The CY8C25xxx/26xxx family of Programmable System-on-Chip (PSoC™) microcontrollers replaces many MCU-based system components with *one* single-chip, programmable device. A single PSoC microcontroller offers a fast core, Flash program memory, and SRAM data memory with configurable analog and digital peripheral blocks in a range of convenient pin-outs and memory sizes. The driving force behind this innovative programmable system-on-a-chip comes from user configurability of analog and digital arrays, the PSoC blocks.

Powerful Harvard Architecture Processor with Fast Multiply/Accumulate

- Processor speeds to 24MHz
- Register speed memory transfers
- Instruction set that is easy to learn and use
- Flexible addressing modes
- Bit manipulation on I/O and memory
- 8x8 multiply, 32-bit accumulate

Flexible On-Chip Memory

- FLASH memory, 4k to 16 kbytes, depending on device
- 100,000 erase/write cycles
- SRAM memory, 128 to 256 bytes, depending on device
- Serial programming capability
- Partial Flash updates
- Flexible protection model
- EEPROM emulation in Flash

Programmable System-on-Chip (PSoC™) Blocks

- On-chip, user configurable analog and digital peripheral blocks
- PSoC blocks can be used individually or in combination
- Analog PSoC blocks provide:
  - Up to 12 bit Delta-Sigma ADC
  - Up to 8 bit Successive Approximation ADC
  - Up to 12 bit Incremental ADC
  - Up to 8 bit direct DAC
  - Programmable gain
  - Sample and hold
  - Programmable filters
  - Differential comparators
  - On-chip temperature sensor
- Digital PSoC blocks provide:
  - Multipurpose timers: event timing, real-time clock, pulse width modulation (PWM) and PWM with deadband
  - CRC modules
  - Full-duplex UARTs
  - SPI™ master or slave configuration
  - Complex clocking sources for analog PSoC blocks

Programmable Pin Configurations

- Schmitt trigger TTL I/O pins
- Configurable output drive to 25 mA with internal pull-up or pull-down resistors, open drain, or active driver
- Interrupt on Pin Change

Precision, Programmable Clocking

- Internal 48/24MHz oscillator (+/- 2.5%, no external components)
- External 32.768kHz crystal oscillator (optional precision source for PLL)
- Internal Low Speed Oscillator for Watchdog and Sleep

Dedicated Peripherals

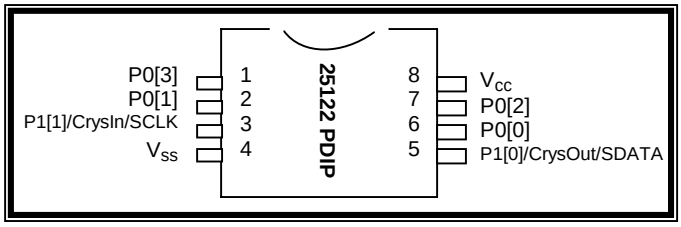
- Watchdog/Sleep Timers
- 5V and 3V Brownout protection with user-configurable trip voltages
- On-chip voltage reference
- On-chip temperature sensor

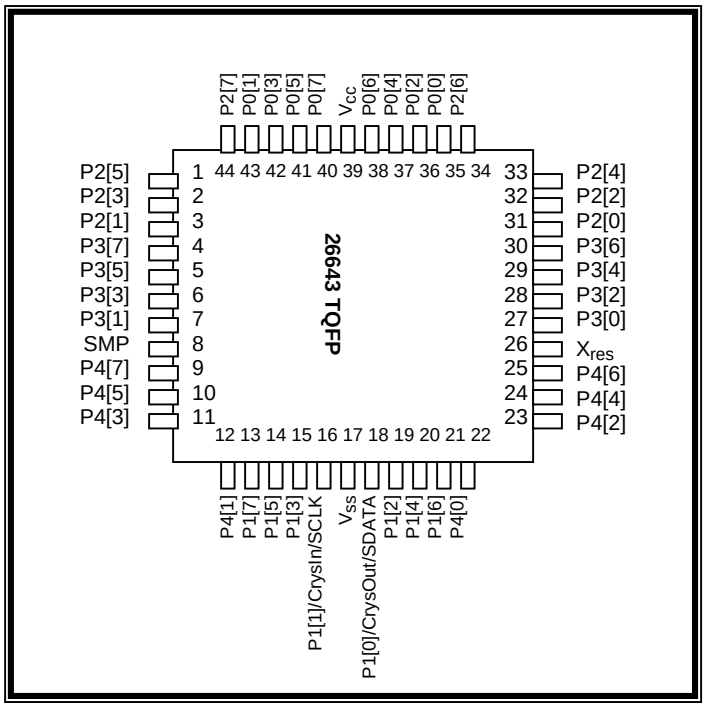
Fully Static CMOS Devices utilizing advanced FLASH technology

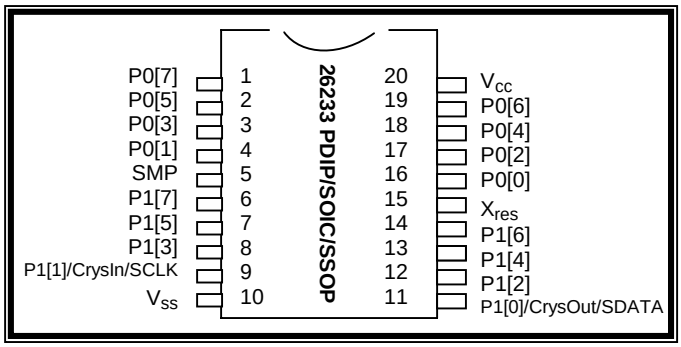
- Low power at high speed
- Operating voltage from 3.0 to 5.5 VDC
- Operating voltages from 0.9V to 5.5 VDC using on-chip switch mode voltage pump
- Wide temperature range: -40 ^{o}C to + 85 ^{o}C

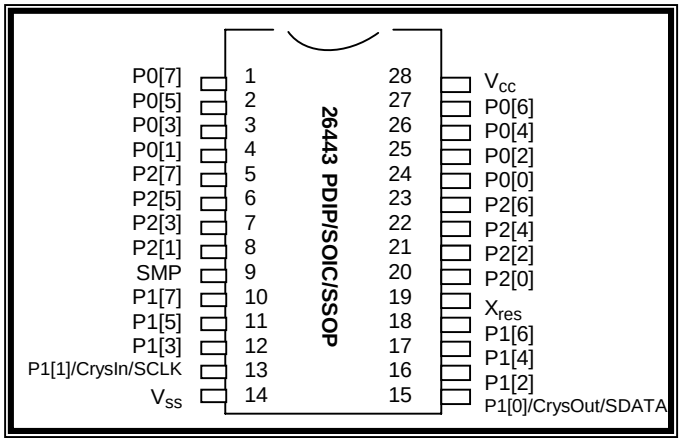
Complete Development Tools

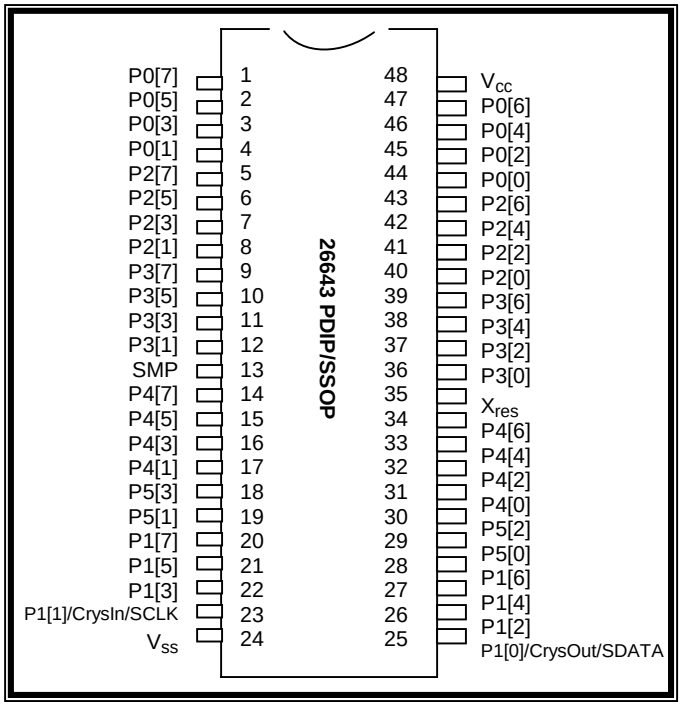
- Powerful integrated development environment (PSoC Designer™)
- Low-cost, in-circuit emulator and programmer











1 Functional Overview

The CPU heart of the 8C25K family is a high performance, 8-bit, next generation M8C Harvard architecture microprocessor. Separate program and memory busses allow for faster overall throughput. Processor clock speeds to 24MHz are available. The processor may also be run at lower clock speeds for power-sensitive applications. A rich instruction set allows for efficient high-level language support as well as bit-manipulation capabilities.

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

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

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

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

Several different device types in this family will provide various amounts of code and data memory. The code space ranges in size from 4K to 16K bytes of user programmable Flash memory. This memory is programmed serially in either a programming station or on the user board. The endurance on the Flash memory is 100,000 erase/write cycles. The data space ranges in size from 128 to 256 bytes of user SRAM.

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

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

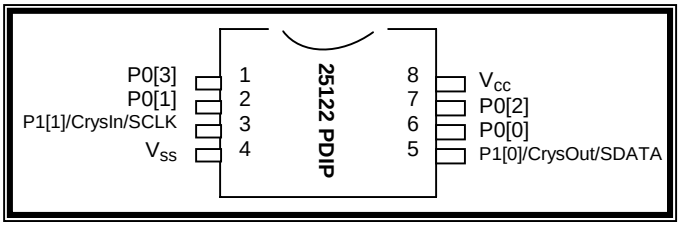
1.1 Key Features

	8C25122A	8C26233A	8C26443A	8C26643A
Operating Frequency	93.7kHz - 24MHz	93.7kHz - 24MHz	93.7kHz - 24MHz	93.7kHz - 24MHz
Operating Voltage	3.0 - 5.5v	3.0 - 5.5v	3.0 - 5.5v	3.0 - 5.5v
Program Memory (KBytes)	4	8	16	16
Data Memory (Bytes)	128	256	256	256
Digital PSoC Blocks	8	8	8	8
Analog PSoC Blocks	12	12	12	12
I/O Pins	6	16	24	40/44
External Switch Mode Pump	No	Yes	Yes	Yes
Available Packages	8 PDIP	20 PDIP	28 PDIP	48 PDIP
		20 SOIC	28 SOIC	48 SSOP
		20 SSOP	28 SSOP	44 TQFP

1.2 Pin-out Descriptions

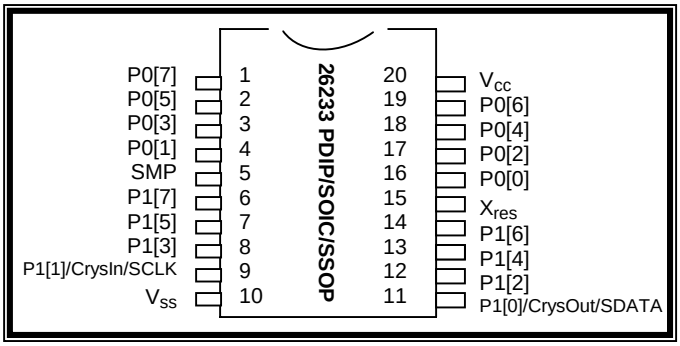
1.2.1 Pin-out 8 Pin

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



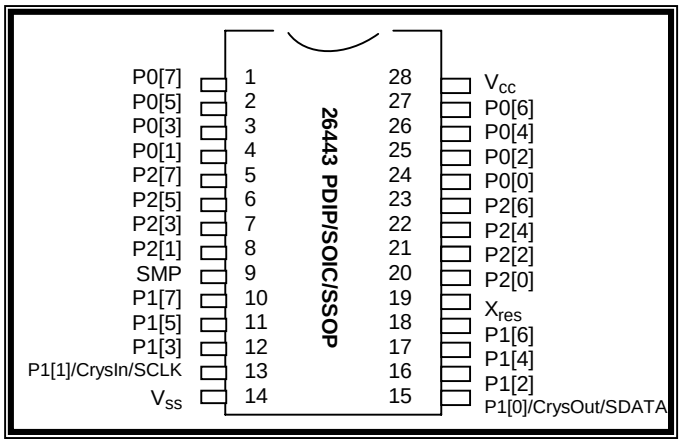
1.2.2 Pin-out 20 Pin

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



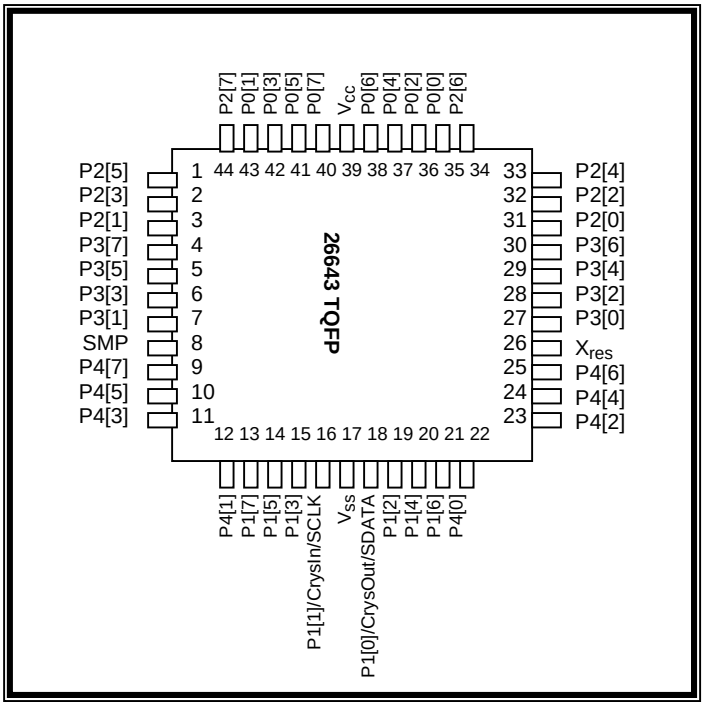
1.2.3 Pin-out 28 Pin

Name	I/O	Pin #	Description
P0[7]	I/O	1	Port 0 bit 7 (Analog Input)
P0[5]	I/O	2	Port 0 bit 5 (Analog Input/Output)
P0[3]	I/O	3	Port 0 bit 3 (Analog Input/Output)
P0[1]	I/O	4	Port 0 bit 1 (Analog Input)
P2[7]	I/O	5	Port 2 bit 7
P2[5]	I/O	6	Port 2 bit 5
P2[3]	I/O	7	Port 2 bit 3 (Non-Multiplexed Analog Input)
P2[1]	I/O	8	Port 2 bit 1 (Non-Multiplexed Analog Input)
SMP	O	9	Switch Mode Pump
P1[7]	I/O	10	Port 1 bit 7
P1[5]	I/O	11	Port 1 bit 5
P1[3]	I/O	12	Port 1 bit 3
P1[1]	I/O	13	Port 1 bit 1 / CrysIn / SCLK
Vss	Power	14	Ground
P1[0]	I/O	15	Port 1 bit 0 / CrysOut / SDATA
P1[2]	I/O	16	Port 1 bit 2
P1[4]	I/O	17	Port 1 bit 4
P1[6]	I/O	18	Port 1 bit 6
XRES	I	19	External Reset
P2[0]	I/O	20	Port 2 bit 0 (Non-Multiplexed Analog Input)
P2[2]	I/O	21	Port 2 bit 2 (Non-Multiplexed Analog Input)
P2[4]	I/O	22	Port 2 bit 4
P2[6]	I/O	23	Port 2 bit 6
P0[0]	I/O	24	Port 0 bit 0 (Analog Input)
P0[2]	I/O	25	Port 0 bit 2 (Analog Input/Output)
P0[4]	I/O	26	Port 0 bit 4 (Analog Input/Output)
P0[6]	I/O	27	Port 0 bit 6 (Analog Input)
Vcc	Power	28	Supply Voltage



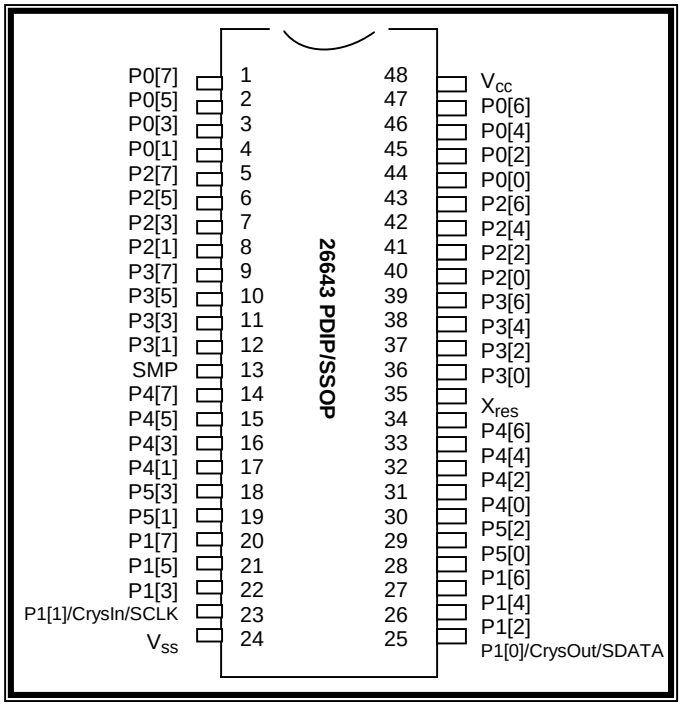
1.2.4 Pin-out 44 Pin

Name	I/O	Pin #	Description
P0[7]	I/O	40	Port 0 bit 7 (Analog Input)
P0[5]	I/O	41	Port 0 bit 5 (Analog Input/Output)
P0[3]	I/O	42	Port 0 bit 3 (Analog Input/Output)
P0[1]	I/O	43	Port 0 bit 1 (Analog Input)
P2[7]	I/O	44	Port 2 bit 7
P2[5]	I/O	1	Port 2 bit 5
P2[3]	I/O	2	Port 2 bit 3 (Non-Multiplexed Analog Input)
P2[1]	I/O	3	Port 2 bit 1 (Non-Multiplexed Analog Input)
P3[7]	I/O	4	Port 3 bit 7
P3[5]	I/O	5	Port 3 bit 5
P3[3]	I/O	6	Port 3 bit 3
P3[1]	I/O	7	Port 3 bit 1
SMP	O	8	Switch Mode Pump
P4[7]	I/O	9	Port 4 bit 7
P4[5]	I/O	10	Port 4 bit 5
P4[3]	I/O	11	Port 4 bit 3
P4[1]	I/O	12	Port 4 bit 1
P1[7]	I/O	13	Port 1 bit 7
P1[5]	I/O	14	Port 1 bit 5
P1[3]	I/O	15	Port 1 bit 3
P1[1]	I/O	16	Port 1 bit 1 / CrysIn / SCLK
Vss	Power	17	Ground
P1[0]	I/O	18	Port 1 bit 0 / CrysOut / SDATA
P1[2]	I/O	19	Port 1 bit 2
P1[4]	I/O	20	Port 1 bit 4
P1[6]	I/O	21	Port 1 bit 6
P4[0]	I/O	22	Port 4 bit 0
P4[2]	I/O	23	Port 4 bit 2
P4[4]	I/O	24	Port 4 bit 2
P4[6]	I/O	25	Port 4 bit 6
XRES	I	26	External Reset
P3[0]	I/O	27	Port 3 bit 0
P3[2]	I/O	28	Port 3 bit 2
P3[4]	I/O	29	Port 3 bit 4
P3[6]	I/O	30	Port 3 bit 6
P2[0]	I/O	31	Port 2 bit 0 (Non-Multiplexed Analog Input)
P2[2]	I/O	32	Port 2 bit 2 (Non-Multiplexed Analog Input)
P2[4]	I/O	33	Port 2 bit 4
P2[6]	I/O	34	Port 2 bit 6
P0[0]	I/O	35	Port 0 bit 0 (Analog Input)
P0[2]	I/O	36	Port 0 bit 2 (Analog Input/Output)
P0[4]	I/O	37	Port 0 bit 4 (Analog Input/Output)
P0[6]	I/O	38	Port 0 bit 6 (Analog Input)
Vcc	Power	39	Supply Voltage

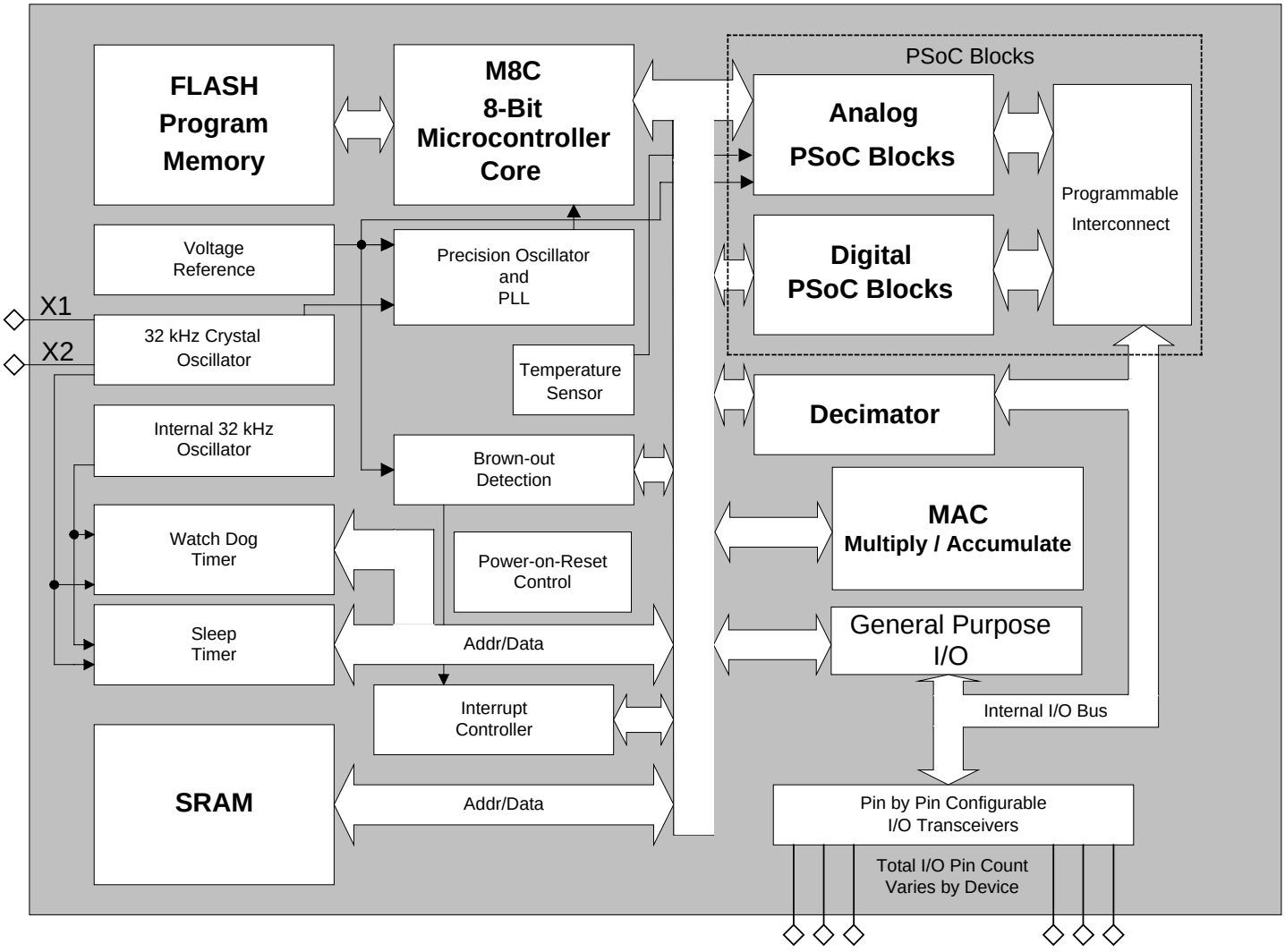


1.2.5 Pin-out 48 Pin

Name	I/O	Pin #	Description
P0[7]	I/O	1	Port 0 bit 7 (Analog Input)
P0[5]	I/O	2	Port 0 bit 5 (Analog Input/Output)
P0[3]	I/O	3	Port 0 bit 3 (Analog Input/Output)
P0[1]	I/O	4	Port 0 bit 1 (Analog Input)
P2[7]	I/O	5	Port 2 bit 7
P2[5]	I/O	6	Port 2 bit 5
P2[3]	I/O	7	Port 2 bit 3 (Non-Multiplexed Analog Input)
P2[1]	I/O	8	Port 2 bit 1 (Non-Multiplexed Analog Input)
P3[7]	I/O	9	Port 3 bit 7
P3[5]	I/O	10	Port 3 bit 5
P3[3]	I/O	11	Port 3 bit 3
P3[1]	I/O	12	Port 3 bit 1
SMP	O	13	Switch Mode Pump
P4[7]	I/O	14	Port 4 bit 7
P4[5]	I/O	15	Port 4 bit 5
P4[3]	I/O	16	Port 4 bit 3
P4[1]	I/O	17	Port 4 bit 1
P5[3]	I/O	18	Port 5 bit 3
P5[1]	I/O	19	Port 5 bit 1
P1[7]	I/O	20	Port 1 bit 7
P1[5]	I/O	21	Port 1 bit 5
P1[3]	I/O	22	Port 1 bit 3
P1[1]	I/O	23	Port 1 bit 1 / CrysIn / SCLK
Vss	Power	24	Ground
P1[0]	I/O	25	Port 1 bit 0 / CrysOut / SDATA
P1[2]	I/O	26	Port 1 bit 2
P1[4]	I/O	27	Port 1 bit 4
P1[6]	I/O	28	Port 1 bit 6
P5[0]	I/O	29	Port 5 bit 0
P5[2]	I/O	30	Port 5 bit 2
P4[0]	I/O	31	Port 4 bit 0
P4[2]	I/O	32	Port 4 bit 2
P4[4]	I/O	33	Port 4 bit 2
P4[6]	I/O	34	Port 4 bit 6
XRES	I	35	External Reset
P3[0]	I/O	36	Port 3 bit 0
P3[2]	I/O	37	Port 3 bit 2
P3[4]	I/O	38	Port 3 bit 4
P3[6]	I/O	39	Port 3 bit 6
P2[0]	I/O	40	Port 2 bit 0 (Non-Multiplexed Analog Input)
P2[2]	I/O	41	Port 2 bit 2 (Non-Multiplexed Analog Input)
P2[4]	I/O	42	Port 2 bit 4
P2[6]	I/O	43	Port 2 bit 6
P0[0]	I/O	44	Port 0 bit 0 (Analog Input)
P0[2]	I/O	45	Port 0 bit 2 (Analog Input/Output)
P0[4]	I/O	46	Port 0 bit 4 (Analog Input/Output)
P0[6]	I/O	47	Port 0 bit 6 (Analog Input)
Vcc	Power	48	Supply Voltage



1.3 Block Diagram



2 Family Architecture

2.1 Introduction

This family is based on a high performance, 8-bit, Harvard architecture microprocessor. There are six registers that control the primary operation of the CPU core. These six registers are affected by various instructions, but are not directly accessible by the user. For more details on addressing with the register space, see section 4 (Register Organization).

Following, is the list of CPU registers and their mnemonics:

Register	Mnemonic
Flag	(CPU_F)
Program Counter High	(CPU_PCH)
Program Counter Low	(CPU_PCL)
Accumulator	(CPU_A)
Stack Pointer	(CPU_SP)
Index	(CPU_X)

The pair of Program Counter registers (CPU_PCH and CPU_PCL) form a 16-bit address that allows for direct addressing of the full 16Kbytes of program memory space available in the largest members of this family. This forms one contiguous program space, and no paging is required.

The Accumulator Register (CPU_A) is the general-purpose register that holds the results of all instructions that specify any of the source addressing modes.

The Index Register (CPU_X) holds an offset value that is used in the indexed addressing modes. Typically, this is used to address a block of data within the data memory space.

The Stack Pointer Register (CPU_SP) holds the address of the current top-of-stack value in the data memory space. It is affected by the PUSH, POP, CALL, RETI, and RET instructions, which manage the software stack.

The Flag Register (CPU_F) has three status bits: Global Interrupt Enable bit [0]; Zero Flag bit [1]; Carry Flag bit [2]. An extended I/O space address, bit [4], is used to determine which bank of the register space is in use. The user cannot manipulate the Supervisory State status bit. The flags are affected by arithmetic, logic, and shift operations. The manner in which each flag is changed is dependent upon the instruction being executed.

2.2 Registers

2.2.1 Flags Register

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

Bit 7: **Reserved**

Bit 6: **Reserved**

Bit 5: **Reserved**

Bit 4: **XIO** is set by the user to select between the register banks. Can only be set or reset with logical instruction
0 = Bank 0
1 = Bank 1

Bit 3: **Super** is set by the CPU to indicate whether the CPU is executing user code or supervisor code
0 = User Code
1 = Supervisor Code

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

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

Bit 0: **Global IE** determines whether all interrupts are enabled or disabled. Can only be set or reset with logical instructions
0 = Disabled
1 = Enabled

Flags Register (CPU_F)

2.2.2 Accumulator Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	System*	System*	System*	System*	System*	System*	System*	System*
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]** 8-bit data value holds the result of any logical/arithmetic instruction that uses a source addressing mode

Accumulator Register (CPU_A) * = System - not directly accessible by the user

2.2.3 Index Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	System*	System*	System*	System*	System*	System*	System*	System*
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]** 8-bit data value holds an index for any instruction that uses an indexed addressing mode

Index Register (CPU_X) * = System - not directly accessible by the user

2.2.4 Stack Pointer Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	System*	System*	System*	System*	System*	System*	System*	System*
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]** 8-bit data value holds a pointer to the current top-of-stack

Stack Pointer Register (CPU_SP) * = System - not directly accessible by the user

2.2.5 Program Counter High Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	System*	System*	System*	System*	System*	System*	System*	System*
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]** 8-bit data value is the high-order byte of the program counter

Program Counter High Register (CPH_PCH) * = System - not directly accessible by the user

2.2.6 Program Counter Low Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	System*	System*	System*	System*	System*	System*	System*	System*
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]** 8-bit data value is the low-order byte of the program counter

Program Counter Low Register (CPU_PCL) * = System - not directly accessible by the user

2.3 Addressing Modes

2.3.1 Source Immediate

The result of an instruction using this addressing mode is placed in the A register, the F register or the X register, which is specified as part of the instruction opcode. Operand 1 is an immediate value that serves as a source for the instruction. Arithmetic instructions require two sources, the second source is the A register or X register specified in the opcode. Instructions using this addressing mode are two bytes in length.

Opcode	Operand 1
Instruction	Immediate Value

Examples:

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

2.3.2 Source Direct

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

Opcode	Operand 1
Instruction	Source Address

Examples:

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

2.3.3 Source Indexed

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

Opcode	Operand 1
Instruction	Source Index

Examples:

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

2.3.4 Destination Direct

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

Opcode	Operand 1
Instruction	Destination Address

Examples:

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

2.3.5 Destination Indexed

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

Opcode	Operand 1
Instruction	Destination Index

Examples:

```
ADD     [x+7]   A       ; In this case, the value in the memory location at address X+7 is added
                        with the Accumulator, and the result is placed in the memory location at
                        address x+7. The Accumulator is unchanged.
```

2.3.6 Destination Direct Immediate

The result of an instruction using this addressing mode is placed within either the RAM memory space or the register space. Operand 1 is the address of the result. The source for the instruction is Operand 2, which is an immediate value. Arithmetic instructions require two sources, the second source is the location specified by Operand 1. Instructions using this addressing mode are three bytes in length.

Opcode	Operand 1	Operand 2
Instruction	Destination Address	Immediate Value

Examples:

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

2.3.7 Destination Indexed Immediate

The result of an instruction using this addressing mode is placed within either the RAM memory space or the register space. Operand 1 is added to the X register to form the address of the result. The source for the instruction is Operand 2, which is an immediate value. Arithmetic instructions require two sources, the second source is the location specified by Operand 1 added with the X register. Instructions using this addressing mode are three bytes in length.

Opcode	Operand 1	Operand 2
Instruction	Destination Index	Immediate Value

Examples:

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

2.3.8 Destination Direct Direct

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

Opcode	Operand 1	Operand 2
Instruction	Destination Address	Source Address

Examples:

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

2.3.9 Source Indirect Post Increment

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

Opcode	Operand 1
Instruction	Source Address Address

Examples:

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

2.3.10 Destination Indirect Post Increment

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

Opcode	Operand 1
Instruction	Destination Address Address

Examples:

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

2.4 Instruction Set Summary

Program Flow Instructions

				Flags	Cycles
CALL		**Call (relative)**			
9xh	CALL addr	LSB Address Byte (MSN in opcode, x)			11
HALT		**Halt**			NA
30h	HALT				
JACC		**Jump Accumulator (relative)**			
Exh	JACC addr	LSB Address Byte (MSN in opcode, x)			7
JC		**Jump if Carry (relative)**			
Cxh	JC addr	LSB Address Byte (MSN in opcode, x)			5/4
JMP		**Jump (relative)**			
8xh	JMP addr	LSB Address Byte (MSN in opcode, x)			5
JNC		**Jump if No Carry (relative)**			
Dxh	JNC addr	LSB Address Byte (MSN in opcode, x)			5/4
JNZ		**Jump if Not Zero (relative)**			
Bxh	JNZ addr	LSB Address Byte (MSN in opcode, x)			5/4
JZ		**Jump if Zero (relative)**			
Axh	JZ addr	LSB Address Byte (MSN in opcode, x)			5/4
LCALL		**Long Call**			
7Ch	LCALL addrl	addr MSB	addr LSB		13
LJMP		**Long Jump**			
7Dh	LJMP addrl	addr MSB	addr LSB		7
NOP		**No Operation**			
40h	NOP				4
RET		**Return from Call**			
7Fh	RET				8
RETI		**Return from Interrupt**			
7Eh	RETI				10
SSC		**System Supervisor Call**			NA
00h	SSC				

Non Destructive Test Instructions

				Flags	Cycles
CMP		**Non Destructive Compare**			
39h	CMP A,expr	Immediate		c z	5
3Ah	CMP A,[expr]	Direct Address		c z	7
3Bh	CMP A,[X+exr]	Index		c z	8
3Ch	CMP [expr],expr	Direct Address	Immediate	c z	8
3Dh	CMP [X+expr],expr	Index	Immediate	c z	9
TST		**Test with Mask**			
47h	TST [expr],expr	Direct Address	Bit Mask	z	8
48h	TST [X+expr],expr	Index	Bit Mask	z	9
49h	TST REG[expr],expr	Reg Direct Address	Bit Mask	z	8
4Ah	TST REG[X+expr],expr	Reg Index	Bit Mask	z	9

Arithmetic Instructions

				Flags	Cycles
ADC		**Add with Carry**			
09h	ADC A,expr	Immediate		c z	4
0Ah	ADC A,[expr]	Direct Address		c z	6
0Bh	ADC A,[X+expr]	Index		c z	7
0Ch	ADC [expr],A	Direct Address		c z	7
0Dh	ADC [X+expr],A	Index		c z	8
0Eh	ADC [expr],expr	Direct Address	Immediate	c z	9
0Fh	ADC [X+expr],expr	Index	Immediate	c z	10
ADD		**Add without Carry**			
01h	ADD A,expr	Immediate		c z	4
02h	ADD A,[expr]	Direct Address		c z	6
03h	ADD A,[X+expr]	Index		c z	7
04h	ADD [expr],A	Direct Address		c z	7
05h	ADD [X+expr],A	Index		c z	8
06h	ADD [expr],expr	Direct Address	Immediate	c z	9
07h	ADD [X+expr],expr	Index	Immediate	c z	10
38h	ADD SP,expr	Immediate		c z	5
SBB		**Subtract with Borrow**			
19h	SBB A,expr	Immediate		c z	4
1Ah	SBB A,[expr]	Direct Address		c z	6
1Bh	SBB A,[X+expr]	Index		c z	7
1Ch	SBB [expr],A	Direct Address		c z	7
1Dh	SBB [X+expr],A	Index		c z	8
1Eh	SBB [expr],expr	Direct Address	Immediate	c z	9
1Fh	SBB [X+expr],expr	Index	Immediate	c z	10
SUB		**Subtract without Borrow**			
11h	SUB A,expr	Immediate		c z	4
12h	SUB A,[expr]	Direct Address		c z	6
13h	SUB A,[X+expr]	Index		c z	7
14h	SUB [expr],A	Direct Address		c z	7
15h	SUB [X+expr],A	Index		c z	8
16h	SUB [expr],expr	Direct Address	Immediate	c z	9
17h	SUB [X+expr],expr	Index	Immediate	c z	10
DEC		**Decrement**			
78h	DEC A			c z	4
79h	DEC X			c z	4
7Ah	DEC [expr]	Direct Address		c z	7
7Bh	DEC [X+expr]	Index		c z	8
INC		**Increment**			
74h	INC A			c z	4
75h	INC X			c z	4
76h	INC [expr]	Direct Address		c z	7
77h	INC [X+expr]	Index		c z	8
ASL		**Arithmetic Shift Left**			
64h	ASL A			c z	4
65h	ASL [expr]	Direct Address		c z	7
66h	ASL [X+expr]	Index		c z	8
ASR		**Arithmetic Shift Right**			
67h	ASR A			c z	4
68h	ASR [expr]	Direct Address		c z	7
69h	ASR [X+expr]	Index		c z	8

Movement Instructions

				Flags	Cycles
INDEX		**Table Read (relative)**			
Fxh	INDEX addr	LSB Address Byte (MSN in opcode, x)		z	13
MOV		**Move**			
4Fh	MOV X,SP				4
50h	MOV A,expr	Immediate		z	4
51h	MOV A,[expr]	Direct Address		z	5
52h	MOV A,[X+expr]	Index		z	6
53h	MOV [expr],A	Direct Address			5
54h	MOV [X+expr],A	Index			6
55h	MOV [expr],expr	Direct Address	Immediate		8
56h	MOV [X+expr],expr	Index	Immediate		9
57h	MOV X,expr	Immediate			4
58h	MOV X,[expr]	Direct Address			6
59h	MOV X,[X+expr]	Index			7
5Ah	MOV [expr],X	Direct Address			5
5Bh	MOV A,X			z	4
5Ch	MOV X,A				4
5Dh	MOV A,REG[expr]	Reg Direct Address		z	6
5Eh	MOV A,REG[X+expr]	Reg Index		z	7
5Fh	MOV [expr],[expr]	Direct Address	Direct Address		10
60h	MOV REG[expr],A	Reg Direct Address			5
61h	MOV REG[X+expr],A	Reg Index			6
62h	MOV REG[expr],expr	Reg Direct Address	Immediate		8
63h	MOV REG[X+expr],expr	Reg Index	Immediate		9
MVI		**Move Indirect, Post Increment to Memory**			
3Eh	MVI A,[expr]	Direct Address (Page 0)		z	10
3Fh	MVI [expr],A	Direct Address (Page 0)			10
POP		**Pop Stack into Register**			
18h	POP A			z	5
20h	POP X				5
PUSH		**Push Register onto Stack**			
08h	PUSH A				4
10h	PUSH X				4
ROMX		**Table Read**			
28h	ROMX			z	11
SWAP		**Swap**			
4Bh	SWAP A,X			z	5
4Ch	SWAP A,[expr]	Direct Address		z	7
4Dh	SWAP X,[expr]	Direct Address		z	7
4Eh	SWAP A,SP			z	5

Logical Instructions

				Flags	Cycles
AND		**Bitwise AND**			
21h	AND A,expr	Immediate		z	4
22h	AND A,[expr]	Direct Address		z	6
23h	AND A,[X+expr]	Index		z	7
24h	AND [expr],A	Direct Address		z	7
25h	AND [X+expr],A	Index		z	8
26h	AND [expr],expr	Direct Address	Immediate	z	9
27h	AND [X+expr],expr	Index	Immediate	z	10
41h	AND REG[expr],expr	Reg Direct Address	Immediate	z	9
42h	AND REG[X+expr],expr	Reg Index	Immediate	z	10
70h	AND F,expr	Immediate		c z	4
OR		**Bitwise OR**			
29h	OR A,expr	Immediate		z	4
2Ah	OR A,[expr]	Direct Address		z	6
2Bh	OR A,[X+expr]	Index		z	7
2Ch	OR [expr],A	Direct Address		z	7
2Dh	OR [X+expr],A	Index		z	8
2Eh	OR [expr],expr	Direct Address	Immediate	z	9
2Fh	OR [X+expr],expr	Index	Immediate	z	10
43h	OR REG[expr],expr	Reg Direct Address	Immediate	z	9
44h	OR REG[X+expr],expr	Reg Index	Immediate	z	10
71h	OR F,expr	Immediate		c z	4
XOR		**Bitwise XOR**			
31h	XOR A,expr	Immediate		z	4
32h	XOR A,[expr]	Direct Address		z	6
33h	XOR A,[X+expr]	Index		z	7
34h	XOR [expr],A	Direct Address		z	7
35h	XOR [X+expr],A	Index		z	8
36h	XOR [expr],expr	Direct Address	Immediate	z	9
37h	XOR [X+expr],expr	Index	Immediate	z	10
45h	XOR REG[expr],expr	Reg Direct Address	Immediate	z	9
46h	XOR REG[X+expr],expr	Reg Index	Immediate	z	10
72h	XOR F,expr	Immediate		c z	4
CPL		**Complement Accumulator**			
73h	CPL A			z	4
RLC		**Rotate Left through Carry**			
6Ah	RLC A			c z	4
6Bh	RLC [expr]	Direct Address		c z	7
6Ch	RLC [X+expr]	Index		c z	8
RRC		**Rotate Right through Carry**			
6Dh	RRC A			c z	4
6Eh	RRC [expr]	Direct Address		c z	7
6Fh	RRC [X+expr]	Index		c z	8

3 Memory Organization

3.1 Flash Program Memory Organization

3.1.1 Flash Program Memory Map

After Reset	Address	Description
PC ⇨	0x0000	Reset Vector
	0x0002	Supply Monitor Interrupt Vector
	0x0004	DBA 00 PSoC Block Interrupt Vector
	0x0006	DBA 01 PSoC Block Interrupt Vector
	0x0008	DCA 02 PSoC Block Interrupt Vector
	0x000A	DCA 03 PSoC Block Interrupt Vector
	0x000C	DBA 04 PSoC Block Interrupt Vector
	0x000E	DBA 05 PSoC Block Interrupt Vector
	0x0010	DCA 06 PSoC Block Interrupt Vector
	0x0012	DCA 07 PSoC Block Interrupt Vector
	0x0014	Analog Column 0 Interrupt Vector
	0x0016	Analog Column 1 Interrupt Vector
	0x0018	Analog Column 2 Interrupt Vector
	0x001A	Analog Column 3 Interrupt Vector
	0x001C	GPIO Interrupt Vector
	0x001E	Sleep Timer Interrupt Vector
	0x0020	On-Chip User Program Memory Starts Here
		***
		***
		***
	0x3FFF	16K Flash Maximum Depending on Version

3.2 RAM Data Memory Organization

3.2.1 RAM Data Memory Map

Extension Bits	Address	Description
000	0x00	Bottom of Hardware Stack
000	0x01	⇩⇩⇩
000	0x02	Stack Grows This Way
000	0x03	⇩⇩⇩
000	0x04	***
000	0xXX	User Defined Top of Stack
000	0xXX	First General Purpose RAM Location
000	0xXX	***
000	0xXX	***
000	0xFF	Top of First Data Memory Page

4 Register Organization

4.1 Introduction

There are two register banks implemented on these devices. Each bank contains 256 bytes. The purpose of these register banks is to personalize and parameterize the on-chip resources as well as read and write data values.

The user selects between the two banks by setting the XIO bit in the CPU_F Flag Register.

4.2 Register Bank 0 Map

Register Name	Address	Page	Access	Register Name	Address	Page	Access	Register Name	Address	Page	Access	Register Name	Address	Page	Access
PRT0DR	00h	25	RW	Reserved	40h			ASA10CR0	80h	62	RW	Reserved	C0h		
PRT0IE	01h	25	W	Reserved	41h			ASA10CR1	81h	64	RW	Reserved	C1h		
PRT0GS	02h	25	W	Reserved	42h			ASA10CR2	82h	65	RW	Reserved	C2h		
Reserved	03h			Reserved	43h			ASA10CR3	83h	67	RW	Reserved	C3h		
PRT1DR	04h	25	RW	Reserved	44h			ASB11CR0	84h	68	RW	Reserved	C4h		
PRT1IE	05h	25	W	Reserved	45h			ASB11CR1	85h	70	RW	Reserved	C5h		
PRT1GS	06h	25	W	Reserved	46h			ASB11CR2	86h	71	RW	Reserved	C6h		
Reserved	07h			Reserved	47h			ASB11CR3	87h	73	RW	Reserved	C7h		
PRT2DR	08h	25	RW	Reserved	48h			ASA12CR0	88h	62	RW	Reserved	C8h		
PRT2IE	09h	25	W	Reserved	49h			ASA12CR1	89h	64	RW	Reserved	C9h		
PRT2GS	0Ah	25	W	Reserved	4Ah			ASA12CR2	8Ah	65	RW	Reserved	CAh		
Reserved	0Bh			Reserved	4Bh			ASA12CR3	8Bh	67	RW	Reserved	CBh		
PRT3DR	0Ch	25	RW	Reserved	4Ch			ASB13CR0	8Ch	68	RW	Reserved	CCh		
PRT3IE	0Dh	25	W	Reserved	4Dh			ASB13CR1	8Dh	70	RW	Reserved	CDh		
PRT3GS	0Eh	25	W	Reserved	4Eh			ASB13CR2	8Eh	71	RW	Reserved	CEh		
Reserved	0Fh			Reserved	4Fh			ASB13CR3	8Fh	73	RW	Reserved	CFh		
PRT4DR	10h	25	RW	Reserved	50h			ASB20CR0	90h	68	RW	Reserved	D0h		
PRT4IE	11h	25	W	Reserved	51h			ASB20CR1	91h	70	RW	Reserved	D1h		
PRT4GS	12h	25	W	Reserved	52h			ASB20CR2	92h	71	RW	Reserved	D2h		
Reserved	13h			Reserved	53h			ASB20CR3	93h	73	RW	Reserved	D3h		
PRT5DR	14h	25	RW	Reserved	54h			ASA21CR0	94h	62	RW	Reserved	D4h		
PRT5IE	15h	25	W	Reserved	55h			ASA21CR1	95h	64	RW	Reserved	D5h		
PRT5GS	16h	25	W	Reserved	56h			ASA21CR2	96h	65	RW	Reserved	D6h		
Reserved	17h			Reserved	57h			ASA21CR3	97h	67	RW	Reserved	D7h		
Reserved	18h			Reserved	58h			ASB22CR0	98h	68	RW	Reserved	D8h		
Reserved	19h			Reserved	59h			ASB22CR1	99h	70	RW	Reserved	D9h		
Reserved	1Ah			Reserved	5Ah			ASB22CR2	9Ah	71	RW	Reserved	DAh		
Reserved	1Bh			Reserved	5Bh			ASB22CR3	9Bh	73	RW	Reserved	DBh		
Reserved	1Ch			Reserved	5Ch			ASA23CR0	9Ch	62	RW	Reserved	DCh		
Reserved	1Dh			Reserved	5Dh			ASA23CR1	9Dh	64	RW	Reserved	DDh		
Reserved	1Eh			Reserved	5Eh			ASA23CR2	9Eh	65	RW	Reserved	DEh		
Reserved	1Fh			Reserved	5Fh			ASA23CR3	9Fh	67	RW	Reserved	DFh		
DBA00DR0	20h	41	*	AMX_IN	60h	76	RW	Reserved	A0h			INT_MSK0	E0h	35	RW
DBA00DR1	21h	41	*	Reserved	61h			Reserved	A1h			INT_MSK1	E1h	36	RW
DBA00DR2	22h	41	*	Reserved	62h			Reserved	A2h			INT_VC	E2h	36	RW
DBA00CR0	23h	42	*	ARF_CR	63h	78	RW	Reserved	A3h			RES_WDT	E3h	87	RW
DBA01DR0	24h	41	*	CMP_CR	64h	74	R	Reserved	A4h			DEC_DH/DEC_CL	E4h	85	RW
DBA01DR1	25h	41	*	ASY_CR	65h	74	*	Reserved	A5h			DEC_DL	E5h	85	R
DBA01DR2	26h	41	*	Reserved	66h			Reserved	A6h			DEC_CR	E6h	84	RW
DBA01CR0	27h	42	*	Reserved	67h			Reserved	A7h			Reserved	E7h		
DCA02DR0	28h	41	*	Reserved	68h			Reserved	A8h			MUL_X	E8h	81	W
DCA02DR1	29h	41	*	Reserved	69h			Reserved	A9h			MUL_Y	E9h	82	W
DCA02DR2	2Ah	41	*	Reserved	6Ah			Reserved	AAh			MUL_DH	EAh	82	R
DCA02CR0	2Bh	42	*	Reserved	6Bh			Reserved	ABh			MUL_DL	EBh	82	R
DCA03DR0	2Ch	41	*	Reserved	6Ch			Reserved	ACh			ACC_DR1/MAC_X	ECh	82	RW
DCA03DR1	2Dh	41	*	Reserved	6Dh			Reserved	ADh			ACC_DR0/MAC_Y	EDh	83	RW
DCA03DR2	2Eh	41	*	Reserved	6Eh			Reserved	AEh			ACC_DR3/MAC_CL0	EEh	83	RW
DCA03CR0	2Fh	42	*	Reserved	6Fh			Reserved	AFh			ACC_DR2/MAC_CL1	EFh	83	RW
DBA04DR0	30h	41	*	Reserved	70h			Reserved	B0h			Reserved	F0h		
DBA04DR1	31h	41	*	ACA00CR0	71h	56	RW	Reserved	B1h			Reserved	F1h		
DBA04DR2	32h	41	*	ACA00CR1	72h	58	RW	Reserved	B2h			Reserved	F2h		
DBA04CR0	33h	42	*	ACA00CR2	73h	59	RW	Reserved	B3h			Reserved	F3h		
DBA05DR0	34h	41	*	Reserved	74h			Reserved	B4h			Reserved	F4h		
DBA05DR1	35h	41	*	ACA01CR0	75h	56	RW	Reserved	B5h			Reserved	F5h		
DBA05DR2	36h	41	*	ACA01CR1	76h	58	RW	Reserved	B6h			Reserved	F6h		
DBA05CR0	37h	42	*	ACA01CR2	77h	59	RW	Reserved	B7h			Reserved	F7h		
DCA06DR0	38h	41	*	Reserved	78h			Reserved	B8h			Reserved	F8h		
DCA06DR1	39h	41	*	ACA02CR0	79h	56	RW	Reserved	B9h			Reserved	F9h		
DCA06DR2	3Ah	41	*	ACA02CR1	7Ah	58	RW	Reserved	BAh			Reserved	FAh		
DCA06CR0	3Bh	42	*	ACA02CR2	7Bh	59	RW	Reserved	BBh			Reserved	FBh		
DCA07DR0	3Ch	41	*	Reserved	7Ch			Reserved	BCh			Reserved	FCh		
DCA07DR1	3Dh	41	*	ACA03CR0	7Dh	56	RW	Reserved	BDh			Reserved	FDh		
DCA07DR2	3Eh	41	*	ACA03CR1	7Eh	58	RW	Reserved	BEh			Reserved	FEh		
DCA07CR0	3Fh	42	*	ACA03CR2	7Fh	59	RW	Reserved	BFh			CPU_SCR	FFh	86	*

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

4.3 Register Bank 1 Map

Register Name	Address	Page	Access	Register Name	Address	Page	Access	Register Name	Address	Page	Access	Register Name	Address	Page	Access
PRT0DM0	00h	26	W	Reserved	40h			ASA10CR0	80h	62	RW	Reserved	C0h		
PRT0DM1	01h	26	W	Reserved	41h			ASA10CR1	81h	64	RW	Reserved	C1h		
PRT0IC0	02h	26	W	Reserved	42h			ASA10CR2	82h	65	RW	Reserved	C2h		
PRT0IC1	03h	27	W	Reserved	43h			ASA10CR3	83h	67	RW	Reserved	C3h		
PRT1DM0	04h	26	W	Reserved	44h			ASB11CR0	84h	68	RW	Reserved	C4h		
PRT1DM1	05h	26	W	Reserved	45h			ASB11CR1	85h	70	RW	Reserved	C5h		
PRT1IC0	06h	26	W	Reserved	46h			ASB11CR2	86h	71	RW	Reserved	C6h		
PRT1IC1	07h	27	W	Reserved	47h			ASB11CR3	87h	73	RW	Reserved	C7h		
PRT2DM0	08h	26	W	Reserved	48h			ASA12CR0	88h	62	RW	Reserved	C8h		
PRT2DM1	09h	26	W	Reserved	49h			ASA12CR1	89h	64	RW	Reserved	C9h		
PRT2IC0	0Ah	26	W	Reserved	4Ah			ASA12CR2	8Ah	65	RW	Reserved	CAh		
PRT2IC1	0Bh	27	W	Reserved	4Bh			ASA12CR3	8Bh	67	RW	Reserved	CBh		
PRT3DM0	0Ch	26	W	Reserved	4Ch			ASB13CR0	8Ch	68	RW	Reserved	CCh		
PRT3DM1	0Dh	26	W	Reserved	4Dh			ASB13CR1	8Dh	70	RW	Reserved	CDh		
PRT3IC0	0Eh	26	W	Reserved	4Eh			ASB13CR2	8Eh	71	RW	Reserved	CEh		
PRT3IC1	0Fh	27	W	Reserved	4Fh			ASB13CR3	8Fh	73	RW	Reserved	CFh		
PRT4DM0	10h	26	W	Reserved	50h			ASB20CR0	90h	68	RW	Reserved	D0h		
PRT4DM1	11h	26	W	Reserved	51h			ASB20CR1	91h	70	RW	Reserved	D1h		
PRT4IC0	12h	26	W	Reserved	52h			ASB20CR2	92h	71	RW	Reserved	D2h		
PRT4IC1	13h	27	W	Reserved	53h			ASB20CR3	93h	73	RW	Reserved	D3h		
PRT5DM0	14h	26	W	Reserved	54h			ASA21CR0	94h	62	RW	Reserved	D4h		
PRT5DM1	15h	26	W	Reserved	55h			ASA21CR1	95h	64	RW	Reserved	D5h		
PRT5IC0	16h	26	W	Reserved	56h			ASA21CR2	96h	65	RW	Reserved	D6h		
PRT5IC1	17h	27	W	Reserved	57h			ASA21CR3	97h	67	RW	Reserved	D7h		
Reserved	18h			Reserved	58h			ASB22CR0	98h	68	RW	Reserved	D8h		
Reserved	19h			Reserved	59h			ASB22CR1	99h	70	RW	Reserved	D9h		
Reserved	1Ah			Reserved	5Ah			ASB22CR2	9Ah	71	RW	Reserved	DAh		
Reserved	1Bh			Reserved	5Bh			ASB22CR3	9Bh	73	RW	Reserved	DBh		
Reserved	1Ch			Reserved	5Ch			ASA23CR0	9Ch	62	RW	Reserved	DCh		
Reserved	1Dh			Reserved	5Dh			ASA23CR1	9Dh	64	RW	Reserved	DDh		
Reserved	1Eh			Reserved	5Eh			ASA23CR2	9Eh	65	RW	Reserved	DEh		
Reserved	1Fh			Reserved	5Fh			ASA23CR3	9Fh	67	RW	Reserved	DFh		
DBA00FN	20h	38	RW	CLK_CR0	60h	32	RW	Reserved	A0h			OSC_CR0	E0h	31	RW
DBA00IN	21h	39	RW	CLK_CR1	61h	33	RW	Reserved	A1h			OSC_CR1	E1h	31	RW
DBA00OU	22h	40	RW	ABF_CR	62h	77	W	Reserved	A2h			Reserved	E2h		
Reserved	23h			AMD_CR	63h	79	RW	Reserved	A3h			VLT_CR	E3h	89	RW
DBA01FN	24h	38	RW	Reserved	64h			Reserved	A4h			Reserved	E4h		
DBA01IN	25h	39	RW	Reserved	65h			Reserved	A5h			Reserved	E5h		
DBA01OU	26h	40	RW	Reserved	66h			Reserved	A6h			Reserved	E6h		
Reserved	27h			Reserved	67h			Reserved	A7h			Reserved	E7h		
DCA02FN	28h	38	RW	Reserved	68h			Reserved	A8h			IMO_TR	E8h	28	W
DCA02IN	29h	39	RW	Reserved	69h			Reserved	A9h			ILO_TR	E9h	28	W
DCA02OU	2Ah	40	RW	Reserved	6Ah			Reserved	AAh			BDG_TR	EAh	89	W
Reserved	2Bh			Reserved	6Bh			Reserved	ABh			ECO_TR	EBh	29	W
DCA03FN	2Ch	38	RW	Reserved	6Ch			Reserved	ACh			Reserved	ECh		
DCA03IN	2Dh	39	RW	Reserved	6Dh			Reserved	ADh			Reserved	EDh		
DCA03OU	2Eh	40	RW	Reserved	6Eh			Reserved	AEh			Reserved	EEh		
Reserved	2Fh			Reserved	6Fh			Reserved	AFh			Reserved	EFh		
DBA04FN	30h	38	RW	Reserved	70h			Reserved	B0h			Reserved	F0h		
DBA04IN	31h	39	RW	ACA00CR0	71h	56	RW	Reserved	B1h			Reserved	F1h		
DBA04OU	32h	40	RW	ACA00CR1	72h	58	RW	Reserved	B2h			Reserved	F2h		
Reserved	33h			ACA00CR2	73h	59	RW	Reserved	B3h			Reserved	F3h		
DBA05FN	34h	38	RW	Reserved	74h			Reserved	B4h			Reserved	F4h		
DBA05IN	35h	39	RW	ACA01CR0	75h	56	RW	Reserved	B5h			Reserved	F5h		
DBA05OU	36h	40	RW	ACA01CR1	76h	58	RW	Reserved	B6h			Reserved	F6h		
Reserved	37h			ACA01CR2	77h	59	RW	Reserved	B7h			Reserved	F7h		
DCA06FN	38h	38	RW	Reserved	78h			Reserved	B8h			Reserved	F8h		
DCA06IN	39h	39	RW	ACA02CR0	79h	56	RW	Reserved	B9h			Reserved	F9h		
DCA06OU	3Ah	40	RW	ACA02CR1	7Ah	58	RW	Reserved	BAh			Reserved	FAh		
Reserved	3Bh			ACA02CR2	7Bh	59	RW	Reserved	BBh			Reserved	FBh		
DCA07FN	3Ch	38	RW	Reserved	7Ch			Reserved	BCh			Reserved	FCh		
DCA07IN	3Dh	39	RW	ACA03CR0	7Dh	56	RW	Reserved	BDh			Reserved	FDh		
DCA07OU	3Eh	40	RW	ACA03CR1	7Eh	58	RW	Reserved	BEh			Reserved	FEh		
Reserved	3Fh			ACA03CR2	7Fh	59	RW	Reserved	BFh			CPU_SCR	FFh	86	*

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

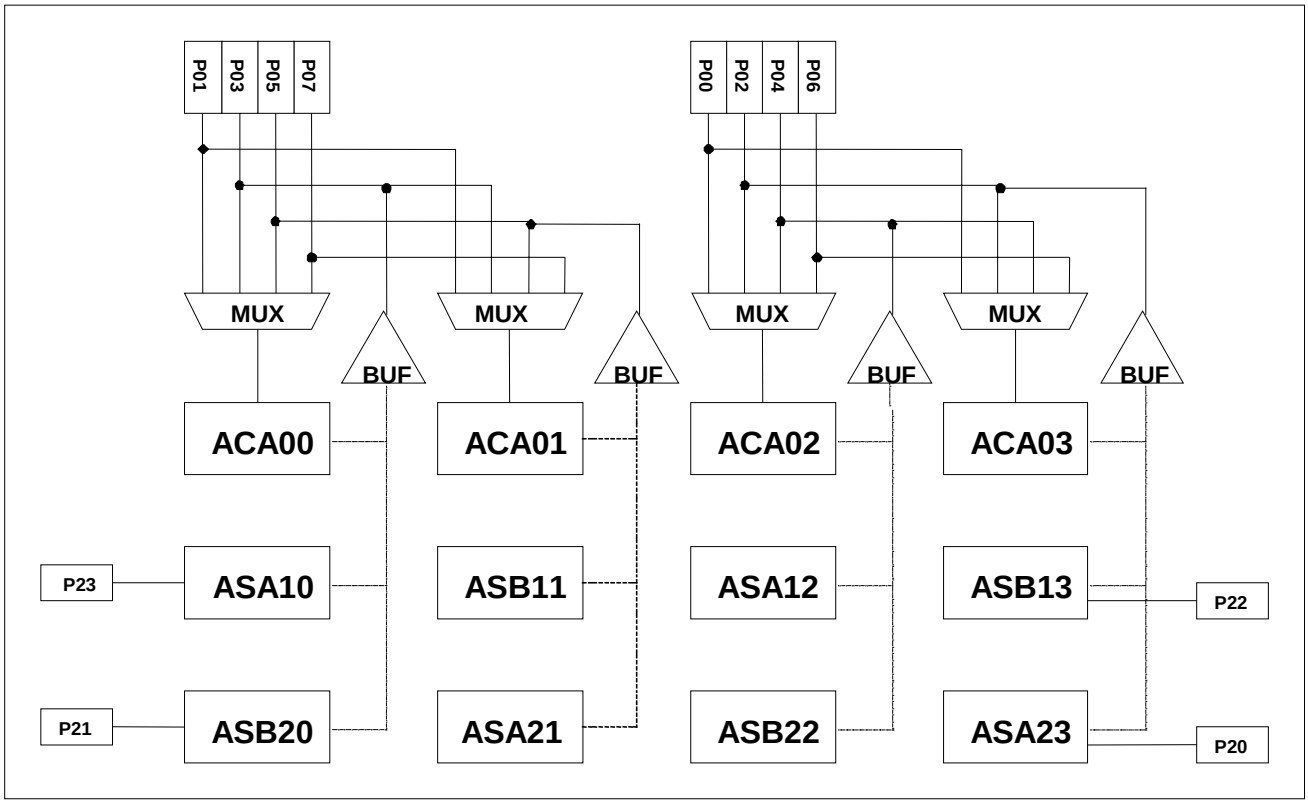
5 I/O Ports

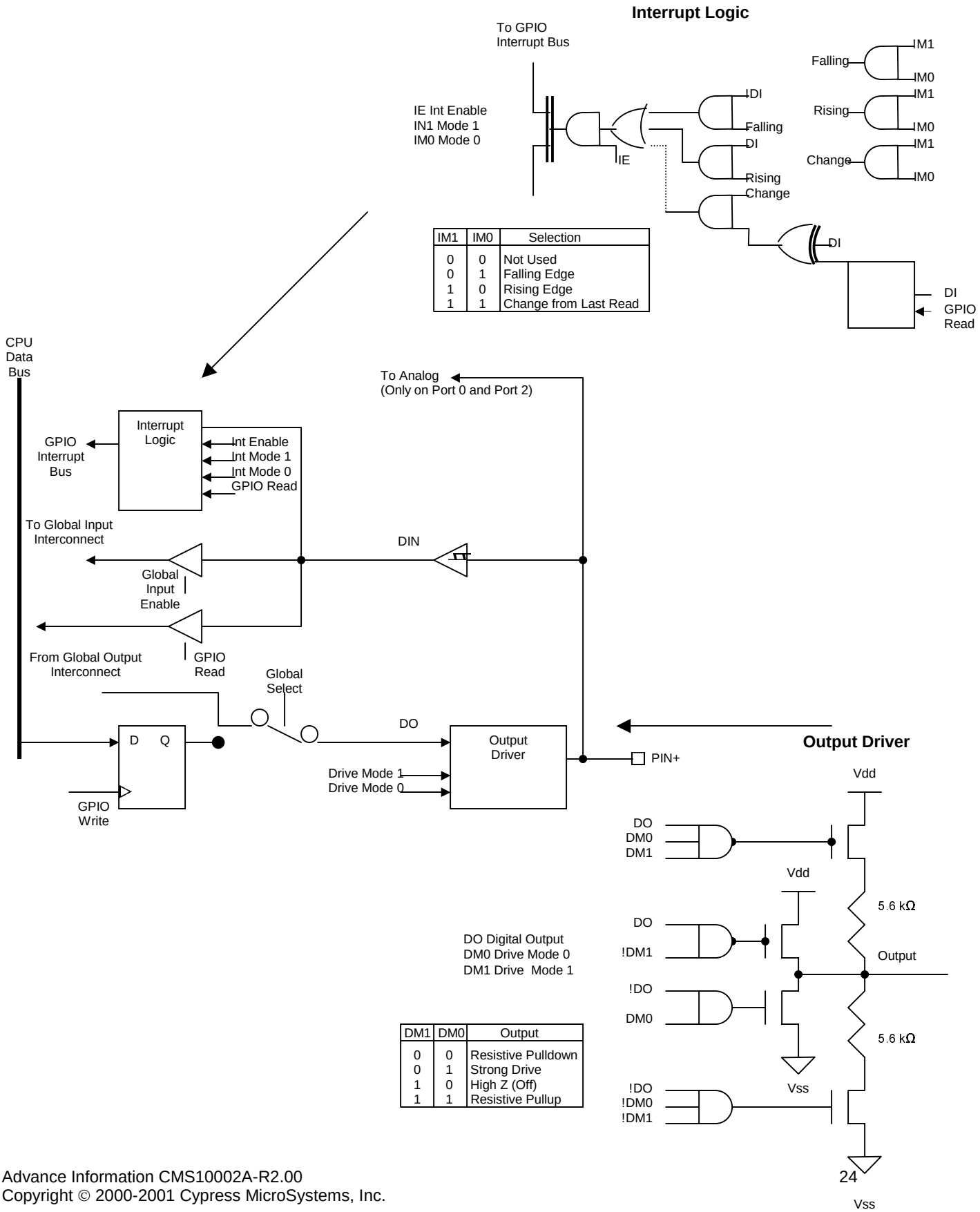
5.1 Introduction

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

- General-purpose digital input readable by the CPU.
- General-purpose digital output writeable by the CPU.
- Independent control of data direction for each port bit.
- Independent access for each port bit to Global Input and Global Output busses.
- Interrupt programmable to assert on rising edge, falling edge, or change from last pin state.
- Output drive strength programmable in logic 0 and 1 states as strong, resistive (pull-up or pull-down), or high impedance.

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





5.2 I/O Registers

5.2.1 Port x Data Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]** When written is the bits for output on port pins. When read is the state of the port pins.

Port 0 Data Register (PRT0DR, Address = Bank 0, 00h)
Port 1 Data Register (PRT1DR, Address = Bank 0, 04h)
Port 2 Data Register (PRT2DR, Address = Bank 0, 08h)
Port 3 Data Register (PRT3DR, Address = Bank 0, 0Ch)
Port 4 Data Register (PRT4DR, Address = Bank 0, 10h)
Port 5 Data Register (PRT5DR, Address = Bank 0, 14h) **Note**: If implemented, Port 5 is 4-bits wide

5.2.2 Port x Interrupt Enable Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W
Bit Name	Int En [7]	Int En [6]	Int En [5]	Int En [4]	Int En [3]	Int En [2]	Int En [1]	Int En [0]

Bit [7:0]: **Int En [7:0]** When written sets the pin interrupt state
0 = Interrupt disabled for pin
1 = Interrupt enabled for pin

Port 0 Interrupt Enable Register (PRT0IE, Address = Bank 0, 01h)
Port 1 Interrupt Enable Register (PRT1IE, Address = Bank 0, 05h)
Port 2 Interrupt Enable Register (PRT2IE, Address = Bank 0, 09h)
Port 3 Interrupt Enable Register (PRT3IE, Address = Bank 0, 0Dh)
Port 4 Interrupt Enable Register (PRT4IE, Address = Bank 0, 11h)
Port 5 Interrupt Enable Register (PRT5IE, Address = Bank 0, 15h) **Note**: If implemented, Port 5 is 4-bits wide

5.2.3 Port x Global Select Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W
Bit Name	GlobSel [7]	GlobSel [6]	GlobSel [5]	GlobSel [4]	GlobSel [3]	GlobSel [2]	GlobSel [1]	GlobSel [0]

Bit [7:0]: **Global Select [7:0]** When written determines whether a pin is connected to the Global Input Bus and Global Output Bus
0 = Not Connected
1 = Connected
Drive Mode xx = Global Select Register 0 = Standard CPU controlled port (Default).
Drive Mode 1 0 (High Z) = Global Select Register 1 = Direct Drive of associated Global Input line.
Drive Mode 0 0, 0 1, 1 1 = Global Select Register 1 = Direct Receive from associated Global Output line.

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

5.2.4 Port x Drive Mode 0 Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W
Bit Name	DM0 [7]	DM0 [6]	DM0 [5]	DM0 [4]	DM0 [3]	DM0 [2]	DM0 [1]	DM0 [0]

Bit [7:0]: **DM0 [7:0]** The two Drive Mode bits that control a particular port pin are treated as a pair and are decoded as follows:
Output State 0 = Drive Mode 0 0 = 0 Resistive (Default)
Output State 0 = Drive Mode 0 1 = 0 Strong
Output State 0 = Drive Mode 1 0 = High Z
Output State 0 = Drive Mode 1 1 = 0 Strong
Output State 1 = Drive Mode 0 0 = 1 Strong (Default)
Output State 1 = Drive Mode 0 1 = 1 Strong
Output State 1 = Drive Mode 1 0 = High Z
Output State 1 = Drive Mode 1 1 = 1 Resistive

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

5.2.5 Port x Drive Mode 1 Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W
Bit Name	DM1 [7]	DM1 [6]	DM1 [5]	DM1 [4]	DM1 [3]	DM1 [2]	DM1 [1]	DM1 [0]

Bit [7:0]: **DM1 [7:0]** See truth table for Port x Drive Mode 0 Register (PRT0DM0-PRT5DM0)

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

5.2.6 Port x Interrupt Control 0 Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W
Bit Name	IC0 [7]	IC0 [6]	IC0 [5]	IC0 [4]	IC0 [3]	IC0 [2]	IC0 [1]	IC0 [0]

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

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

5.2.7 Port x Interrupt Control 1 Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W
Bit Name	IC1 [7]	IC1 [6]	IC1 [5]	IC1 [4]	IC1 [3]	IC1 [2]	IC1 [1]	IC1 [0]

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

Port 0 Interrupt Control 1 Register (PRT0IC1, Address = Bank 1, 03h)
Port 1 Interrupt Control 1 Register (PRT1IC1, Address = Bank 1, 07h)
Port 2 Interrupt Control 1 Register (PRT2IC1, Address = Bank 1, 0Bh)
Port 3 Interrupt Control 1 Register (PRT3IC1, Address = Bank 1, 0Fh)
Port 4 Interrupt Control 1 Register (PRT4IC1, Address = Bank 1, 13h)
Port 5 Interrupt Control 1 Register (PRT5IC1, Address = Bank 1, 17h) **Note**: If implemented, Port 5 is 4-bits wide

6 Clocking

6.1 Oscillator Options

6.1.1 Internal Main Oscillator

The Internal Main Oscillator outputs two frequencies, 48MHz and 24MHz. In the absence of a high-precision input source from the external oscillator, the accuracy of this circuit will be +/- 2.5% over two voltage ranges and the entire temperature range. No external components are required to achieve this level of accuracy. However, there is a Main Oscillator Trim Register (IMO_TR) used to calibrate this oscillator into specified tolerance. A factory-programmed value is available, loaded into this register at reset. This register must be adjusted when operating voltage outside the range for which factory calibration was set.

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.

6.1.1.1 Main Oscillator Trim Register

Bit #	7	6	5	4	3	2	1	0
POR	FS*	FS*	FS*	FS*	FS*	FS*	FS*	FS*
Read/Write	W	W	W	W	W	W	W	W
Bit Name	IMO Trim [7]	IMO Trim [6]	IMO Trim [5]	IMO Trim [4]	IMO Trim [3]	IMO Trim [2]	IMO Trim [1]	IMO Trim [0]

Bit [7:0]: **IMO Trim [7:0]** Data value stored will alter the trimmed frequency of the Internal Main Oscillator. A larger value in this register will increase the speed of the Internal Main Oscillator

Main Oscillator Trim Register (IMO_TR, Address = Bank 1, E8h) *FS = Factory set trim value

6.1.2 Internal Low Speed Oscillator

An internal low speed oscillator of nominally 32.7kHz is available to generate Sleep wake-up interrupts and Watchdog resets if the user does not wish to attach a 32.768kHz watch crystal. This oscillator can also be used as a clocking source for the Digital PSoC blocks.

The oscillator operates in two different modes. A trim value applied to the Internal Low Speed Oscillator Trim Register (ILO_TR), shown below, will guarantee 32.7kHz with +/-20% accuracy across voltage and temperature while the IC is *not* in sleep mode. When the IC is put into sleep mode this oscillator drops into an ultra low current state and the accuracy is reduced to +/-50%.

6.1.2.1 Internal Low Speed Oscillator Trim Register

This register sets the adjustment for the internal low speed oscillator. The value placed in this register is based on factory testing. It is recommended that the user not alter this value.

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

Bit 7: **Reserved**

Bit 6: **Reserved**

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

Internal Low Speed Oscillator Trim Register (ILO_TR, Address = Bank 1, E9h) *FS = Factory set trim value

6.1.3 External Crystal Oscillator

The CrysIn and CrysOut pins support connection of a 32.768kHz watch crystal. To run from the external crystal, Bit 7 of the Oscillator Control 0 Register (OSC_CR0) must be set (default is off). Note that the internal low speed oscillator continues to run when this external function is selected until the oscillator is automatically switched over by the sleep timer interrupt. Feedback capacitors and bias circuitry for this oscillator are implemented internally.

The firmware steps involved in switching between the internal low speed oscillator to External Crystal Oscillator are as follows:

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

Note: The internal oscillator switches back instantaneously by writing the control bit to 0.

Transitions between oscillator domains may produce glitches on this clock bus.

6.1.3.1 External Crystal Oscillator Trim Register

This register sets the adjustment for the External Crystal Oscillator. The value placed in this register at reset is based on factory testing. It is recommended that the user not alter this value.

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	FS*	FS*	FS*	FS*
Read/Write	--	--	W	W	W	W	W	W
Bit Name	Reserved	Reserved	Reserved	Reserved	Amp [1]	Amp [0]	Bias [1]	Bias [0]

Bit 7: **Reserved**

Bit 6: **Reserved**

Bit 5: **Reserved**

Bit 4: **Reserved**

Bit [3:2]: **Amp [1:0]** Sets the amplitude of the adjustment, (not recommended for customer alteration)

Bit [1:0]: **Bias [1:0]** Sets the bias of the adjustment, (not recommended for customer alteration)

External Crystal Oscillator Trim Register (ECO_TR, Address = Bank 1, EBh) *FS = Factory set trim value

6.2 System Clocking Signals

There are twelve system-clocking signals that are used throughout the device. Referenced frequencies are based on use of 32.768kHz crystal. The names of these signals and their definitions are as follows:

Signal	Definition
48M	The direct 48MHz output from the internal main oscillator.
24M	The direct 24MHz output from the internal main oscillator.
24V1	The 24M output from the internal main oscillator that has been passed through a user-selectable 1 to 16 divider {F = 24MHz / (1 to 16) = 24MHz to 1.5MHz}. The divider value is found in the Oscillator Control 1 Register (OSC_CR1). Note that the divider will be N+1, based on a value of N written into the register bits.
24V2	The 24V1 signal that has been passed through an additional user-selectable 1 to 16 divider {F = 24MHz / ((1 to 16) * (1 to 16)) = 24MHz to 93.7kHz}. The divider value is found in the Oscillator Control 1 Register (OSC_CR1). Note that the divider will be N+1, based on a value of N written into the register bits.
32K	The multiplexed output of either the internal low speed oscillator or the external crystal oscillator.
CPU	The output from the internal main oscillator that has been passed through a divider that has eight user selectable ratios ranging from 1:1 to 1:256, yielding frequencies ranging from 24MHz to 93.7kHz.
SLP	The **32K** system-clocking signal that has been passed through a divider that has four user selectable ratios ranging from $1:2^6$ to $1:2^{15}$, yielding frequencies ranging from 512Hz to 1Hz. This signal is used to clock the sleep timer period.
ACLK0	A system-clocking signal that is driven by the clock output of a digital PSoC block and can be selected by the user to drive the clocking signal to an analog column. Any of the eight digital PSoC blocks can be muxed into this line using the ACLK0[2:0] bits in the Analog Clock Select Register (CLK_CR1).
ACLK1	A system-clocking signal that is driven by the clock output of a digital PSoC block and can be selected by the user to drive the clocking signal to an analog column. Any of the eight digital PSoC blocks can be muxed into this line using the ACLK1[2:0] bits in the Analog Clock Select Register (CLK_CR1).
ACLMN0_CLK	A system-clocking signal that can drive all the Analog PSoC blocks in Analog Column 0. This signal is derived from the muxed input of the **24V1**, **24V2**, **ACLK0**, and **ACLK1** system clock signals. The output of this mux is then passed through a 1:4 divider to reduce the frequency by a factor of four.
ACLMN1_CLK	A system-clocking signal that can drive all the Analog PSoC blocks in Analog Column 1. This signal is derived from the muxed input of the **24V1**, **24V2**, **ACLK0**, and **ACLK1** system clock signals. The output of this mux is then passed through a 1:4 divider to reduce the frequency by a factor of four.
ACLMN2_CLK	A system-clocking signal that can drive all the Analog PSoC blocks in Analog Column 2. This signal is derived from the muxed input of the **24V1**, **24V2**, **ACLK0**, and **ACLK1** system clock signals. The output of this mux is then passed through a 1:4 divider to reduce the frequency by a factor of four.
ACLMN3_CLK	A system-clocking signal that can drive all the Analog PSoC blocks in Analog Column 3. This signal is derived from the muxed input of the **24V1**, **24V2**, **ACLK0**, and **ACLK1** system clock signals. The output of this mux is then passed through a 1:4 divider to reduce the frequency by a factor of four.

6.2.1 CPU and Sleep Timer Clock Options

The CPU is clocked off the **CPU** system-clocking signal, which can be configured to run at one of 8 rates. This selection is independent from all other clock selection functions. It is completely safe for the CPU to change its clock rate without a timing hazard. The CPU clock period is determined by setting the CPU[2:0] bits in the Oscillator Control 0 Register (OSC_CR0).

The sleep timer is clocked off the **SLP** system-clocking signal. Setting the SLEEP1 and SLEEP0 bits in the Oscillator Control 0 Register (OSC_CR0) allow the user to select from the four available periods.

6.2.1.1 Oscillator Control 0 Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	--	RW	RW	RW	RW	RW	RW
Bit Name	32k Select	PLL Mode	Reserved	Sleep [1]	Sleep [0]	CPU [2]	CPU [1]	CPU [0]

Bit 7: **32k Select**
0 = Internal low precision 32kHz oscillator
1 = External crystal oscillator

Bit 6: **PLL Mode**
0 = Disabled
1 = Enabled, Internal Main is locked to External Crystal Oscillator

Bit 5: **Reserved**

Bit [4:3]: **Sleep [1:0]**
0 0 = N = 6 ((512Hz) or (1.95 ms period))
0 1 = N = 9 ((64 Hz) or (15.6 ms period))
1 0 = N = 12 ((8 Hz) or (125 ms period))
1 1 = N = 15 ((1 Hz) or (1 s period))

Bit [2:0]: **CPU [2:0]**
0 0 0 = 3MHz
0 0 1 = 6MHz
0 1 0 = 12MHz
0 1 1 = 24MHz
1 0 0 = 1.5MHz
1 0 1 = 750kHz
1 1 0 = 187.5kHz
1 1 1 = 93.7kHz

Oscillator Control 0 Register (OSC_CR0, Address = Bank 1, E0h)

6.2.1.2 Oscillator Control 1 Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	24V1 [3]	24V1 [2]	24V1 [1]	24V1 [0]	24V2 [3]	24V2 [2]	24V2 [1]	24V2 [0]

Bit [7:4]: **24V1 [3:0]** 4-bit data value determines the divider value for the **24V1** system clocking signal

Bit [3:0]: **24V2 [3:0]** 4-bit data value determines the divider value for the **24V2** system clocking signal

Oscillator Control 1 Register (OSC_CR1, Address = Bank 1, E1h)

6.2.2 Digital PSoC Block Clocking Options

All Digital PSoC block clocks are a user-selectable choice of **48M**, **24V1**, **24V2**, or **32K**, as well as clocking signals from other Digital PSoC blocks or general purpose I/O pins. There are a total of 16 possible clock options for each Digital PSoC block. See Digital PSoC block section for details.

6.2.3 Analog PSoC Block Clocking Options

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

1. First, if the user wants to use the **ACLK0** and **ACLK1** system-clocking signals, the digital PSoC blocks that serve as the source for these signals must be selected. This selection is made in the Analog Clock Select Register (CLK_CR1).
2. Next, the user must select the source for the **ACLMN0_CLK**, **ACLMN1_CLK, ACLMN2_CLK**, and **ACLMN3_CLK** system-clocking signals. The user will choose the clock for Acolumnx[1:0] bits in the Analog Column Clock Select Register (CLK_CR0) column. Each of the analog PSoC blocks in a particular Analog Column is clocked from the **ACLMN_CLKx** system-clocking signal for that column. (Note that the ACLMNx signals have a 1:4 divider on them.)

6.2.3.1 Analog Column Clock Select Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Acolumn3 [1]	Acolumn3 [0]	Acolumn2 [1]	Acolumn2 [0]	Acolumn1 [1]	Acolumn1 [0]	Acolumn0 [1]	Acolumn0 [0]

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

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

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

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

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

6.2.3.2 Analog Clock Select Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Reserved	SHDIS	ACLK1 [2]	ACLK1 [1]	ACLK1 [0]	ACLK0 [2]	ACLK0 [1]	ACLK0 [0]

Bit 7: **Reserved**

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

0 = Sample and hold function disabled
1 = Sample and hold function enabled

Bit [5:3]: **ACLK1 [2:0]**
0 0 0 = Digital Basic Type A Block 00
0 0 1 = Digital Basic Type A Block 01
0 1 0 = Digital Communications Type A Block 02
0 1 1 = Digital Communications Type A Block 03
1 0 0 = Digital Basic Type A Block 04
1 0 1 = Digital Basic Type A Block 05
1 1 0 = Digital Communications Type A Block 06
1 1 1 = Digital Communications Type A Block 07

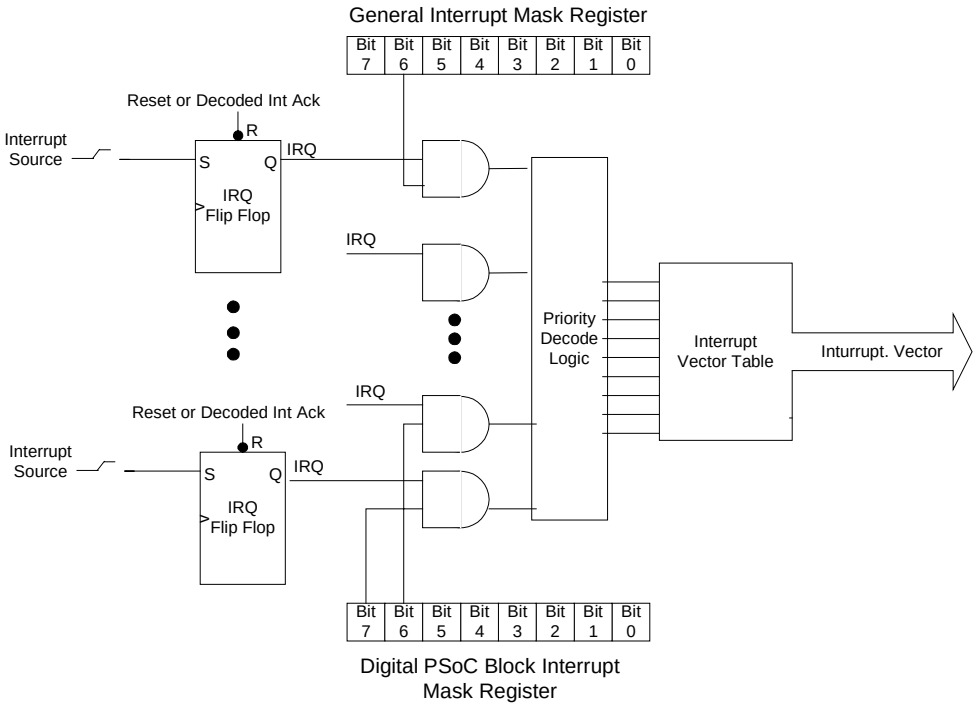
Bit [2:0]: **ACLK0 [2:0]** Same configurations as ACLK1 [2:0]
0 0 0 = Digital Basic Type A Block 00
0 0 1 = Digital Basic Type A Block 01
0 1 0 = Digital Communications Type A Block 02
0 1 1 = Digital Communications Type A Block 03
1 0 0 = Digital Basic Type A Block 04
1 0 1 = Digital Basic Type A Block 05
1 1 0 = Digital Communications Type A Block 06
1 1 1 = Digital Communications Type A Block 07

Analog Clock Select Register (CLK_CR1, Address = Bank 1, 61h)

7 Interrupts

7.1 Overview

Interrupts can be generated by the General Purpose I/O lines, the Power monitor, the internal Sleep Timer, the eight Digital PSoC blocks, and the four analog columns. Every interrupt has a separate enable bit, which are contained in the General Interrupt Mask Register (INT_MSK0) and the Digital PSoC Block Interrupt Mask Register (INT_MSK1). When the user writes a "1" to a particular bit position, this enables the interrupt associated with that position. There is a single Global Interrupt Enable bit in the Flags Register (CPU_F), which can disable all interrupts, or enable those interrupts that also have their individual interrupt bit enabled. During a reset, the enable bits in the General Interrupt Mask Register (INT_MSK0) and the Global Interrupt Enable bit, is cleared. The Interrupt Vector Register (INT_VC) holds the interrupt vector for the highest priority pending interrupt when read, and when written will clear all pending interrupts.



Interrupt Control Architecture

The interrupt controller contains a separate flip-flop for each interrupt. When an interrupt is generated, it is registered as a pending interrupt. It will stay pending until it is serviced or a reset occurs. A pending interrupt will only generate an interrupt request when enabled by the appropriate mask bit in the Interrupt Enable Register (PRT0IE-PRT5IE), and the Global IE bit in the CPU_F register is set.

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

Nested interrupts can be accomplished by re-enabling interrupts inside an interrupt service routine. The user must store sufficient information to maintain machine state if this is done.

Each Digital PSoC block has its own unique Interrupt Vector and Interrupt Enable bit. There are also individual interrupt vectors for each of the Analog columns, Supply Voltage Monitor, Sleep Timer and General Purpose I/Os. Functions can be stopped/started by the Enable bit within the Function Register (DBA00FN-DBA07FN).

GPIO Interrupts are polarity configurable and pin-wise maskable (within each Port's pin configuration registers). They all share the same interrupt priority and vector.

The Interrupt Vector Register (INT_VC) holds the interrupt vector for the highest priority pending interrupt when read, and when written will clear all pending interrupts.

7.2 Interrupt Vectors

FLASH-based Interrupt Vector Table

Address	Interrupt Priority Number	Description
0x0002	1	Supply Monitor Interrupt Vector
0x0004	2	DBA 00 PSoC Block Interrupt Vector
0x0006	3	DBA 01 PSoC Block Interrupt Vector
0x0008	4	DCA 02 PSoC Block Interrupt Vector
0x000A	5	DCA 03 PSoC Block Interrupt Vector
0x000C	6	DBA 04 PSoC Block Interrupt Vector
0x000E	7	DBA 05 PSoC Block Interrupt Vector
0x0010	8	DCA 06 PSoC Block Interrupt Vector
0x0012	9	DCA 07 PSoC Block Interrupt Vector
0x0014	10	Acolumn 0 Interrupt Vector
0x0016	11	Acolumn 1 Interrupt Vector
0x0018	12	Acolumn 2 Interrupt Vector
0x001A	13	Acolumn 3 Interrupt Vector
0x001C	14	GPIO Interrupt Vector
0x001E	15	Sleep Timer Interrupt Vector
0x0020		On-Chip Program Memory Starts Here

Important: The interrupt vectors are instructions **not** addresses. Typically these would be JUMP instructions to the start of the interrupt handling routine for that interrupt.

7.3 Interrupt Masks

7.3.1 General Interrupt Mask Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Reserved	Sleep	GPIO	Acolumn3	Acolumn2	Acolumn1	Acolumn0	Voltage Monitor

Bit 7: **Reserved**

Bit 6: **Sleep**
0 = Disabled
1 = Enabled

Bit 5: **GPIO**
0 = Disabled
1 = Enabled

Bit [4:1]: **Acolumn [3:0]**
0 = Disabled
1 = Enabled

Bit 0: **Voltage Monitor**
0 = Disabled
1 = Enabled

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

7.3.2 Digital PSoC Block Interrupt Mask Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	DCA07	DCA06	DBA05	DBA04	DCA03	DCA02	DBA01	DBA00

Bit 7: **DCA07**
0 = Disabled
1 = Enabled

Bit 6: **DCA06**
0 = Disabled
1 = Enabled

Bit 5: **DBA05**
0 = Disabled
1 = Enabled

Bit 4: **DBA04**
0 = Disabled
1 = Enabled

Bit 3: **DCA03**
0 = Disabled
1 = Enabled

Bit 2: **DCA02**
0 = Disabled
1 = Enabled

Bit 1: **DBA01**
0 = Disabled
1 = Enabled

Bit 0: **DBA00**
0 = Disabled
1 = Enabled

Digital PSoC Block Interrupt Mask Register (INT_MSK1, Address = Bank 0, E1h)

7.3.3 Interrupt Vector Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Data[7]	Data[6]	Data[5]	Data[4]	Data[3]	Data[2]	Data[1]	Data[0]

Bit [7:0] Data [7:0]
8-bit data value holds the interrupt vector for the highest priority pending interrupt. Writing to this register will clear all pending interrupts

Interrupt Vector Register (INT_VC, Address = Bank 0, E2h)

7.4 Interrupt on Pin Change

Any general purpose I/O can be used as an interrupt source. The appropriate enable bits in the Interrupt Vector Register (INT_VC) for a particular pin must be set in order to activate interrupts. There are user selectable options to generate an interrupt on 1) any change from previous state, 2) rising edge, and 3) falling edge.

8 PSoC Blocks

8.1 Overview

PSoC blocks are user configurable system resources. On-chip digital and analog PSoC blocks reduce the need for many MCU part types and external peripheral components. Each PSoC block can be configured to provide a wide variety of user modules. The PSoC Designer provides automated configuration of PSoC blocks by simply selecting the desired functions. PSoC Designer then generates the proper configuration information and can print a device data sheet unique to that configuration.

Digital PSoC blocks provide up to eight, 8-bit multipurpose timers/counters supporting multiple event timers, real-time clocks, Pulse Width Modulators (PWM), and CRCs. PSoC blocks may be configured independently or combined to provide longer functions. Communications configured PSoC blocks support full-duplex UARTs and SPI master or slave functions. If a digital PSoC block with more than 8 bits is needed, more than one may be chained together to form lengths of 1 through 8 bytes using configuration bits.

Twelve 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 like battery chargers and data acquisition without requiring external components. There are three Analog PSoC block types ContinuousTime (CT) blocks, and Type 1 and Type 2 Switch Capacitor (SC) blocks. CT blocks provide continuous time analog functions. SC blocks provide ADC and DAC analog functions. Currently, supported analog functions are 14 bit Multi-Slope and 12 bit Delta-Sigma ADC, successive approximation ADCs up to 9 bits, DACs up to 9 bits, programmable gain stages, sample and hold circuits, programmable filters, differential comparators, and temperature sensor.

8.2 Digital PSoC Blocks

8.2.1 Introduction

There are a total of eight 8-bit Digital PSoC blocks in this device family. Four of theses are the digital Basic Type A blocks and four are the Digital Communications Type A blocks. Each of these digital PSoC blocks can be configured independently for function, or used in combination. There are three configuration registers for each digital PSoC block that control I/O source sinking and block functions. There are also four data registers within each digital PSoC block.

Each digital PSoC block has a unique Interrupt Vector and Interrupt Enable bit. Functions can be stopped/started by a user-accessible Enable bit.

The Timer/Counter/Shifter/CRC/PRS/Deadband functions are available on the Digital Basic Type A blocks and also the Digital Communications Type A blocks. The UART and SPI communications functions are available on the Digital Communications Type A blocks.

The three configuration registers are; the Function Register (DBA00FN-DCA07FN) to select the block function and mode, the Input Register (DBA00IN-DCA07IN) to select data input and clock selection, and the Output Register (DBA00OU-DCA07OU) to select and enable function outputs.

The three data registers are designated Data 0, Data 1, Data 2. The function of these registers and their bit mapping is dependent on the overall block function selected by the user.

The one Control Register (DBA00CR0-DCA07CR0) is designated Control 0. The function of this register and its bit mapping is dependent on the overall block function selected by the user.

8.2.2 Registers

8.2.2.1 Digital Basic Type A/ Communications Type A Block xx Function Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Reserved	Reserved	End	Mode [1]	Mode [0]	Function [2]	Function [1]	Function [0]

Bit 7: **Reserved**

Bit 6: **Reserved**

Bit 5: **End**
0 = PSoC block is not the end of a chained function
1 = PSoC block is the end of a chained function, or unchained PSoC block

Bit 4: **Mode [1]** The definition of the Mode [1] bit depends on the block function selected
Timer: The Mode [1] bit signifies the Compare Type
0 = Less Than or Equal
1 = Less Than
Counter: The Mode [1] bit signifies the Compare Type
0 = Less Than or Equal
1 = Less Than
CRC/PRS: The Mode [1] bit is unused in this function
Dead Band: The Mode [1] bit is unused in this function
UART: The Mode [1] bit is unused in this function
SPI: The Mode [1] bit is unused in this function

Bit 3: **Mode [0]** The definition of the Mode [0] bit depends on the block function selected
Timer: The Mode [0] bit signifies Interrupt Type
0 = Terminal Count
1 = Compare True
Counter: The Mode [0] bit signifies Interrupt Type
0 = Terminal Count
1 = Compare True
CRC/PRS: The Mode [0] bit is unused in this function
Dead Band: The Mode [0] bit is unused in this function
UART: The Mode [0] bit signifies the Direction
0 = Receive
1 = Transmit
SPI: The Mode [0] bit signifies the Type
0 = Master
1 = Slave

Bit [2:0]: **Function [2:0]** The Function [2:0] bits select the block function which determines the basic hardware configuration
0 0 0 = Timer (chainable)
0 0 1 = Counter (chainable)
0 1 0 = CRC/PRS (Cyclical Redundancy Checker or Psuedo Random Sequencer) (chainable)
0 1 1 = Reserved
1 0 0 = Deadband for Pulse Width Modulator
1 0 1 = UART (function only available on DCA type blocks)
1 1 0 = SPI (function only available on DCA type blocks)
1 1 1 = Reserved

Digital Basic Type A Block 00 Function Register	(DBA00FN, Address = Bank 1, 20h)
Digital Basic Type A Block 01 Function Register	(DBA01FN, Address = Bank 1, 24h)
Digital Communications Type A Block 02 Function Register	(DCA02FN, Address = Bank 1, 28h)
Digital Communications Type A Block 03 Function Register	(DCA03FN, Address = Bank 1, 2Ch)
Digital Basic Type A Block 04 Function Register	(DBA04FN, Address = Bank 1, 30h)
Digital Basic Type A Block 05 Function Register	(DBA05FN, Address = Bank 1, 34h)
Digital Communications Type A Block 06 Function Register	(DCA06FN, Address = Bank 1, 38h)
Digital Communications Type A Block 07 Function Register	(DCA07FN, Address = Bank 1, 3Ch)

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

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

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Data 3	Data 2	Data 1	Data 0	Clock 3	Clock 2	Clock 1	Clock 0

Bit [7:4]: **Data [3:0]**

0 0 0 0 = Data = 0
0 0 0 1 = Data = 1
0 0 1 0 = Digital Basic Type A Block 00
0 0 1 1 = Chain Function to Previous PSoC block
0 1 0 0 = Analog Column Out 0
0 1 0 1 = Analog Column Out 1
0 1 1 0 = Analog Column Out 2
0 1 1 1 = Analog Column Out 3
1 0 0 0 = Select Global Output[0] (for Digital Blocks 00 to 03) **or** Select Global Output[4] (for Digital Blocks 04 to 07)
1 0 0 1 = Select Global Output[1] (for Digital Blocks 00 to 03) **or** Select Global Output[5] (for Digital Blocks 04 to 07)
1 0 1 0 = Select Global Output[2] (for Digital Blocks 00 to 03) **or** Select Global Output[6] (for Digital Blocks 04 to 07)
1 0 1 1 = Select Global Output[3] (for Digital Blocks 00 to 03) **or** Select Global Output[7] (for Digital Blocks 04 to 07)
1 1 0 0 = Select Global Input[0] (for Digital Blocks 00 to 03) **or** Select Global Input[4] (for Digital Blocks 04 to 07)
1 1 0 1 = Select Global Input[1] (for Digital Blocks 00 to 03) **or** Select Global Input[5] (for Digital Blocks 04 to 07)
1 1 1 0 = Select Global Input[2] (for Digital Blocks 00 to 03) **or** Select Global Input[6] (for Digital Blocks 04 to 07)
1 1 1 1 = Select Global Input[3] (for Digital Blocks 00 to 03) **or** Select Global Input[7] (for Digital Blocks 04 to 07)

Bit [3:0]: **Clock [3:0]**

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

Digital Basic Type A Block 00 Input Register	(DBA00IN, Address = Bank 1, 21h)
Digital Basic Type A Block 01 Input Register	(DBA01IN, Address = Bank 1, 25h)
Digital Communications Type A Block 02 Input Register	(DCA02IN, Address = Bank 1, 29h)
Digital Communications Type A Block 03 Input Register	(DCA03IN, Address = Bank 1, 2Dh)
Digital Basic Type A Block 04 Input Register	(DBA04IN, Address = Bank 1, 31h)
Digital Basic Type A Block 05 Input Register	(DBA05IN, Address = Bank 1, 35h)
Digital Communications Type A Block 06 Input Register	(DCA06IN, Address = Bank 1, 39h)
Digital Communications Type A Block 07 Input Register	(DCA07IN, Address = Bank 1, 3Dh)

The Data/Enable inputs to each Digital PSoC block serve as Clock Enables or Data Input depending on the Digital PSoC block's programmed function. If Previous Digital PSoC block is selected for Data/Enable then the selected DSCM receives its Data, Enable, Zero Detect, and all chaining information from the previous Digital PSoC block.

The Clock[3:0] bits select multiple sources for the clock for each Digital PSoC block. The sources for each Digital PSoC block clock can form the Global Input lines, System Clocks, and other neighboring Digital PSoC blocks. As shown in the table, Digital PSoC block 0-3 can only interface to Global I/Os 0-3, and Digital PSoC block 4-7 can only interface to Global I/Os 4-7.

8.2.2.3 Digital Basic Type A/ Communications Type A Block xx Output Register

A Digital PSoC block may have 0, 1, or 2 outputs depending on its function. Each Digital PSoC block's output can be selected to drive an associated Global signal line via the Output Select bits. In addition, the output drive can be selectively disabled in this register.

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Reserved	Reserved	AUX Out Enable	AUX IO Sel [1]	AUX IO Sel [0]	Out Enable	Out Sel [1]	Out Sel [0]

Bit 7: **Reserved**

Bit 6: **Reserved**

Bit 5: **AUX Out Enable**
0 = Disable Auxiliary Output
1 = Enable Auxiliary Output (function dependant)

Bit [4:3]: **AUX IO Sel [1:0]**
0 0 = Input from Global Input[0] or Drive Global Output[0] (for Digital Blocks 00 to 03) **or** Input from Global Input[4] or Drive Global Output [4] (for Digital Blocks 04 to 07)
0 1 = Input from Global Input[1] or Drive Global Output[1] (for Digital Blocks 00 to 03) **or** Input from Global Input[5] or Drive Global Output[5] (for Digital Blocks 04 to 07)
1 0 = Input from Global Input[2] or Drive Global Output[2] (for Digital Blocks 00 to 03) **or** Input from Global Input[6] or Drive Global Output[6] (for Digital Blocks 04 to 07)
1 1 = Input from Global Input[3] or Drive Global Output[3] (for Digital Blocks 00 to 03) **or** Input from Global Input[7] or Drive Global Output[7] (for Digital Blocks 04 to 07)

Bit 2: **Out Enable**
0 = Disable Primary Output
1 = Enable Primary Output (function dependant)

Bit [1:0]: **Out Sel [1:0]**
0 0 = Drive Global Output[0] (for Digital Blocks 00 to 03) **or** Drive Global Output[4] (for Digital Blocks 04 to 07)
0 1 = Drive Global Output[1] (for Digital Blocks 00 to 03) **or** Drive Global Output[5] (for Digital Blocks 04 to 07)
1 0 = Drive Global Output[2] (for Digital Blocks 00 to 03) **or** Drive Global Output[6] (for Digital Blocks 04 to 07)
1 1 = Drive Global Output[3] (for Digital Blocks 00 to 03) **or** Drive Global Output[7] (for Digital Blocks 04 to 07)

Digital Basic Type A Block 00 Output Register	(DBA00OU, Address = Bank 1, 22h)
Digital Basic Type A Block 01 Output Register	(DBA01OU, Address = Bank 1, 26h)
Digital Communications Type A Block 02 Output Register	(DCA02OU, Address = Bank 1, 2Ah)
Digital Communications Type A Block 03 Output Register	(DCA03OU, Address = Bank 1, 2Eh)
Digital Basic Type A Block 04 Output Register	(DBA04OU, Address = Bank 1, 32h)
Digital Basic Type A Block 05 Output Register	(DBA05OU, Address = Bank 1, 36h)
Digital Communications Type A Block 06 Output Register	(DCA06OU, Address = Bank 1, 3Ah)
Digital Communications Type A Block 07 Output Register	(DCA07OU, Address = Bank 1, 3Eh)

8.2.3 User Registers

There are four user registers within each Digital PSoC block, three data registers, and one status/control register. The three data registers are Data 0, which is a shifter/counter, and Data 1 and Data 2 registers, which contain data used during the operation. The status/control register contains an enable bit that is used for all configurations. In addition, it contains function-specific status and control, which is outlined below.

8.2.3.1 Digital Basic Type A/Communications Type A Block xx Data Register 0,1,2

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	VF*	VF*	VF*	VF*	VF*	VF*	VF*	VF*
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]**

Digital Basic Type A Block 00 Data Register 0 (DBA00DR0, Address = Bank 0, 20h)
Digital Basic Type A Block 00 Data Register 1 (DBA00DR1, Address = Bank 0, 21h)
Digital Basic Type A Block 00 Data Register 2 (DBA00DR2, Address = Bank 0, 22h)
Digital Basic Type A Block 01 Data Register 0 (DBA01DR0, Address = Bank 0, 24h)
Digital Basic Type A Block 01 Data Register 1 (DBA01DR1, Address = Bank 0, 25h)
Digital Basic Type A Block 01 Data Register 2 (DBA01DR2, Address = Bank 0, 26h)
Digital Communications Type A Block 02 Data Register 0 (DCA02DR0, Address = Bank 0, 28h)
Digital Communications Type A Block 02 Data Register 1 (DCA02DR1, Address = Bank 0, 29h)
Digital Communications Type A Block 02 Data Register 2 (DCA02DR2, Address = Bank 0, 2Ah)
Digital Communications Type A Block 03 Data Register 0 (DCA03DR0, Address = Bank 0, 2Ch)
Digital Communications Type A Block 03 Data Register 1 (DCA03DR1, Address = Bank 0, 2Dh)
Digital Communications Type A Block 03 Data Register 2 (DCA03DR2, Address = Bank 0, 2Eh)
Digital Basic Type A Block 04 Data Register 0 (DBA04DR0, Address = Bank 0, 30h)
Digital Basic Type A Block 04 Data Register 1 (DBA04DR1, Address = Bank 0, 31h)
Digital Basic Type A Block 04 Data Register 2 (DBA04DR2, Address = Bank 0, 32h)
Digital Basic Type A Block 05 Data Register 0 (DBA05DR0, Address = Bank 0, 34h)
Digital Basic Type A Block 05 Data Register 1 (DBA05DR1, Address = Bank 0, 35h)
Digital Basic Type A Block 05 Data Register 2 (DBA05DR2, Address = Bank 0, 36h)
Digital Communications Type A Block 06 Data Register 0 (DCA06DR0, Address = Bank 0, 38h)
Digital Communications Type A Block 06 Data Register 1 (DCA06DR1, Address = Bank 0, 39h)
Digital Communications Type A Block 06 Data Register 2 (DCA06DR2, Address = Bank 0, 3Ah)
Digital Communications Type A Block 07 Data Register 0 (DCA07DR0, Address = Bank 0, 3Ch)
Digital Communications Type A Block 07 Data Register 1 (DCA07DR1, Address = Bank 0, 3Dh)
Digital Communications Type A Block 07 Data Register 2 (DCA07DR2, Address = Bank 0, 3Eh)

*** = Varies by function/ User Module selection (See table ahead)**

Function	DR0	R/W	DR1	R/W	DR2	R/W
Timer	Count	R*	Period Value	W	Capture Value	RW
Counter	Count	R*	Period Value	W	Compare Value	RW
CRC	Current Value/CRC Residue	R*	Polynomial Mask Value	W	Seed Value	RW
PRS	Current Value	R*	Polynomial Mask Value	W	Seed Value	RW
Dead Band	Count	R*	Period Value	W	Not Used	RW
UART	Shifter	NA	TX Data Register	W	RX Data Register	R
SPI	Shifter	NA	TX Data Register	W	RX Data Register	R

R* = Each time the register is read, its value is written to the DR2 register

8.2.3.2 Digital Basic Type A/Communications Type A Block xx Control Register 0

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	VF*	VF*	VF*	VF*	VF*	VF*	VF*	VF*
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]**

Digital Basic Type A Block 00 Control Register 0 (DBA00CR0, Address = Bank 0, 23h)* VF = Varies by function
Digital Basic Type A Block 01 Control Register 0 (DBA01CR0, Address = Bank 0, 27h)
Digital Communications Type A Block 02 Control Register 0 (DCA02CR0, Address = Bank 0, 2Bh)
Digital Communications Type A Block 03 Control Register 0 (DCA03CR0, Address = Bank 0, 2Fh)
Digital Basic Type A 04 Control Register 0 (DBA04CR0, Address = Bank 0, 33h)
Digital Basic Type A 05 Control Register 0 (DBA05CR0, Address = Bank 0, 37h)
Digital Communications Type A Block 06 Control Register 0 (DCA06CR0, Address = Bank 0, 3Bh)
Digital Communications Type A Block 07 Control Register 0 (DCA07CR0, Address = Bank 0, 3Fh)

8.2.3.2.1 Digital Basic Type A/Communications Type A Block xx Control Register 0 When Used as Timer, Counter, CRC, and Dead Band

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

Bit 7: **Reserved**

Bit 6: **Reserved**

Bit 5: **Reserved**

Bit 4: **Reserved**

Bit 3: **Reserved**

Bit 2: **Reserved**

Bit 1: **Reserved**

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

8.2.3.2.2 Digital Basic Type A/Communications Type A Block xx Control Register 0 When Used as UART Transmitter

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	--	--	R	R	--	RW	RW	RW
Bit Name	Reserved	Reserved	TX Complete	TX Reg Empty	Reserved	Parity Type	Parity Enable	Enable

Bit 7: **Reserved**

Bit 6: **Reserved**

Bit 5: **TX Complete**
0 = Indicates that if a transmission has been initiated, it is still in progress
1 = Indicates that the current transmission is complete (including framing bits)

Bit 4: **TX Reg Empty**
0 = Indicates the TX Data register is not available to accept another byte
1 = Indicates the TX Data register is available to accept another byte (interrupt)
Note that the interrupt does not occur until at least 1 byte has been previously written to the TX Reg Empty

Bit 3: **Reserved**

Bit 2: **Parity Type**
0 = Even
1 = Odd

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

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

8.2.3.2.3 Digital Basic Type A/Communications Type A Block xx Control Register 0 When Used as UART Receiver

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	RW	RW	RW
Bit Name	Parity Error	Overrun	Framing Error	RX Active	RX Reg Full	Parity Type	Parity Enable	Enable

Bit 7: **Parity Error**
0 = Indicates no parity error detected in the last byte received
1 = Indicates a parity error detected in the last byte received (reset when the register is read)

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

Bit 5: **Framing Error**
0 = Indicates correct stop bit
1 = Indicates a missing STOP bit (reset when the register is read)

Bit 4: **RX Active**
0 = Indicates no communication currently in progress
1 = Indicates a start bit has been received and a byte is currently being received (reset when the register is read)

Bit 3: **RX Reg Full**
0 = Indicates the RX Data register is empty
1 = Indicates a byte has been loaded into the RX Data register (interrupt)

Bit 2: **Parity Type**
0 = Even
1 = Odd

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

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

8.2.3.2.4 Digital Basic Type A/Communications Type A Block xx Control Register 0 When Used as SPI Transceiver

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

Bit 7: **Reserved**

Bit 6: **Reserved**

Bit 5: **SPI Complete**
0 = Indicates the byte is in process of shifting out
1 = Indicates the completion of the byte has been shifted out (reset when the register is read)

Bit 4: **TX Reg Empty**
0 = Indicates the TX data register is not available to accept another byte
1 = Indicates the TX data register is available to accept another byte

Bit 3: **LSB First**
0 = LSB First
1 = MSB First

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

Bit 1: **Clock Polarity**
0 = Non-inverted
1 = Inverted

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

8.3 Global Inputs and Outputs

Global Inputs and Outputs provide additional capability to route clock and data signals to the Digital PSoC blocks. Digital PSoC blocks are connected to the global input and output lines by configuring the PSoC block Input and Output registers (DBA00IN-DBA07IN, DBA00OU-DBA07OU). These global input and output lines form an 8-bit global input bus and an 8-bit global output bus. Four Digital PSoC blocks have access to the upper half of these buses, while the other four access the lower half, per the configuration register. These global input/output buses may be connected to the I/O pins on a per-pin basis using the pin configuration registers. This allows Digital PSoC blocks to route their inputs and outputs to pins using the global I/O buses.

8.3.1 Input Assignments

PSoC block I/O Select Bits[3:0] define the use of global inputs to Digital PSoC blocks. Digital PSoC blocks may also use any of the CPU clocks as inputs or the output of the previous Digital PSoC blocks, when chaining PSoC blocks for more complex functions. The PSoC block Global Configuration Register along with the Input Assignment Register control input assignments.

Input Assignments

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

8.3.2 Output Assignments

PSoC block I/O Select Bits[3:0] define the use of global outputs from Digital PSoC blocks. Digital PSoC blocks may output data to the next Digital PSoC blocks, when chaining PSoC blocks for more complex functions. The PSoC block Global Configuration Register along with the Output Assignments Register controls output assignments.

Output Assignments

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

8.4 Potential Digital User Modules

8.4.1 Timer with Optional Capture

8.4.1.1 Summary

The timer function continuously measures the amount of time in “ticks” between two events, and provides a rate generator. A down counter lies at the heart of the timer functions. Rate generators divide their clock source by an integer value. Hardware or software generated events trigger capture operations that permit calculation of elapsed “ticks.” Timer-configured PSoC blocks may be chained to arbitrary lengths in 8 bit increments.

8.4.1.2 Registers

Data Register 1 establishes the period or integer clock division value. Data Register 0 holds the current state of the down counter. It is automatically loaded when the function is enabled and on the input clock cycle after it reaches zero, the terminal count value. When a capture event occurs, the current value of Data Register 0 is transferred to Data Register 2. A software capture event is generated by reading Data Register 0.

8.4.1.3 Inputs

There are two inputs, the Source Clock and the Hardware Capture signal. The down counter is decremented on the rising-edge of the Source Clock. A hardware capture event is signaled by a rising edge of the Hardware Capture signal. This is synchronized to the 24.5MHz system clock and the data is transferred to Data Register 2 on the falling edge of the System Clock. The multiplexers selecting these input sources are controlled by the PSoC block Input Register (DBA00IN-DBA07IN).

8.4.1.4 Outputs

The Terminal Count signal exhibits a duty cycle that is the reciprocal of the period value contained in Data Register 1. In other words, it is high during the source clock cycle when the value in Data Register 0 is zero and low otherwise. The Terminal Count can be routed to additional analog or digital PSoC blocks or via Global Input or Global Output lines. Output options are controlled by the PSoC block Output Register (DBA00OU-DBA07OU).

8.4.1.5 Interrupts

Interrupts may be generated in either of two ways. First, the PSoC block may optionally generate an interrupt on the rising edge of Terminal Count or the rising edge of a compare signal. When so configured, the comparison operator conditions the output based on the values in Data Register 2 and Data Register 0. The selection of interrupt source is determined by the MODE[0] bit of the PSoC block Function Register (DBA00FN-DBA07FN). The MODE[1] bit controls whether the comparison operation is “less than” or “less than or equal to.” If capture events are disabled, Data Register 2 can be used to create a periodic interrupt with a particular offset from the terminal count.

8.4.2 Counter with Optional Compare (Pulse-Width) Output

8.4.2.1 Summary

Conceptually, a counter measures the number of events between “ticks,” however, this distinction between counter and timer blurs because both functions provide a complete range of clock selections. The counter trades the timer’s hardware capture for a clock gate or ”enable” and provides a means of adjusting the duty cycle of its output so that it can double as a pulse-width modulator. A down counter lies at the heart of the counter function. Counter-configured PSoC blocks may be chained to arbitrary lengths in 8 bit increments.

8.4.2.2 Registers

Data Register 0 holds the current state of the down counter and automatically loads a new value from Data Register 1 after it reaches zero, the terminal count value. In other words, Data Register 1 establishes the maximum count value or period of the counter. Reading Data Register 0 to obtain the current value of the down counter should occur only when the function is disabled. The value in Data Register 2 is compared to Data Register 1 to establish the output pulse-width (duty cycle).

8.4.2.3 Inputs

There are two primary inputs, the Source Clock and the Enable signal. When the Enable signal is high, the down counter is decremented on the rising-edge of the Source Clock. The multiplexers selecting these inputs are controlled by the PSoC block Input Register (DBA00IN-DBA07IN).

8.4.2.4 Outputs

The counter function drives its output signal, Compare, high on the falling edge of the Source Clock when the value in Data Register 0 is less (or less than or equal to) the value in Data Register 2. The duty cycle of the pulse-width modulator formed in this way is the ratio of Data Register 2 (or Data Register 2 minus one) to Data Register 1. The choice of compare operators is determined by the MODE[1] bit. Output options are controlled by the PSoC block Output Register (DBA00OU-DBA07OU).

8.4.2.5 Interrupts

The counter generates an interrupt when the MODE[0] bit is set and the compare first becomes true, that is, as the output pulse goes high. The counter function inhibits further interrupts until after Data Register 0 reload occurs.

8.4.3 Dead-Band Generator

8.4.3.1 Summary

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

8.4.3.2 Registers

Data Register 1 stores the count that controls the elapsed dead time. Data Register 0 holds the current state of the dead-time down counter. The Dead-band function loads the initial value of Data Register 0 from Data Register 1 when the function is enabled and again each time it reaches its terminal count. Data Register 2 is unused.

8.4.3.3 Inputs

The “data” input controls the period and duty cycle of the dead-band generator outputs. If this signal is pulse-width modulated, the dead-band outputs will be similarly modulated. The F0 output corresponds to the duty cycle of the input (less the dead time) and F1 to the duty cycle of the inverted input (again, less the dead time). The clock input to the dead-band generator controls the rate at which the down counter is decremented. The multiplexers selecting these inputs are controlled by the PSoC block Input Register (DBA00IN-DBA07IN).

8.4.3.4 Outputs

Both the F0 and F1 outputs can be driven onto the global output bus. If the next PSoC block selects “Previous PSoC block” for its input, it only “sees” the F0 output of the dead-band function.

8.4.3.5 Interrupts

No interrupts are generated by the dead-band function.

8.4.3.6 Constraints

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

8.4.4 PRS - Pseudo-Random Sequence Generator

8.4.4.1 Summary

The PRS function generates an output waveform corresponding to a sequence of pseudo-random numbers. A linear-feedback shift register generates the sequence according to a user-specified polynomial. The width of the numbers in the sequence is variable and the initial value is determined by a user-defined “seed” value. PRS PSoC blocks can be chained to increase the width of the numbers and, hence, the length of the sequence. A chain of N PSoC blocks can generate numbers from 2- to 8N-bits wide and sequences of up to n^{8N}-1 distinct values.

8.4.4.2 Registers

Data Register 2 holds the “seed” value. Data Register 0 implements a linear-feedback shift register and is loaded with the seed value when the function is enabled. Data Register 1 specifies the polynomial and width of the numbers in the sequence (see “Determining the Polynomial,” below).

8.4.4.3 Inputs

The clock input determines the rate at which the output sequence is produced. The data input is ignored by the PRS function. The multiplexer selecting the clock input is controlled by the PSoC block Input Register (DBA00IN-DBA07IN).

8.4.4.4 Outputs

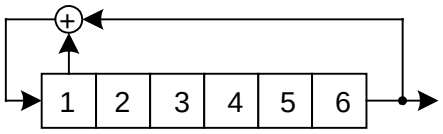
The PRS function drives the output serial data stream synchronous with the input clock. The output bits are valid on the falling edge of the input clock. The output may be driven on the global output bus or to the subsequent digital PSoC block.

8.4.4.5 Interrupts

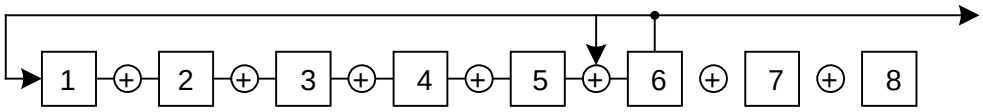
The PRS function does not provide interrupts.

8.4.4.6 Determining the Polynomial

A simple linear-feedback shift register, or LFSR, uses an XOR gate to “add” the values of one or more bits and feed the result back into the least-significant bit. One possible realization of a 6-bit LFSR providing a maximal sequence of 63 six-bit values is shown here:



The PRS function utilizes a different “modular” architecture with one XOR gate between each bit of the shift register. A maximal sequence equivalent to that produced by the previous realization is generated by the following modular LFSR:



Denote the first implementation as a (6, 1) LFSR, where 6 gives the length of the output codes and 1 indicates the tap which feeds the XOR gate along with the final bit. Then the modular form just shown is denoted as a [6, 5] LFSR. In general, the equivalent modular form of a simple N bit LFSR with M taps denoted by (N, t_1, t_2, …, t_M) is given by the notation [N, N-t_1, N-t_2, …, N-t_M]. Once the form (and thus the notation) is determined, the value of Data Register 1 is easily determined. The bit corresponding to the length and all tap bits are turned on; the others are zero. Thus, the polynomial specification for Data Register 1 to implement a [6, 5] LFSR is 00110000b, or 30h. A maximal sequence PRS for 8-bits giving 255 codes is [8, 4, 3, 2] with polynomial 10001110b or 8Eh.

8.4.5 CRC - Cyclic Redundancy Check

8.4.5.1 Summary

The CRC uses a shift register and XOR gates like the PRS function; however, instead of an output bit stream, the CRC function expects an input bit stream. A polynomial specification permits the length of the input sequence over which the cyclic redundancy check computes a result to be varied. CRC-configured PSoC blocks can be chained to form longer results.

REGISTERS: Data Register 0 is the shift register used in the computation. Data Register 1 holds the polynomial specification. Data Register 2 holds the latch and may be read anytime after the input bit stream is complete.

8.4.5.2 Registers

The PSoC block Input Register determines the multiplexer settings for the input serial data stream and the bit clock. The data is assumed to be valid on the rising edge of the bit clock.

8.4.5.3 Outputs

None (see Data Register 2).

8.4.5.4 Interrupts

None.

8.4.5.5 Specifying the Polynomial

Computation of an N-bit result is generally specified by a polynomial with N+1 terms, the last of which is the X^0 term, where X^0=1. For example, the widely used CRC-CCIT 16-bit polynomial is $X^{16}+X^{12}+X^5+1$. The PSoC block CRC function assumes the presence of the X^0 term so that the polynomial for an N-bit result can be expressed by an N-bit rather than N+1 bit specification. To obtain the PSoC block register specification, write an N+1 bit binary number corresponding to the full polynomial, with 1's for each term present. The CRC-CCIT polynomial would be 10001000000100001b. Simply drop the right-most bit (the X^0 term) to obtain the register specification for the PSoC block. To implement the CRC-CCIT example, two PSoC blocks must be chained together. Data Register 1 in the high-order PSoC block would take the value 10001000b (88h) and the corresponding register in the low-order PSoC block would take 00010000b (10h).

8.4.6 Universal Asynchronous Receiver

8.4.6.1 Summary

The Universal Asynchronous Receiver implements the input half of a basic 8-bit UART. Start and Stop bits are recognized and stripped. Parity type and parity validation are configurable features. This function requires a Digital Communications Type PSoC block and cannot be chained for longer data words.

8.4.6.2 Registers

The function shifts incoming data into Data Register 0. Once complete, the byte is transferred to Data Register 2 from which it may be read. Data Register 1 is not used by this function. Control Register 0 (DBA00CR0-DBA07CR0) enables the function, provides the means to configure parity checking, and a full set of status indications. See the register definition for full details.

8.4.6.3 Inputs

The serial data and clock inputs are obtained from the PSoC block input and clock multiplexers controlled by the Input Register (DBA00IN-DBA07IN). The clock signal must run 8 times faster than the input bit rate.

8.4.6.4 Outputs

None (see Data Register 2, above).

8.4.6.5 Interrupts

The function can be configured to generate an interrupt on receive register (Data Register 2) full.

8.4.7 Universal Asynchronous Transmitter

8.4.7.1 Summary

The Universal Asynchronous Transmitter implements the output half of a basic 8-bit UART. Start and Stop bits are generated. Parity bit generation and type are configurable features. This function requires a Digital Communications Type PSoC block. It cannot be chained for longer data words.

8.4.7.2 Registers

When Data Register 0 is empty and a new byte has been written to Data Register 1, the function transfers the byte to Data Register 0 and shifts it out along with a start bit, possibly a parity bit and a stop bit. Data Register 2 is not used by this function. The PSoC block's Control Register 0 (DBA00CR0-DBA07CR0) configures the parity type and enable. It also provides status information to enable detection of transmission complete.

8.4.7.3 Inputs

A baud-rate clock running at 8 times the desired output bit rate is selected by the clock-input multiplexer controlled by the PSoC block Input Register (DBA00IN-DBA07IN). The output of the Data Input multiplexer is ignored by this function.

8.4.7.4 Outputs

The transmitter's serial data output appears at the PSoC block output and may be driven onto one of the global bus lines.

8.4.7.5 Interrupts

If enabled, the function will raise an interrupt when the transmit data register (Data Register 1) is empty. This will happen immediately after the first byte is written to Data Register 1 as it is immediately transferred to Data Register 0.

8.4.8 SPI Master - Serial Peripheral Interface (SPIM)

8.4.8.1 Summary

The SPI Master function provides a full-duplex synchronous data transceiver that also generates a bit clock for the data. This function requires a Digital Communications Type PSoC block. It cannot be chained for longer data words.

8.4.8.2 Registers

Data Register 0 provides a shift register for both incoming and outgoing data. Data Register 1 is the transmit register. When data is written to Data Register 1, it is transferred to Data Register 0. New data bits are shifted in as the transmit bits are shifted out. After the 8 bits are transmitted and received by Data Register 0, its content is transferred to Data Register 2 from which it may be read. The function's Control Register 0 (DBA00CR0-DBA07CR0) provides status information and configures the function for one of the four standard modes, which configure the interface based on clock polarity and phase with respect to data.

8.4.8.3 Inputs

MISO (master-in, slave-out) is selected by the input multiplexer. The clock input multiplexer selects a clock that runs at twice the desired data rate. The SPIM function divides the input clock by 2 to obtain the 50% duty-cycle required for proper timing.

8.4.8.4 Outputs

There are two outputs, both of which can be enabled onto the global output bus. The MOSI (master-out, slave-in) data line provides the output serial data. The second output is the bit-clock derived by dividing the input clock by 2 to ensure a 50% duty-cycle.

Note: The SPIM function does not provide the SS signal that may be used by a corresponding SPI Slave. However, this can be implemented with a GPIO pin if desired.

8.4.8.5 Interrupts

When enabled, the function raises the PSoC block interrupt on transmit register (Data Register 1) empty.

8.4.9 SPI Slave - Serial Peripheral Interface (SPIS)

8.4.9.1 Summary

The SPI Slave function provides a full-duplex bi-directional synchronous data transceiver that requires an externally provided bit clock for the data. This function requires a Digital Communications Type PSoC block. It cannot be chained for longer data words.

8.4.9.2 Registers

Data Register 0 provides a shift register for both incoming and outgoing data. Output data is written to Data Register 1. When Data Register 0 is empty, its value is updated from Data Register 2. As new data bits are shifted in, the transmit bits are shifted out. After the 8 bits are transmitted and received by Data Register 0, its content is transferred to Data Register 2 from which it may be read and Data Register 0 is marked “empty.” Control Register 0 (DBA00CR0-DBA07CR0) provides status information and configures the function for one of the four standard modes, which configure the interface based on clock polarity and phase with respect to data.

8.4.9.3 Inputs

The SPIS function is a bit unusual in that there are three inputs. The Input Register (DBA00IN-DBA07IN) controls the input multiplexer, which selects the MOSI data stream. It also controls the clock selection multiplexer from which the function obtains the master’s bit clock. The AUX-IO bits of the Output Register (DBA00OU-DBA07OU) select the global input line from which the SS signal is obtained.

Important: The AUX Out Enable bit (bit 5) of the Output Register (DBA00OU-DBA07OU) must be set to 0 to disable it.

8.4.9.4 Outputs

The function output is the MISO (master-in, slave-out) signal, which may be driven on the global output bus.

8.4.9.5 Interrupts

When enabled, the function raises the PSoC block interrupt on transmit register (Data Register 1) empty.

8.5 Analog PSoC Blocks

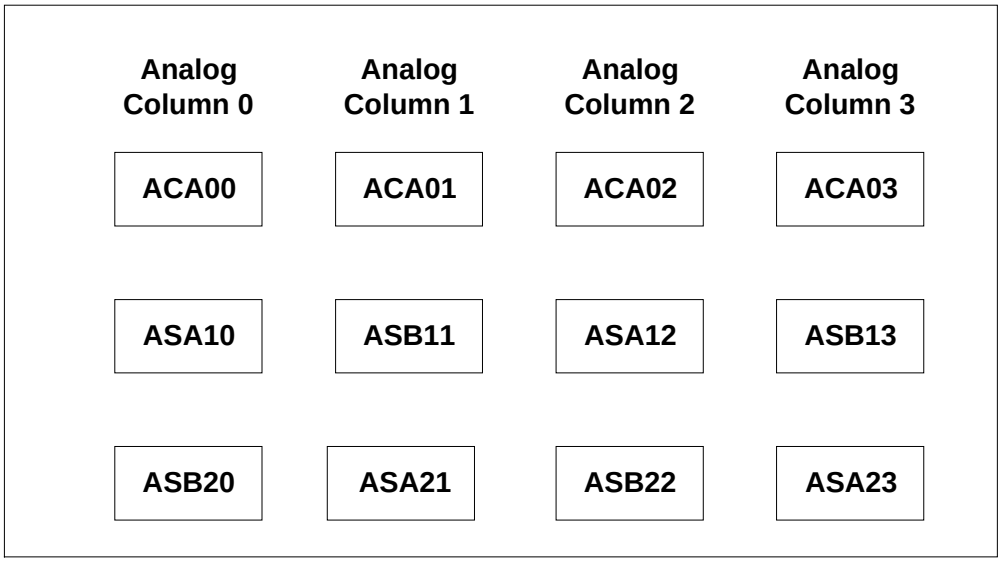
8.5.1 Introduction

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 / accuracy, and provides simple interface to most real-world analog inputs and outputs.
- Support is provided for sensor interfaces, audio codecs, embedded modems, and general-purpose op amp circuits.
- Flexible, System on-a-Chip programmability, providing variations in functions.
- For a given function, easily selected tradeoffs 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 programs gain 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.

8.4.10 Array of Analog PSoC Blocks

There are twelve analog PSoC blocks implemented for each of the following types; Analog Continuous Time Type A (ACAxx), Analog Switch Cap Type A (ASAxx), and Analog Switch Cap Type B (ASBxx). These blocks are arranged in an array of three rows by four columns. Each column has one of each type of PSoC block, and the individual PSoC blocks are identified by the row and column in which they reside.

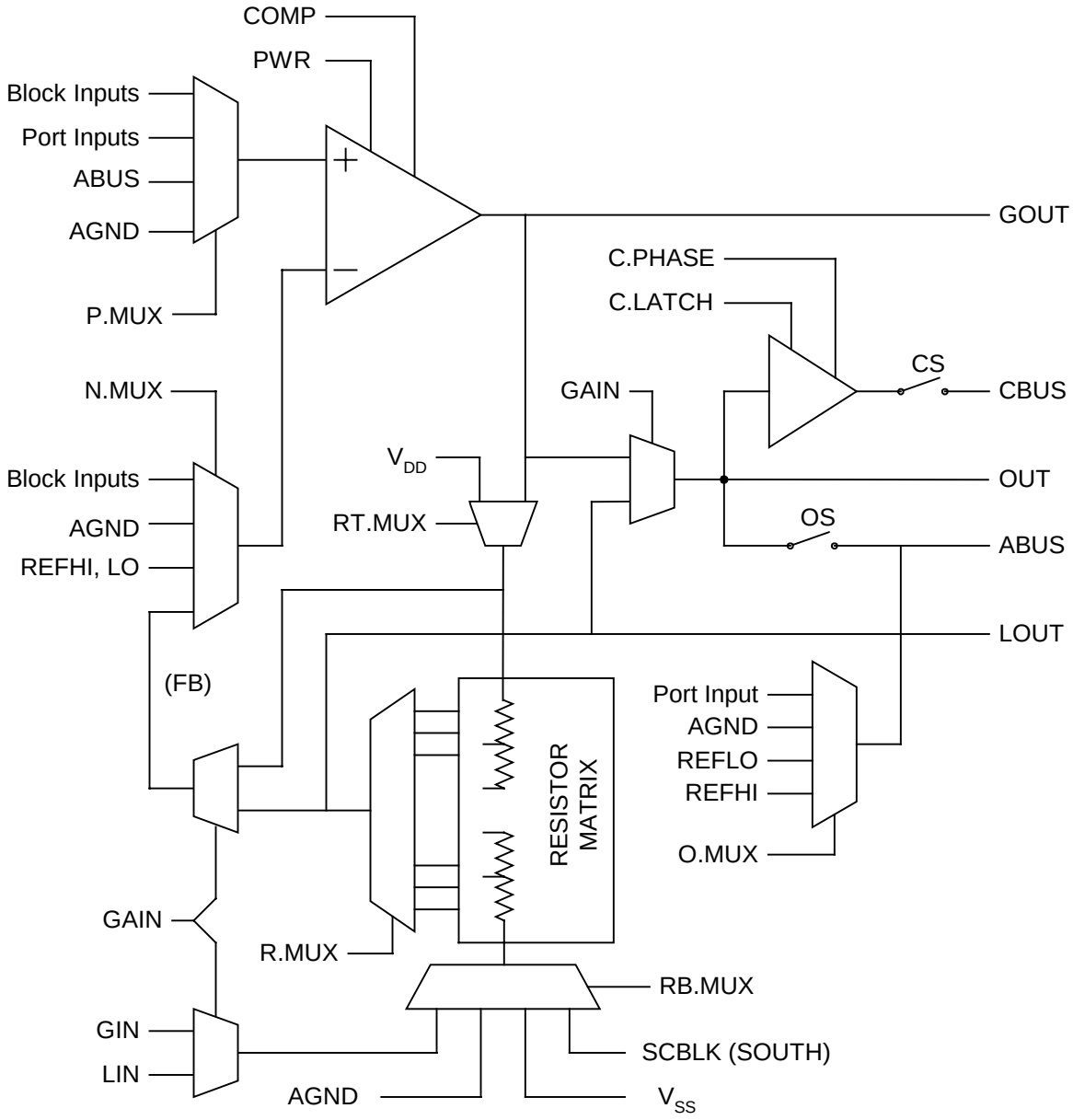


There are two primary types of analog PSoC blocks. Both types contain one op-amp and one comparator but their principles of operation are quite different. Continuous-time PSoC blocks employ three configuration registers and use resistors to condition amplifier response. Switched capacitor blocks have four configuration registers operate as discrete-time sampling operators. In both types, the configuration registers are divided into distinct bit fields. Some bit fields set the PSoC block's resistor ratios or capacitor values. Others configure switches and multiplexers that form connections between internal block nodes. Additionally, a block may be connected via local interconnection resources to neighboring analog PSoC blocks, reference voltage sources, input multiplexers and output busses. Specific advantages and applications of each type are treated separately below.

8.5.2 Analog Continuous Time PSoC Blocks

8.5.2.1 Introduction

Supports Programmable Gain or attenuation Op Amp Circuits, (Differential Gain) Instrumentation Amplifiers (using two CT Blocks), Continuous time high frequency antialiassing filters, and modest response-time analog comparators.

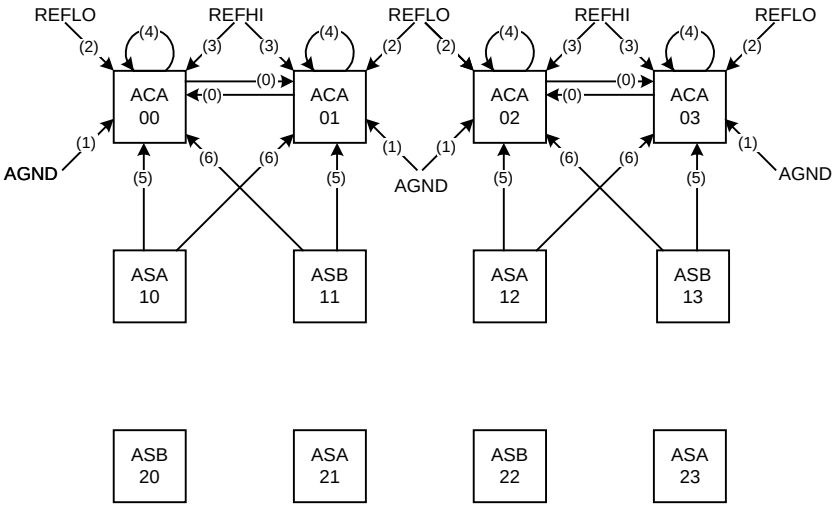


8.5.2.2 Local Interconnect

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

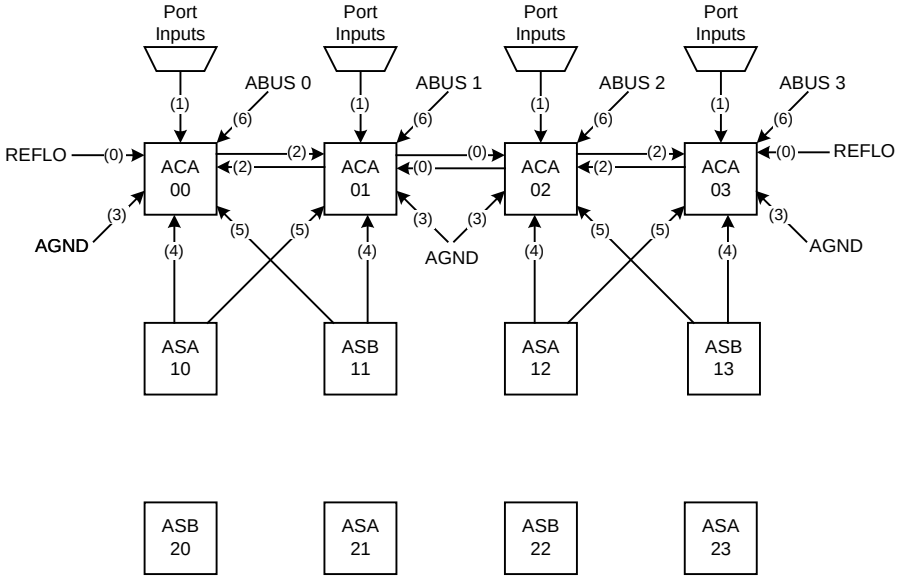
N.MUX

N (Inverting) Input Multiplexer Connections



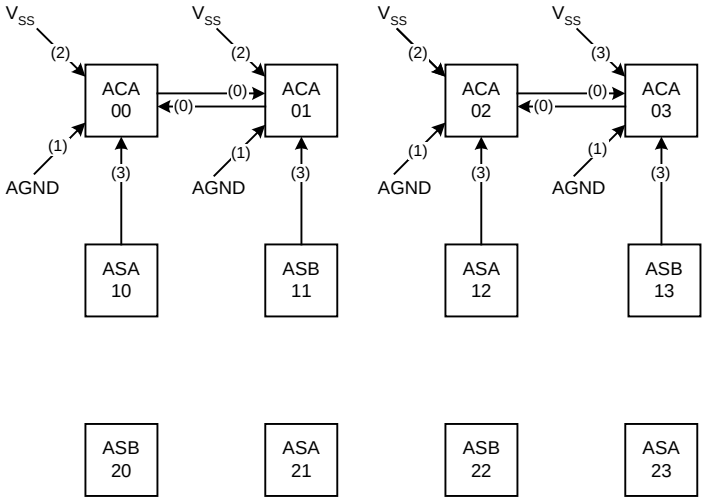
P.MUX

P (Non-inverting) Input Multiplexer Connections



RB.MUX

RB Input Multiplexer Connections



8.5.2.3 Registers

8.5.2.3.1 Analog Continuous Time Block xx Control 0 Register

The RT.MUX bits control the connection of the two ends of the resistor string. The RT.MUX bits control the top end of the resistor string, which can either be connected to Vcc or to the op-amp output. The RB.MUX bits control the connection of the bottom end of the resistor string.

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

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

The GAIN bit controls whether the resistor string is connected around the op-amp as for gain (center tap to inverting op-amp input) or for loss (center tap to output of the block). Note that setting GAIN alone does not guarantee that you will have a gain or loss block. Routing of the other ends of the resistor determine this.

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	R.MUX3	R.MUX2	R.MUX1	R.MUX0	GAIN	RT.MUX	RB.MUX1	RB.MUX0

Bit [7:4]: **R.MUX [3:0]** Encoding for selecting 1 of 16 resistor taps
0 0 0 0 = Rf 15 = Ri 01 = Loss .0625 / Gain 16.00
0 0 0 1 = Rf 14 = Ri 02 = Loss .1250 / Gain 8.000
0 0 1 0 = Rf 13 = Ri 03 = Loss .1875 / Gain 5.333
0 0 1 1 = Rf 12 = Ri 04 = Loss .2500 / Gain 4.000
0 1 0 0 = Rf 11 = Ri 05 = Loss .3125 / Gain 3.200
0 1 0 1 = Rf 10 = Ri 06 = Loss .3750 / Gain 2.667
0 1 1 0 = Rf 09 = Ri 07 = Loss .4375 / Gain 2.286
0 1 1 1 = Rf 08 = Ri 08 = Loss .5000 / Gain 2.000
1 0 0 0 = Rf 07 = Ri 09 = Loss .5625 / Gain 1.778
1 0 0 1 = Rf 06 = Ri 10 = Loss .6250 / Gain 1.600
1 0 1 0 = Rf 05 = Ri 11 = Loss .6875 / Gain 1.455
1 0 1 1 = Rf 04 = Ri 12 = Loss .7500 / Gain 1.333
1 1 0 0 = Rf 03 = Ri 13 = Loss .8125 / Gain 1.231
1 1 0 1 = Rf 02 = Ri 14 = Loss .8750 / Gain 1.143
1 1 1 0 = Rf 01 = Ri 15 = Loss .9375 / Gain 1.067
1 1 1 1 = Rf 00 = Ri 16 = Loss 1.000 / Gain 1.000

Bit 3: **GAIN** Select gain or loss configuration for output tap
0 = Loss
1 = Gain

Bit 2: **RT.MUX [2]** Encoding for feedback resistor select
0 = Rtop to V_{CC}
1 = Rtop to Amp's Output

Bit [1:0]: **RB.MUX [1:0]** Encoding for feedback resistor select

	ACA00	ACA01	ACA02	ACA03
0 0 =	ACA01	ACA00	ACA03	ACA02
0 1 =	AGND	AGND	AGND	AGND
1 0 =	V_{SS}	V_{SS}	V_{SS}	V_{SS}
1 1 =	ASA10	ASB11	ASA12	ASB13

Analog Continuous Time Block 00 Control 0 Register (ACA00CR0, Address = Bank 0/1, 71h)
Analog Continuous Time Block 01 Control 0 Register (ACA01CR0, Address = Bank 0/1, 75h)
Analog Continuous Time Block 02 Control 0 Register (ACA02CR0, Address = Bank 0/1, 79h)
Analog Continuous Time Block 03 Control 0 Register (ACA03CR0, Address = Bank 0/1, 7Dh)

8.4.10.1.1 Analog Continuous Time Block xx Control 1 Register

The P.MUX bits control the multiplexing of inputs to the non-inverting input of the op-amp. There are physically only 7 inputs. The 8th code (111) will leave the input floating. This is not desireable, and should be avoided.

The N.MUX bits control the multiplexing of inputs to the inverting input of the op-amp. There are physically only 7 inputs. The 8th code (111) will leave the input floating. This is not desireable, and should be avoided.

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

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

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	OS	CS	N.MUX2	N.MUX1	N.MUX0	P.MUX2	P.MUX1	P.MUX0

Bit 7: **OS** Enable output to the analog bus (if high and PS=0 S/H clock gates to ABUS, if high and PS=1 the always on)
0 = Disable analog bus driven by this block
1 = Enable analog bus driven by this block

Bit 6: **CS** Enable output to the comparator bus
0 = Disable comaprator bus driven by this block
1 = Enable comparator bus driven by this block

Bit [5:3]: **N.MUX [2:0]** Encoding for negative input select

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

Bit [2:0]: **P.MUX [2:0]** Encoding for positive input select

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

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

8.5.2.3.2 Analog Continuous Time Type A Block xx Control 2 Register

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

O.MUX – selects block bypass mode for testing and characterization purposes.

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

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

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

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	C.PHASE	C.LATCH	COMP	O.MUX2	O.MUX1	O.MUX0	PWR1	PWR0

Bit 7: **C.PHASE**
0 = Comparator Control latch transparent on PHI1
1 = Comparator Control latch transparent on PHI2

Bit 6: **C.LATCH**
0 = Comparator Control latch is always transparent
1 = Comparator Control latch is active

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

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

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

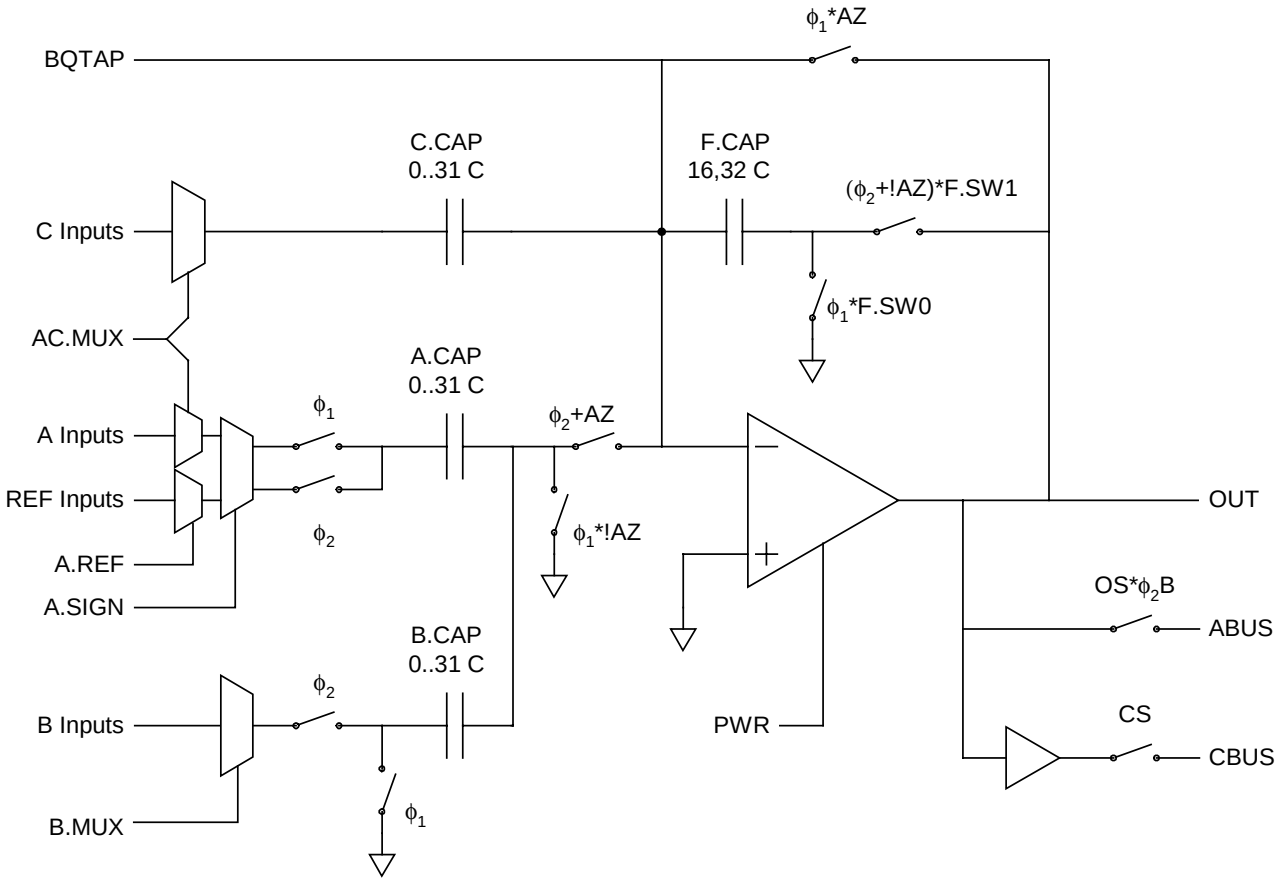
Bit [1:0]: **PWR [1:0]** Encoding for selecting 1 of 4 power levels
0 0 = Off
0 1 = Low (10 µA)
1 0 = Med (50 µA)
1 1 = High (200 µA)

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

8.5.3 Analog Switch Cap Type A PSoC Blocks

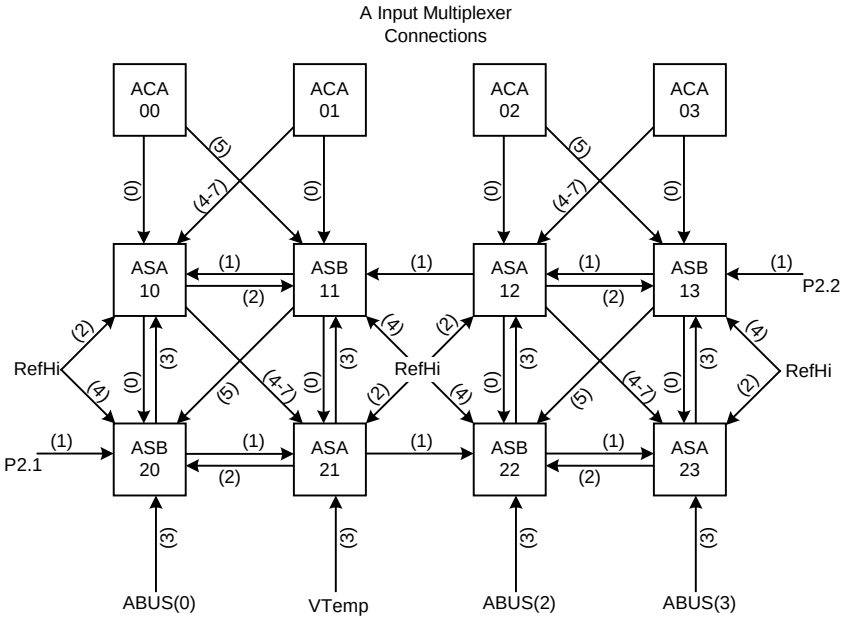
8.5.3.1 Introduction

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

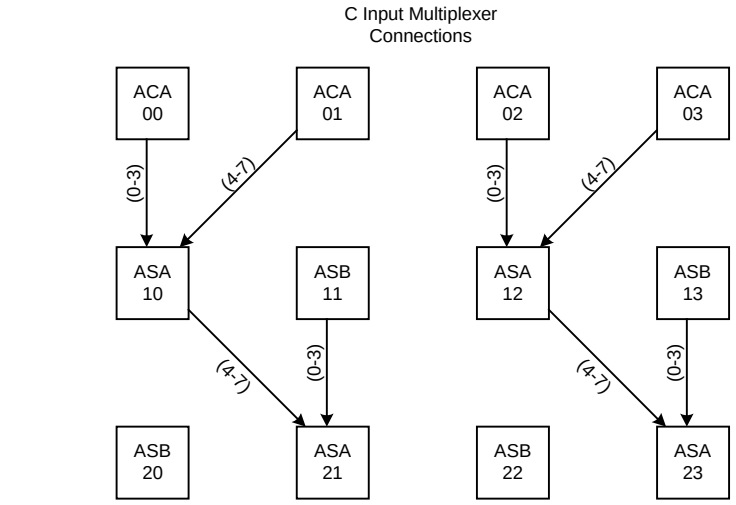


8.5.3.2 Local Interconnect

A.MUX



C.MUX

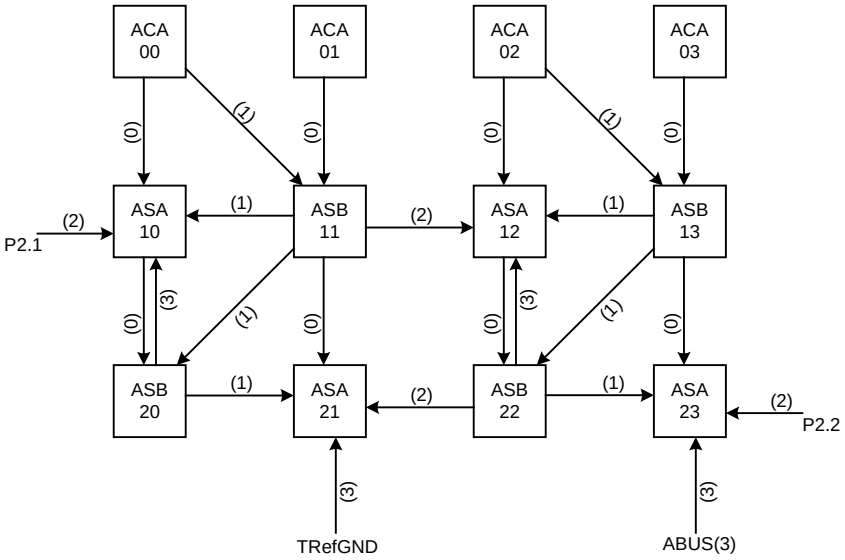


AC.MUX

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

B.MUX

B Input Multiplexer Connections



8.5.3.3 Registers

8.5.3.3.1 Analog Switch Cap Type A Block xx Control 0 Register

F.CAP controls the size of the switched feedback capacitor in the integrator.

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

A.SIGN controls the switch phasing of the switches on the bottom plate of the A.CAP capacitor. The bottom plate samples the input or the reference.

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

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	F.CAP	PHASE	A.SIGN	A.CAP4	A.CAP3	A.CAP2	A.CAP1	A.CAP0

Bit 7: **F.CAP** Binary encoding for capacitor size
0 = 16 capacitor units
1 = 32 capacitor units

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

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

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

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

Bit [4:0]: **A.CAP [4:0]** Binary encoding for 32 possible capacitor sizes
0 0 0 0 0 = 0 Capacitor units in array
0 0 0 0 1 = 1 Capacitor units in array
0 0 0 1 0 = 2 Capacitor units in array
0 0 0 1 1 = 3 Capacitor units in array
0 0 1 0 0 = 4 Capacitor units in array
0 0 1 0 1 = 5 Capacitor units in array
0 0 1 1 0 = 6 Capacitor units in array
0 0 1 1 1 = 7 Capacitor units in array
0 1 0 0 0 = 8 Capacitor units in array
0 1 0 0 1 = 9 Capacitor units in array
0 1 0 1 0 = 10 Capacitor units in array
0 1 0 1 1 = 11 Capacitor units in array
0 1 1 0 0 = 12 Capacitor units in array
0 1 1 0 1 = 13 Capacitor units in array
0 1 1 1 0 = 14 Capacitor units in array
0 1 1 1 1 = 15 Capacitor units in array
1 0 0 0 0 = 16 Capacitor units in array
1 0 0 0 1 = 17 Capacitor units in array
1 0 0 1 0 = 18 Capacitor units in array
1 0 0 1 1 = 19 Capacitor units in array
1 0 1 0 0 = 20 Capacitor units in array
1 0 1 0 1 = 21 Capacitor units in array
1 0 1 1 0 = 22 Capacitor units in array
1 0 1 1 1 = 23 Capacitor units in array
1 1 0 0 0 = 24 Capacitor units in array
1 1 0 0 1 = 25 Capacitor units in array
1 1 0 1 0 = 26 Capacitor units in array
1 1 0 1 1 = 27 Capacitor units in array
1 1 1 0 0 = 28 Capacitor units in array
1 1 1 0 1 = 29 Capacitor units in array
1 1 1 1 0 = 30 Capacitor units in array
1 1 1 1 1 = 31 Capacitor units in array

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

8.5.3.3.2 Analog Switch Cap Type A Block xx Control 1 Register

AC.MUX controlls the input muxing for both the A and C capacitor branches. The high order bit, AC.MUX[2], selects one of two inputs for the C branch. However, when the bit is high, it also overrides the two low order bits, forcing the A and C branches to the same source. The resulting condition is used to construct lowpass biquad filters.

The B.CAP bits set the value of the capacitor in the B path.

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	AC.MUX2	AC.MUX1	AC.MUX0	B.CAP4	B.CAP3	B.CAP2	B.CAP1	B.CAP0

Bit [7:5] **AC.MUX [2:0]** Encoding for selecting A and C inputs

	ASA10		ASA21		ASA12		ASA23	
	A Inputs	C Inputs	A Inputs	C Inputs	A Inputs	C Inputs	A Inputs	C Inputs
0 0 0 =	ACA00	ACA00	ASB11	ASB11	ACA02	ACA02	ASB13	ASB13
0 0 1 =	ASB11	ACA00	ASB20	ASB11	ASB13	ACA02	ASB22	ASB13
0 1 0 =	REFHI	ACA00	REFHI	ASB11	REFHI	ACA02	REFHI	ASB13
0 1 1 =	ASB20	ACA00	V_{temp}	ASB11	ASB22	ACA02	$A_{out}3$	ASA12
1 0 0 =	ACA01	ACA01	ASA10	ASA10	ACA03	ACA03	ASA12	ASA12
1 0 1 =	ACA01	ACA01	ASA10	ASA10	ACA03	ACA03	ASA12	ASA12
1 1 0 =	ACA01	ACA01	ASA10	ASA10	ACA03	ACA03	ASA12	ASA12
1 1 1 =	ACA01	ACA01	ASA10	ASA10	ACA03	ACA03	ASA12	ASA12

Bit [4:0]: **B.CAP [4:0]** Binary encoding for 32 possible capacitor sizes

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

Analog Switch Cap Type A Block 10 Control 1 Register (ASA10CR1, Address = Bank 0/1, 81h)
Analog Switch Cap Type A Block 12 Control 1 Register (ASA12CR1, Address = Bank 0/1, 89h)
Analog Switch Cap Type A Block 21 Control 1 Register (ASA21CR1, Address = Bank 0/1, 95h)
Analog Switch Cap Type A Block 23 Control 1 Register (ASA23CR1, Address = Bank 0/1, 9Dh)

8.5.3.3.3 Analog Switch Cap Type A Block xx Control 2 Register

OS gates the output to the analog column bus. The output on the analog column bus is affected by the state of the PHASE bit in Control 0 Register (ASA10CR0, ASA12CR0, ASA21CR0, ASA23CR0). If OS is set to 0, the output to the analog column bus is tri-stated. If OS is set to 1, the signal that is output to the analog column bus is selected by the PHASE bit. If the PHASE bit is 0, the block output is gated by sampling clock on last part of PHI2. If the PHASE bit is 1, the block output continuously drives the analog column bus.

CS controls the output to the column comparator bus. Note that logic was added so 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.

AZ controls the shorting of the output to the inverting input of the op-amp. When shorted, the op-amp is basically a follower. The output is the op-amp offset. By using the feedback capacitor of the integrator, the user can memorize the offset and create an offset cancellation scheme. AZ also controls a pair of switches between the A and B branches and the summing node of the op-amp. If AZ is enabled, then the pair of switches is active. AZ also affects the function of the F.SW1 bit in control word 4.

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

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	OS	CS	AZ	C.CAP4	C.CAP3	C.CAP2	C.CAP1	C.CAP0

Bit 7: **OS** Enable output to the analog bus
0 = Disable output to analog column bus
1 = Enable output to analog column bus
(The output on the analog column bus is affected by the state of the PHASE bit in Control 0 Register (ASA10CR0, ASA12CR0, ASA21CR0, ASA23CR0). If OS is set to 0, the output to the analog column bus is tri-stated. If OS is set to 1, the signal that is output to the analog column bus is selected by the PHASE bit. If the PHASE bit is 0, the block output is gated by sampling clock on last part of PHI2. If the PHASE bit is 1, the block output continuously drives the analog column bus.)

Bit 6: **CS** Enable output to the comparator bus
0 = Disable output to comparator bus
1 = Enable output to comparator bus

Bit 5: **AZ** Bit for controlling gated switches
0 = Shorting switch is not active. Input cap branches shorted to op-amp input
1 = Shorting switch is enabled during internal PHI1. Input cap branches shorted to analog ground during internal PHI1 and to op-amp input during internal PHI2.

Bit [4:0]: **C.CAP [4:0]** Binary encoding for 32 possible capacitor sizes
0 0 0 0 0 = 0 Capacitor units in array
0 0 0 0 1 = 1 Capacitor units in array
0 0 0 1 0 = 2 Capacitor units in array
0 0 0 1 1 = 3 Capacitor units in array
0 0 1 0 0 = 4 Capacitor units in array
0 0 1 0 1 = 5 Capacitor units in array
0 0 1 1 0 = 6 Capacitor units in array
0 0 1 1 1 = 7 Capacitor units in array
0 1 0 0 0 = 8 Capacitor units in array
0 1 0 0 1 = 9 Capacitor units in array
0 1 0 1 0 = 10 Capacitor units in array
0 1 0 1 1 = 11 Capacitor units in array
0 1 1 0 0 = 12 Capacitor units in array
0 1 1 0 1 = 13 Capacitor units in array
0 1 1 1 0 = 14 Capacitor units in array
0 1 1 1 1 = 15 Capacitor units in array
1 0 0 0 0 = 16 Capacitor units in array
1 0 0 0 1 = 17 Capacitor units in array
1 0 0 1 0 = 18 Capacitor units in array
1 0 0 1 1 = 19 Capacitor units in array
1 0 1 0 0 = 20 Capacitor units in array
1 0 1 0 1 = 21 Capacitor units in array
1 0 1 1 0 = 22 Capacitor units in array
1 0 1 1 1 = 23 Capacitor units in array
1 1 0 0 0 = 24 Capacitor units in array
1 1 0 0 1 = 25 Capacitor units in array
1 1 0 1 0 = 26 Capacitor units in array
1 1 0 1 1 = 27 Capacitor units in array
1 1 1 0 0 = 28 Capacitor units in array
1 1 1 0 1 = 29 Capacitor units in array
1 1 1 1 0 = 30 Capacitor units in array
1 1 1 1 1 = 31 Capacitor units in array

Analog Switch Cap Type A Block 10 Control 2 Register (ASA10CR2, Address = Bank 0/1, 82h)
Analog Switch Cap Type A Block 12 Control 2 Register (ASA12CR2, Address = Bank 0/1, 8Ah)
Analog Switch Cap Type A Block 21 Control 2 Register (ASA21CR2, Address = Bank 0/1, 96h)
Analog Switch Cap Type A Block 23 Control 2 Register (ASA23CR2, Address = Bank 0/1, 9Eh)

8.5.3.3.4 Analog Switch Cap Type A Block xx Control 3 Register

A.REF selects the reference input of the A capacitor branch.

F.SW1 is used to control a switch between in the integrator capacitor path. It connects the output of the op-amp to the integrating cap. The state of the switch is affected by the state of the AZ bit in Control 2 Register (ASA10CR2, ASA12CR2, ASA21CR2, ASA23CR2). If the F.SW1 bit is set to 0, the switch is always disabled. If the F.SW1 bit is set to 1, the state of the switch is determined by the AZ bit. If the AZ bit is 0, the switch is enabled at all times. If the AZ bit is 1, the switch is enabled only when the internal PHI2 is high.

F.SW0 is used to control a switch in the integrator capacitor path. It connects the output of the op-amp to analog ground.

B.MUX controls the muxing to the input of the B capacitor branch.

PWR – encoding for selecting 1 of 4 power levels. The block always powers up in the off state.

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	A.REF1	A.REF0	F.SW1	F.SW0	B.MUX1	B.MUX0	PWR1	PWR0

Bit [7:6]: **A.REF [1:0]** Encoding for selecting reference input
0 0 = Analog ground is selected
0 1 = REFHI input selected (This is usually the high reference)
1 0 = REFLO input selected (This is usually the low reference)
1 1 = Reference selection is driven by the comparator.

Bit [5]: **F.SW1** Bit for controlling gated switches
0 = Switch is disabled
1 = If the F.SW1 bit is set to 1, the state of the switch is determined by the AZ bit. If the AZ bit is 0, the switch is enabled at all times. If the AZ bit is 1, the switch is enabled only when the internal PHI2 is high.

Bit [4]: **F.SW0** Bits for controlling gated switches
0 = Switch is disabled
1 = Switch is enabled when PHI2 is high

Bit [3:2] **B.MUX [1:0]** Encoding for selecting B inputs

	ASA10	ASA21	ASA12	ASA23
0 0 =	ACA00	ASB11	ACA02	ASB13
0 1 =	ASB11	ASB20	ASB13	ASB22
1 0 =	P2.1	ASB22	ASB11	P2.2
1 1 =	ASB20	T_{ref}GND	ASB22	ABUS3

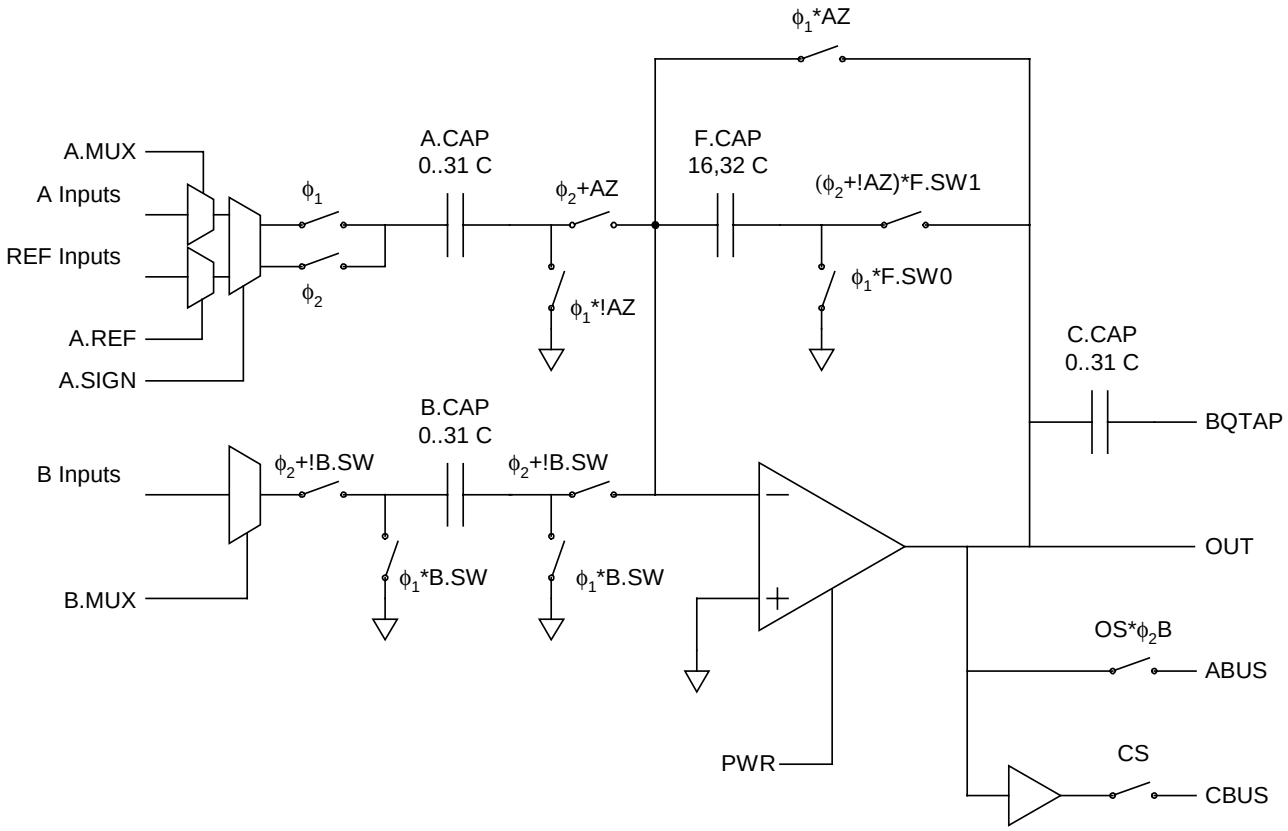
Bit [1:0]: **PWR [1:0]** Encoding for selecting 1 of 4 power levels
0 0 = Off
0 1 = 10 µA, typical
1 0 = 50 µA, typical
1 1 = 200 µA, typical

Analog Switch Cap Type A Block 10 Control 3 Register (ASA10CR3, Address = Bank 0/1, 83h)
Analog Switch Cap Type A Block 12 Control 3 Register (ASA12CR3, Address = Bank 0/1, 8Bh)
Analog Switch Cap Type A Block 21 Control 3 Register (ASA21CR3, Address = Bank 0/1, 97h)
Analog Switch Cap Type A Block 23 Control 3 Register (ASA23CR3, Address = Bank 0/1, 9Fh)

8.5.4 Analog Switch Cap Type B PSoC Blocks

8.5.4.1 Introduction

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



8.5.4.2 Registers

8.5.4.2.1 Analog Switch Cap Type B Block xx Control 0 Register

F.CAP controls the size of the switched feedback capacitor in the integrator.

PHASE controls the internal clock phasing relative to the input clock phasing. PHASE affects the output of the analog column bus which is controlled by the OS bit in Control 2 Register (ASB11CR2, ASB13CR2, ASB20CR2, ASB22CR2).

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

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

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	F.CAP	PHASE	A.SIGN	A.CAP4	A.CAP3	A.CAP2	A.CAP1	A.CAP0

Bit 7: **F.CAP** Bits for controlling gated switches
0 = 16 capacitor units
1 = 32 capacitor units

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

Following are the exceptions: 1) If the PHASE bit in CR0 (for the SC block in question) is set to one, then the output is enabled for the whole of PHI2. 2) If the SHDIS signal is set in bit 6 of the Analog Column Select Register, then sample and hold operation is disabled for all columns and all enabled outputs of SC blocks are connected to their respective output busses for the entire period of their respective PHI2s
0 = Internal PHI1 = External PHI1
1 = Internal PHI1 = External PHI2

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

Bit [4:0]: **A.CAP [4:0]** Binary encoding for 32 possible capacitor sizes
0 0 0 0 0 = 0 Capacitor units in array
0 0 0 0 1 = 1 Capacitor units in array
0 0 0 1 0 = 2 Capacitor units in array
0 0 0 1 1 = 3 Capacitor units in array
0 0 1 0 0 = 4 Capacitor units in array
0 0 1 0 1 = 5 Capacitor units in array
0 0 1 1 0 = 6 Capacitor units in array
0 0 1 1 1 = 7 Capacitor units in array
0 1 0 0 0 = 8 Capacitor units in array
0 1 0 0 1 = 9 Capacitor units in array
0 1 0 1 0 = 10 Capacitor units in array
0 1 0 1 1 = 11 Capacitor units in array
0 1 1 0 0 = 12 Capacitor units in array
0 1 1 0 1 = 13 Capacitor units in array
0 1 1 1 0 = 14 Capacitor units in array
0 1 1 1 1 = 15 Capacitor units in array
1 0 0 0 0 = 16 Capacitor units in array
1 0 0 0 1 = 17 Capacitor units in array
1 0 0 1 0 = 18 Capacitor units in array
1 0 0 1 1 = 19 Capacitor units in array
1 0 1 0 0 = 20 Capacitor units in array
1 0 1 0 1 = 21 Capacitor units in array
1 0 1 1 0 = 22 Capacitor units in array
1 0 1 1 1 = 23 Capacitor units in array
1 1 0 0 0 = 24 Capacitor units in array
1 1 0 0 1 = 25 Capacitor units in array
1 1 0 1 0 = 26 Capacitor units in array
1 1 0 1 1 = 27 Capacitor units in array
1 1 1 0 0 = 28 Capacitor units in array
1 1 1 0 1 = 29 Capacitor units in array
1 1 1 1 0 = 30 Capacitor units in array
1 1 1 1 1 = 31 Capacitor units in array

Analog Switch Cap Type B Block 11 Control 0 Register (ASB11CR0, Address = Bank 0/1, 84h)
Analog Switch Cap Type B Block 13 Control 0 Register (ASB13CR0, Address = Bank 0/1, 8Ch)
Analog Switch Cap Type B Block 20 Control 0 Register (ASB20CR0, Address = Bank 0/1, 90h)
Analog Switch Cap Type B Block 22 Control 0 Register (ASB22CR0, Address = Bank 0/1, 98h)

8.5.4.2.2 Analog Switch Cap Type B Block xx Control 1 Register

A.MUX controls the input muxing for the A capacitor branch.

The B.CAP bits set the value of the capcitor in the B path.

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	A.MUX2	A.MUX1	A.MUX0	B.CAP4	B.CAP3	B.CAP2	B.CAP1	B.CAP0

Bit [7:5]: **A.MUX [2:0]** Input muxing select for A capacitor branch

	ASB11	ASB13	ASB20	ASB22
0 0 0 =	ACA01	ACA03	ASA10	ASA12
0 0 1 =	ASA12	P2.2	P2.1	ASA21
0 1 0 =	ASA10	ASA12	ASA21	ASA23
0 1 1 =	ASA21	ASA23	ABUS0	ABUS2
1 0 0 =	REFHI	REFHI	REFHI	REFHI
1 0 1 =	ACA00	ACA02	ASB11	ASB13
1 1 0 =	Reserved	Reserved	Reserved	Reserved
1 1 1 =	Reserved	Reserved	Reserved	Reserved

Bit [4:0]: B.CAP [4:0] Binary encoding for 32 possible capacitor sizes

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

Analog Switch Cap Type B Block 11 Control 1 Register (ASB11CR1, Address = Bank 0/1, 85h)
Analog Switch Cap Type B Block 13 Control 1 Register (ASB13CR1, Address = Bank 0/1, 8Dh)
Analog Switch Cap Type B Block 20 Control 1 Register (ASB20CR1, Address = Bank 0/1, 91h)
Analog Switch Cap Type B Block 22 Control 1 Register (ASB22CR1, Address = Bank 0/1, 99h)

8.5.4.2.3 Analog Switch Cap Type B Block xx Control 2 Register

OS gates the output to the analog column bus. The output on the analog column bus is affected by the state of the PHASE bit in Control 0 Register (ASB11CR0, ASB13CR0, ASB20CR0, ASB22CR0). If OS is set to 0, the output to the analog column bus is tri-stated. If OS is set to 1, the signal that is output to the analog-column bus is selected by the PHASE bit. If the PHASE bit is 0, the block output is gated by sampling clock on last part of PHI2. If the PHASE bit is 1, the block output continuously drives the analog column bus.

CS controls the output to the column comparator bus. Note that logic was added so 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.

AZ controls the shorting of the output to the inverting input of the op-amp. When shorted, the op-amp is basically a follower. The output is the op-amp offset. By using the feedback capacitor of the integrator, the user can memorize the offset and create an offset cancellation scheme. AZ also controls a pair of switches between the A and B branches and the summing node of the op-amp. If AZ is enabled, then the pair of switches is active. AZ also affects the function of the F.SW1 bit in control word 4.

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

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	OS	CS	AZ	C.CAP4	C.CAP3	C.CAP2	C.CAP1	C.CAP0

Bit 7: **OS** Enable output to the analog bus
0 = Disable output to analog column bus
1 = Enable output to analog column bus
(The output on the analog column bus is affected by the state of the PHASE bit in Control 0 Register (ASB11CR0, ASB13CR0, ASB20CR0, ASB22CR0). If OS is set to 0, the output to the analog column bus is tri-stated. If OS is set to 1, the signal that is output to the analog column bus is selected by the PHASE bit. If the PHASE bit is 0, the block output is gated by sampling clock on last part of PHI2. If the PHASE bit is 1, the block output continuously drives the analog column bus)

Bit 6: **CS** Enable output to the comparator bus
0 = Disable output to comparator bus
1 = Enable output to comparator bus

Bit 5: **AZ** Bit for controlling gated switches
0 = Shorting switch is not active. Input cap branches shorted to op-amp input
1 = Shorting switch is enabled during internal PHI1. Input cap branches shorted to analog ground during internal PHI1 and to op-amp input during internal PHI2.

Bit [4:0]: **C.CAP [4:0]** Binary encoding for 32 possible capacitor sizes
0 0 0 0 0 = 0 Capacitor units in array
0 0 0 0 1 = 1 Capacitor units in array
0 0 0 1 0 = 2 Capacitor units in array
0 0 0 1 1 = 3 Capacitor units in array
0 0 1 0 0 = 4 Capacitor units in array
0 0 1 0 1 = 5 Capacitor units in array
0 0 1 1 0 = 6 Capacitor units in array
0 0 1 1 1 = 7 Capacitor units in array
0 1 0 0 0 = 8 Capacitor units in array
0 1 0 0 1 = 9 Capacitor units in array
0 1 0 1 0 = 10 Capacitor units in array
0 1 0 1 1 = 11 Capacitor units in array
0 1 1 0 0 = 12 Capacitor units in array
0 1 1 0 1 = 13 Capacitor units in array
0 1 1 1 0 = 14 Capacitor units in array
0 1 1 1 1 = 15 Capacitor units in array
1 0 0 0 0 = 16 Capacitor units in array
1 0 0 0 1 = 17 Capacitor units in array
1 0 0 1 0 = 18 Capacitor units in array
1 0 0 1 1 = 19 Capacitor units in array
1 0 1 0 0 = 20 Capacitor units in array
1 0 1 0 1 = 21 Capacitor units in array
1 0 1 1 0 = 22 Capacitor units in array
1 0 1 1 1 = 23 Capacitor units in array
1 1 0 0 0 = 24 Capacitor units in array
1 1 0 0 1 = 25 Capacitor units in array
1 1 0 1 0 = 26 Capacitor units in array
1 1 0 1 1 = 27 Capacitor units in array
1 1 1 0 0 = 28 Capacitor units in array
1 1 1 0 1 = 29 Capacitor units in array
1 1 1 1 0 = 30 Capacitor units in array
1 1 1 1 1 = 31 Capacitor units in array

Analog Switch Cap Type B Block 11 Control 2 Register (ASB11CR2, Address = Bank 0/1, 86h)
Analog Switch Cap Type B Block 13 Control 2 Register (ASB13CR2, Address = Bank 0/1, 8Eh)
Analog Switch Cap Type B Block 20 Control 2 Register (ASB20CR2, Address = Bank 0/1, 92h)
Analog Switch Cap Type B Block 22 Control 2 Register (ASB22CR2, Address = Bank 0/1, 9Ah)

8.5.4.2.4 Analog Switch Cap Type B Block xx Control 3 Register

A.REF selects the reference input of the A capacitor branch.

F.SW1 is used to control a switch between in the integrator capacitor path. It connects the output of the op-amp to the integrating cap. The state of the switch is affected by the state of the AZ bit in Control 2 Register (ASB11CR2, ASB13CR2, ASB20CR2, ASB22CR2). If the F.SW1 bit is set to 0, the switch is always disabled. If the F.SW1 bit is set to 1, the state of the switch is determined by the AZ bit. If the AZ bit is 0, the switch is enabled at all times. If the AZ bit is 1, the switch is enabled only when the internal PHI2 is high.

F.SW0 is used to control a switch in the integrator capacitor path. It connects the output of the op-amp to analog ground.

B.SW 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.

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

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	A.REF1	A.REF0	F.SW1	F.SW0	B.SW	B.MUX	PWR1	PWR0

Bit [7:6]: **A.REF [1:0]** Encoding for selecting reference input
0 0 = Analog ground is selected
0 1 = REFHI input selected (This is usually the high reference)
1 0 = REFLO input selected (This is usually the low reference)
1 1 = Reference selection is driven by the comparator

Bit 5: **F.SW1** Bit for controlling gated switches
0 = Switch is disabled
1 = If the DO2 bit is set to 1, the state of the switch is determined by the AZ bit. If the AZ bit is 0, the switch is enabled at all times. If the AZ bit is 1, the switch is enabled only when the internal PHI2 is high

Bit 4: **F.SW0** Bits for controlling gated switches
0 = Switch is disabled
1 = Switch is enabled when PHI2 is high

Bit 3: **B.SW** Enable switching in branch
0 = B branch is a continuous time path
1 = B branch is switched with internal PHI2 sampling

Bit 2: **B.MUX** Encoding for selecting B input

	ASB11	ASB13	ASB20	ASB22
0 =	ACA01	ACA03	ASB10	ASB12
1 =	ACA00	ACA02	ASA11	ASA13

Bit [1:0]: **PWR [1:0]** Encoding for selecting 1 of 4 power levels
0 0 = Off
0 1 = 10 µA, typical
1 0 = 50 µA, typical
1 1 = 200 µA, typical

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

8.6 Analog Comparator Bus

8.6.1 Analog Comparator Control Register

The ACLKx bits return the state of the PHI1 write window, which is the first half of the high period of the PHI1 for Analog Column x. This bit, when high, represents the optimal time to write to the registers in an analog PSoC block in Analog Column x.

The COMPx bits return the state of the analog comparator bus output from the Analog Column x.

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R
Bit Name	ACLK3	COMP3	ACLK2	COMP2	ACLK1	COMP1	ACLK0	COMP0

Bit 7: **ACLK3** Returns the state of PHI1 write window for Analog Column 3

Bit 6: **COMP3** Returns the state of the analog comparator bus for Analog Column 3

Bit 5: **ACLK2** Returns the state of PHI1 write window for Analog Column 2

Bit 4: **COMP2** Returns the state of the analog comparator bus for Analog Column 2

Bit 3: **ACLK1** Returns the state of PHI1 write window for Analog Column 1

Bit 2: **COMP1** Returns the state of the analog comparator bus for Analog Column 1

Bit 1: **ACLK0** Returns the state of PHI1 write window for Analog Column 0

Bit 0: **COMP0** Returns the state of the analog comparator bus for Analog Column 0

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

8.7 Analog Synchronization

8.7.1 Analog Synchronization Control Register

For high precision analog operation, it may be necessary to precisely time when updated register values are available to the analog PSoC blocks. There is a window of time, which is the first half of the PHI1 active period, when it is optimal to update values in the registers. The user has the ability to determine the state of the write window by reading the ACLKx bit in the Analog Comparator Control 1 Register (CMP_CR). If the user sets the SYNCEN bit, and a write occurs outside the write window, the CPU will be stalled until the beginning of the PHI1 write window. If the write takes place inside the PHI1 write window, it will proceed without delay.

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

Bit 7: **Reserved**

Bit 6: **Reserved**

Bit 5: **Reserved**

Bit 4: **Reserved**

Bit 3: **Reserved**

Bit 2: **Reserved**

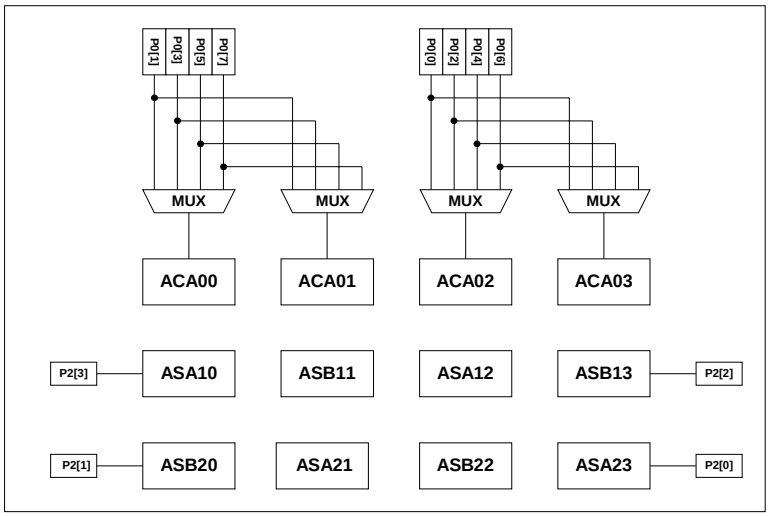
Bit 1: **Reserved**

Bit 0: **SYNCEN** If set to 1 by the user, will stall the CPU if a write to a register within an analog PSoC block takes place.

Analog Synchronization Control Register (ASY_CR, Address = Bank 0, 65h)

8.8 Analog I/O

8.8.1 Analog Input Muxing



8.8.1.1 Analog Input Select Register

This register controls the analog muxes that feed signals in from port pins into each Analog Column. Each of the Analog Columns can have up to four port pins connected to its muxed input. There are four additional analog inputs that go directly into the Switch Capacitor PSoC blocks.

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	ACI3 [1]	ACI3 [0]	ACI2 [1]	ACI2 [0]	ACI1 [1]	ACI1 [0]	ACI0 [1]	ACI0 [0]

Bit [7:6]: **ACI3 [1:0]**
0 0 = Port Pin P0[0]
0 1 = Port Pin P0[2]
1 0 = Port Pin P0[4]
1 1 = Port Pin P0[6]

Bit [5:4]: **ACI2 [1:0]**
0 0 = Port Pin P0[0]
0 1 = Port Pin P0[2]
1 0 = Port Pin P0[4]
1 1 = Port Pin P0[6]

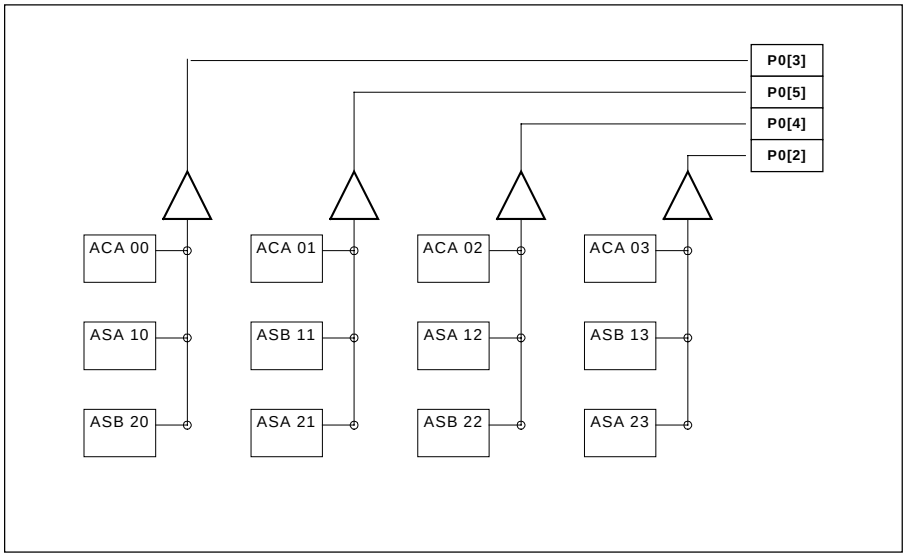
Bit [3:2]: **ACI1 [1:0]**
0 0 = Port Pin P0[1]
0 1 = Port Pin P0[3]
1 0 = Port Pin P0[5]
1 1 = Port Pin P0[7]

Bit [1:0]: **ACI0 [1:0]**
0 0 = Port Pin P0[1]
0 1 = Port Pin P0[3]
1 0 = Port Pin P0[5]
1 1 = Port Pin P0[7]

Analog Input Select Register (AMX_IN, Address = Bank 0, 60h)

8.8.2 Analog Output Buffers

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



8.8.2.1 Analog Output Buffer Control Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	--	--	W	W	W	W	--	W
Bit Name	Reserved	Reserved	ABUF1EN	ABUF2EN	ABUF0EN	ABUF3EN	Reserved	PWR

Bit 7: **Reserved**

Bit 6: **Reserved**

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

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

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

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

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

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

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

8.9 Analog Reference and Bias Control

The references in the analog array are driven by single op-amps. A single ground referred signal is taken as the reference input and then offset with respect to analog ground. The reference can be input on a pin, it can be taken from the bandgap, or it can be set to be the supplies. A series of op-amps are used to do the level shifting and buffering for driving the array. As more loads are added on the reference lines, the response will slow down. Settling time will be roughly linear with load.

A separate bias circuit controls the 3 rows. The first row is to be controlled independently. The second and third rows have their bias control tied together.

8.9.1 Analog Reference Control Register

RS2 and RS1 control the muxing of the input to the reference circuit.

HBE controls the bias level. This is essentially a boost bit. There is a trade-off. At high bias levels, the op-amps swings are more limited but the op-amp can be faster. At low bias levels, wider swings (and hence lower supply voltages) are possible, but the op-amp is slower.

RP2 and RP1 control the power levels of the reference drivers. The control is similar to the power levels in the array blocks.

R32EN controls the bias circuits in rows 2 and 3 of the array.

PEN is the global power enable. When PEN is 0, it is possible to go to 0 current consumption in the array.

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	RS2	RS1	HBE	Reserved	RP1	RP0	R32EN	PE

Bit [7:6]: **RS2,RS1** Input muxing for reference circuit
0 0 = Vref High = Agnd + Bandgap Vref Low = Agnd - Bandgap
0 1 = Vref High = Agnd + P0[0] Vref Low = Agnd – P0[0]
1 0 = Vref High = Agnd + P2[2] Vref Low = Agnd – P2[2]
1 1 = Vref High = Vcc Vref Low = Vss

Bit 5: **HBE** Bias level control for op-amps
0 = Low bias mode for analog array
1 = High bias mode for analog array

Bit 4: **Reserved**

Bit [3:2]: **RP1,RP0** Sets power level of the reference drivers
0 0 = Off
0 1 = Low
1 0 = Medium
1 1 = High

Bit 1: **R32EN** Controls bias circuit in rows 2 and 3 of the array
0 = Rows 2 and 3 are off
1 = Rows 2 and 3 are on

Bit 0: **PE** Global power enable for analog array
0 = Array power is off
1 = Array power is on

Analog Reference Control Register (ARF_CR, Address = Bank 0, 63h)

8.10 Analog Modulator

The user has the capability to use switched-cap analog PSoC blocks as amplitude modulators. This is implemented by controlling the sign bit in the switched-cap A-capacitor at the carrier rate. Two dedicated routing resources bring the outputs of user-selectable muxes to Analog Columns 0 and 2. The Analog Modulator Control Register (AMD_CR) allows the user to select the appropriate signal.

8.10.1 Analog Modulator Control Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Reserved	Reserved	Reserved	Reserved	AMOD21	AMOD20	AMOD01	AMOD00

Bit 7: **Reserved**

Bit 6: **Reserved**

Bit 5: **Reserved**

Bit 4: **Reserved**

Bit [3:2]: **AMOD21**, **AMOD20** Selects the modulation signal for Analog Column 2
0 0 = No Modulation
0 1 = Global Output [0]
1 0 = Global Output [4]
1 1 = Digital Basic Type A Block 00

Bit [1:0]: **AMOD01**, **AMOD00** Selects the modulation signal for Analog Column 0
0 0 = No Modulation
0 1 = Global Output [0]
1 0 = Global Output [4]
1 1 = Digital Basic Type A Block 00

Analog Modulator Control Register (AMD_CR, Address = Bank 1, 63h)

8.11 Potential Analog User Modules

8.11.1 Delta-Sigma A/D converters

8.11.2 Successive Approximation A/D converters

8.11.3 Incremental A/D converters

8.11.4 Programmable gain/loss stage

8.11.5 Analog comparators

8.11.6 Zero-crossing detectors

8.11.7 Filters

8.11.8 Amplitude modulators,

8.11.9 Amplitude demodulators

8.11.10 Sine-wave generators

8.11.11 Sine-wave detectors

8.11.12 Sideband detection

8.11.13 Sideband stripping

8.11.14 Frequency modulation

8.11.15 Frequency demodulation

8.11.16 Audio coding, audio decoding,

8.11.17 Audio output drive

8.11.18 Audio compress/expansion.

8.12 Temperature Sensing Capability

A temperature-sensitive voltage derived from the Band Gap sensing on the die is buffered and available as an analog input into the Analog Switchcap Type A Block 21. 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 out the internal temperature rise based on a known current consumption.

9 Special Features of the CPU

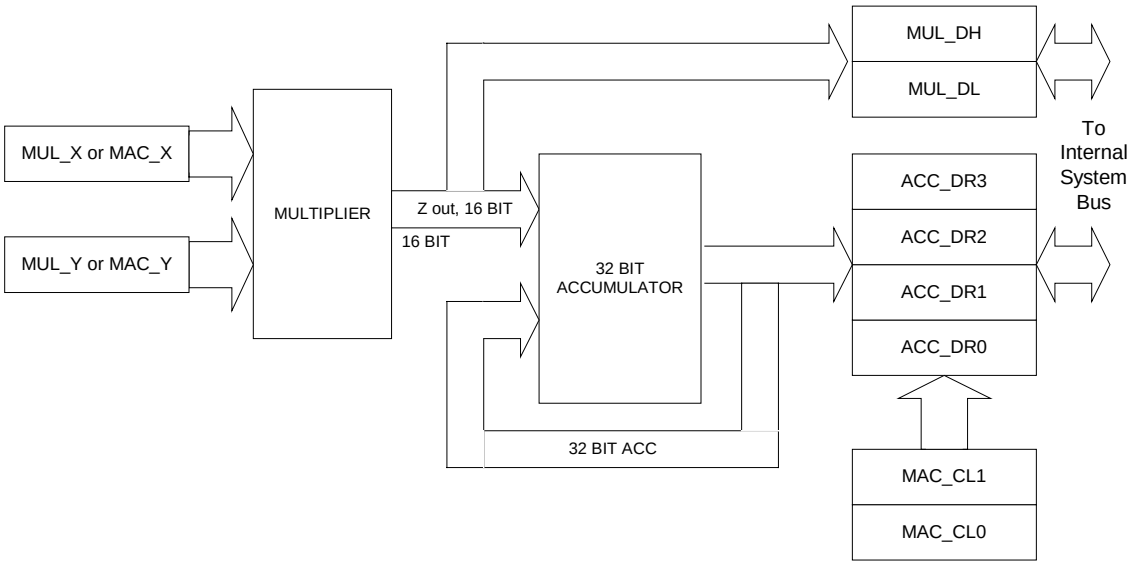
9.1 Multiplier / Accumulator

A fast, on-chip signed 2's complement MAC (Multiply/Accumulate) function is provided to assist the main CPU with digital signal processing applications. MAC results are available immediately after the input registers are written. The MAC function is tied directly on the internal data bus, and is mapped into the register space. The following MAC block diagram provides data flow information. The user has the choice to either cause a multiply/accumulate function to take place, or a multiply only function. The user selects which operation is performed by the choice of input register. The multiply function occurs immediately whenever the MUL_X or the MUL_Y multiplier input registers are written, and the result is available in the MUL_DH and MUL_DL multiplier result registers. The multiply/accumulate function is executed whenever there is a write to the MAC_X or the MAC_Y multiply/accumulate input registers, and the result is available in the ACC_DR3, ACC_DR2, ACC_DR1, and ACC_DR0 accumulator result registers. A write to MUL_X or MAC_X is input as the X value to both the multiply and multiply/accumulate functions. A write to MUL_Y or MAC_Y is input as the Y value to both the multiply and multiply/accumulate functions. A write to the MAC_CL0 or MAC_CL1 registers will clear the value in the four accumulate registers.

Operation of the multiply/accumulate function relies on proper multiplicand input. The first value of each multiplicand must be placed into MUL_X (or MUL_Y) register to avoid causing a multiply/accumulate to occur. The second multiplicand must be placed into MAC_Y (or MAC_X) thereby triggering the multiply/accumulate function.

MUL_X, MUL_Y, MAC_X, and MAC_Y are 8-bit signed input registers. MUL_DL and MUL_DH form a 16-bit signed output. ACC_DR0, ACC_DR1, ACC_DR2 and ACC_DR3 form a 32-bit signed output.

Multiply/Accumulate Block Diagram



9.1.1.1.1 Multiply Input X Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W
Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]** 8-bit data is the input value for X multiplier

Multiply Input X Register (MUL_X, Address = Bank 0, E8h)

9.1.1.1.2 Multiply Input Y Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]** 8-bit data is the input value for Y multiplier

Multiply Input Y Register (MUL_Y, Address = Bank 0, E9h)

9.1.1.1.3 Multiply Result High Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]** 8-bit data value is the high order result of the multiply function

Multiply Result High Register (MUL_DH, Address = Bank 0, EAh)

9.1.1.1.4 Multiply Result Low Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]** 8-bit data value is the low order result of the multiply function

Multiply Result Low Register (MUL_DL, Address = Bank 0, EBh)

9.1.1.1.5 Accumulator Result 1 / Multiply/Accumulator Input X Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]**
8-bit data value when read is the next to lowest order result of the multiply/accumulate function
8-bit data value when written is the X multiplier input to the multiply/accumulate function

Accumulator Result 1 / Multiply/Accumulator Input X Register (ACC_DR1 / MAC_X, Address = Bank 0, ECh)

9.1.1.1.6 Accumulator Result 0 / Multiply/Accumulator Input Y Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]**
8-bit data value when read is the lowest order result of the multiply/accumulate function
8-bit data value when written is the Y multiplier input to the multiply/accumulate function

Accumulator Result 0 / Multiply/Accumulator Input Y Register (ACC_DR0 / MAC_Y, Address = Bank 0, EDh)

9.1.1.1.7 Accumulator Result 3 / Multiply/Accumulator Clear 0 Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]**
8-bit data value when read is the highest order result of the multiply/accumulate function
Any 8-bit data value when written will cause all four Accumulator result registers to clear

Accumulator Result 3 / Multiply/Accumulator Clear 0 Register (ACC_DR3 / MAC_CL0, Address = Bank 0, EEh)

9.1.1.1.8 Accumulator Result 2 / Multiply/Accumulator Clear 1 Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

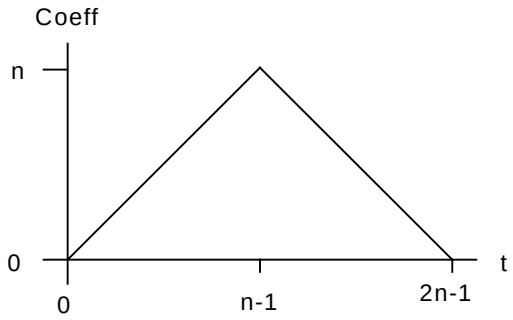
Bit [7:0]: **Data [7:0]**
8-bit data value when read is next to highest order result of the multiply/accumulate function
Any 8-bit data value when written will cause all four Accumulator result registers to clear

Accumulator Result 2 / Multiply/Accumulator Clear 1 Register (ACC_DR2 / MAC_CL1, Address = Bank 0, EFh)

9.2 Decimator

The output of a Δ–Σ modulator is a high-speed, single bit A/D converter. A single bit A/D converter is of little use to anyone and must be converted to a lower speed multiple bit output. Converting this high-speed single bit data stream to a lower speed multiple bit data stream requires a data decimator.

A "divide by n" decimator is a digital filter that takes the single bit data at a fast rate and outputs multiple bits at one n^{th} the speed. For a single stage Δ–Σ converter, the optimal filter has a $sinc^2$ response. This filter can be implemented as a finite impulse response (FIR) filter and for a "divide by n" implementation should have the following coefficients:



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

9.2.1 Decimator Control Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	--	--	--	--	--	RW	RW	RW
Bit Name	Reserved	Reserved	Reserved	Reserved	Reserved	Col[1]	Col[0]	EN

Bit 7: **Reserved**

Bit 6: **Reserved**

Bit 5: **Reserved**

Bit 4: **Reserved**

Bit 3: **Reserved**

Bit [2:1]: **Col [1:0]** Selects analog column comparator source
0 0 = Analog column comparator 0
0 1 = Analog column comparator 1
1 0 = Analog column comparator 2
1 1 = Analog column comparator 3

Bit 0: **EN** Enables the decimation function
0 = Decimator disabled
1 = Decimator enabled

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

9.2.2 Decimator Data High Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]**
8-bit data value when read is the high order byte within the 16-bit decimator data registers
Any 8-bit data value when written will cause both the Decimator Data High (DEC_DH) and Decimator Data Low (DEC_DL) registers to be cleared

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

9.2.3 Decimator Data Low Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]

Bit [7:0]: **Data [7:0]**
8-bit data value when read is the low order byte within the 16 bit decimator data registers

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

9.3 Reset

9.3.1 Overview

The micro-controller supports two types of resets. When reset is initiated, all registers are restored to their default states and all interrupts are disabled.

Reset Types: Power On Reset (POR), External Reset (X_{res}), and Watchdog Reset (WDR).

The occurrence of a reset is recorded in the Status and Control Register (CPU_SCR). Bits within this register record the occurrence of POR and WDR Reset respectively. The firmware can interrogate these bits to determine the cause of a reset.

The micro-controller resumes execution from ROM address 0x0000 after a reset. The internal clocking mode is active after a reset, until changed by user firmware. In addition, the Sleep / Watchdog Timer is reset to its minimum interval count.

Important: The CPU clock defaults to divide by 8 mode at POR to guarantee operation at the low Vcc that is present during the supply ramp.

9.3.2 Processor Status and Control Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	--	--	R/C*	R/C*	RW	--	--	RW
Bit Name	Reserved	Reserved	WDRS	PORS	Sleep	Reserved	Reserved	Stop

Bit 7: **Reserved**

Bit 6: **Reserved**

Bit 5: **WDRS**
WDRS is set by the CPU to indicate that a Watch Dog Reset event has occurred. The user can read this bit to determine the type of reset that has occurred. The user cannot set this bit, but can clear it
0 = No WDRS
1 = Power On Reset Event

Bit 4: **PORS**
PORS is set by the CPU to indicate that a Power On Reset event has occurred. The user can read this bit to determine the type of reset that has occurred. The user cannot set this bit, but can clear it
0 = No POR event has taken place
1 = A POR event has taken place (Note that WDRS events will not take place until this bit is cleared)

Bit 3: **Sleep** Set by the user to enable the CPU sleep state. The CPU will remain in sleep mode until any interrupt has taken place
0 = Normal operation
1 = Sleep

Bit 2: **Reserved**

Bit 1: **Reserved**

Bit 0: **Stop** Set by the user to halt the CPU. The CPU will remain halted until a reset (WDR or POR) has taken place
0 = Normal CPU operation
1 = CPU is halted (not recommended)

Status and Control Register (CPU_SCR, Address = Bank 0/1, FFh) *C = Clear

9.3.3 Power On Reset (POR)

Power On Reset (POR) occurs every time the power to the device is switched on; POR is released when the supply is 2.2V +/-10% for the upward supply transition. Bit 4 of the Control and Status Register (CPU_SCR) is set to record this event (the register contents are set to 00010001 by the POR). After a POR, the microprocessor is suspended for 64mS. This provides time for the Vcc supply to stabilize after the POR trip, before CPU operation begins. If the Vcc voltage drops below the POR downward supply trip point (2V +/-10%, 200mV of hysteresis between upward and downward transitions), POR is reasserted.

Important: The PORS status bit is set at POR and can only be cleared by the user, and cannot be set by firmware.

9.3.4 External Reset (X_{res})

Pulling the X_{res} pin high for a minimum of 1 mS forces the microcontroller to perform a Power On Reset (POR). Engineering sample parts requires that X_{res} be tied to ground through a 330-1 KΩ register.

9.3.5 Watchdog Timer Reset (WDR)

The user has the option to enable the WDR. The WDR is enabled by clearing the PORS bit. Once the PORS bit is cleared, the Watch Dog Timer cannot be disabled. The only exception to this is if a POR event takes place, which will disable the WDR.

The sleep timer is used to generate the sleep time period and the watchdog time period.

The sleep timer divides down the **32K** system clock, and thereby produces the sleep time period. The user can program the sleep time period to be one of 4 multiples of the period of the **32K** clock. When it overflows, an interrupt to the Sleep Timer Interrupt Vector will be generated.

The Watch Dog Timer period is automatically set to be 3 times the sleep timer period. When this timer overflows, a WDR is generated.

The user can either clear the WDT, or the WDT and the Sleep Timer. Whenever the user writes to the Reset WDT Register (RES_WDT), the WDT will be cleared. If the data that is written is the hex value 38H, the Sleep Timer will also be cleared at the same time.

This timer chain is also used to time the startup for the external 32kHz crystal oscillator. When selecting the external 32kHz oscillator, a value of 1 second must be selected as the sleep interval. When the sleep interrupt occurs, the 32kHz oscillator source will switch from internal to the crystal. The device does not have to be put into sleep for this event to occur. Note that if too short of a sleep interval is given, the crystal oscillator will not be stable prior to switch over and the results will be unpredictable.

9.3.6 Reset WDT Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	Data [7]	Data [6]	Data [5]	Data [4]	Data [3]	Data [2]	Data [1]	Data [0]
Bit [7:0]: **Data [7:0]** Any write to this register will clear the Watch Dog Timer, a write of 38H will also clear the Sleep Timer as well.								

Reset WDT Register (RES_WDT, Address = Bank 0, E3h)

9.4 Sleep States

There are three sleep states that can be used to lower the overall power consumption on the device. The three states are CPU Sleep, Analog Sleep, and Full Sleep.

The CPU can only be put to sleep by the firmware. This is accomplished by setting the Sleep Bit in the Status and Control Register (CPU_SCR). This stops the CPU from executing instructions, and the CPU will remain asleep until an interrupt occurs, or there is a reset event (either a Power On Reset, or a Watch Dog Timer Reset). While in the CPU Sleep state, allclocking signals derived from the Internal Main Oscillator are inactivated, including the **48M**, **24M**, **24V1**, and **24V2** system clocking signals. The Internal Low Speed Oscillator will continue to operate during the CPU Sleep state. The function of any Analog or Digital PSoC block that clocked from these system-clocking signals will stop during the CPU Sleep state.

The user can also put all the Analog PSoC block circuits to sleep. This is accomplished by resetting the Power Enable bit in the Reference Control Register Two (REF_CR2), which overrides the individual enable bits within each Analog PSoC block. Setting the Power Enable bit will restore the function to those Analog PSoC blocks that were previously in use. The user should take into account the required setting time after an Analog PSoC block is enabled before it will provide the maximum precision.

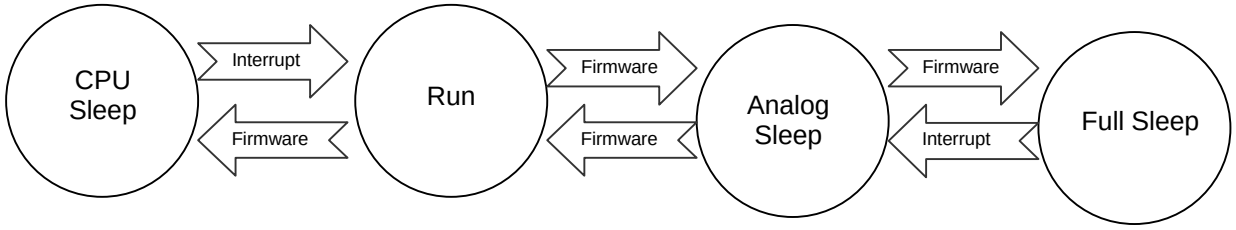
For greatest power savings, the user should put the device in the Full Sleep state. This is accomplished by first transitioning to the Analog Sleep state, and then to the Full Sleep state. The CPU will be stopped at this point, and either an interrupt or reset event is required to transition back to the Analog Sleep state.

The Voltage Reference and Supply Voltage Monitor drop into (fully functional) power-reduced states. All interrupts remain active. The Internal Low Speed Oscillator remains running (it will however drop into a less accurate, low-power state). If enabled, the External Crystal Oscillator will continue running throughout sleep (the Internal Low Speed Oscillator is disabled if the External Crystal Oscillator is selected). Only the occurrence of an interrupt will wake the part from sleep. The Run bit must be set for a part to resume out of sleep.

Any digital PSoC block that is clocked by a System Clock other than the **32K** system-clocking signal or external pins will be stopped, as these clocks do not run in sleep mode.

The Internal Main Oscillator restarts immediately on exiting sleep mode. Analog functions must be re-enabled by firmware. If the External Crystal Oscillator is used and the internal PLL is enabled, the PLL will take many cycles to change from its initial 2% accuracy to track that of the External Crystal Oscillator. If the PLL is enabled, there will be a 30µS (one full **32K** cycle) delay hold-off time for the CPU to let the VCO and PLL stabilize. If the PLL is not enabled, the hold-off time is one half of the **32K** cycle.

The Sleep interrupt allows the microcontroller to wake-up periodically and poll system components while maintaining very low average power consumption.



Three Sleep States

9.5 Supply Voltage Monitor

The Supply Voltage Monitor detector generates an interrupt whenever Vcc drops below a pre-programmed value set by the Voltage Monitor Control Register (VLT_CR). It covers two supply voltage ranges: 3.3 V +/- 5% with 4 steps and 5.0 V +/- 5% with 4 steps. The Supply Voltage Monitor will remain active when the device enters sleep mode.

9.5.1.1.1 Voltage Monitor Control Register

Bit #	7	6	5	4	3	2	1	0
POR	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Bit Name	SMP	Reserved	Reserved	Reserved	Reserved	VM [2]	VM [1]	VM [0]

Bit 7: **SMP** Disables SMP function
0 = (Switch Mode Pump Enabled, default)
1 = (Switch Mode Pump Disabled)

Bit 6: **Reserved**

Bit 5: **Reserved**

Bit 4: **Reserved**

Bit 3: **Reserved**

Bit 2: **VM [2]**
0 = Low Voltage range (3.3V)
1 = High Voltage range (5.0V)

Bit [2:0]: **VM [2:0]**
The supply voltage monitor also puts out a 4.4V trip signal for use by the flash in starting its internal boost pumps at points where supply conditions require this.
0 0 0 = 2.64 (80% of 3.3V) Trip Voltage (+/- 3.0 %)
0 0 1 = 2.77 (84% of 3.3V) Trip Voltage (+/- 3.0 %)
0 1 0 = 2.90 (88% of 3.3V) Trip Voltage (+/- 3.0 %)
0 1 1 = 3.04 (92% of 3.3V) Trip Voltage (+/- 3.0 %)
1 0 0 = 4.00 (80% of 5.0V) Trip Voltage (+/- 3.0 %)
1 0 1 = 4.20 (84% of 5.0V) Trip Voltage (+/- 3.0 %)
1 1 0 = 4.40 (88% of 5.0V) Trip Voltage (+/- 3.0 %)
1 1 1 = 4.60 (92% of 5.0V) Trip Voltage (+/- 3.0 %)

Voltage Monitor Control Register (VLT_CR, Address = Bank 1, E3h)

9.6 Internal Voltage Reference

An internal bandgap voltage reference source is provided on-chip. This reference is used for the Supply Voltage Monitor, and can also be accessed by the user as a reference voltage for analog operations.

9.6.1 Bandgap Trim Register

Bit #	7	6	5	4	3	2	1	0
POR	0	--	--	--	--	--	--	0
Read/Write	--	W	W	W	W	W	W	W
Bit Name	Reserved	BGT[2]	BGT[1]	BGT[0]	BGO[3]	BGO[2]	BGO[1]	BGO[0]

Bit 7: **Reserved**

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

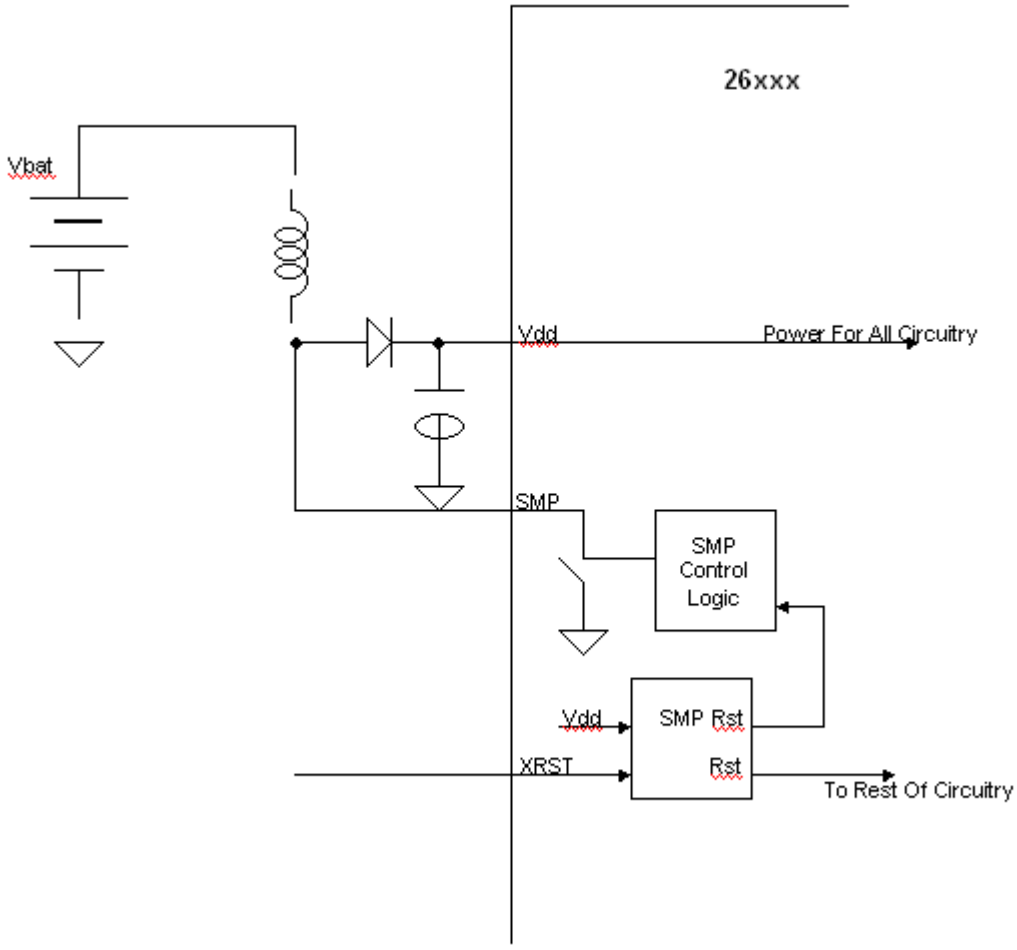
Bit [3:0]: **BGO [3:0]** Provides +/- 5% Offset Trim to center Vbg to 1.30V, to +/- 1% final accuracy

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

9.7 Switch Mode Pump

This feature is available on the CY8C26xxx versions within this family. During the time Vcc is ramping from 0 to V_{trip} (2.2V +/- 10%), IC operation is held off by the POR circuit (and the switch mode pump is forced on) pin SMP is driven to support a Switch Mode voltage pump. The pump is realized by connecting an external inductor between V_{bat} and SMP, with an external diode pointing from SMP to the Vcc pin (which must have a bypass capacitance of at least 0.1uF connected to Vss). This circuitry will pump Vcc to 5% greater than the current Supply Voltage Monitor setting and attempt to regulate to this level. V_{bat} values down to 0.9 V during operation are supported, but this circuitry is not guaranteed to start for battery voltages below 1.2 V. Once the IC is enabled after its power up and boot sequence, firmware can disable the SMP function by writing Voltage Monitor Control Register (VLT_CR): bit 7 to a 1.

When the IC is put into sleep mode, the power supply pump will remain running to maintain voltage. This may result in higher than specification sleep current depending upon application. If the user desires, the pump may be disabled during precision measurements (such as A/D conversions) and then re-enabled (writing B7 to 1 and then back to 0 again). The user, however, is responsible for making the operation happen quickly enough to guarantee supply holdup (by the bypass capacitor) sufficient for continued operation.



9.8 Supervisor ROM / System Supervisor Call Instruction

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

The System Supervisor Call instruction (SSC, opcode 00) provides the method for the user to access the pre-existing routines in the Supervisor ROM to implement these functions. This instruction sets the Flags Register (CPPU_F) bit 3 to a 1 and performs an interrupt to address 0000 into the Supervisory ROM. The flag and old PC are pushed onto the Stack. The fact that the flag pushed has F[3] = 1 is irrelevant as the RETI instruction always clears F[3]. The Supervisory code at 0000 does a JACC table lookup based on the Accumulator value, which is effectively another level of instruction encoding. This service table implements the vectors to the various supervisory functions. The user must set several parameters when utilizing these functions. The parameters are written to a five-byte block near the top of RAM memory space. The highest byte address within the block is the three less than the highest address within the entire array (FCh for devices with 256 bytes and 7Ch for devices with 128 bytes). The lowest byte address within the block is the seven less than the highest address within the entire array. The following table documents each function, as well as the required parameter values:

Operation	Function	A Data	SRAM Data									
			F8h or 78h		F9h or 79h		FAh or 7Ah		FBh or 7Bh		FCh or 7Ch	
			Input	Output	In	Out	In	Out	In	Out	In	Out
Reset		00	NA	*	NA	*	NA	*	NA	*	NA	*
Read Block	Move block of 64 bytes of FLASH data into SRAM	01	3Ah	0	SP+3	0	Block ID	0	Pointer	0	NA	0
Program Block	Program block of FLASH with data from SRAM	02	3Ah	0	SP+3	0	Block ID	0	Pointer	0	Clock	0
Erase Block	Erase block of FLASH	03	3Ah	0	SP+3	0	Block ID	0	Pointer	0	Clock	0
Protect Block	[1]	04	3Ah	0	SP+3	0	NA	0	NA	0	Clock	0
Erase All	Erase all FLASH data[2]	05	3Ah	0	SP+3	0	NA	0	NA	0	Clock	0
Read Product ID	Read device type code	06	3Ah	34h	SP+3	8Ch	NA	25h	NA	XXh	NA	*
Checksum	Calculate FLASH checksum for data range specified	07	3Ah	CS4	SP+3	CSL	Block Counter	*	NA	*	NA	*

* = Indeterminate
XXh = Returned ID information
Block ID = Number of 64-byte block within FLASH memory space
Clock = CPU system clocking signal value (in MHz) x 5
NA = Not Applicable
Pointer = Address of first byte of 64-byte block within SRAM memory space

Notes:
[1] Always uses 080h as RAM buffer
[2] Erase All will erase the security bits and all of the code in the FLASH

9.9 Flash Program Memory Protection

The user has the option to define the access to the Flash memory. A flexible system allows the user to select one of four protection modes for each 64-byte block within the Flash based on the particular application. The protection mechanism is implemented using the System Supervisor Call instruction. When this command is executed, two bits within the data programmed into the Flash will select the protection mode. The following table lists the available protection options:

Mode Bits	Mode Name	External Read	External Write	Internal Write
00	Unprotected	Enabled	Enabled	Enabled
01	Factory Upgrade	Disabled	Enabled	Enabled
10	Field Upgrade	Disabled	Disabled	Enabled
11	Full Protection	Disabled	Disabled	Disabled

9.10 Programming Requirements, Flow Chart, and Step Descriptions

9.10.1 Programmer Requirements

The pins in the following table are critical for the programmer:

Pin Name	Function	Programmer HW Pin Requirements
SDATA	Serial Data In/Out	Drive TTL levels, Read TTL, High-Z
SCLK	Serial Clock	Drive TTL level clock signal
Vss	Power Supply Ground Connection	Low resistance ground connection
Vcc	Power Supply Positive Voltage	0V, 2.8V, 5V, and 5.4V. 0.1V accuracy. 20mA current capability

9.10.2 Data File Read

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

9.10.3 Programmer Flow

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

9.10.3.1 Verify Silicon ID

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

This test will detect a bad connection to the programmer or an incorrect device selection on the programmer.
The silicon ID test is required to be first in the flow and cannot be bypassed. The sequence is as follows:

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

Notes: See "DC Specifications" table in section 11 for value of Vccp and VILP. See "AC Specifications" table in section 11 for value of frequency for the SCLK driver (Fsclk).

9.10.3.2 Erase

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

SET-CLK-FREQ(num_MHz_times_5)
ERASE
WAIT-AND-POLL

Notes: Sequence is performed with Vcc=Vccp; see "DC Specifications" table in section 11 for value of Vccp.

9.10.3.3 Program

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

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

Notes: Sequence is done with Vcc=Vccp.

9.10.3.4 Verify (at Low Vcc and High Vcc)

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

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

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

9.10.3.5 Set Security

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

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

Notes: This sequence is done at Vcc=Vccp.

9.10.3.6 Device Checksum (at Low Vcc and High Vcc)

The device checksum is retrieved from the device and compared to the "Device Checksum" from the user's file (Note that this is NOT the same thing as the "Record Checksum.") The checksum is retrieved from the device with the following sequence:

```
CHECKSUM-SETUP(max_data_block)
WAIT-AND-POLL
READ-CHECKSUM(data)
```

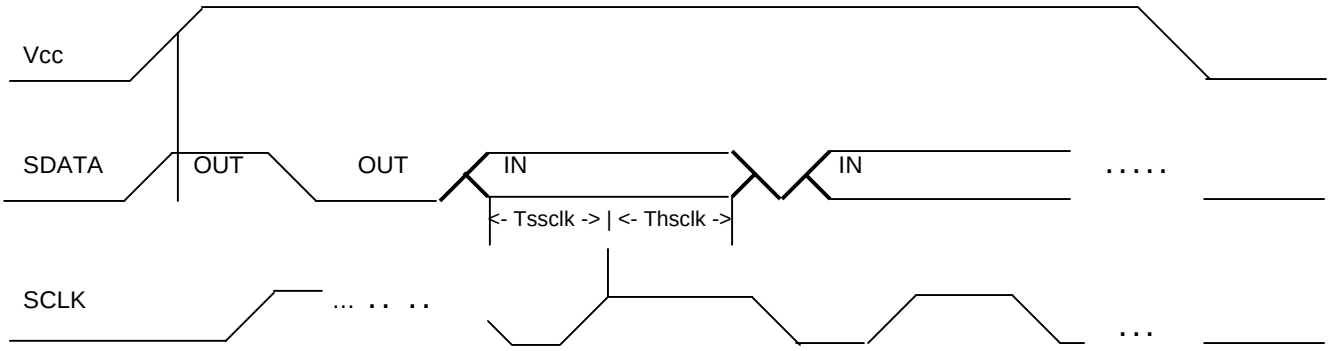
Notes: This should be done 2 times; once at Vcc=Vcchv and once at Vcc=Vcclv.

9.10.3.7 Power Down

The last step is to power down the device. This is accomplished by the following sequence:

```
Set SDATA=HighZ (float pin P1[0])
Set SCLK=0V (Vin on pin P1[1]=Vilp)
Set Vcc = 0V
```

9.11 Programming Wave Forms



Notes:

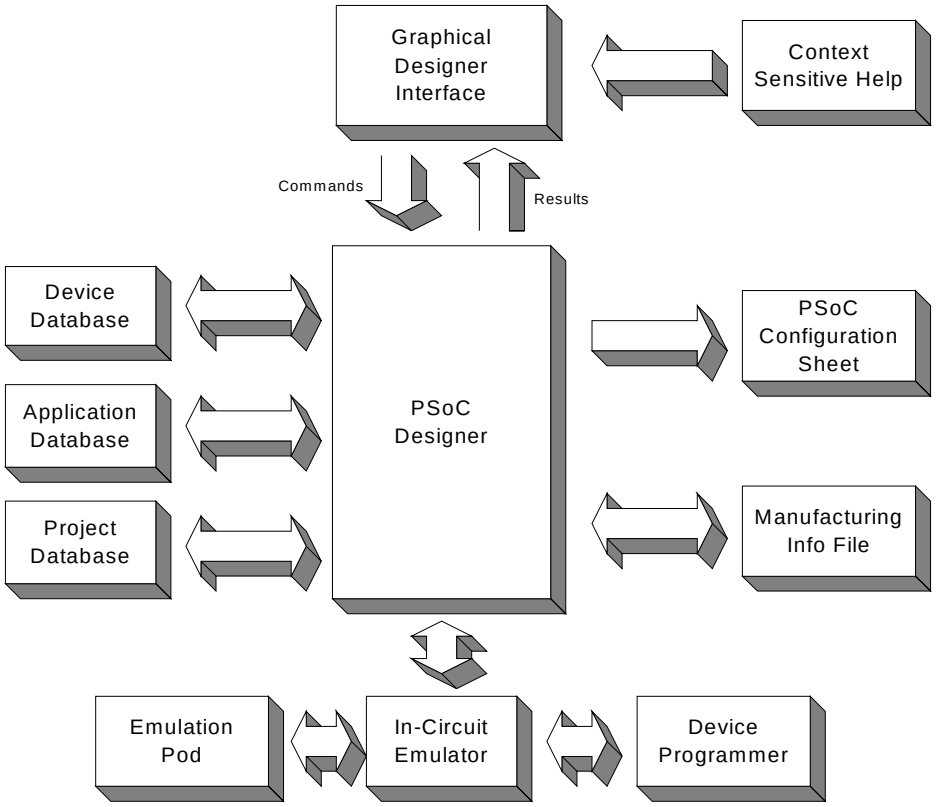
1. Vcc is only turned off (0V) at the very beginning and the very end of the flow - not within the programming flow.
2. When the programmer puts the driver on SDATA in a high Z (floating) state, the SDATA pin will float to a low due to an internal device pull down circuit.
3. SCLK is set to VILP during the power up and power down; at other times the SCLK is “free running.” The frequency of the hardware’s SCLK signal must be known by the software because the value (entered in the number of MegaHertz multiplied by the number 5) must be passed into the device with the SET-CLK-FREQ() mnemonic.

9.12 Programming File Format

The programming file is created by PSoC™ Designer, the Cypress MicroSystems development tool. This tool generates this file in a format similar to IntelHex.

The programmer should assume the data is 00 if it is not specified in the user’s data file.

10 Development Tools



10.1 Overview

The Cypress MicroSystems PSoC Designer is a Microsoft ® Windows-based, integrated development environment for the 8C2000 Programmable System-on-Chip (PSoC) devices. The PSoC Designer runs on Windows 98, Windows NT 4.0, Windows 2000, or Windows Millennium edition.

PSoC Designer helps the customer to select an operating configuration for the microcontroller, write application code that uses the microcontroller, and debug the application. This system provides design database management by project, an integrated debugger with In-Circuit Emulator, in-system programming support, and the CYASM macro assembler for the 8C2000 family CPUs.

PSoC Designer also supports a high-level C language compiler developed specifically for the 8C2000 PSoC family of devices.

10.2 Integrated Development Environment Subsystems

10.2.1 Online Help System

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

10.2.2 Device Editor

PSoC Designer has several main functions. The Device Editor subsystem lets the user select different onboard analog and digital component configurations for the PSoC blocks. PSoC Designer sets up power-on initialization tables for selected PSoC block configurations and creates source code for an application framework. The framework contains software to operate the selected components and, if the project uses more than one operating configuration, contains routines to switch between different sets of PSoC block configurations at runtime. PSoC Designer can print out a configuration sheet for given CSM 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.

10.2.3 Assembler

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

10.2.4 C Language Software Development

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

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

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

10.2.5 Debugger

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

10.3 Hardware Tools

10.3.1 In-Circuit Emulator

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

11 DC and AC Characteristics

Please note that the following data specifications are preliminary.

11.1 Absolute Maximum Ratings

Symbol	Absolute Maximum Ratings	Minimum	Typical	Maximum	Unit
	Storage Temperature	–65		+100 [1]	°C
	Ambient Temperature with Power Applied	–40		+85	°C
	Supply Voltage on V_{cc} Relative to V_{ss}	–0.5		+6.0	V
	DC Input Voltage	–0.5		$+V_{cc}$+0.5	V
	DC voltage Applied to Tri-state	V_{ss}–0.5		V_{cc}+0.5	V
	Maximum Output Current into Port Pins		60		mA
	Power Dissipation		[2]		mW
	Static Discharge Voltage	>2000			V
	Latch-up Current [1]	>200			mA

Notes:
1. Higher storage temperatures will degrade data retention
2. Package specific

11.1.1 Temperature Specifications During Programming

Symbol	Temperature Specifications	Minimum	Typical	Maximum	Unit
T_{ap}	Ambient Temperature During Programming	–40	24	+85	°C

11.2 DC Characteristics

11.2.1 DC Operating Specifications

Symbol	DC Operating Specifications	Minimum	Typical	Maximum	Unit
V_{cc}	Supply Voltage	3.0		5.5	V
I_{cc}	Supply Current			20 [1]	mA
I_{sb}	Standby Current			5 [2]	µA
V_{ref}	Reference Voltage	1.275		1.325	V
V_{il}	Input Low Voltage			0.8	V
V_{ih}	Input High Voltage	2.0			V
V_h	Hysterisis Voltage	0.02		0.1	V
V_{ol}	Output Low Voltage			V_{ss}+0.75 [3]	V
V_{oh}	Output High Voltage	V_{cc}-1.0 [4]			V
V_{rip}	[5]	2.0		2.4	V
R_{pu}	Pull Up Resistor Value	4500	5600	6900	Ω
R_{pd}	Pull Down Resistor Value	4500	5600	6900	Ω
I_{il}	Input Leakage	-41	0	33	nA
C_{in}	Capacitive Load on Pins as Input	0.5	1.7	5 [6]	pF
C_{out}	Capacitive Load on Pins as Output	0.5	1.7	5	pF

Notes:
1. 0°C and V_{cc} = 5.5V, 24.5MHz CPU, 8 Digital PSoC blocks at 48.5MHz, **no** IO sourcing current
2. Without crystal oscillator (preliminary specifications)
3. I_{sink} = 25 mA, V_{cc} = 4.5 V (maximum of 8 IO sinking)
4. I_{source} =10 mA, V_{cc} = 4.5V (maximum of 8 IO sourcing)
5. 0 °C to 70 °C
6. Package dependent

11.2.2 DC Analog PSoC Block Specifications

Symbol	DC Analog PSoC Block Specifications	Minimum	Typical	Maximum	Unit
	Input Offset Voltage	-15	0	15	mV
	Average Input Offset Voltage Drift			40	µV/°C
	Input Bias Current			1	nA
	Input Resistance - CT Block		100		MΩ
	Input Capacitance - CT Block		0.1	0.2	pF
	Input Resistance - SC Block (f = 0)		100		MΩ
	Effective input resistance = 1/(f x c)		461		KΩ
	Input Capacitance - SC Block		Variable	2	pF
	Input Voltage Range	0.5		V_{cc}-1	V
	Large Signal Voltage Gain	5	10		V/mV
	Max Output Voltage	0		V_{cc}	V
	Output Voltage Swing	0		V_{cc}	V
	Output Short Circuit Current	-140			µA
	Output Current Source Sink		 140		 µADC µADC
	Amplifier-to-Amplifier Coupling				dB
	Common-Mode Voltage Range	0.5		V_{cc}-1	VDC
	Common Mode Rejection Ratio		80		dB
	Differential Input Voltage			V_{cc}	VDC
	Supply Voltage Rejection Ratio		80		dB
	Supply Current Bias = Low Bias = Medium Bias = High		 50 120 400		 µA µA µA
	Resistor Unit Value	38	45	54	KΩ
	Capacitor Unit Value	60	70	80	fF
	Open Loop Gain				dB

11.2.3 DC Analog Input Pin with Multiplexer Specifications

Symbol	DC Analog Input Pin with Multiplexer Specifications	Minimum	Typical	Maximum	Unit
	Input Resistance	10			MΩ
	Input Capacitance	0.5	1.7	5	pF
	Bandwidth		1		MHz
	Input Voltage Range	0		V_{cc}	V
	Input-to-Input Coupling			-60	dB

11.2.4 DC Analog Input Pin to Switch Cap Block Specifications

Symbol	DC Analog Input Pin to SC Block Specifications	Minimum	Typical	Maximum	Unit
	Effective input resistance = 1/(f x c)		461		KΩ
	Input Capacitance (Mostly due to package)	0.5	Variable	7	pF
	Bandwidth		1		MHz
	Input Voltage Range	0		V_{cc}	V

11.2.5 DC Analog Output Buffer Specifications

Symbol	DC Analog Output Buffer Specifications	Minimum	Typical	Maximum	Unit
	Output Current Source Sink		 45[1] 45[1]		 mADC mADC
	Supply Current		0.5		mA
	Bandwidth		1		MHz
	Offset Voltage	-9	0	9	mV
	Output Resistance		1		Ω
	Output Voltage Swing	0.5		V_{cc}-1	V
	Output Short Circuit Current			>100	mA
	Output Buffer Offset Voltage Temperature Drift	-1	0	+1[2]	mV

Notes:
1. V_{cc} = 5V, swing around 2.5V of ± 1.3V
2. Over full temperature range

11.2.6 DC Switch Mode Pump Specifications

Symbol	DC Switch Mode Pump Specifications	Minimum	Typical	Maximum	Unit
	Output Voltage	3.0		4.9	V
	Output Current V_i = 1.5 V, V_o = 3.3 V V_i = 1.5 V, V_o = 4.9 V			 12 7	 mA mA
	Short Circuit Current (V_i = 3.3 V)		12		mA
	Input Voltage Range	1.0		3.3	V
	Input Current (Over V_i Range)	30		45	mA
	Startup Voltage	1.2			
	Output Voltage Tolerance (Over V_i Range)		5		$\%V_o$
	Line Regulation (Over V_i Range)		5		$\%V_o$
	Load Regulation		5		$\%V_o$
	Output Voltage Ripple (Depends on Capacitor)				mV_{pp}
	Transient Response 50% Load Change V_o Over/Undershoot		 1 10[2]		 µSec $\%V_o$
	Efficiency	60[3]		70	%
	Switching Frequency		1.3		MHz
	Switching Duty Cycle		50		%

Notes:
1. Transient delay/response is proportional to capacitor size
2. Larger cap reduces overshoot
3. For lighter loads, efficiency increases when a larger inductor is used

11.2.7 DC Programming Specifications

Symbol	DC Programming Specifications	Minimum	Typical	Maximum	Unit
V_{ccp}	V_{CC} for Programming	3.0	5	5.5	V
V_{cclv}	Low V_{cc} for Verify	3.0	3.1	3.2	V
V_{cchv}	High V_{cc} for Verify	5.3	5.4	5.5	V
I_{ccp}	Supply Current During Programming or Verify			20	mA
V_{ilp}	Input Low Voltage During Programming or Verify	-0.3	0	0.8	V
V_{ihp}	Input High Voltage During Programming or Verify	2.0	V_{cc}	V_{cc}+0.3	V
I_{ilp}	Input Current when Applying Vilp to P1[0] or P1[1] During Programming or Verify			0.2	mA
I_{ihp}	Input Current when Applying Vihp to P1[0] or P1[1] During Programming or Verify			1	
V_{olv}	Output Low Voltage During Programming or Verify			V_{ss}+0.75	V
V_{ohv}	Output High Voltage During Programming or Verify	V_{cc}-1.0		V_{cc}	V
I_{olv}	Current Sinking Capability for Output Low Voltage During Programming or Verify	10			mA
I_{ohv}	Current Sourcing Capability for Output High Voltage During Programming or Verify	5			mA
$Flash_{en}$	Flash Endurance	100,000			R/W Cycles
$Flash_{dr}$	Flash Data Retention	10			Years

11.3 AC Characteristics

11.3.1 AC Operating Specifications

Symbol	AC Operating Specifications	Minimum	Typical	Maximum	Unit
F_{max}	Operating Frequency	50 [1]			MHz
F_{cpu}	Processing Frequency				
T_f	Output Fall Time	25 [2]		100	
T_r	Output Rise Time	25 [2]		100	ns
F_{32K}	Internal Low Speed Oscillator Frequency	24 [2, 3]		40	kHz
F_{pllmin}			24		MHz
$T_{pllslew}$		1		10	ms
S_{Vdd}	V_{dd} Rise Time for Successful POR			100 [4]	ms
T_{os}	Oscillator Startup Timer Period		1	15 [5]	S
	External Reset Pulse Width	1			µs

Notes:

1. 100°C and V_{cc} = 4.5V (Digital PSoC blocks only)
2. V_{cc} 4.5 to 5.5 V, 0 °C to 100 °C
3. Only valid when not sleeping
4. To 3.0V, assuming linear ramp
5. Minimum and maximum times depend on settling precision

11.3.2 AC Analog PSoC Block Specifications

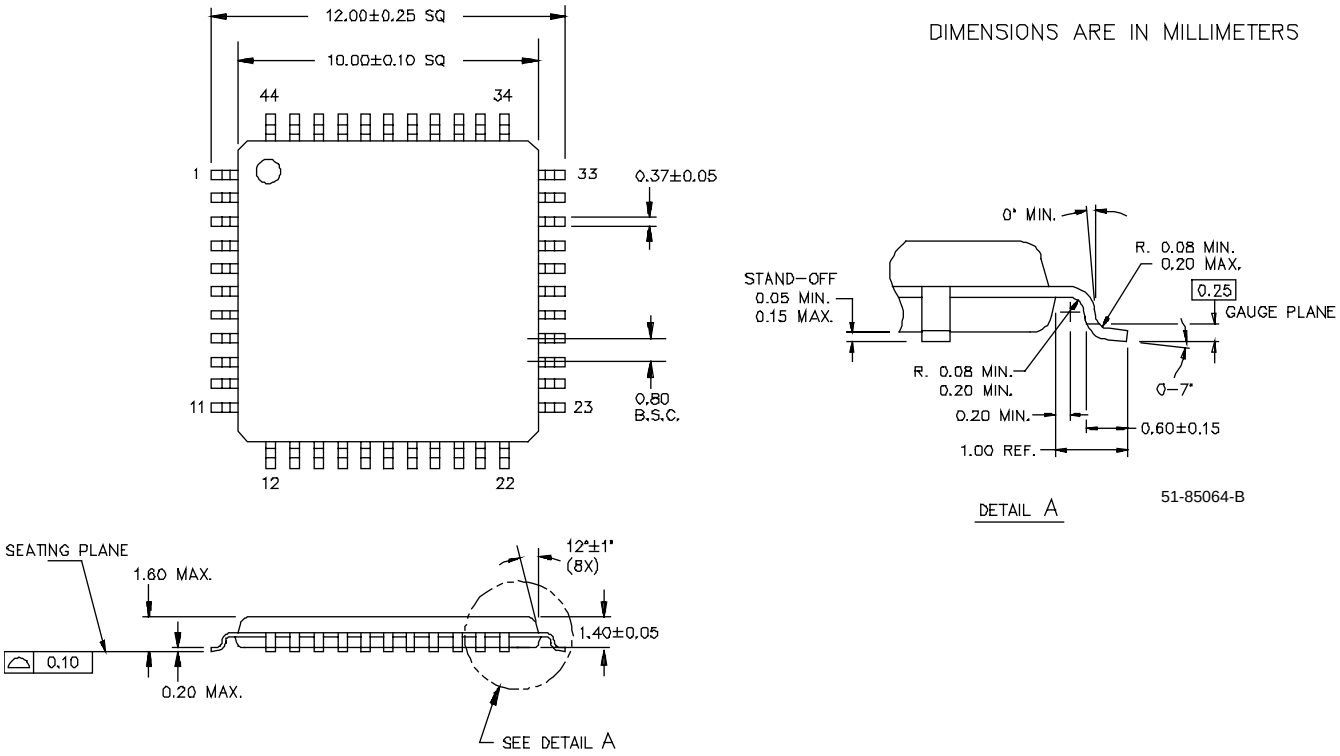
Symbol	AC Analog PSoC Block Specifications	Minimum	Typical	Maximum	Unit
	Transient Response Rise Time Overshoot				 µs %
	Settling Time to 0.1%				µs
	Gain Bandwidth Product Bias = Low Bias = Medium Bias = High		 3.5 9 15		 MHz MHz MHz
	Slew Rate - No load Bias = Low Bias = Medium Bias = High		 3.3 13 40		 V/µs V/µs V/µs
	Equivalent Input Noise Voltage (Follower configuration neglecting flicker)		20		nV/√¯Hz
	Equivalent Input Noise Current (MOS)		N/A		pA/√¯Hz
	Switch Capacitor Frequency	30		100	KHz

11.3.3 AC Programming Specifications

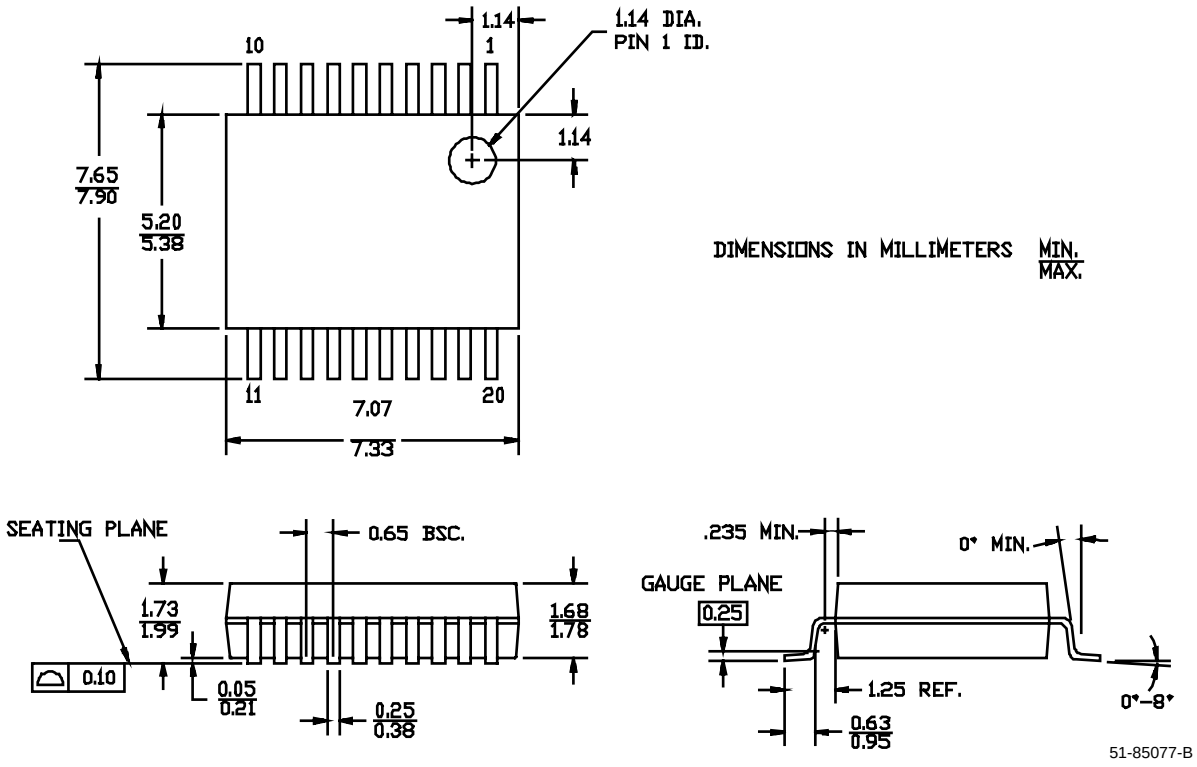
Symbol	AC Programming Specifications	Minimum	Typical	Maximum	Unit
T_{rsclk}	Rise Time of SCLK	1		20	ns
T_{fsclk}	Fall Time of SCLK	1		20	ns
T_{ssclk}	Data Set up Time to Rising Edge of SCLK	25			ns
T_{hsclk}	Data Hold Time from Rising Edge of SCLK	25			ns
F_{sclk}	Frequency of SCLK	0.2		20	MHz
T_{eraseb}	Flash Erase Time (Block)		10		ms
T_{erasef}	Flash Erase Time (Full)		40		ms
T_{write}	Flash Block Write Time		10		ms

12 Packaging Information

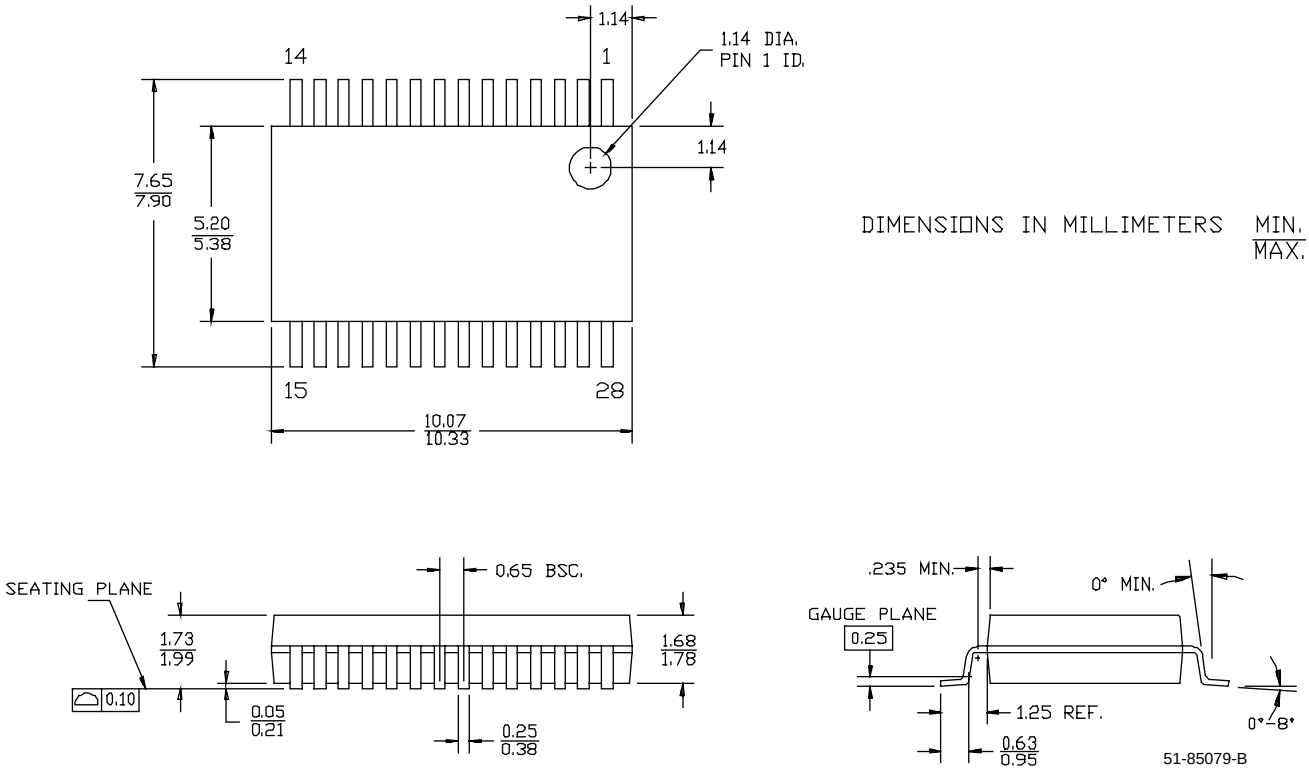
44-Lead Thin Plastic Quad Flat Pack A44



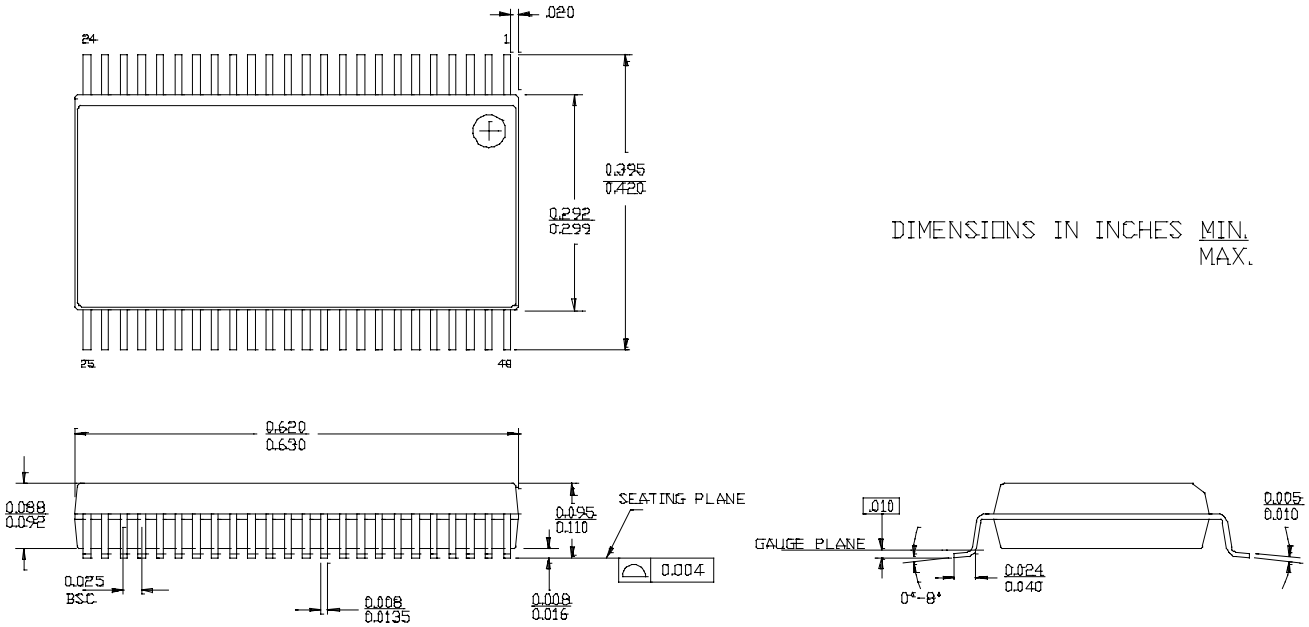
20-Pin Shrunk Small Outline Package O20



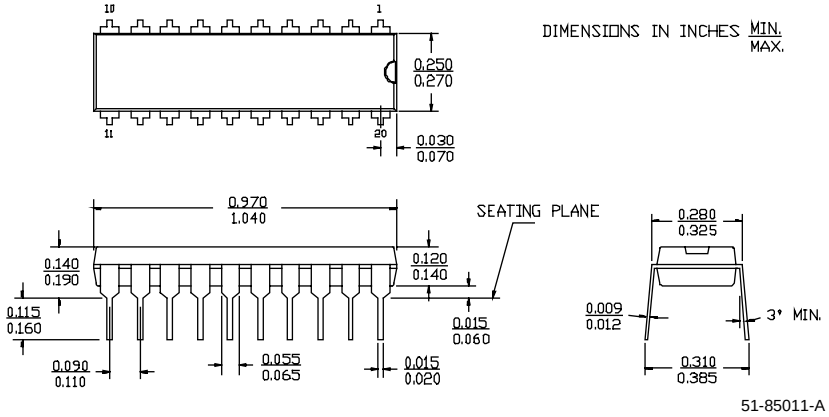
28-Lead (210-Mil) Shrunk Small Outline Package O28



48-Lead Shrunk Small Outline Package O48

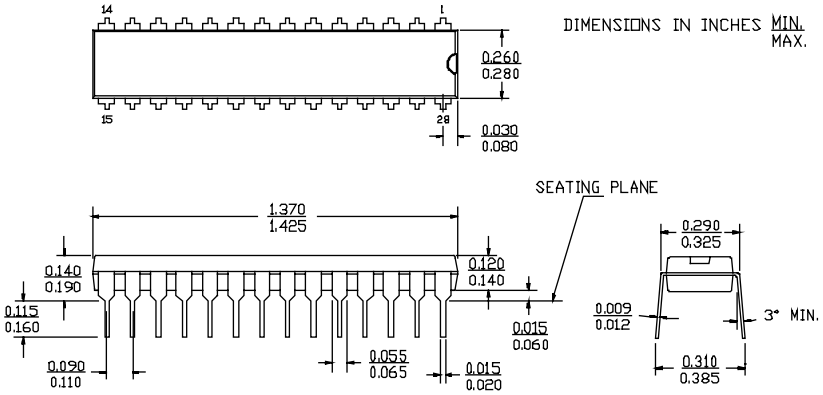


20-Lead (300-Mil) Molded DIP P5



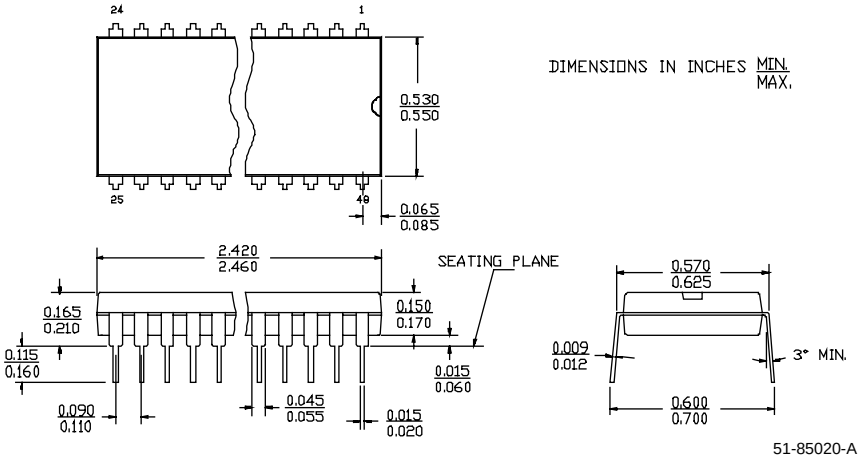
51-85011-A

28-Lead (300-Mil) Molded DIP P21



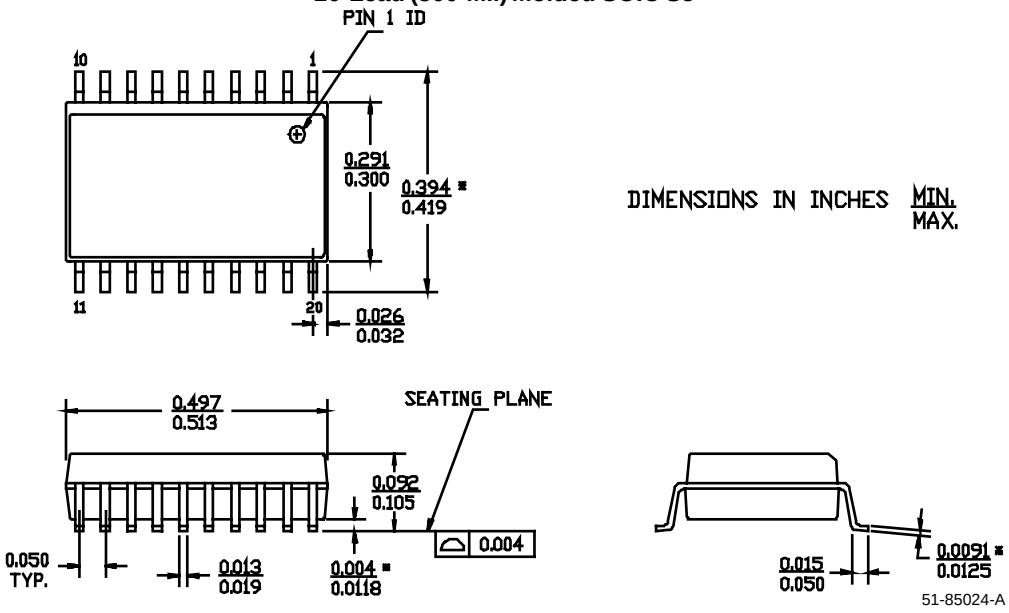
51-85014-B

48-Lead (600-Mil) Molded DIP P25

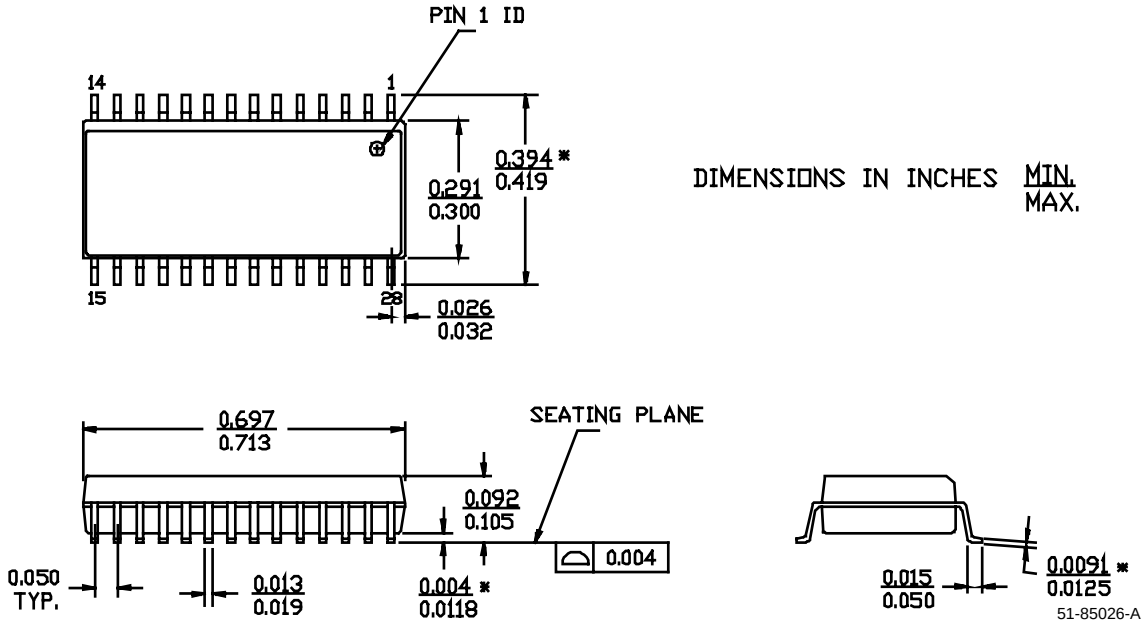


51-85020-A

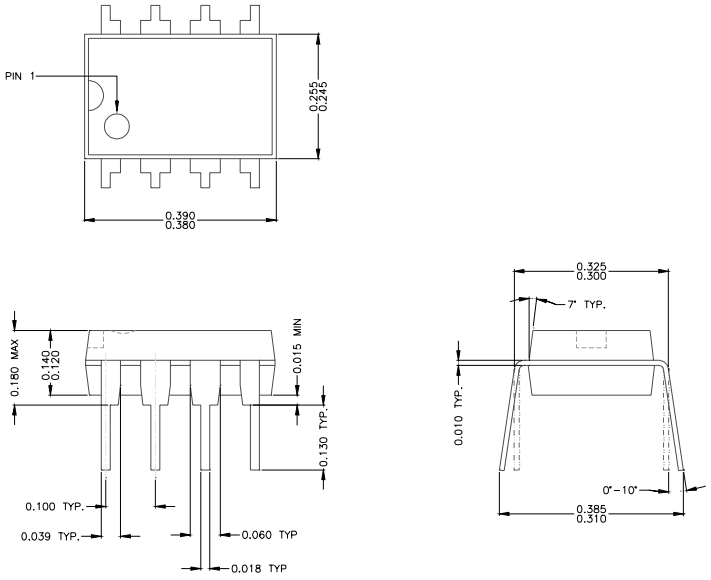
20-Lead (300-Mil) Molded SOIC S5



28-Lead (300-Mil) Molded SOIC S21



8-Lead (300-Mil) Molded DIP



13 Ordering Guide

Ordering Code	Flash Size (KBytes)	RAM Size (Bytes)	Switch Mode Pump	Package Name	Temperature Range
CY8C25122-24PI	4KB	128	No	8 PDIP	Ind. -40C to +85C
CY8C26233-24PI	8KB	256	Yes	P5	Ind. -40C to +85C
CY8C26233-24SI	8KB	256	Yes	S5	Ind. -40C to +85C
CY8C26233-24PVI	8KB	256	Yes	O20	Ind. -40C to +85C
CY8C26443-24PI	16KB	256	Yes	P21	Ind. -40C to +85C
CY8C26443-24SI	16KB	256	Yes	S21	Ind. -40C to +85C
CY8C26443-24PVI	16KB	256	Yes	O28	Ind. -40C to +85C
CY8C26643-24PI	16KB	256	Yes	P25	Ind. -40C to +85C
CY8C26643-24PVI	16KB	256	Yes	O48	Ind. -40C to +85C
CY8C26643-24AI	16KB	256	Yes	A44	Ind. -40C to +85C

Package Name	Package Type
8 PDIP	8 Pin (300 Mil) Molded DIP
P5	20 Pin (300 Mil) Molded DIP
S5	20 Pin (300 Mil) Molded SOIC
O20	20 Pin (210 Mil) Shrunk Small Outline Package
P21	28 Pin (300 Mil) Molded DIP
S21	28 Pin (300 Mil) Molded SOIC
O28	28 Pin (210 Mil) Shrunk Small Outline Package
P25	48 Pin (600 Mil) Molded DIP
O48	48 Pin (300 Mil) Shrunk Small Outline Package
A44	44 Pin Thin Plastic Quad Flatpack