over a range of CPU clock speeds (Table 24-3), 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 254 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-2 divide circuit, which are selected by a 3-bit code. At any given time, the CPU 8:1 clock multiplexer is selecting one of the available frequencies, which is re-synchronized 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 will be 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.

Table 24-3. OSC_CR0[2:0] Bits: CPU Speed

	Divider Source Clock	
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, reference the OSC_CR0 register on page 165.

24.2.4 OSC_CR1 Register

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 24 MHz system clock; however, if the device is configured to use an external clock, the input to the divider will be 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 24-4 lists all values for the VC1 clock net.

Table 24-4. OSC_CR1[7:4] Bits: VC1 Divider Value

	Divider Source Clock	
Bits	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

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 24-5 lists all values for the VC2 clock net.

Table 24-5. OSC_CR1[3:0] Bits: VC2 Divider Value

	Divider Source Clock	
Bits	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, reference the OSC_CR1 register on page 166.

24.2.5 OSC_CR2 Register

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.

Bits 6 to 3: Reserved.

Bit 2: EXTCLKEN. When the EXTCLKEN bit is set, the external clock becomes the source for the internal clock tree, SYSCLK, which drives most chip 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.

Bit 1: IMODIS. When set, the Internal Main Oscillator is disabled. If the doubler is enabled (SYSCLKX2DIS=0), the Internal Main oscillator will be forced on.

Bit 0: SYSCLKX2DIS. When set, the Internal Main Oscillator's doubler is disabled. This will result 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, reference the OSC_CR2 register on page 167.

24.2.6 OSC_CR3 Register

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 24-6. OSC_CR3[7:0] Bits: VC3 Divider Value

	Divider Source Clock			
Bits	**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

As mentioned previously 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 needs to 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 would occur every six periods of the VC3's input clock. Another example would be if the divider value was 00h (divide by 1), an interrupt would be 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 will always be posting. See the Interrupt Controller chapter for more information on posting and pending.

For additional information, reference the OSC_CR3 register on page 164.

24.2.7 OSC_CR4 Register

Bits 7 to 2: Reserved.

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 VC3 is most similar to the VC2 in that its input clock comes from a configurable source. As shown in Figure 24-1 on page 253, a 4-to-1 multiplexer determines the clock that will be used as in the input to the VC3 divider. The multiplexer 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 24-6.

Table 24-7. OSC_CR4[1:0] Bits: VC3

Bits	Multiplexer Output
00b	SYSCLK
01b	VC1
10b	VC2
11b	SYSCLKX2

It is important to remember that even though the VC3 divider has four choices for 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 multiplexer must be considered.

For additional information, reference the OSC_CR4 register on page 163

25. Decimator

This chapter briefly explains the PSoC Decimator and its associated registers. It serves as an overview of the clocking options available in the PSoC devices. The decimator block is a hardware assist for digital signal processing applications. The decimator may be used for sigma-delta analog to digital converters and incremental analog to digital converters.

Table 25-1. Decimator Registers

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
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			ICLKS1			DCLKS1	RW : 00

LEGEND
C: Cearable register or bits.
X: The value for power on reset is unknown.

25.1 Architectural Description

The decimator may perform 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 may be up to 16-bits long and is 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:

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 needs to 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).

25.2 Register Definitions

25.2.1 DEC_DH Register

The Decimator Data High register (DEC_DH) is a dual purpose register. 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 will be cleared regardless of the value written. Either the DEC_DH or DEC_DL register 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, reference the DEC_DH register on page 138.

25.2.2 DEC_DL Register

The Decimator Data Low register (DEC_DL) is a dual purpose register. When the register is read the least 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 lower byte of the 16-bit counter. The second function of the DEC_DL register is activated when ever the register is written: That function is to clear the decimator value. When the DEC_DL register is written the decimator's value will be cleared regardless of the value written. Either the DEC_DL or DEC_DH register 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, reference the DEC_DL register on page 139.

25.2.3 DEC_CR0 Register

This register contains control bits to access hardware support for both the Incremental ADC and the DELISG ADC. For Incremental support, the upper four bits, IGEN[3:0], select which column comparator bit will be 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 ICLKS1 in DEC_CR1. Up to one of 16 digital blocks may be selected, depending on your specific chip resources.

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).

In addition, 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 16 digital blocks (depending on your specific chip resources) with bit DCLKS0 and DCLKS1 in DEC_CR1.

For additional information, reference the DEC_CR0 register on page 140.

25.2.4 DEC_CR1 Register

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.

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 will cause the decimator output to be sampled. However, when the IDEC bit is set, the negative edge of the selected digital block input will cause the decimator value to be sampled.

Bits 5 and 4: Reserved.

Bit 3: ICLKS1. The ICLKS1 bit in this register selects the digital block sources for Incremental and DELSIGN ADC hardware support (see the DEC_CR0 register).

Bits 2 and 1: Reserved.

Bit 0: DCLKS1. The DCLKS1 bit in this register selects the digital block sources for Incremental and DELSIGN ADC hardware support (see the DEC_CR0 register).

For additional information, reference the DEC_CR1 register on page 141.

26. I²C

This chapter explains the I2C block and its associated registers. The I2C communications block is a serial processor designed to implement a complete I2C Slave and/or Master.

Table 26-1. I2C Registers

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		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

The I2C communications block is a serial to parallel processor designed to interface to the two-wire I2C serial communications bus. The block provides I2C specific support for status detection and generation of framing bits, to eliminate the need for excessive host processor intervention and overhead.

The I2C block will directly control the data (SDA) and clock (SCL) signals to the external I2C interface, through connections to two dedicated GPIO pins. The host firmware will interact with the block through IO register reads and writes, and firmware synchronization will be implemented through polling and/or interrupts.

Functionality requirements 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:

1. There is no hardware support for automatic address comparison. When Slave mode is enabled, every slave address will cause the block to interrupt the host and possibly stall the bus.
2. Since receive and transmitted data is not buffered, there is no support for automatic receive acknowledge. The host processor must intervene at the boundary of each byte and either send a byte or ACK received bytes.

26.1 Architectural Description

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 host will set up control options and issue commands to the unit through IO Writes and obtain 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 can work 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 Multi 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.

26.1.1 Basic I2C Data Transfer

Figure 26-1 shows the basic form of data transfers on the I2C bus with a 7-bit address format. (For a more detailed description, see the I2C Specification, Version 2.1, by Phillips Semiconductor).

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 R/W bit. The R/W 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 9th bit time. If the ACK is received, the transfer may proceed and the master can transmit or receive an indeterminate number of bytes, depending on the R/W 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 retry attempt.

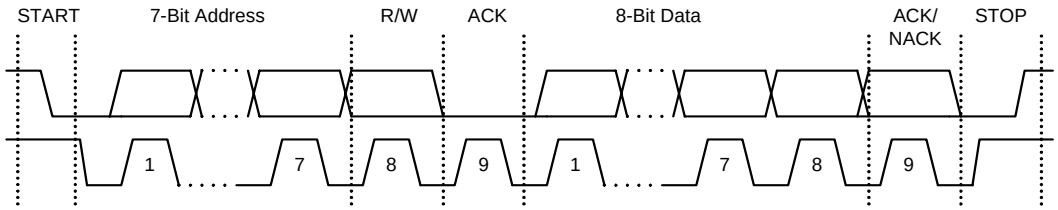


Figure 26-1. Basic I2C Data Transfer with 7-Bit Address Format

26.2 Application Description

26.2.1 Slave Operation

Assuming Slave mode is enabled, it is continually listening on the bus for a Start condition. When detected, the transmitted Address/R/W byte is received and read from the unit by firmware. At the point where eight bits of the address has been 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 host has had a chance to read the address byte and compare it to its own address. It will issue an ACK or NACK command based on that comparison.

If there is an address match, the R/W bit determines how the host will sequence the data transfer in Slave mode as shown in the two branches of Figure 26-2. I2C handshaking methodology (Slave holds the SCL line low to "stall" the bus) will be used as necessary, to give the host time to respond to the events and conditions on the bus. Figure 26-2 is a graphical representation of a typical data transfer from the slave perspective.

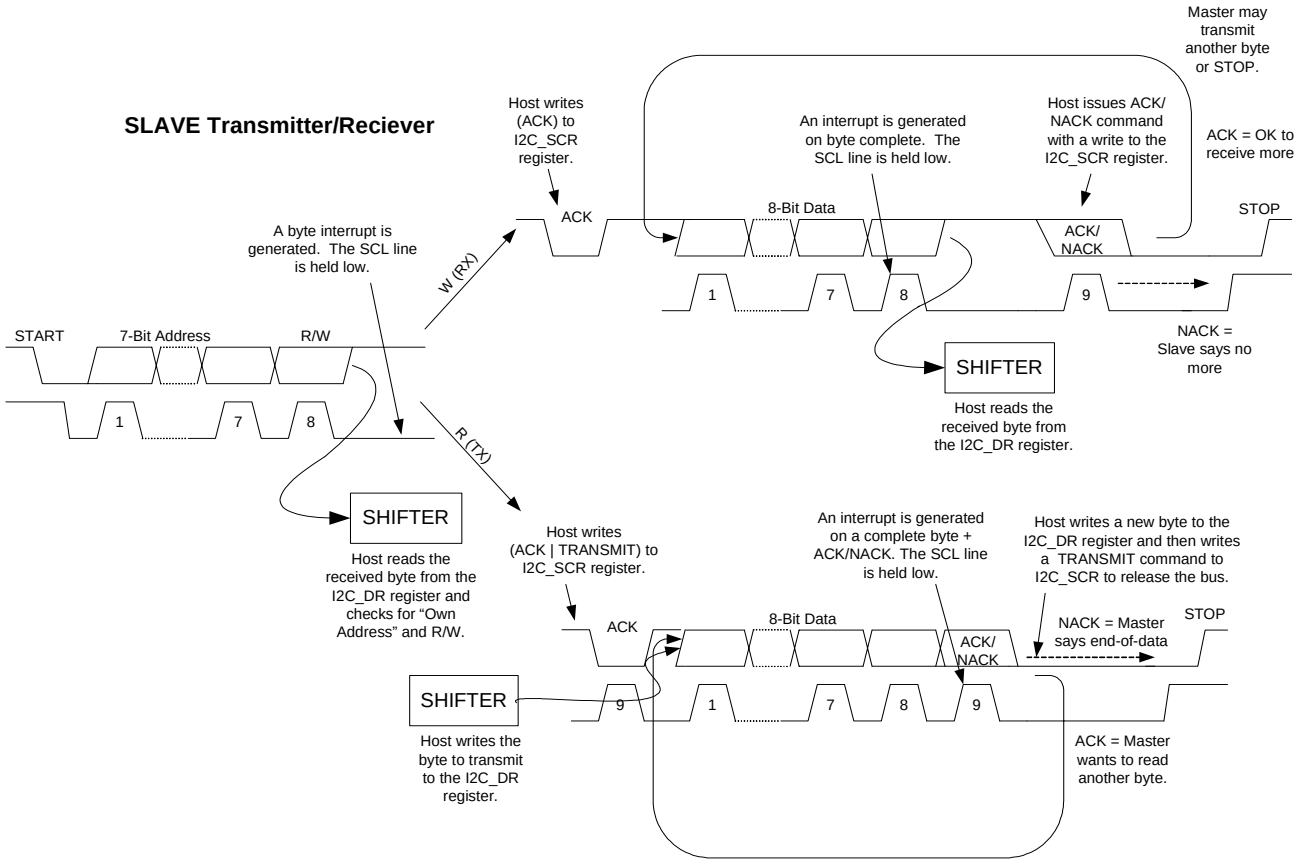


Figure 26-2. Slave Operation

26.2.2 Master Operation

To prepare for a Master mode transaction, the host must determine if the bus is free. This can be done by polling the BusBusy status. If busy, interrupts can be enabled to detect a Stop condition. Once it is determined that the bus is available, firmware should write the address byte into the I2C_DR register and set 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 bus on externally generated Start conditions.

In a Multi-Master environment there are two additional outcomes possible:

1. The host was too late to reserve the bus as a Master and another Master may have generated a Start and sent an Address/R/W byte. In this case, the unit as a Master will fail to generate a Start and will be forced into Slave mode. The Start will be pending and eventually occur at a later time when the bus becomes free. When the interrupt occurs in Slave mode, the host can determine that the Start command was unsuccessful by reading the I2C_MSCR register Start bit, which will be 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 will have the master's address data.
2. If another Master starts a transmission at the same time as this unit, arbitration will occur. If this unit loses the arbitration, the LostArb status bit will be set. In this case, the block will release the bus and switch to Slave operation. When the Start/Address interrupt occurs, the data register will have the winning master's address data.

Figure 26-3 is a graphical representation of a typical data transfer from the master perspective.

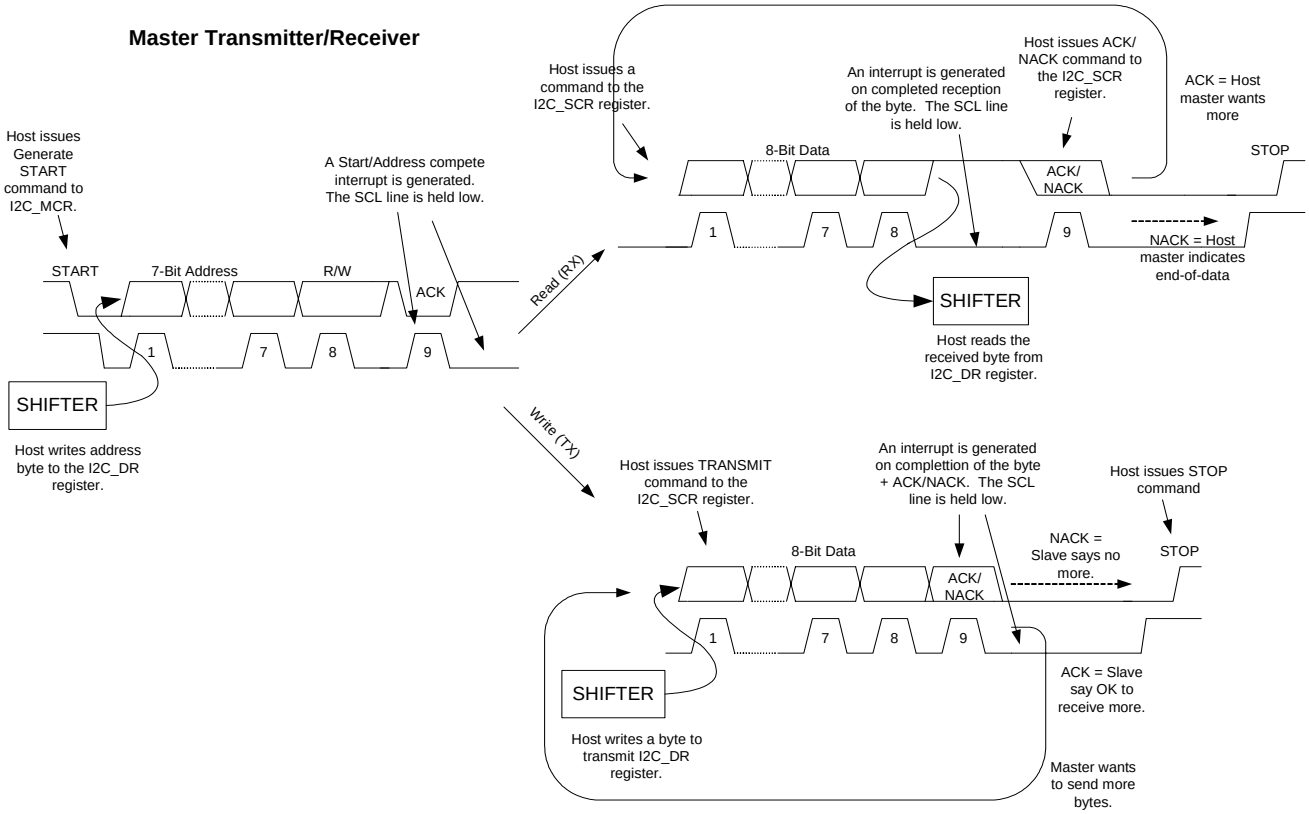


Figure 26-3. Master Operation

26.3 Register Definitions

The I2C block contains four registers, all of which reside in User IO space: Configuration Register (I2C_CFG), Status and Control Register (I2C_SCR), Master Status and Control Register (I2C_MSCR), and Data Register (I2C_DR).

The Configuration register is used to set the basic operating modes, baud rate, and selection of interrupts. The Status and Control 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. Data and address bytes are written to and read from the Data Register. The Master Status and Control register implements I2C framing controls and provides Bus Busy status.

26.3.1 I2C_CFG Register

This register is the I2C configuration register and contains the configuration bits for both Master and Slave mode operation. These bits control baud rate selection and optional interrupts. These values are typically set once for a given configuration. The bits in this register are all R/W.

Table 26-2. I2C_CFG Configuration Register

Bit	Access	Description	Mode
0	R/W	Enable Slave '0' = Disabled '1' = Enabled	Master/ Slave
1	R/W	Enable Master '0' = Disabled '1' = Enabled	Master/ Slave
3:2	R/W	Clock Rate 00 = 100K, Standard Mode 01 = 400K Fast Mode 10 = 50K Standard Mode 11 = Reserved	Master/ Slave
4	R/W	Stop IE Stop interrupt enable. 0 = Disabled. 1 = Enabled. An interrupt is generated on the detection of a Stop Condition.	Master Only
5	R/W	Bus Error IE Stop interrupt enable. 0 = Disabled. 1 = Enabled. An interrupt is generated on the detection of a Bus Error.	Master/ Slave
6	R/W	I2C Pin Select 0 = P1[7], P1[5] 1 = P1[1], P1[0]	Master/ Slave

Bit 7: Reserved.

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 I2C may be used on 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-circuit System Serial Programming (ISSP) is to be 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 going to be used, pins P1[1] and P1[0] will respond differently to a POR or XRES event than other IO pins. After an XRES, event both pins will be pulled down to the ground by going into the resistive zero drive mode before reaching the High-Z drive mode. After a POR event, P1[0] will drive out a one, then go to the resistive zero state for some time, and finally reach the High-Z drive mode state. After POR, P1[1] will go 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 will generate an interrupt. A Bus Error is typically a misplaced Start or Stop. See the Bus Error status bit description for a definition.

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) will reset the bus interface and synchronize to this signal. However, when the hardware detects a Bus Error in Master mode operation, the device will release the bus and transition to an idle state. In this case, a Master operation in progress will never have any further status or interrupts associated with it and therefore the Master may not be able to determine the status of that transaction. An immediate bus error interrupt will inform 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 will be generated until it is cleared by firmware and a subsequent Stop condition is generated.

Bits 3 and 2: Clock Rate. The Clock Rate bits offer a selection of four sampling and bit rates. All block clocking is based on the SYSCLK input, which is nominally 24 MHz. 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 26-3.

When clocking the input with a frequency other than 24 MHz (e.g., clocking the PSOC chip with an external clock), the baud rates and sampling rates will scale accordingly. Whether the block will work 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 I2C Specification, Version 2.1, by Phillips Semiconductor, for minimum Start and Stop hold times.)

Table 26-3. I2C 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 us
01b	Fast	/4	16	6 MHz/167 ns	375 kHz	1.33 us
10b	Standard	/16	32	1.5 MHz/667 ns	46.8 kHz	10.7 us
11b	Reserved					

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 will clock the I2C bus at one of four 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 will operate in Master Only mode. All external Start conditions will be ignored (although the Bus Busy status bit will still keep track of bus activity). Block enable will be synchronized to the SYSCLK clock input (see “Timing Diagrams” on page 270).

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, which indicates the beginning of an I2C transfer. When operating as a Slave, the block is clocked from an external Master and therefore, will work 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 setup 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 is cleared. See Figure 26-4 for a description of the interaction between the Master/Slave Enable bits. Block enable will be synchronized to the SYSCLK clock input (see “Timing Diagrams” on page 270).

For additional information, reference the I2C_CFG register on page 125.

Table 26-4. 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 will cause 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 will cause the block to revert to an idle state The I2C_MSCR register is held in reset.
Yes	No	Master Only Mode: In this mode, external Start conditions are ignored. 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 will generate a clock based on the setting in the Clock Rate register
Yes	Yes	Master/Slave Mode: In this mode, both Master and Slave may be operational. 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 will revert to Slave mode and the received byte will generate a Slave address interrupt.

26.3.2 I2C_SCR Register

This register is the I2C status and control register and is used to control both Master and Slave data transfer. It contains status bits for determining the state of the current I2C transfer, and control bits, which determine the actions for the next byte transfer. At the end of each byte transfer, the I2C hardware will interrupt the host processor and stall the bus on the subsequent low of the clock until the host intervenes with the next command. This register may be read as many times as necessary, but on a subsequent write, the bus stall will be released and the current transfer will continue.

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 26-5.

There are two control bits: Transmit and ACK. These bits have R/W access. These bits may also be cleared by hardware, as noted.

Table 26-5. I2C_SCR Status and Control Register

Bit	Access	Description	Mode
0	R/C	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 will automatically clear this bit.	Master/ Slave
1	R/C	LRB Last Received Bit. The value of the 9th bit in a Transmit sequence, which is acknowledge bit from the receiver. 0 = Last transmitted byte was ACKed by the receiver. 1 = Last transmitted byte was NACKed by the receiver. Any Start detect will automatically clear this bit.	Master/ Slave
2	R/W	Transmit 0 = Receive Mode. 1 = Transmit Mode. This bit is set by firmware to define the direction of the byte transfer. Any Start detect will automatically clear this bit.	Master/ Slave
3	R/C	Address 1 = The transmitted or received byte is an address. This status bit must be cleared by firmware with write of '0' to the bit position.	Master/ Slave
4	R/W	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
5	R/C	Stop Status 1 = A Stop condition was detected. This status bit must be cleared by firmware with write of '0' to the bit position. It is never cleared by the hardware.	Master/ Slave
6	R/C	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 will automatically clear this bit.	Master Only
7	R/C	Bus Error 1 = A misplaced Start or Stop condition was detected. This status bit must be cleared by firmware with write of '0' to the bit position. It is never cleared by the hardware.	Master Only

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 mentioned previously, 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 will initiate an address reception and a Stop will idle the Slave. In the case of a Master, this event will force 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.

Bit 6: Lost Arb. This bit is set when I2C bus contention is detected, during a Master mode transfer. Contention will occur when a Master is writing a '1' to the SDA output line and reading back a '0' on the SDA input line at 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.

The sequence occurs differently between Master transmitter and Master receiver. As a transmitter, the contention will occur on a data bit. On the subsequent Byte Complete interrupt, the Lost Arbitration status will be set. In receiver mode, the contention will occur on the ACK bit. The Master that NACKed the last reception will lose the arbitration. However, the hardware will shift 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 points 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 will remain set until a '0' is written back to it by the firmware. This bit may only be cleared if Byte Complete status is set. If the Stop Interrupt Enable bit is set, an interrupt will also be 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 will be 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 will determine the next bit of data that will be transmitted. It is active high. A '1' will send an ACK and a '0' will send 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 will read the received address from the data register and compare it with its own address. If the address does not match, the firmware will write a NACK indication to this register. No further interrupts will occur, until the next Address is received. If the address does match, firmware must ACK the received byte, then Byte Complete interrupts will be generated on subsequent bytes of the transfer.

This bit will also be set 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 will revert 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 will still occur 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 R/W 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 will send a byte from the data register and clock in an acknowledge bit from the receiver. On the subsequent byte complete interrupt, firmware will check 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 may send 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 may clear the transmit bit in the I2C_SCR register. This will automatically send a Stop. If the Start or Restart bits are set in the I2C_MSCR register, the Stop may be 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 should load the next byte into the I2C_DR register and set the transmit bit in the I2C_SCR register, to continue the transfer.

'1': NACK, the Master is done reading bytes. The Slave will revert 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 have been 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 host responds with a write back to this register before the subsequent falling edge of SCL (which is approximately one-half bit time), the transfer will continue without interruption. However, if the host is unable to respond within that time, the hardware will hold the SCL line low, stalling the I2C bus. In both Master and Slave mode, a subsequent write to the I2C_SCR register will release the stall.

For additional information, reference the I2C_SCR register on page 126.

26.3.3 I2C_DR Register

This register is the I2C data register and provides read/write access to the shift register. It is not buffered and therefore, writes and valid data reads may 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 be generated and the address shifted 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, reference the I2C_DR register on page 127.

26.3.4 I2C_MSCR Register

This register is the I2C master status and control register.

Table 26-6. I2C_MSCR Master Status and Control Register

Bit	Access	Description	Mode
0	R/W	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
1	R/W	Restart Gen 1 = Generate a Restart condition. This bit is cleared by hardware when the Start generation is complete.	Master Only

Table 26-6. I2C_MSCR Master Status and Control Register *(continued)*

Bit	Access	Description	Mode
2	RO	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
3	RO	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

Bits 7 to 4: Reserved.

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 may be polled by firmware to determine when a bus transfer may be 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 above 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 NACKed by the Slave, a Restart condition will be 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 will be generated after the Start and the address byte has been transmitted. If the address was ACKed by the receiver, the firmware may then proceed 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 will be generated. When reading the I2C_MSCR, the Master will see the Start Gen bit still set and the I2C_SCR will have the Address bit set, indicating that the block has been addressed as a Slave.

Slave mode is not enabled: The Start Gen bit will remain set and the Start will be queued, until the bus becomes free and the Start condition will be subsequently generated. An interrupt will be generated at a later time, when the Start and address byte has 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 will be generated. When reading the I2C_MSCR, the Master will see the Start Gen bit cleared, indicating that the Start was generated. However, the Lost Arb bit will be set in the I2C_SCR register. The Address status will also be set, indicating that the block has been addressed as a Slave. The firmware may then ACK or NACK the address to continue the transfer.

   Slave mode is not enabled: A Start and address byte interrupt will be generated. The Start Gen bit will be cleared and the Lost Arb bit will be set. The hardware will wait for command input, stalling the bus if necessary. In this case, the Master will clear the I2C_SCR register, to release the bus and allow the transfer to continue, and the block will idle.

Other cases where the Start bit may be used to generate a Start condition are as follows:

1. When a Master is finished with a transfer, a NACK will be written to the I2C_SCR register, in the case of the Master receiver, or the transmit bit will be cleared, in case of a Master transmitter. Normally, the action will free the stall and generate 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), will be generated. In this way, messages may be chained.
2. When a Master transmitter is NACKed, an automatic Stop condition is generated on the subsequent I2C_SCR write. However, if the Start Gen bit has previously been set, the Stop will be immediately followed by a Start conditon.

For additional information, reference the I2C_MSCR register on page 128.

26.4 Timing Diagrams

26.4.1 Clock Generation

Figure 26-4 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. As an additional option, the block may be clocked directly from SYSCLK, to achieve the highest baud rate. 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.

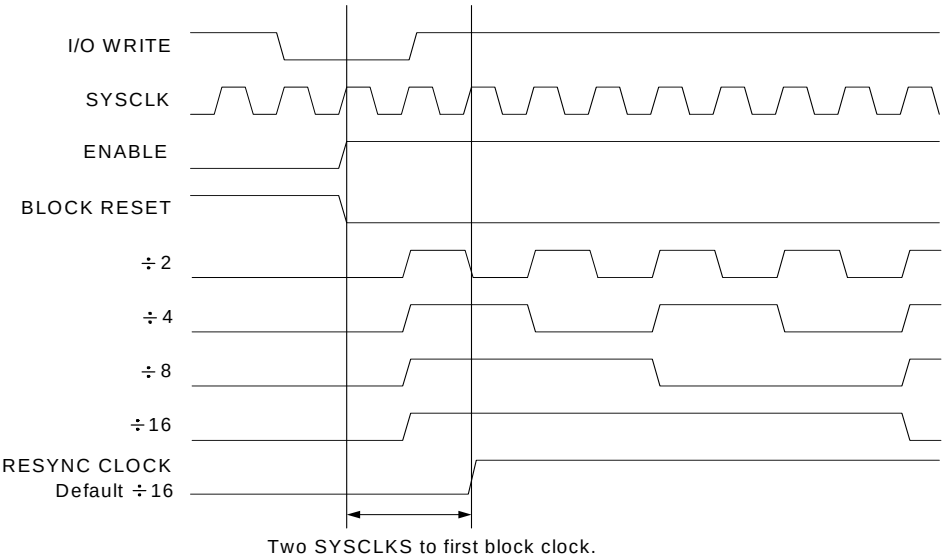


Figure 26-4. I2C Input Clocking

26.4.2 Enable and Command Synchronization

Figure 26-5 illustrates an all block reset (except for the I2C_CFG register) is asserted when the block is disabled. When either the Enable Master or Enable Slave bit is set, the block reset is negated on the next positive edge of SYSCLK, which ensures a full SYSCLK cycle of setup time on the next clock edge.

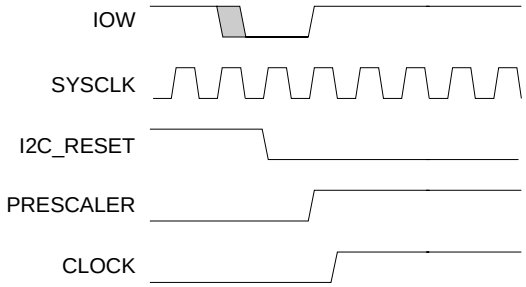


Figure 26-5. I2C Enable

Figure 26-6 illustrates a Start Gen command or I2C_SCR write after Byte Complete is resynchronized to the following block clock edge. I2C processing continues on the selected block clock following this resync clock edge.

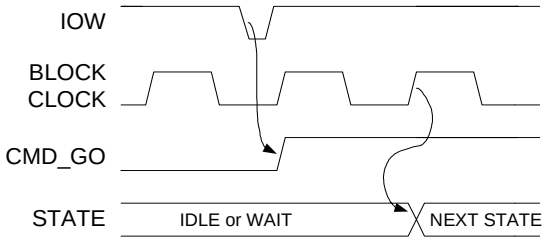


Figure 26-6. I2C Command

26.4.3 Basic Input/Output Timing

Figure 26-7 illustrates basic input output timing that is valid for both 16x sampling and 32x sampling. For 16x sampling, N=4 and for 32x sampling N=12. N is derived from the half bit rate sampling of eight and 16 clocks, respectively, minus the input latency of three (count of 4 and 12 correspond to 5 and 13 clocks).

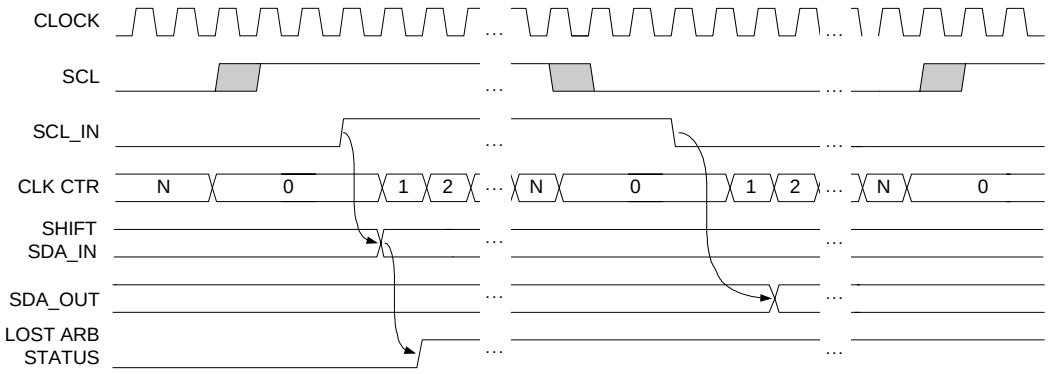


Figure 26-7. Basic Input/Output Timing

26.4.4 Status Timing

Figure 26-8 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.

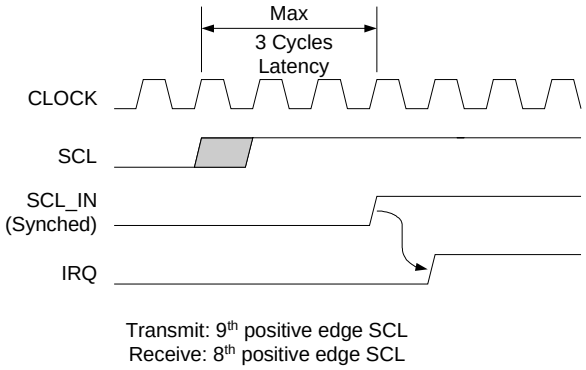


Figure 26-8. Byte Complete, Address, LRB Timing

Figure 26-9 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.

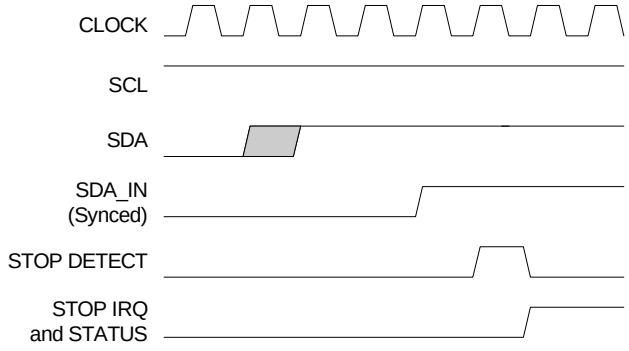


Figure 26-9. Stop Status and Interrupt Timing

Figure 26-10 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).

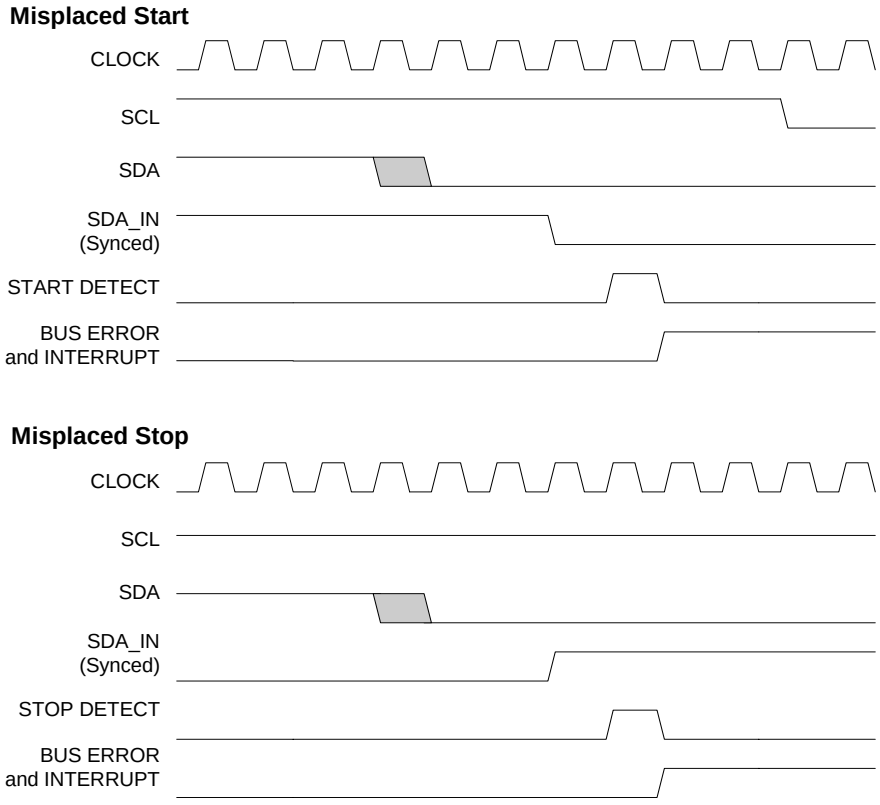


Figure 26-10. Bus Error Interrupt Timing

26.4.5 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 setup time to the next clock edge. When the Start is initiated, the SCL line is left high for 6/14 clocks (corresponding to 16/32X 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 will automatically initiate a new Start sequence.

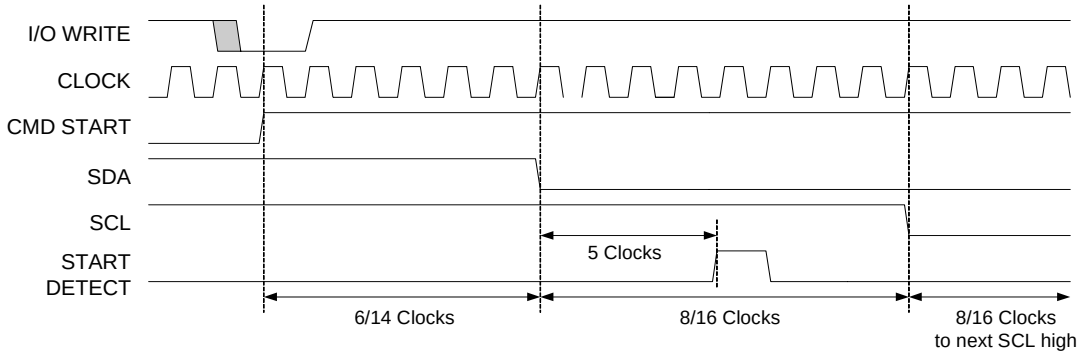


Figure 26-11. Basic Master Start Timing

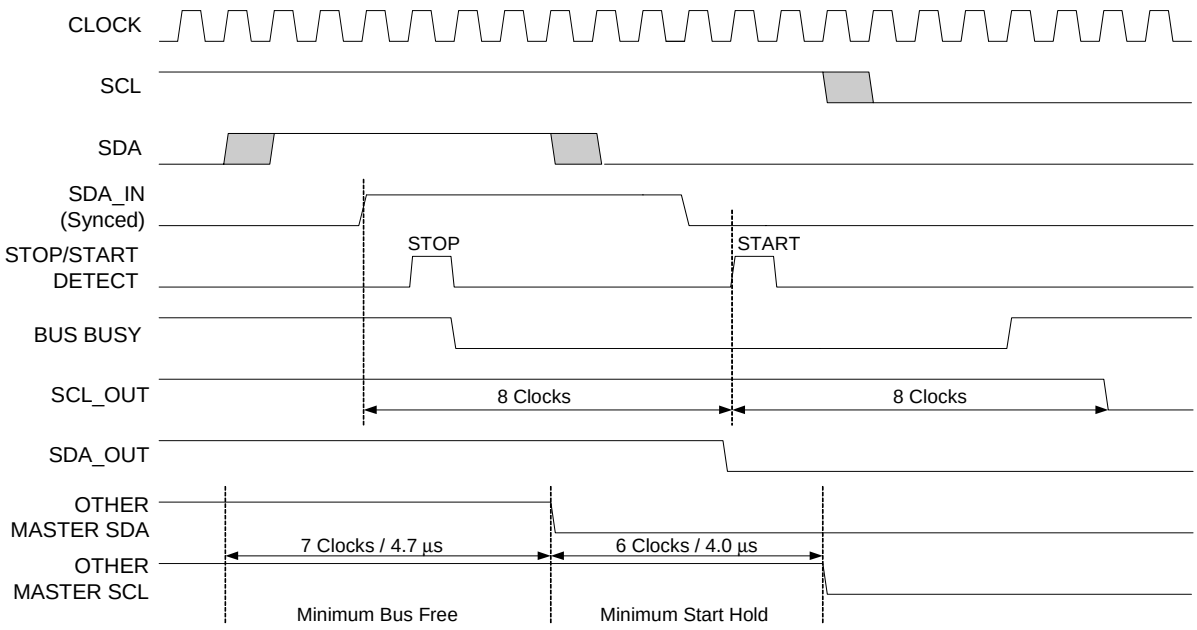


Figure 26-12. Start Timing with a Pending Start

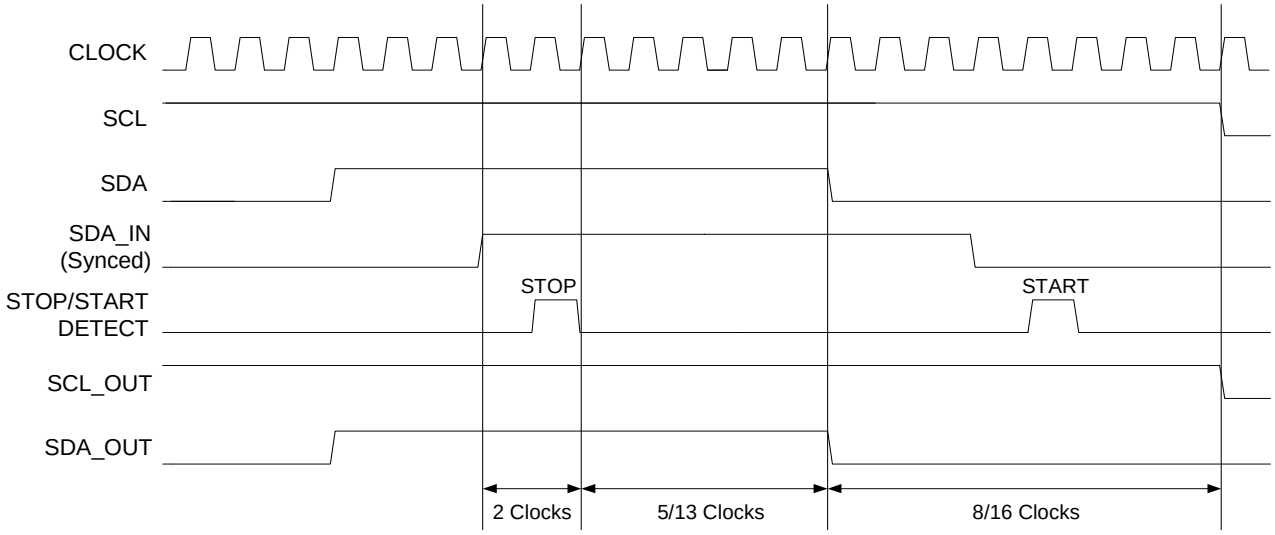


Figure 26-13. Master Stop/Start Chaining

26.4.6 Master Restart Timing

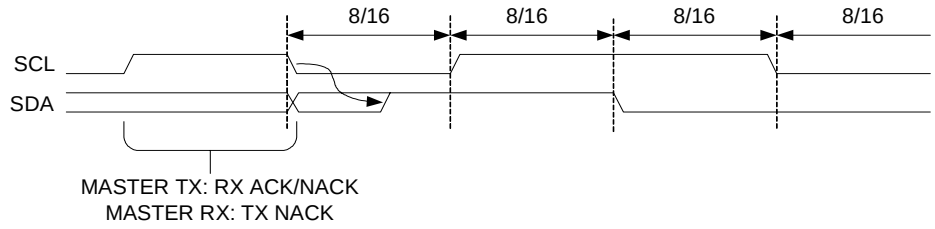


Figure 26-14. Master Restart Timing

26.4.7 Master Stop Timing

Figure 26-15 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 one-half bit time before it is pulled high to signal the Stop.

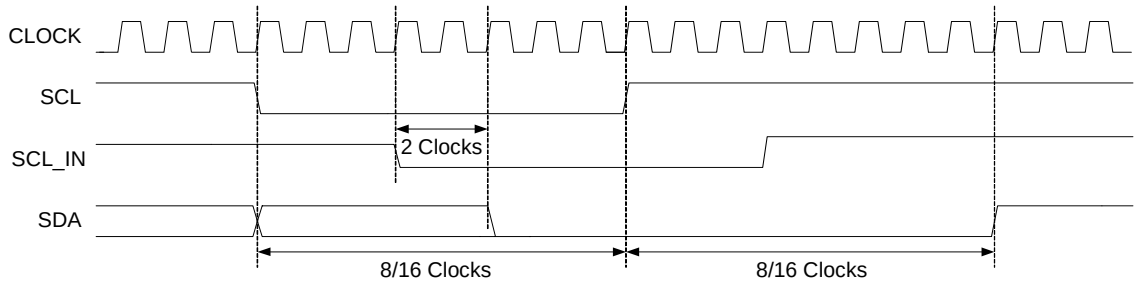


Figure 26-15. Master Stop Timing

26.4.8 Master/Slave Stall Timing

When a Byte Complete interrupt occurs, the host 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). As illustrated in Figure 26-16, firmware has until one clock after the falling edge of SCL_IN to write to the I2C_SCR register; otherwise, a stall will occur. Once stalled, the IO write will release the stall. The setup time between data output and the next rising edge of SCL will always be N-1 clocks.

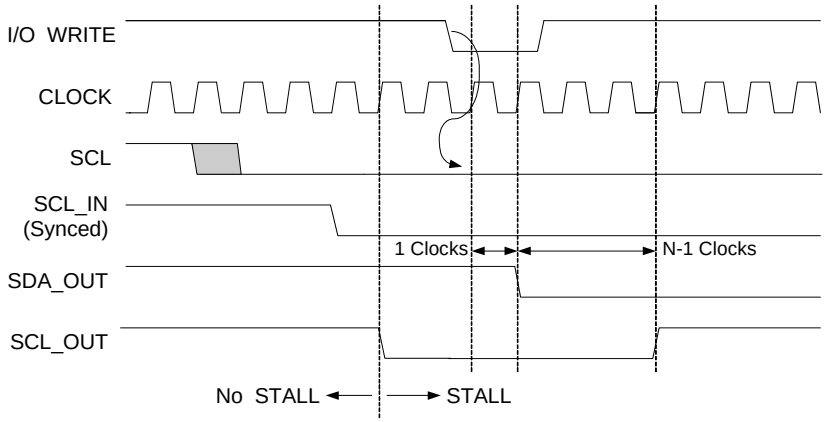


Figure 26-16. Master/Slave Stall Timing

26.4.9 Master Lost Arbitration Timing

Figure 26-17 shows a Lost Arbitration sequence. When contention is detected at the input (SDA_IN) sampling point, the SDA output is immediately released to an IDLE1 state. However, the master will continue clocking until the Byte Complete interrupt, which is processed in the usual way. Any write to the I2C_SCR register will result in the master reverting to an IDLE state, one clock after the next positive edge of the SCL_IN clock.

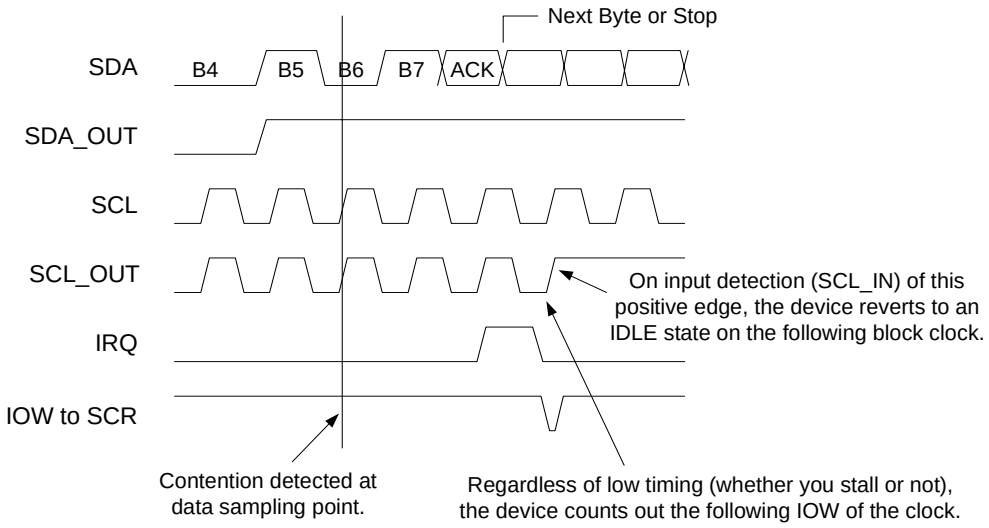


Figure 26-17. Lost Arbitration Timing (Transmitting Address or Data)

26.4.10 Master Clock Synchronization

Figure 26-18 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' will be seen by all Masters. When the hardware asserts a '0' to the output, it is immediately fed back from the chip 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 will hold 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.

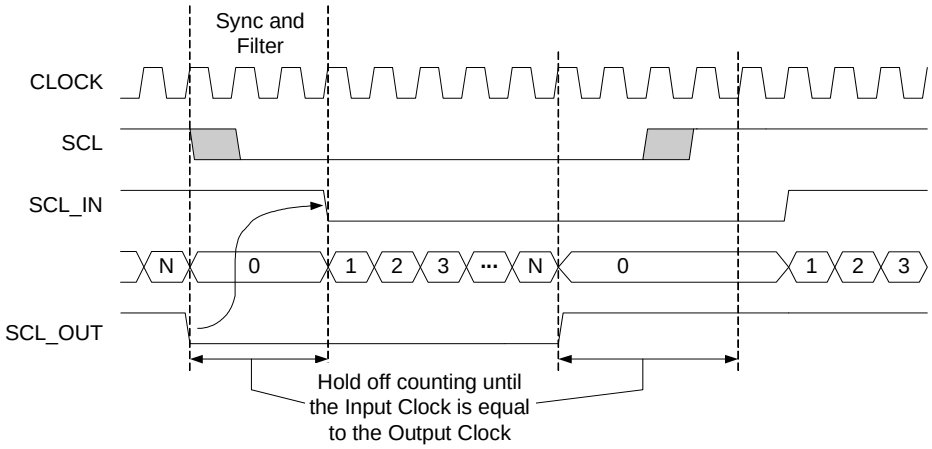


Figure 26-18. Master Clock Synchronization

27. POR and LVD

This chapter briefly discusses the POR and LVD circuits and their associated registers.

Table 27-1. POR and LVD Registers

Address	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
1,E4h	VLT_CMP							LVD	PPOR	R : 00

27.1 Architectural Description

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 will support operation to spec. 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.

27.2 Register Definitions

This block contains two registers: VLT_CR and VLT_CMP (read only status bits).

27.2.1 VLT_CR Register

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 will clear 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 7 and 6: Reserved.

Bits 5 and 4: PORLEV[1:0]. PORLEV[1:0] sets the Vdd level at which PPOR switches.

Bit 3: LVDTBEN. LVDTBEN is AND'ed with LVD to produce a throttle-back signal that reduces CPU clock speed when low voltage conditions are detected.

Bits 2, 1, and 0: VM[2:0]. VM[2:0] sets the Vdd level of the LVD Comparator switch.

For additional information, reference the VLT_CR register on page 168.

27.2.2 VLT_CMP Register

Bits 7 and 2: Reserved.

Bit 1: LVD. LVD reads the state of the low voltage detect comparator. The trip point for LVD is set by VM[2:0] in the VLT_CR register.

Bit 0: PPOR. The PPOR 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, reference the VLT_CMP register on page 169.

28. Internal Voltage Reference

This chapter briefly 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.

Table 28-1. Internal Voltage Reference Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,EAh	BDG_TR			TC[1:0]		V[3:0]				RW:00

28.1 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 bandgap voltage, to produce a 1.30V reference. A simplified schematic is illustrated in Figure 28-1. 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.

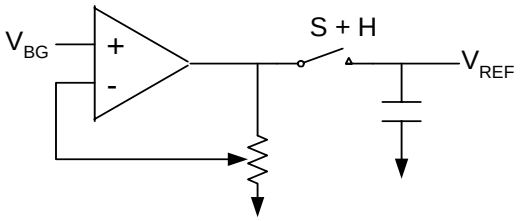


Figure 28-1. Voltage Reference Schematic

A temperature proportional voltage is also produced in this block for use in temperature sensing.

28.2 Register Definitions

The Internal Voltage Reference is trimmed for gain and temperature coefficient with a single write-only register, BDG_TR.

28.2.1 BDG_TR Register

Bits 7 and 6: Reserved.

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.

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.

For additional information, reference the BDG_TR register on page 172.

29. System Resets

This chapter discusses the System Resets and its associated registers. The M8C supports 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.

Table 29-1. System Reset Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,FEh	CPU_SCR1								IRAMDIS	RW : 00
0,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	RW : XX

LEGEND
XX: The reset value is 10h after POR/XRES and 20h after a watchdog reset.

When reset is initiated, all registers are restored to their default states. Minor exceptions are explained below.

The following types of resets can occur:

- 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 chip, on parts that contain an Xres pin.
- Watchdog Reset (WDR). This optional reset occurs when a timer expires, before being cleared by user firmware.
- Internal Reset (IRES). This occurs during the boot sequence, if the SROM code determines that Flash reads are not valid.

The occurrence of a reset is recorded in the Status and Control Register (CPU_SCR, for POR/XRES/WDR), or in the System Status and Control Register 1 (CPU_SCR1, for IRESS). Firmware can interrogate these registers to determine the cause of a reset.

29.1 Register Definitions

29.1.1 CPU_SCR0 Register

The bits of the CPU_SCR0 register are used to convey 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 will service interrupts.

Bit 6: Reserved.

Bit 5: WDRS. The WatchDog Reset Status bit is normally zero, but 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. This bit may not be set.

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

Bit 3: Sleep. The Sleep bit is used to enter low power Sleep mode when set, as described in this chapter.

Bits 2 and 1: Reserved.

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

For additional information, reference the CPU_SCR0 register on page 144.

29.1.2 CPU_SCR1 Register

The CPU_SCR1 register is used to convey status and control of events related to internal resets and watchdog reset.

Bits 7 to 1: Reserved.

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 set, 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 in the SROM chapter.

For additional information, reference the CPU_SCR1 register on page 143.

29.2 Timing Diagrams

29.2.1 Power On Reset (POR)

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 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 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 29-1).

POR configures register reset status bits as shown in Table 29-2. PPOR does not affect the BandGap Trim Register (BDG_TR), but IPOR does reset this register.

29.2.2 External Reset (XRES)

A XRES reset is caused by pulling the Xres pin high. The Xres pin has an always-on pull down resistor, so it does not require an external pull down for operation and can be tied directly to ground or left open. Behavior after XRES is similar to POR.

During XRES (XRES=1), the IMO is powered off for low power during start-up. Once XRES de-asserts, the IMO is started (see Figure 29-1).

XRES configures register reset status bits as shown in Table 29-2.

29.2.3 Watchdog Timer Reset (WDR)

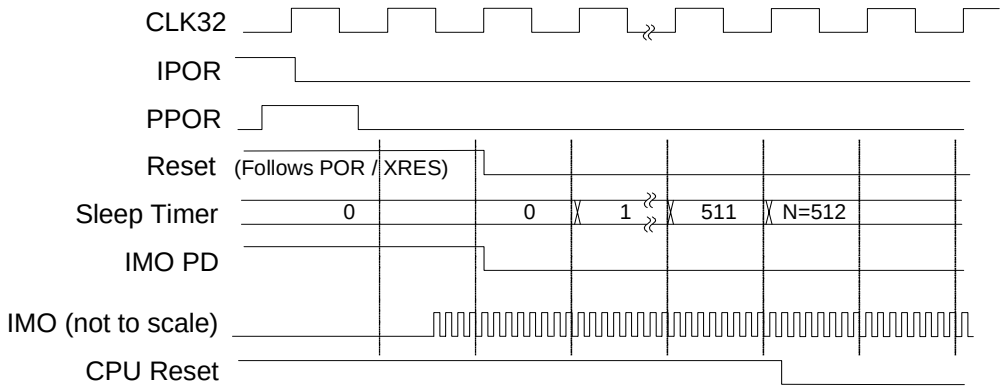
The user has the option to enable the WDR, by clearing the PORS bit in the CPU_SCR0 register. Once the PORS bit is cleared, the Watchdog Timer cannot be disabled. The only exception to this is if a POR/XRES event takes place, which will disable the WDR. See "Watchdog Timer (WDT)" on page 79 for details of the watchdog operation.

When the watchdog timer expires, a watchdog event occurs resulting in the reset sequence. Some WDR unique items are as follows.

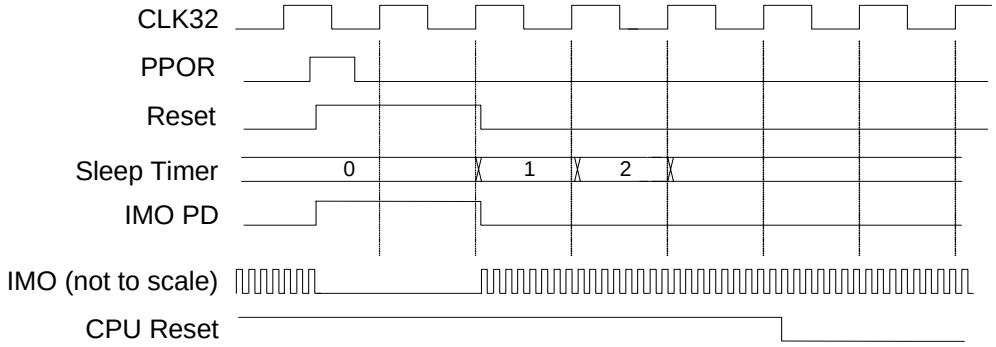
- Chip reset asserts for one cycle of the CLK32K clock (at its reset state).
- The IMO is not halted during or after WDR, i.e., the part does not go through a low power phase.
- CPU operation re-starts one CLK32K cycle after the internal reset de-asserts (see Figure 29-2).

WDR configures register reset status bits as shown in Table 29-2.

POR (IPOR followed by PPOR): Reset while POR high (IMO off), then 511(+) cycles (IMO on), and then CPU reset released. **XRES** is the same, with N=8.



PPOR (with no IPOR): Reset while PPOR high and to the end of the next 32K cycle (IMO off); 1 cycle IMO on before CPU reset released. Note that at the 5V level, PPOR will tend to be brief, because the reset clears the POR range register (VLT_CR) back to the default 3V setting.



XRES: Reset while XRES high (IMO off), then 7(+) cycles (IMO on), and then CPU reset released.

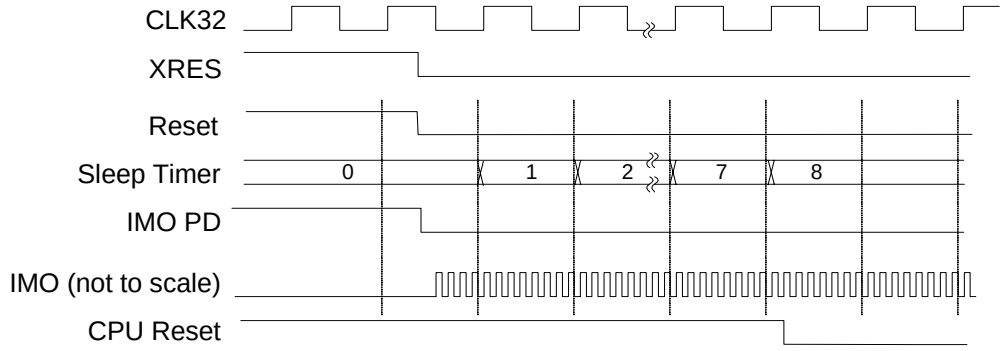
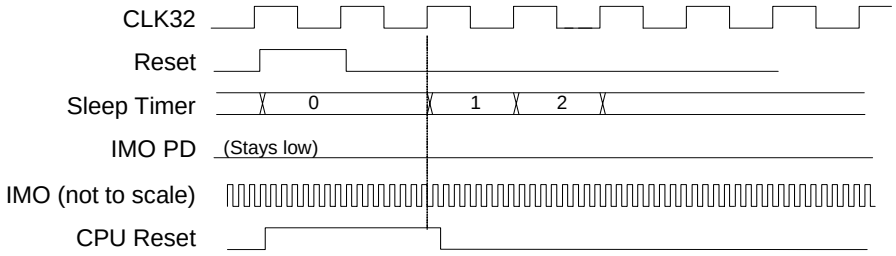


Figure 29-1. Key Signals During POR and XRES

WDR: Reset 1 cycle; then one additional cycle before CPU reset released.



IRES: Reset 1 cycle; then 2048 additional cycles low power hold-off, then 1 cycle with IMO on before CPU reset released.

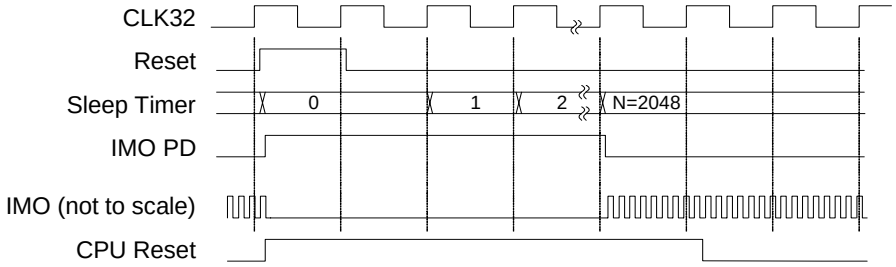


Figure 29-2. Key Signals During WDR and IRES

29.2.4 Reset Details

Timing and functionality details are summarized in Table 29-2. Figure 29-1 shows some of the relevant signals for IPOR, PPOR, and XRES.

Table 29-2. 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 us (1-2 clocks)	While XRES=1	30 us (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**	512*	1	8*	1
Register Reset (see next line for CPU_SCR, CPU_SCR1)	All	All, except PPOR does not reset Bandgap Trim Register	All	All
Reset status bits in CPU_SCR, CPU_SCR1	Set PORS Clear WDRS Clear IRAMDIS	Set PORS	Set PORS Clear WDRS Clear IRAMDIS	Set WDRS
Bandgap Power	On	On	On	On

* This count can be up to one CLK32K cycle less, depending on relative timing between the reset and the CLK32K clock.

** CPU reset is released after synchronization with CPU Clock.

*** **Note** If IPOR and PPOR occur together on a power up, key acquire will occur due to the IPOR.

29.3 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 assert, the IMO is powered off to reduce start-up power consumption.

During and following IRES (for 64 ms nominally), the IMO is powered off for low average power, during slow supply ramps.

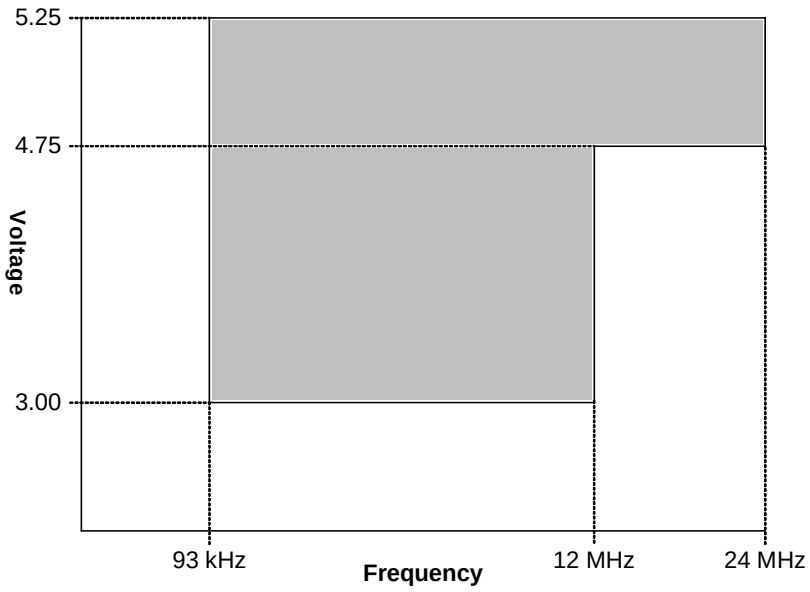
During and after POR or XRES, the bandgap circuit is powered up.

Following IRES, the bandgap circuit is only powered up occasionally, to refresh the sampled bandgap voltage value. This sampling follows the same process used during sleep mode.

The IMO is always on for at least one CLK32K cycle, before CPU reset is de-asserted.

SECTION G ELECTRICAL SPECIFICATIONS

The Electrical Specifications section presents the DC and AC electrical specifications of the PSoC device. Specifications are valid for -40°C ≤ T_A ≤ 85°C and T_J ≤ 100°C as specified, except where noted. Specifications for devices running at 24 MHz are valid at -40°C ≤ T_A ≤ 70°C and T_J ≤ 82°C.



Voltage Frequency Graph

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

Symbol	Units of Measure	Symbol	Units of Measure
°C	degree Celsius	µs	microsecond
AC	alternating current	µV	microvolts
dB	decibels	µVrms	microvolts root-mean-square
DC	direct current	mA	milliampere
fF	femto Farad	ms	millisecond
Hz	hertz	mV	millivolts
k	kilo, 1000	ns	nanosecond
K	2^{10}, 1024	nV	nanovolts
KB	1024 bytes	Ω	ohm
Kbit	1024 bits	pF	pico Farad
kHz	kilohertz	pp	peak-to-peak
kΩ	kilohm	ppm	parts per million
MHz	megahertz	sps	samples per second
MΩ	megaohm	σ	sigma: one standard deviation
µA	microampere	V	volts

Absolute Maximum Ratings

Absolute Maximum Ratings

Symbol	Description	Min	Typ	Max	Units	Notes
T_{STG}	Storage Temperature	-65	–	+100	°C	Higher storage temperatures will reduce data retention time.
T_A	Ambient Temperature with Power Applied	-40	–	+85	°C	
Vdd	Supply Voltage on Vdd Relative to Vss	-0.5	–	+6.0	V	
V_{IO}	DC Input Voltage	Vss-0.5	–	Vdd+0.5	V	
–	DC Voltage Applied to Tri-state	Vss-0.5	–	Vdd+0.5	V	
I_{MIO}	Maximum Current into any Port Pin	-25	–	+50	mA	
I_{MAIO}	Maximum Current into any Port Pin Configured as Analog Driver	-50	–	+50	mA	
–	Static Discharge Voltage	2000	–	–	V	
–	Latch-up Current	–	–	200	mA	

Operating Temperature

Operating Temperature

Symbol	Description	Min	Typ	Max	Units	Notes
T_A	Ambient Temperature	-40	–	+85	°C	
T_J	Junction Temperature	-40	–	+100	°C	The temperature rise from ambient to junction is package specific. See “Thermal Impedances” on page 30. The user must limit the power consumption to comply with this requirement.

DC Electrical Characteristics

DC Chip-Level Specifications

The following table lists guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

DC Chip-Level Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
Vdd	Supply Voltage	3.00	–	5.25	V	
I_{DD}	Supply Current	–	5	8	mA	Conditions are 5.0V, 25 °C, 3 MHz, 48 MHz disabled. VC1=1.5 MHz, VC2=93.75 kHz, VC3=93.75 kHz.
I_{SB}	Sleep (Mode) Current with POR, LVD, Sleep Timer, and WDT.[a]	–	3	6.5	µA	Conditions are with internal slow speed oscillator, Vdd = 3.3 V, -40 °C <=T_A <= 55 °C.
I_{SBH}	Sleep (Mode) Current with POR, LVD, Sleep Timer, and WDT.[a]	–	4	25	µA	Conditions are with internal slow speed oscillator, Vdd = 3.3 V, 55 °C < T_A <= 85 °C.
I_{SBXTL}	Sleep (Mode) Current with POR, LVD, Sleep Timer, and WDT.[a]	–	4	7.5	µA	Conditions are with properly loaded, 1 uW max, 32.768 kHz crystal. Vdd = 3.3 V, -40 °C <= T_A <= 55 °C.
I_{SBXTLH}	Sleep (Mode) Current with POR, LVD, Sleep Timer, and WDT.[a]	–	5	26	µA	Conditions are with properly loaded, 1 uW max, 32.768 kHz crystal. Vdd = 3.3 V, 55 °C < T_A <= 85 °C.
V_{REF}	Reference Voltage (Bandgap)	1.275	1.3	1.325	V	Trimmed for appropriate Vdd.

a. Standby current includes all functions (POR, LVD, WDT, Sleep Time) needed for reliable system operation. This should be compared with devices that have similar functions enabled.

DC General Purpose IO (GPIO) Specifications

The following table lists guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

DC GPIO Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
R_{PU}	Pull up Resistor	4	5.6	8	kΩ	
R_{PD}	Pull down Resistor	4	5.6	8	kΩ	
V_{OH}	High Output Level	Vdd - 1.0	–	–	V	IOH = 10 mA, Vdd = 4.75 to 5.25V (80 mA maximum combined IOH budget)
V_{OL}	Low Output Level	–	–	0.75	V	IOL = 25 mA, Vdd = 4.75 to 5.25V (150 mA maximum combined IOL budget)
V_{IL}	Input Low Level	–	–	0.8	V	Vdd = 3.0 to 5.5
V_{IH}	Input High Level	2.2	–		V	Vdd = 3.0 to 5.5
V_H	Input Hysterisis	–	60	–	mV	
I_{IL}	Input Leakage (Absolute Value)	–	1	–	nA	Gross tested to 1 µA.
C_{IN}	Capacitive Load on Pins as Input	–	3.5	10	pF	Package and pin dependent. Temp = 25°C.
C_{OUT}	Capacitive Load on Pins as Output	–	3.5	10	pF	Package and pin dependent. Temp = 25°C.

DC Operational Amplifier Specifications

The following tables list guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

The Operational Amplifier is a component of both the Analog Continuous Time PSoC blocks and the Analog Switched Cap PSoC blocks. The guaranteed specifications are measured in the Analog Continuous Time PSoC block. Typical parameters apply to 5V at 25°C and are for design guidance only.

5V DC Operational Amplifier Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
V_{OSOA}	Input Offset Voltage (absolute value) Low Power	–	1.6	10	mV	
	Input Offset Voltage (absolute value) Mid Power	–	1.3	8	mV	
	Input Offset Voltage (absolute value) High Power	–	1.2	7.5	mV	
TCV_{OSOA}	Average Input Offset Voltage Drift	–	7.0	35.0	µV/ºC	
I_{EBOA}	Input Leakage Current (Port 0 Analog Pins)	–	20	–	pA	Gross tested to 1 µA.
C_{INOA}	Input Capacitance (Port 0 Analog Pins)	–	4.5	9.5	pF	Package and pin dependent. Temp = 25ºC.
V_{CMOA}	Common Mode Voltage Range	0.0	–	Vdd	V	The common-mode input voltage range is measured through an analog output buffer. The specification includes the limitations imposed by the characteristics of the analog output buffer.
	Common Mode Voltage Range (high power or high opamp bias)	0.5	–	Vdd-0.5		
G_{OLOA}	Open Loop Gain		–	–	dB	Specification is applicable at high power. For all other bias modes (except high power, high opamp bias), minimum is 60 dB.
	Power=Low	60				
	Power=Medium	60				
	Power=High	80				
$V_{OHIGHOA}$	High Output Voltage Swing (worst case internal load)					
	Power=Low	Vdd-0.2	–	–	V	
	Power=Medium	Vdd-0.2	–	–	V	
	Power=High	Vdd-0.5	–	–	V	
V_{OLOWOA}	Low Output Voltage Swing (worst case internal load)					
	Power=Low	–	–	0.2	V	
	Power=Medium	–	–	0.2	V	
	Power=High	–	–	0.5	V	
I_{SOA}	Supply Current (including associated AGND buffer)					
	Power=Low	–	150	200	µA	
	Power=Low, Opamp Bias=High	–	300	400	µA	
	Power=Medium	–	600	800	µA	
	Power=Medium, Opamp Bias=High	–	1200	1600	µA	
	Power=High	–	2400	3200	µA	
	Power=High, Opamp Bias=High	–	4600	6400	µA	
$PSRR_{OA}$	Supply Voltage Rejection Ratio	60	–	–	dB	

3.3V DC Operational Amplifier Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
V_{OSOA}	Input Offset Voltage (absolute value) Low Power	–	1.65	10	mV	
	Input Offset Voltage (absolute value) Mid Power	–	1.32	8	mV	
	High Power is 5 Volt Only					
TCV_{OSOA}	Average Input Offset Voltage Drift	–	7.0	35.0	µV/°C	
I_{EBOA}	Input Leakage Current (Port 0 Analog Pins)	–	20	–	pA	Gross tested to 1 µA.
C_{INOA}	Input Capacitance (Port 0 Analog Pins)	–	4.5	9.5	pF	Package and pin dependent. Temp = 25°C.
V_{CMOA}	Common Mode Voltage Range	0.2	–	Vdd-0.2	V	The common-mode input voltage range is measured through an analog output buffer. The specification includes the limitations imposed by the characteristics of the analog output buffer.
G_{OLOA}	Open Loop Gain		–	–	dB	Specification is applicable at high power. For all other bias modes (except high power, high opamp bias), minimum is 60 dB.
	Power=Low	60				
	Power=Medium	60				
	Power=High	80				
$V_{OHIGHOA}$	High Output Voltage Swing (worst case internal load)					
	Power=Low	Vdd-0.2	–	–	V	
	Power=Medium	Vdd-0.2	–	–	V	
	Power=High is 5V only	Vdd-0.2	–	–	V	
V_{OLOWOA}	Low Output Voltage Swing (worst case internal load)					
	Power=Low	–	–	0.2	V	
	Power=Medium	–	–	0.2	V	
	Power=High	–	–	0.2	V	
I_{SOA}	Supply Current (including associated AGND buffer)					
	Power=Low	–	150	200	µA	
	Power=Low, Opamp Bias=High	–	300	400	µA	
	Power=Medium	–	600	800	µA	
	Power=Medium, Opamp Bias=High	–	1200	1600	µA	
	Power=High	–	2400	3200	µA	
	Power=High, Opamp Bias=High	–	4600	6400	µA	
$PSRR_{OA}$	Supply Voltage Rejection Ratio	50	–	–	dB	

DC Analog Output Buffer Specifications

The following tables list guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

5V DC Analog Output Buffer Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
V_{OSOB}	Input Offset Voltage (Absolute Value)	–	3	12	mV	
TCV_{OSOB}	Average Input Offset Voltage Drift	–	+6	–	µV/°C	
V_{CMOB}	Common-Mode Input Voltage Range	.5	–	Vdd - 1.0	V	
R_{OUTOB}	Output Resistance					
	Power = Low	–	1	–	Ω	
	Power = High	–	1	–	Ω	
$V_{OHIGHOB}$	High Output Voltage Swing (Load = 32 ohms to Vdd/2)					
	Power = Low	.5 x Vdd + 1.1	–	–	V	
	Power = High	.5 x Vdd + 1.1	–	–	V	
V_{OLOWOB}	Low Output Voltage Swing (Load = 32 ohms to Vdd/2)					
	Power = Low	–	–	.5 x Vdd - 1.3	V	
	Power = High	–	–	.5 x Vdd - 1.3	V	
I_{SOB}	Supply Current Including Bias Cell (No Load)					
	Power = Low	–	1.1	5.1	mA	
	Power = High	–	2.6	8.8	mA	
$PSRR_{OB}$	Supply Voltage Rejection Ratio	60	–	–	dB	

3.3V DC Analog Output Buffer Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
V_{OSOB}	Input Offset Voltage (Absolute Value)	–	3	12	mV	
TCV_{OSOB}	Average Input Offset Voltage Drift	–	+6	–	µV/°C	
V_{CMOB}	Common-Mode Input Voltage Range	.5	-	Vdd - 1.0	V	
R_{OUTOB}	Output Resistance					
	Power = Low	–	1	–	Ω	
	Power = High	–	1	–	Ω	
$V_{OHIGHOB}$	High Output Voltage Swing (Load = 1K ohms to Vdd/2)					
	Power = Low	.5 x Vdd + 1.0	–	–	V	
	Power = High	.5 x Vdd + 1.0	–	–	V	
V_{OLOWOB}	Low Output Voltage Swing (Load = 1K ohms to Vdd/2)					
	Power = Low	–	–	.5 x Vdd - 1.0	V	
	Power = High	–	–	.5 x Vdd - 1.0	V	
I_{SOB}	Supply Current Including Bias Cell (No Load)					
	Power = Low		0.8	2.0	mA	
	Power = High	–	2.0	4.3	mA	
$PSRR_{OB}$	Supply Voltage Rejection Ratio	50	–	–	dB	

DC Analog Reference Specifications

The following tables list guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

The guaranteed specifications are measured through the Analog Continuous Time PSoC blocks. The power levels for AGND refer to the power of the Analog Continuous Time PSoC block. The power levels for RefHi and RefLo refer to the Analog Reference Control register. The limits stated for AGND include the offset error of the AGND buffer local to the Analog Continuous Time PSoC block.

5V DC Analog Reference Specifications

Symbol	Description	Min	Typ	Max	Units
–	AGND = Vdd/2[a] CT Block Power = High	Vdd/2 - 0.043	Vdd/2 - 0.025	Vdd/2 + 0.003	V

a. AGND tolerance includes the offsets of the local buffer in the PSoC block. Bandgap voltage is 1.3V ± 2%.

3.3V DC Analog Reference Specifications

Symbol	Description	Min	Typ	Max	Units
–	AGND = Vdd/2[a] CT Block Power = High	Vdd/2 - 0.037	Vdd/2 - 0.020	Vdd/2 + 0.002	V

a. AGND tolerance includes the offsets of the local buffer in the PSoC block. Bandgap voltage is 1.3V ± 2%

DC Analog PSoC Block Specifications

The following table lists guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

DC Analog PSoC Block Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
R_{CT}	Resistor Unit Value (Continuous Time)	–	12.24	–	kΩ	
C_{SC}	Capacitor Unit Value (Switch Cap)	–	80	–	fF	

DC POR and LVD Specifications

The following table lists guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

DC POR and LVD Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
V_{IPOR}	AVDD Value for IPOR Trip	1.60	1.90	2.30	V	
	AVDD Value for PPOR Trip (positive ramp)					
V_{PPOR0R}	PORLEV1,PORLEV0=00b		2.908		V	
V_{PPOR1R}	PORLEV1,PORLEV0=01b	–	4.394	–	V	
V_{PPOR2R}	PORLEV1,PORLEV0=10b		4.548		V	
	AVDD Value for PPOR Trip (negative ramp)					
V_{PPOR0}	PORLEV1,PORLEV0=00b		2.816		V	
V_{PPOR1}	PORLEV1,PORLEV0=01b	–	4.394	–	V	
V_{PPOR2}	PORLEV1,PORLEV0=10b		4.548		V	
	PPOR Hysteresis					
V_{PH0}	PORLEV1,PORLEV0=00b	–	92	–	mV	
V_{PH1}	PORLEV1,PORLEV0=01b	–	0	–	mV	
V_{PH2}	PORLEV1,PORLEV0=10b	–	0	–	mV	
	AVDD Value for LVD Trip					
V_{LVD0}	VM2,VM1,VM0=000b	2.863	2.921	2.979[a]	V	
V_{LVD1}	VM2,VM1,VM0=001b	2.963	3.023	3.083	V	
V_{LVD2}	VM2,VM1,VM0=010b	3.070	3.133	3.196	V	
V_{LVD3}	VM2,VM1,VM0=011b	3.920	4.00	4.080	V	
V_{LVD4}	VM2,VM1,VM0=100b	4.393	4.483	4.573	V	
V_{LVD5}	VM2,VM1,VM0=101b	4.550	4.643	4.736[b]	V	
V_{LVD6}	VM2,VM1,VM0=110b	4.632	4.727	4.822	V	
V_{LVD7}	VM2,VM1,VM0=111b	4.718	4.814	4.910	V	

a. Always greater than 50 mV above PPOR (PORLEV=00) for falling supply.
b. Always greater than 50 mV above PPOR (PORLEV=10) for falling supply.

DC Programming Specifications

The following table lists guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

DC Programming Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
I_{CCP}	Supply Current During Programming or Verify	–	5	25	mA	
V_{ILP}	Input Low Voltage During Programming or Verify	–	–	0.8	V	
V_{IHP}	Input High Voltage During Programming or Verify	2.2	–	–	V	
I_{ILP}	Input Current when Applying Vilp to P1[0] or P1[1] During Programming or Verify	–	–	0.2	mA	Driving internal pull-down resistor.
I_{IHP}	Input Current when Applying Vihp to P1[0] or P1[1] During Programming or Verify	–	–	1.5	mA	Driving internal pull-down resistor.
V_{OLV}	Output Low Voltage During Programming or Verify	–	–	Vss+0.75	V	
V_{OHV}	Output High Voltage During Programming or Verify	Vdd - 1.0	–	Vdd	V	
$Flash_{ENPB}$	Flash Endurance (per block)	50,000	–	–	–	Erase/write cycles per block.
$Flash_{ENT}$	Flash Endurance (total)[a]	1,800,000	–	–	–	Erase/write cycles.
$Flash_{DR}$	Flash Data Retention	10	–	–	Years	

a. A maximum of 36 x 50,000 block endurance cycles is allowed. This may be balanced between operations on 36x1 blocks of 50,000 maximum cycles each, 36x2 blocks of 25,000 maximum cycles each, or 36x4 blocks of 12,500 maximum cycles each (and so forth to limit the total number of cycles to 36x50,000 and that no single block ever sees more than 50,000 cycles).

The PSoC devices use an adaptive algorithm to enhance endurance over the industrial temperature range (-40°C to +85°C ambient). Any temperature range within a 50°C span between 0°C and 85°C is considered constant with respect to endurance enhancements. For instance, if room temperature (25°C) is the nominal operating temperature, then the range from 0°C to 50°C can be approximated by the constant value 25 and a temperature sensor is not needed.

For the full industrial range, the user must employ a temperature sensor user module (FlashTemp) and feed the result to the temperature argument before writing. Refer to the Flash APIs Application Note AN2015 at http://www.cypress.com under Application Notes for more information.

AC Electrical Characteristics

AC Chip-Level Specifications

The following table lists guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

AC Chip-Level Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
F_{IMO}	Internal Main Oscillator Frequency	23.4	24	24.6[a,c]	MHz	Trimmed. Utilizing factory trim values.
F_{CPU1}	CPU Frequency (5 V Nominal)	0.93	24	24.6[a,b]	MHz	
F_{CPU2}	CPU Frequency (3.3V Nominal)	0.93	12	12.3[b,c]	MHz	
F_{48M}	Digital PSoC Block Frequency	0	48	49.2[a,b,d]	MHz	Refer to the AC Digital Block Specifications below.
F_{24M}	Digital PSoC Block Frequency	0	24	24.6[b,e,d]	MHz	
F_{32K1}	Internal Low Speed Oscillator Frequency	15	32	64	kHz	
F_{32K2}	External Crystal Oscillator	–	32.768	–	kHz	Accuracy is capacitor and crystal dependent. 50% duty cycle.
F_{PLL}	PLL Frequency	–	23.986	–	MHz	Is a multiple (x732) of crystal frequency.
Jitter24M2	24 MHz Period Jitter (PLL)	600	–		ps	
$T_{PLLSLEW}$	PLL Lock Time	0.5	–	10	ms	
$T_{PLLSLEWSLOW}$	PLL Lock Time for Low Gain Setting	0.5	–	50	ms	
T_{OS}	External Crystal Oscillator Startup to 1%	–	1700	2620	ms	
T_{OSACC}	External Crystal Oscillator Startup to 100 ppm	–	2800	3800[f]	ms	
Jitter32k	32 kHz Period Jitter	–	100		ns	
T_{XRST}	External Reset Pulse Width	10	–	–	µs	
DC24M	24 MHz Duty Cycle	40	50	60	%	
Step24M	24 MHz Trim Step Size	–	50	–	kHz	
Fout48M	48 MHz Output Frequency	46.8	48.0	49.2[a,c]	MHz	Trimmed. Utilizing factory trim values.
Jitter24M1	24 MHz Period Jitter (IMO)	–	600		ps	
F_{MAX}	Maximum frequency of signal on row input or row output.	–	–	12	MHz	
T_{RAMP}	Supply Ramp Time	0	–	–	µs	

a. 4.75V < Vdd < 5.25V.
b. Accuracy derived from Internal Main Oscillator with appropriate trim for Vdd range.
c. 3.0V < Vdd < 3.6V.
d. See Application Note AN2012 “Adjusting PSoC Microcontroller Trims for Dual Voltage-Range Operation” for information on maximum frequency for User Modules.
e. 3.0V < 5.25V.
f. The crystal oscillator frequency is within 100 ppm of its final value by the end of the T_{osacc} period. Correct operation assumes a properly loaded 1 uW maximum drive level 32.768 kHz crystal. 3.0V ≤ Vdd ≤ 5.5V, -40 °C ≤ T_A ≤ 85 °C.

AC General Purpose IO (GPIO) Specifications

The following table lists guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

AC GPIO Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
F_{GPIO}	GPIO Operating Frequency	0	–	12	MHz	
TRiseF	Rise Time, Normal Strong Mode, Cload = 50 pF	3	–	18	ns	Vdd = 4.5 to 5.5V, 10% - 90%
TFallF	Fall Time, Normal Strong Mode, Cload = 50 pF	2	–	18	ns	Vdd = 4.5 to 5.5V, 10% - 90%
TRiseS	Rise Time, Slow Strong Mode, Cload = 50 pF	10	27	–	ns	Vdd = 3 to 5.5V, 10% - 90%
TFallS	Fall Time, Slow Strong Mode, Cload = 50 pF	10	22	–	ns	Vdd = 3 to 5.5V, 10% - 90%

AC Operational Amplifier Specifications

The following tables list guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

Settling times, slew rates, and gain bandwidth are based on the Analog Continuous Time PSoC block.

5V AC Operational Amplifier Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
T_{ROA}	Rising Settling Time from 80% of ΔV to 0.1% of ΔV (10 pF load, Unity Gain)					Specification maximums for low power and high opamp bias, medium power, and medium power and high opamp bias levels are between low and high power levels.
	Power = Low	–	–	3.9	µs	
	Power = Low, Opamp Bias = High	–			µs	
	Power = Medium	–			µs	
	Power = Medium, Opamp Bias = High	–	–	0.72	µs	
	Power = High	–			µs	
	Power = High, Opamp Bias = High	–	–	0.62	µs	
T_{SOA}	Falling Settling Time from 20% of ΔV to 0.1% of ΔV (10 pF load, Unity Gain)					Specification maximums for low power and high opamp bias, medium power, and medium power and high opamp bias levels are between low and high power levels.
	Power = Low	–	–	5.9	µs	
	Power = Low, Opamp Bias = High	–			µs	
	Power = Medium	–			µs	
	Power = Medium, Opamp Bias = High	–	–	0.92	µs	
	Power = High	–			µs	
	Power = High, Opamp Bias = High	–	–	0.72	µs	
SR_{ROA}	Rising Slew Rate (20% to 80%)(10 pF load, Unity Gain)					Specification minimums for low power and high opamp bias, medium power, and medium power and high opamp bias levels are between low and high power levels.
	Power = Low	0.15	–		V/µs	
	Power = Low, Opamp Bias = High				V/µs	
	Power = Medium				V/µs	
	Power = Medium, Opamp Bias = High	1.7	–		V/µs	
	Power = High				V/µs	
	Power = High, Opamp Bias = High	6.5	–		V/µs	
SR_{FOA}	Falling Slew Rate (20% to 80%)(10 pF load, Unity Gain)					Specification minimums for low power and high opamp bias, medium power, and medium power and high opamp bias levels are between low and high power levels.
	Power = Low	0.01	–		V/µs	
	Power = Low, Opamp Bias = High				V/µs	
	Power = Medium				V/µs	
	Power = Medium, Opamp Bias = High	0.5	–		V/µs	
	Power = High				V/µs	
	Power = High, Opamp Bias = High	4.0	–		V/µs	
BW_{OA}	Gain Bandwidth Product					Specification minimums for low power and high opamp bias, medium power, and medium power and high opamp bias levels are between low and high power levels.
	Power = Low	0.75	–		MHz	
	Power = Low, Opamp Bias = High				MHz	
	Power = Medium				MHz	
	Power = Medium, Opamp Bias = High	3.1	–		MHz	
	Power = High				MHz	
	Power = High, Opamp Bias = High	5.4	–		MHz	
E_{NOA}	Noise at 1 kHz	–	200	–	nV/rt-Hz	

3.3V AC Operational Amplifier Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
T_{ROA}	Rising Settling Time from 80% of ΔV to 0.1% of ΔV (10 pF load, Unity Gain)					Specification maximums for low power and high opamp bias, medium power, and medium power and high opamp bias levels are between low and high power levels.
	Power = Low	–	–	3.92	µs	
	Power = Low, Opamp Bias = High	–			µs	
	Power = Medium	–			µs	
	Power = Medium, Opamp Bias = High	–	–	0.72	µs	
	Power = High (3.3 Volt High Bias Operation not supported)	–	–	–	µs	
	Power = High, Opamp Bias = High (3.3 Volt High Power, High Opamp Bias not supported)	–	–	–	µs	
T_{SOA}	Falling Settling Time from 20% of ΔV to 0.1% of ΔV (10 pF load, Unity Gain)					Specification maximums for low power and high opamp bias, medium power, and medium power and high opamp bias levels are between low and high power levels.
	Power = Low	–	–	5.41	µs	
	Power = Low, Opamp Bias = High	–			µs	
	Power = Medium	–			µs	
	Power = Medium, Opamp Bias = High	–	–	0.72	µs	
	Power = High (3.3 Volt High Bias Operation not supported)	–	–	–	µs	
	Power = High, Opamp Bias = High (3.3 Volt High Power, High Opamp Bias not supported)	–	–	–	µs	
SR_{ROA}	Rising Slew Rate (20% to 80%)(10 pF load, Unity Gain)					Specification minimums for low power and high opamp bias, medium power, and medium power and high opamp bias levels are between low and high power levels.
	Power = Low	0.31	–		V/µs	
	Power = Low, Opamp Bias = High				V/µs	
	Power = Medium				V/µs	
	Power = Medium, Opamp Bias = High	2.7	–		V/µs	
	Power = High (3.3 Volt High Bias Operation not supported)	–	–	–	V/µs	
	Power = High, Opamp Bias = High (3.3 Volt High Power, High Opamp Bias not supported)	–	–	–	V/µs	
SR_{FOA}	Falling Slew Rate (20% to 80%)(10 pF load, Unity Gain)					Specification minimums for low power and high opamp bias, medium power, and medium power and high opamp bias levels are between low and high power levels.
	Power = Low	0.24	–		V/µs	
	Power = Low, Opamp Bias = High				V/µs	
	Power = Medium				V/µs	
	Power = Medium, Opamp Bias = High	1.8	–		V/µs	
	Power = High (3.3 Volt High Bias Operation not supported)	–	–	–	V/µs	
	Power = High, Opamp Bias = High (3.3 Volt High Power, High Opamp Bias not supported)	–	–	–	V/µs	
BW_{OA}	Gain Bandwidth Product					Specification minimums for low power and high opamp bias, medium power, and medium power and high opamp bias levels are between low and high power levels.
	Power = Low	0.67	–		MHz	
	Power = Low, Opamp Bias = High				MHz	
	Power = Medium				MHz	
	Power = Medium, Opamp Bias = High	2.8	–		MHz	
	Power = High (3.3 Volt High Bias Operation not supported)	–	–	–	MHz	
	Power = High, Opamp Bias = High (3.3 Volt High Power, High Opamp Bias not supported)	–	–	–	MHz	
E_{NOA}	Noise at 1 kHz (Turbo Medium)	–	200	–	nV/rt-Hz	

AC Digital Block Specifications

The following table lists guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

AC Digital Block Specifications

Function	Description	Min	Typ	Max	Units	Notes
Timer	Capture Pulse Width	50[a]	–	–	ns	
	Maximum Frequency, No Capture	–	–	48	MHz	4.75V < Vdd < 5.25V.
	Maximum Frequency, With Capture	–	–	24	MHz	
Counter	Enable Pulse Width	50[a]	–	–	ns	
	Maximum Frequency, No Enable Input	–	–	48	MHz	4.75V < Vdd < 5.25V.
	Maximum Frequency, Enable Input	–	–	24	MHz	
Dead Band	Kill Pulse Width:					
	Asynchronous Restart Mode	20	–	–	ns	
	Synchronous Restart Mode	50[a]	–	–	ns	
	Disable Mode	50[a]	–	–	ns	
	Maximum Frequency	–	–	48	MHz	4.75V < Vdd < 5.25V.
CRCPRS (PRS Mode)	Maximum Input Clock Frequency	–	–	48	MHz	4.75V < Vdd < 5.25V.
CRCPRS (CRC Mode)	Maximum Input Clock Frequency	–	–	24	MHz	
SPIM	Maximum Input Clock Frequency	–	–	8	MHz	
SPIS	Maximum Input Clock Frequency	–	–	4	ns	
	Width of SS_ Negated Between Transmissions	50[a]	–	–	ns	
Transmitter	Maximum Input Clock Frequency	–	–	16	MHz	
Receiver	Maximum Input Clock Frequency	–	16	48	MHz	4.75V < Vdd < 5.25V.

a. 50 ns minimum input pulse width is based on the input synchronizers running at 24 MHz (42 ns nominal period).

AC Analog Output Buffer Specifications

The following tables list guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

5V AC Analog Output Buffer Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
T_{ROB}	Rising Settling Time to 0.1%, 1V Step, 100pF Load					
	Power = Low	–	–	2.5	µs	
	Power = High	–	–	2.5	µs	
T_{SOB}	Falling Settling Time to 0.1%, 1V Step, 100pF Load					
	Power = Low	–	–	2.2	µs	
	Power = High	–	–	2.2	µs	
SR_{ROB}	Rising Slew Rate (20% to 80%), 1V Step, 100pF Load					
	Power = Low	0.65	–	–	V/µs	
	Power = High	0.65	–	–	V/µs	
SR_{FOB}	Falling Slew Rate (80% to 20%), 1V Step, 100pF Load					
	Power = Low	0.65	–	–	V/µs	
	Power = High	0.65	–	–	V/µs	
BW_{OB}	Small Signal Bandwidth, 20mV_{pp}, 3dB BW, 100pF Load					
	Power = Low	0.8	–	–	MHz	
	Power = High	0.8	–	–	MHz	
BW_{OB}	Large Signal Bandwidth, 1V_{pp}, 3dB BW, 100pF Load					
	Power = Low	300	–	–	kHz	
	Power = High	300	–	–	kHz	

3.3V AC Analog Output Buffer Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
T_{ROB}	Rising Settling Time to 0.1%, 1V Step, 100pF Load					
	Power = Low	–	–	3.8	µs	
	Power = High	–	–	3.8	µs	
T_{SOB}	Falling Settling Time to 0.1%, 1V Step, 100pF Load					
	Power = Low	–	–	2.6	µs	
	Power = High	–	–	2.6	µs	
SR_{ROB}	Rising Slew Rate (20% to 80%), 1V Step, 100pF Load					
	Power = Low	.5	–	–	V/µs	
	Power = High	.5	–	–	V/µs	
SR_{FOB}	Falling Slew Rate (80% to 20%), 1V Step, 100pF Load					
	Power = Low	.5	–	–	V/µs	
	Power = High	.5	–	–	V/µs	
BW_{OB}	Small Signal Bandwidth, 20mV_{pp}, 3dB BW, 100pF Load					
	Power = Low	0.7	–	–	MHz	
	Power = High	0.7	–	–	MHz	
BW_{OB}	Large Signal Bandwidth, 1V_{pp}, 3dB BW, 100pF Load					
	Power = Low	200	–	–	kHz	
	Power = High	200	–	–	kHz	

AC External Clock Specifications

The following tables list guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

5V AC External Clock Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
F_{OSCEXT}	Frequency	0	–	24.24	MHz	
–	High Period	20.6	–	–	ns	
–	Low Period	20.6	–	–	ns	
–	Power Up IMO to Switch	150	–	–	µs	

3.3V AC External Clock Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
F_{OSCEXT}	Frequency with CPU Clock divide by 1[a]	0	–	12	MHz	
F_{OSCEXT}	Frequency with CPU Clock divide by 2 or greater[b]	0	–	24	MHz	
–	High Period with CPU Clock divide by 1	41.7	–	–	ns	
–	Low Period with CPU Clock divide by 1	41.7	–	–	ns	
–	Power Up IMO to Switch	150	–	–	µs	

a. Maximum CPU frequency is 12 MHz at 3.3V. With the CPU clock divider set to 1, the external clock must adhere to the maximum frequency and duty cycle requirements.

b. If the frequency of the external clock is greater than 12 MHz, the CPU clock divider must be set to 2 or greater. In this case, the CPU clock divider will ensure that the fifty percent duty cycle requirement is met.

AC Programming Specifications

The following table lists guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

AC Programming Specifications

Symbol	Description	Min	Typ	Max	Units	Notes
T_{RSCLK}	Rise Time of SCLK	1	–	20	ns	
T_{FSCLK}	Fall Time of SCLK	1	–	20	ns	
T_{SSCLK}	Data Set up Time to Falling Edge of SCLK	40	–	–	ns	
T_{HSCLK}	Data Hold Time from Falling Edge of SCLK	40	–	–	ns	
F_{SCLK}	Frequency of SCLK	0	–	8	MHz	
T_{ERASEB}	Flash Erase Time (Block)	–	15	–	ms	
T_{WRITE}	Flash Block Write Time	–	30	–	ms	
T_{DSCLK}	Data Out Delay from Falling Edge of SCLK	–	–	45	ns	

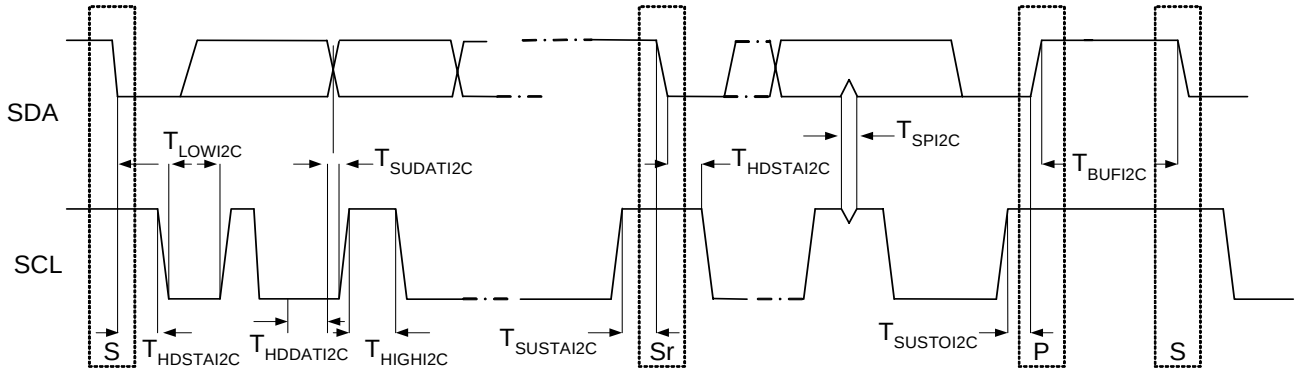
AC I²C Specifications

The following table lists guaranteed maximum and minimum specifications for the voltage and temperature ranges: 4.75V to 5.25V and -40°C ≤ T_A ≤ 85°C, or 3.0V to 3.6V and -40°C ≤ T_A ≤ 85°C, respectively. Typical parameters apply to 5V and 3.3V at 25°C and are for design guidance only or unless otherwise specified.

AC Characteristics of the I²C SDA and SCL Pins

Symbol	Description	Standard Mode		Fast Mode		Units	Notes
		Min	Max	Min	Max		
F_{SCLI2C}	SCL Clock Frequency	0	100	0	400	kHz	
$T_{HDSTAI2C}$	Hold Time (repeated) START Condition. After this period, the first clock pulse is generated.	4.0	–	0.6	–	µs	
T_{LOWI2C}	LOW Period of the SCL Clock	4.7	–	1.3	–	µs	
$T_{HIGHI2C}$	HIGH Period of the SCL Clock	4.0	–	0.6	–	µs	
$T_{SUSTAI2C}$	Set-up Time for a Repeated START Condition	4.7	–	0.6	–	µs	
$T_{HDDATI2C}$	Data Hold Time	0	–	0	–	µs	
$T_{SUDATI2C}$	Data Set-up Time	250	–	100[a]	–	ns	
$T_{SUSTOI2C}$	Set-up Time for STOP Condition	4.0	–	0.6	–	µs	
T_{BUFI2C}	Bus Free Time Between a STOP and START Condition	4.7	–	1.3	–	µs	
T_{SPI2C}	Pulse Width of spikes are suppressed by the input filter.	–	–	0	50	ns	

a. A Fast-Mode I²C-bus device can be used in a Standard-Mode I²C-bus system, but the requirement $t_{SU;DAT} \geq 250$ ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line $t_{rmax} + t_{SU;DAT} = 1000 + 250 = 1250$ ns (according to the Standard-Mode I²C-bus specification) before the SCL line is released.



Definition for Timing for F/S-Mode on the I²C Bus

SECTION H REVISION HISTORY

Document Revision History

Document Title: CY8C22113, CY8C22213 PSoC™ Mixed Signal Array Preliminary Data Sheet **Document Number**: 38-12009				
Revision	**ECN #**	**Issue Date**	**Origin of Change**	**Description of Change**
**	128180	06/30/2003	New Silicon.	New document – Advanced Data Sheet (two page product brief).
*A	129202	09/16/2003	NWJ	New document – Preliminary Data Sheet (300 page product detail).
*B	130127	10/15/2003	NWJ	Revised document for Silicon Revision A.
*C	131679	12/05/2003	NWJ	Changes to Electrical Specifications section, Miscellaneous changes to I2C, GDI, RDI, Registers, and Digital Block chapters.
*D	131803	12/22/2003	NWJ	Changes to Electrical Specifications and miscellaneous small changes throughout the data sheet.
Distribution: External/Public **Posting**: None				