Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

PSoC® 5: CY8C53 Family Datasheet

Programmable System-on-Chip (PSoC®)

General Description

With its unique array of configurable blocks, PSoC® 5 is a true system level solution providing microcontroller unit (MCU), memory, analog, and digital peripheral functions in a single chip. The CY8C53 family offers a modern method of signal acquisition, signal processing, and control with high accuracy, high bandwidth, and high flexibility. Analog capability spans the range from thermocouples (near DC voltages) to ultrasonic signals. The CY8C53 family can handle dozens of data acquisition channels and analog inputs on every GPIO pin. The CY8C53 family is also a high-performance configurable digital system with some part numbers including interfaces such as USB and multi-master I^2C. In addition to communication interfaces, the CY8C53 family has an easy to configure logic array, flexible routing to all I/O pins, and a high-performance 32-bit ARM® Cortex™-M3 microprocessor core. Designers can easily create system-level designs using a rich library of prebuilt components and boolean primitives using PSoC® Creator™, a hierarchical schematic design entry tool. The CY8C53 family provides unparalleled opportunities for analog and digital bill of materials integration while easily accommodating last minute design changes through simple firmware updates.

Features

- 32-bit ARM Cortex-M3 CPU core
  - DC to 67 MHz operation
  - Flash program memory, up to 256 KB, 100,000 write cycles, 20-year retention, multiple security features
  - Up to 64 KB SRAM memory
  - 128 bytes of cache memory
  - 2-KB electrically erasable programmable read-only memory (EEPROM) memory, 1 million cycles, and 20 years retention
  - 24-channel direct memory access (DMA) with multilayer AMBA high-performance bus (AHB) bus access
    - Programmable chained descriptors and priorities
    - High bandwidth 32-bit transfer support
- Low voltage, ultra low power
  - Operating voltage range: 2.7 V to 5.5 V
  - 6 mA at 6 MHz
  - Low power modes including:
    - 2-µA sleep mode
    - 300-nA hibernate mode with RAM retention
- Versatile I/O system
  - 46 to 70 I/Os (60 GPIOs, 8 SIOs, 2 USBIOs))
  - Any GPIO to any digital or analog peripheral routability
  - LCD direct drive from any GPIO, up to 46x16 segments
  - CapSense® support from any GPIO[1]
  - 1.2 V to 5.5 V I/O interface voltages, up to 4 domains
  - Maskable, independent IRQ on any pin or port
  - Schmitt-trigger transistor-transistor logic (TTL) inputs
  - All GPIOs configurable as open drain high/low, pull-up/pull-down, High-Z, or strong output
  - 25 mA sink on SIO
- Digital peripherals
  - 20 to 24 programmable logic device (PLD) based universal digital blocks (UDBs)
  - Full-Speed (FS) USB 2.0 12 Mbps using a 24 MHz external oscillator
  - Up to four 16-bit configurable timer, counter, and PWM blocks
  - Library of standard peripherals
    - 8-, 16-, 24-, and 32-bit timers, counters, and PWMs
    - SPI, UART, and I^2C
    - Many others available in catalog
  - Library of advanced peripherals
    - Cyclic redundancy check (CRC)
    - Pseudo random sequence (PRS) generator
    - Local interconnect network (LIN) bus 2.0
    - Quadrature decoder
- Analog peripherals ($2.7\ V \le V_{DDA} \le 5.5\ V$)
  - 1.024 V ±1% internal voltage reference
  - Successive approximation register (SAR) analog-to-digital converter (ADC), 12-bit at 700 ksps
  - Two 8-bit 5.5 Msps current digital-to-analog converters (DAC) (IDACs) or 1 Msps voltage DACs (VDACs)
  - Four comparators with 95-ns response time
  - Two uncommitted opamps with 10-mA drive capability
  - Two configurable multifunction analog blocks. Example configurations are programmable gain amplifier (PGA), transimpedance amplifier (TIA), mixer, and sample and hold
  - CapSense support
- Programming, debug, and trace
  - Serial wire debug (SWD) and single-wire viewer (SWV) interfaces
  - Cortex-M3 flash patch and breakpoint (FPB) block
  - Cortex-M3 data watchpoint and trace (DWT) generates data trace information
  - Cortex-M3 instrumentation trace macrocell (ITM) can be used for printf-style debugging
  - DWT and ITM blocks communicate with off-chip debug and trace systems via the SWV interface
  - Bootloader programming supportable through I^2C, SPI, UART, USB, and other interfaces
- Precision, programmable clocking
  - 3 to 48 MHz internal oscillator over full temperature and voltage range
  - 4 to 25 MHz crystal oscillator for crystal PPM accuracy
  - Internal PLL clock generation up to 67 MHz
  - 32.768 kHz watch crystal oscillator
  - Low power internal oscillator at 1, 33, and 100 kHz
- Temperature and packaging
  - -40°C to +85°C degrees industrial temperature
  - 68-pin QFN and 100-pin TQFP package options

Note
1. GPIOs with opamp outputs are not recommended for use with CapSense.

Contents

1. Architectural Overview 3
2. Pinouts 5
3. Pin Descriptions 9
4. CPU 10
4.1 ARM Cortex-M3 CPU 10
4.2 Cache Controller 12
4.3 DMA and PHUB 12
4.4 Interrupt Controller 14
5. Memory 16
5.1 Static RAM 16
5.2 Flash Program Memory 16
5.3 Flash Security 16
5.4 EEPROM 16
5.5 Memory Map 17
6. System Integration 18
6.1 Clocking System 18
6.2 Power System 22
6.3 Reset 24
6.4 I/O System and Routing 25
7. Digital Subsystem 32
7.1 Example Peripherals 32
7.2 Universal Digital Block 36
7.3 UDB Array Description 39
7.4 DSI Routing Interface Description 39
7.5 USB 41
7.6 Timers, Counters, and PWMs 41
7.7 I^2C 43
8. Analog Subsystem 44
8.1 Analog Routing 45
8.2 Successive Approximation ADC 47
8.3 Comparators 47
8.4 Opamps 49
8.5 Programmable SC/CT Blocks 49
8.6 LCD Direct Drive 50
8.7 CapSense 51
8.8 Temp Sensor 51
8.9 DAC 51
8.10 Up/Down Mixer 52
8.11 Sample and Hold 53
9. Programming, Debug Interfaces, Resources 53
9.1 Debug Port Acquisition 54
9.2 SWD Interface 54
9.3 Debug Features 55
9.4 Trace Features 55
9.5 SWV Interface 55
9.6 Programming Features 55
9.7 Device Security 55
10. Development Support 56
10.1 Documentation 56
10.2 Online 56
10.3 Tools 56
11. Electrical Specifications 57
11.1 Absolute Maximum Ratings 57
11.2 Device Level Specifications 58
11.3 Power Regulators 60
11.4 Inputs and Outputs 61
11.5 Analog Peripherals 68
11.6 Digital Peripherals 85
11.7 Memory 88
11.8 PSoC System Resources 90
11.9 Clocking 92
12. Ordering Information 96
12.1 Part Numbering Conventions 97
13. Packaging 98
14. Acronyms 100
15. Reference Documents 101
16. Document Conventions 102
16.1 Units of Measure 102
17. Revision History 103
18. Sales, Solutions, and Legal Information 105

1. Architectural Overview

Introducing the CY8C53 family of ultra low-power, flash Programmable System-on-Chip (PSoC®) devices, part of a scalable 8-bit PSoC 3 and 32-bit PSoC 5 platform. The CY8C53 family provides configurable blocks of analog, digital, and interconnect circuitry around a CPU subsystem. The combination of a CPU with a flexible analog subsystem, digital subsystem, routing, and I/O enables a high level of integration in a wide variety of consumer, industrial, and medical applications.

Figure 1-1. Simplified Block Diagram

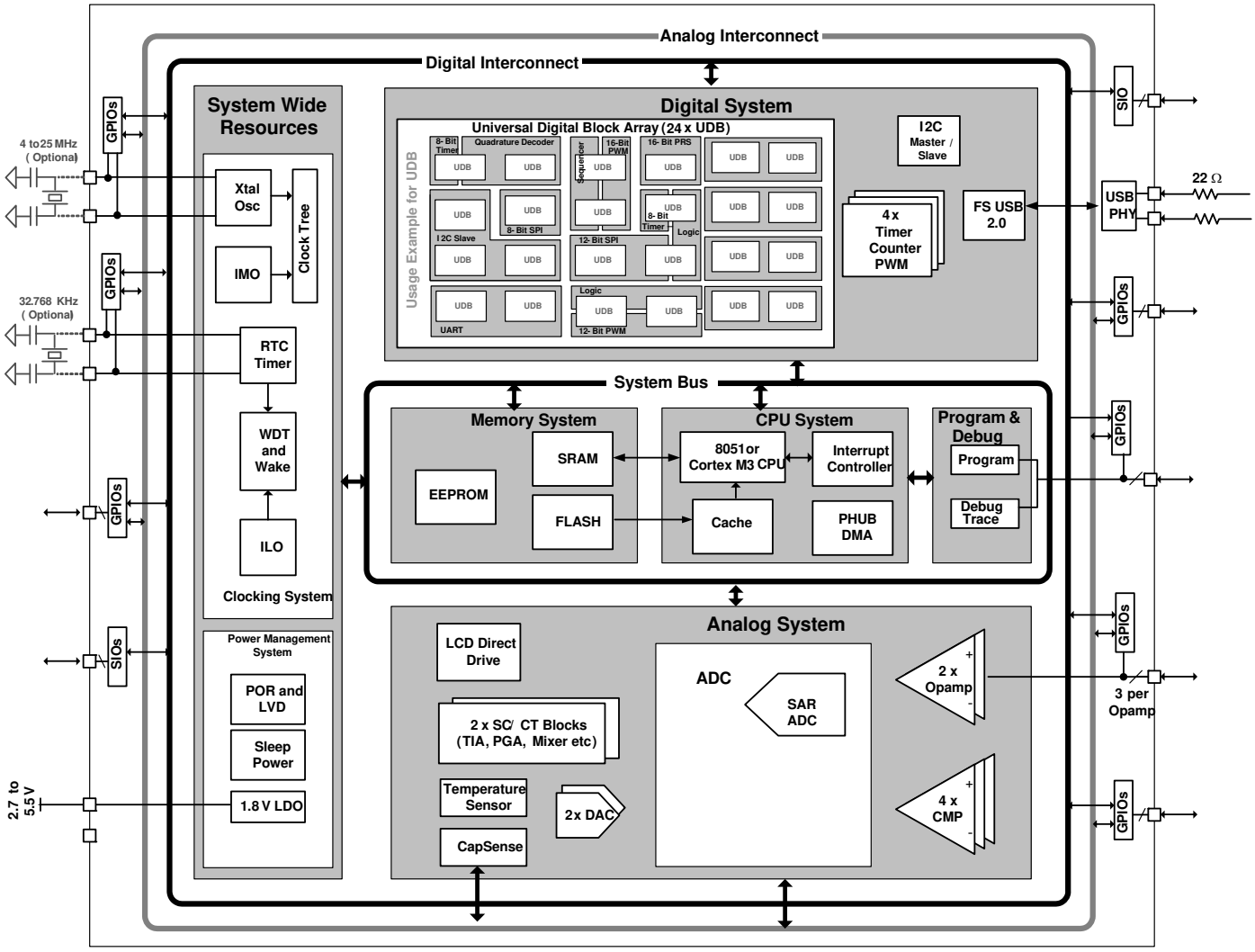


Figure 1-1 illustrates the major components of the CY8C53 family. They are:

- ARM Cortex-M3 CPU subsystem
- Nonvolatile subsystem
- Programming, debug, and test subsystem
- Inputs and outputs
- Clocking
- Power
- Digital subsystem
- Analog subsystem

PSoC's digital subsystem provides half of its unique configurability. It connects a digital signal from any peripheral to any pin through the digital system interconnect (DSI). It also provides functional flexibility through an array of small, fast, low-power UDBs. PSoC Creator provides a library of pre-built and tested standard digital peripherals (UART, SPI, LIN, PRS, CRC, timer, counter, PWM, AND, OR, and so on) that are mapped to the UDB array. The designer can also easily create a digital circuit using boolean primitives by means of graphical design entry. Each UDB contains programmable array logic (PAL)/programmable logic device (PLD) functionality, together with a small state machine engine to support a wide variety of peripherals.

In addition to the flexibility of the UDB array, PSoC also provides configurable digital blocks targeted at specific functions. For the CY8C53 family these blocks can include four 16-bit timer, counter, and PWM blocks; I^2C slave, master, and multimaster and Full-Speed USB.

For more details on the peripherals see the "Example Peripherals" section on page 32 of this data sheet. For information on UDBs, DSI, and other digital blocks, see the "Digital Subsystem" section on page 32 of this data sheet.

PSoC's analog subsystem is the second half of its unique configurability. All analog performance is based on a highly accurate absolute voltage reference with less than 1% error over temperature and voltage. The configurable analog subsystem includes:

- Analog muxes
- Comparators
- Analog mixers
- Voltage references
- Analog-to-Digital Converters (ADC)
- Digital-to-Analog Converters (DACs)

All GPIO pins can route analog signals into and out of the device using the internal analog bus. This allows the device to interface up to 62 discrete analog signals.

The CY8C53 family also offers a SAR ADC. Featuring 12-bit conversions at up to 700 k samples per second, it also offers low nonlinearity and offset errors. It is well suited for a variety of higher speed analog applications.

Two high speed voltage or current DACs support 8-bit output signals at an update rate of up to 5.5 Msps. They can be routed out of any GPIO pin. You can create higher resolution voltage DAC outputs using the UDB array. This can be used to create a pulse width modulated (PWM) DAC of up to 10 bits, at up to 48 kHz. The digital DACs in each UDB support PWM, PRS, or delta-sigma algorithms with programmable widths.

In addition to the ADC and DACs, the analog subsystem provides multiple:

- Comparators
- Uncommitted opamps
- Configurable switched capacitor/continuous time (SC/CT) blocks. These support:
  - Transimpedance amplifiers
  - Programmable gain amplifiers
  - Mixers
  - Other similar analog components

See the "Analog Subsystem" section on page 44 of this data sheet for more details.

PSoC's CPU subsystem is built around a 32-bit three-stage pipelined ARM Cortex-M3 processor running at up to 67 MHz. The Cortex-M3 includes a tightly integrated nested vectored interrupt controller (NVIC) and various debug and trace modules. The overall CPU subsystem includes a DMA controller, cache, and interrupt controller. The NVIC provides low latency, nested interrupts, and tail-chaining of interrupts and other features to increase the efficiency of interrupt handling. The DMA controller enables peripherals to exchange data without CPU involvement. This allows the CPU to run slower (saving power) or use those CPU cycles to improve the performance of firmware algorithms. The presence of cache improves the access speed of instructions by the CPU.

PSoC's nonvolatile subsystem consists of flash and byte-writeable EEPROM. It provides up to 256 KB of on-chip flash. The CPU can reprogram individual blocks of flash, enabling boot loaders. A powerful and flexible protection model secures the user's sensitive information, allowing selective memory block locking for read and write protection. Two KB of byte-writable EEPROM is available on-chip to store application data.

The three types of PSoC I/O are extremely flexible. All I/Os have many drive modes that are set at POR. PSoC also provides up to four I/O voltage domains through the V_{DDIO} pins. Every GPIO has analog I/O, LCD drive, flexible interrupt generation, slew rate control, and digital I/O capability. The SIOs on PSoC allow V_{OH} to be set independently of V_{DDIO} when used as outputs. When SIOs are in input mode they are high impedance. This is true even when the device is not powered or when the pin voltage goes above the supply voltage. This makes the SIO ideally suited for use on an I^2C bus where the PSoC may not be powered when other devices on the bus are. The SIO pins also have high current sink capability for applications such as LED drives. The programmable input threshold feature of the SIO can be used to make the SIO function as a general purpose analog comparator. For devices with Full-Speed USB the USB physical interface is also provided (USBIO). When not using USB these pins may also be used for limited digital functionality and device programming. All the features of the PSoC I/Os are covered in detail in the "I/O System and Routing" section on page 25 of this data sheet.

The PSoC device incorporates flexible internal clock generators, designed for high stability and factory trimmed for high accuracy. The internal main oscillator (IMO) is the master clock base for the system, and has 5% accuracy at 3 MHz. The IMO can be configured to run from 3 MHz up to 48 MHz. Multiple clock derivatives can be generated from the main clock frequency to meet application needs. The device provides a PLL to generate system clock frequencies up to 67 MHz from the IMO, external crystal, or external reference clock. It also contains a separate, very low-power internal low speed oscillator (ILO) for the sleep and watchdog timers. A 32.768 kHz external watch crystal is also supported for use in real time clock (RTC) applications. The clocks, together with programmable clock dividers, provide the flexibility to integrate most timing requirements.

The CY8C53 family supports a wide supply operating range from 2.7 to 5.5 V. This allows operation from regulated supplies such as 3.3 V ± 10% or 5.0 V ± 10%, or directly from a wide range of battery types.

PSoC supports a wide range of low power modes. These include a 300-nA hibernate mode with RAM retention and a 2-µA sleep mode.

Power to all major functional blocks, including the programmable digital and analog peripherals, can be controlled independently by firmware. This allows low-power background processing when some peripherals are not in use. This, in turn, provides a total device current of only 6 mA when the CPU is running at 6 MHz.

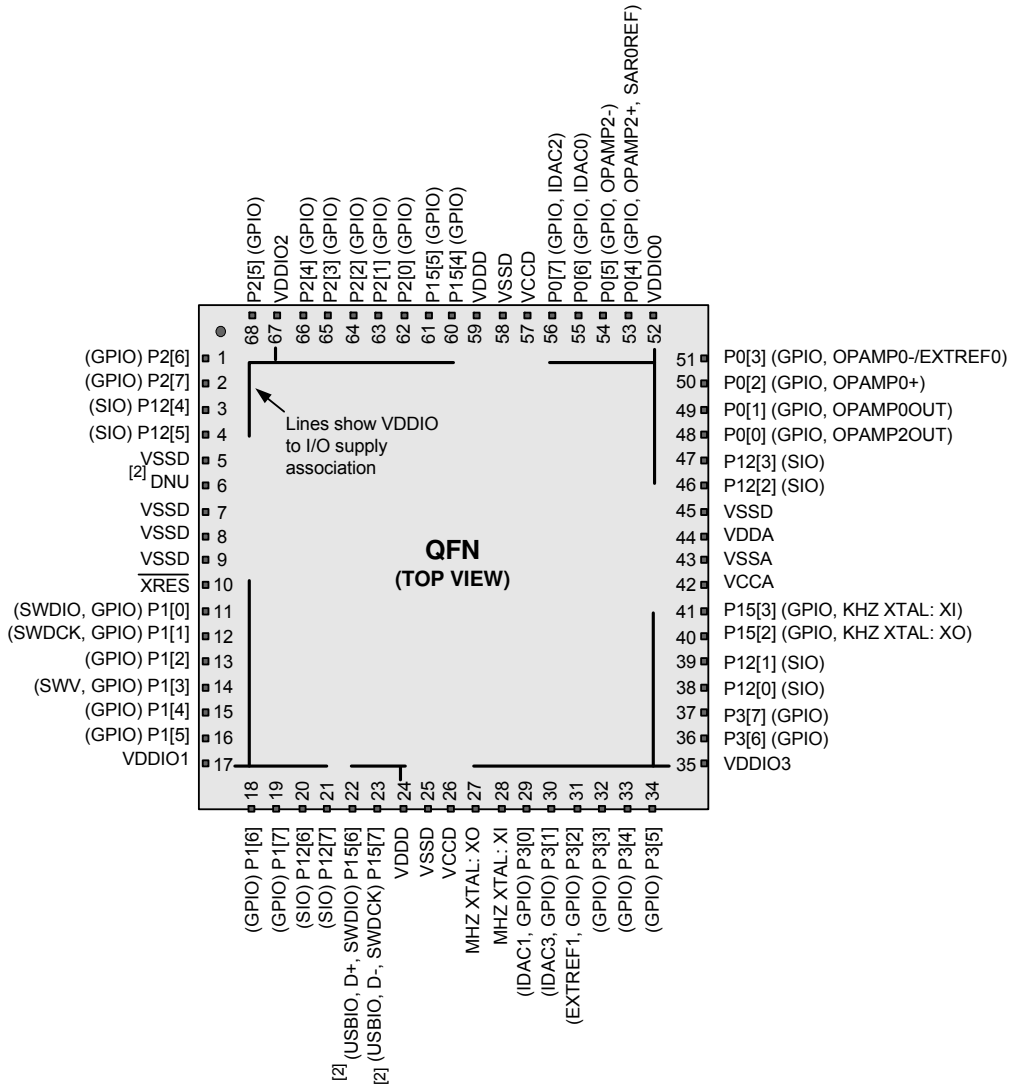
The details of the PSoC power modes are covered in the "Power System" section on page 22 of this data sheet.

PSoC uses a SWD interface for programming, debug, and test. Using this standard interface enables the designer to debug or program the PSoC with a variety of hardware solutions from Cypress or third party vendors. The Cortex-M3 debug and trace modules include FPB, DWT, and ITM. These modules have many features to help solve difficult debug and trace problems. Details of the programming, test, and debugging interfaces are discussed in the "Programming, Debug Interfaces, Resources" section on page 53 of this data sheet.

2. Pinouts

The VDDIO pin that supplies a particular set of pins is indicated by the black lines drawn on the pinout diagrams in Figure 2-1 and Figure 2-2. Using the VDDIO pins, a single PSoC can support multiple interface voltage levels, eliminating the need for off-chip level shifters. Each VDDIO may sink up to 20 mA total to its associated I/O pins and opamps, and each set of VDDIO associated pins may sink up to 100 mA.

Figure 2-1. 68-Pin QFN Part Pinout[3]



Notes
2. Pins labeled Do Not Use (DNU) must be left floating. USB pins on devices without USB are DNU.
3. The center pad on the QFN package should be connected to digital ground (V_{SSD}) for best mechanical, thermal, and electrical performance. If not connected to ground, it should be electrically floated and not connected to any other signal.

Figure 2-2. 100-Pin TQFP Part Pinout

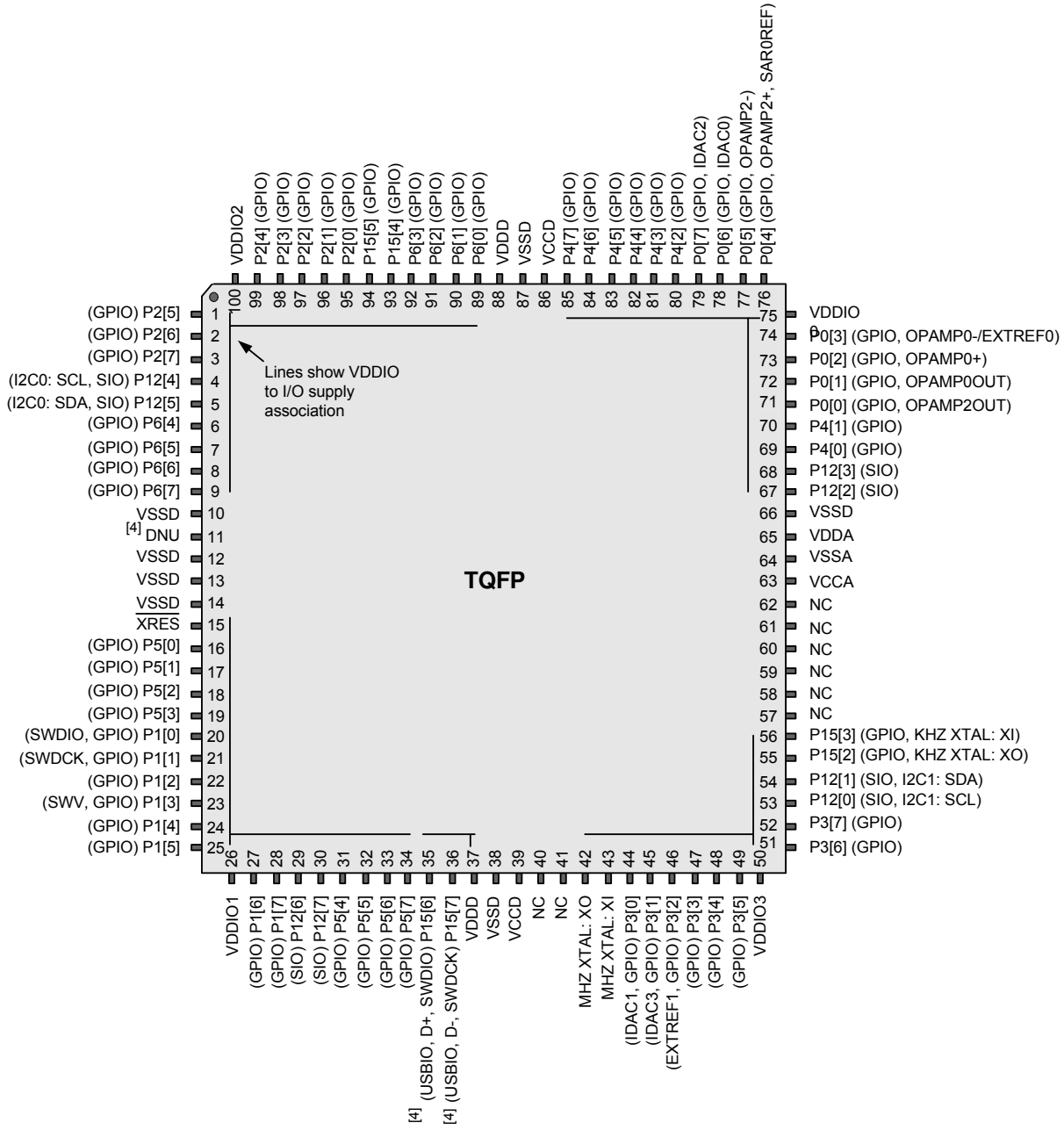


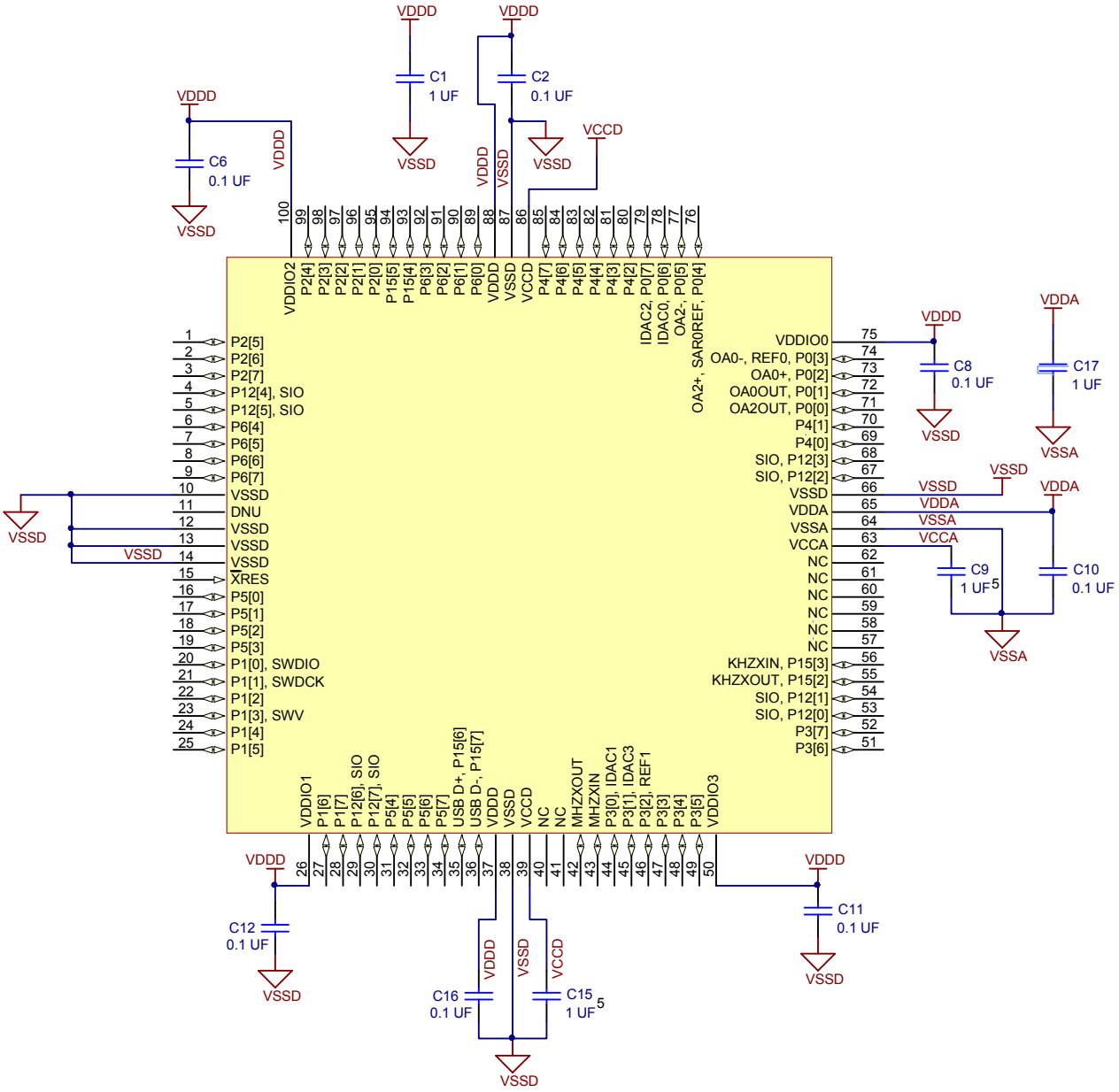
Figure 2-3 and Figure 2-4 on page 9 show an example schematic and an example PCB layout, for the 100-pin TQFP part, for optimal analog performance on a 2-layer board.

- The two pins labeled V_{DDD} must be connected together.
- The two pins labeled V_{CCD} must be connected together, with capacitance added, as shown in Figure 2-3 and Power System on page 22. The trace between the two V_{CCD} pins should be as short as possible.
- The two pins labeled V_{SSD} must be connected together.

Note
4. Pins labeled Do Not Use (DNU) must be left floating. USB pins on devices without USB are DNU.

For information on circuit board layout issues for mixed signals, refer to the application note AN57821 - Mixed Signal Circuit Board Layout Considerations for PSoC® 3 and PSoC 5.

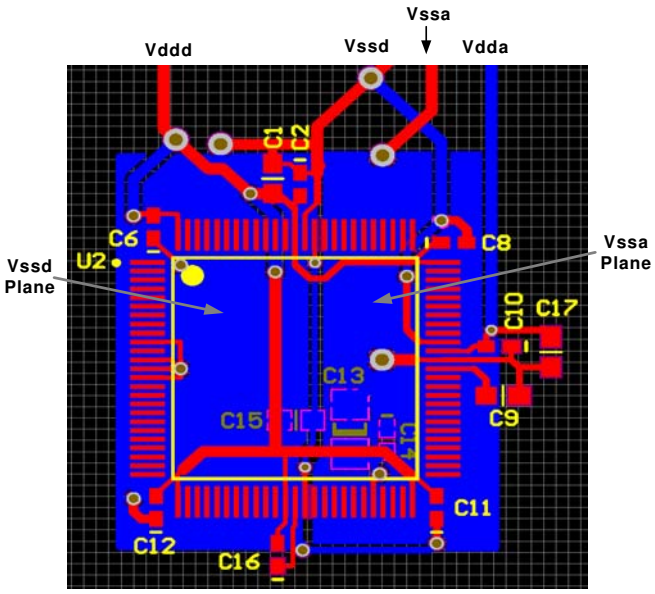
Figure 2-3. Example Schematic for 100-Pin TQFP Part with Power Connections



Note The two V_{CCD} pins must be connected together with as short a trace as possible. A trace under the device is recommended, as shown in Figure 2-4.

Note
5. 10 µF is required for sleep mode. See Table 11-3.

Figure 2-4. Example PCB Layout for 100-Pin TQFP Part for Optimal Analog Performance



3. Pin Descriptions

IDAC0, IDAC2. Low resistance output pin for high current DACs (IDAC).

OpAmp0out, OpAmp2out. High current output of uncommitted opamp[6].

Extref0, Extref1. External reference input to the analog system.

OpAmp0-, OpAmp2-. Inverting input to uncommitted opamp.

OpAmp0+, OpAmp2+. Noninverting input to uncommitted opamp.

SAR0ref. External reference for SAR ADC.

GPIO. General purpose I/O pin provides interfaces to the CPU, digital peripherals, analog peripherals, interrupts, LCD segment drive, and CapSense[6].

kHz XTAL: Xo, kHz XTAL: Xi. 32.768 kHz crystal oscillator pin.

MHz XTAL: Xo, MHz XTAL: Xi. 4 to 25 MHz crystal oscillator pin. If a crystal is not used then Xi must be shorted to ground and Xo must be left floating.

SIO. Special I/O provides interfaces to the CPU, digital peripherals and interrupts with a programmable high threshold voltage, analog comparator, high sink current, and high impedance state when the device is unpowered.

SWDCK. Serial Wire Debug Clock programming and debug port connection. When programming and debugging using SWD is done over USBIOs, the SWDCK pin of port P1[1] is not available for use as a general purpose I/O and should be externally pulled down using a resistor of less than 100 KΩ.

SWDIO. Serial Wire Debug Input and Output programming and debug port connection.

SWV. Single Wire Viewer output.

USBIO, D+. Provides D+ connection directly to a USB 2.0 bus. May be used as a digital I/O pin; it is powered from V_{DDD} instead of from a V_{DDIO}. Pins are Do Not Use (DNU) on devices without USB.

USBIO, D-. Provides D- connection directly to a USB 2.0 bus. May be used as a digital I/O pin; it is powered from V_{DDD} instead of from a V_{DDIO}. Pins are DNU on devices without USB.

V_{CCA}. Output of analog core regulator and input to analog core. Requires a 1 µF capacitor to V_{SSA} (10 µF is required for sleep mode. See Table 11-3). Regulator output not for external use.

V_{CCD}. Output of digital core regulator and input to digital core. The two V_{CCD} pins must be shorted together, with the trace between them as short as possible, and a 1 µF capacitor to V_{SSD} (10 µF is required for sleep mode. See Table 11-3); see Power System on page 22. Regulator output not for external use.

V_{DDA}. Supply for all analog peripherals and analog core regulator. **V_{DDA} must be the highest voltage present on the device. All other supply pins must be less than or equal to V_{DDA}.**[7]

V_{DDD}. Supply for all digital peripherals and digital core regulator. V_{DDD} must be less than or equal to V_{DDA}.[7]

V_{SSA}. Ground for all analog peripherals.

V_{SSD}. Ground for all digital logic and I/O pins.

V_{DDIO0}, V_{DDIO1}, V_{DDIO2}, V_{DDIO3}. Supply for I/O pins. Each V_{DDIO} must be tied to a valid operating voltage (2.7 V to 5.5 V), and must be less than or equal to V_{DDA}.

$\overline{XRES}$. External reset pin. Active low with internal pull-up.

RSVD. Reserved pins.

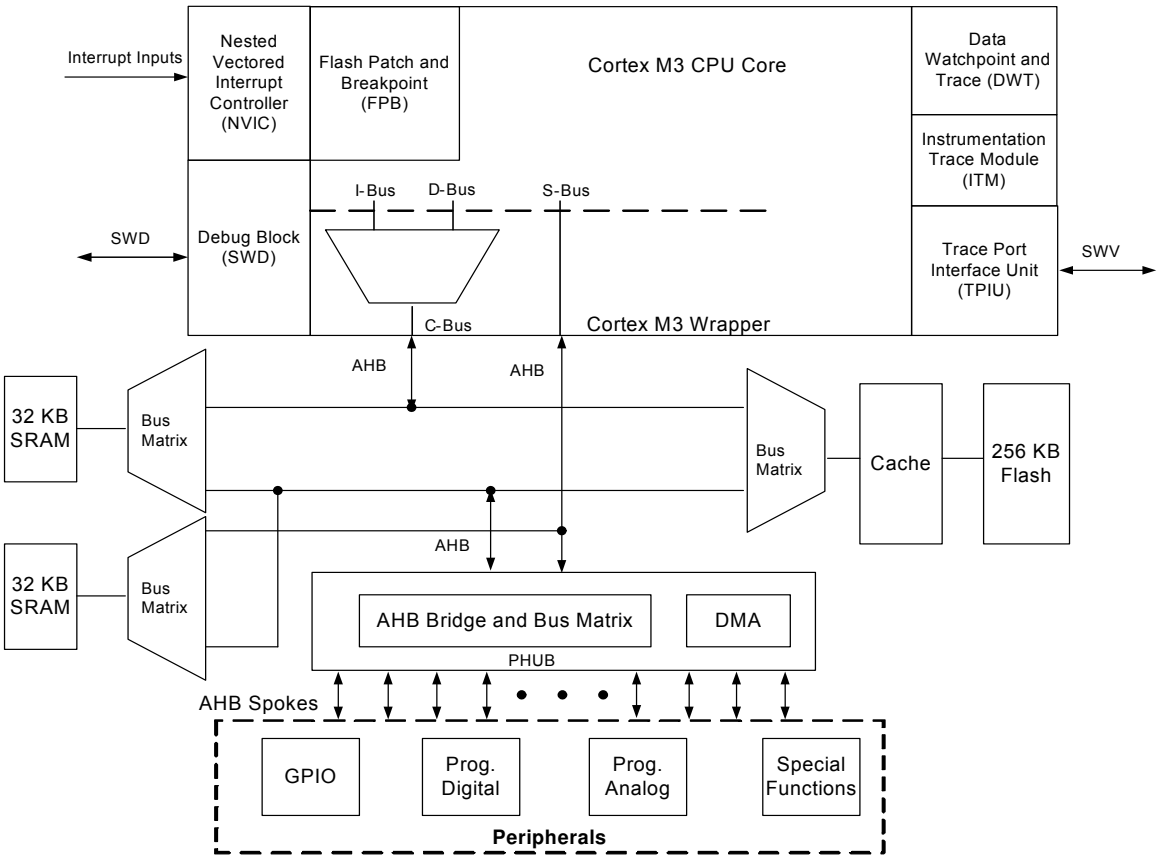
Notes
6. GPIOs with opamp outputs are not recommended for use with CapSense.
7. V_{DDD} and V_{DDA} must be brought up in synchronization with each other, that is, at the same rates and levels. V_{DDA} must be greater than or equal to all other supplies.

4. CPU

4.1 ARM Cortex-M3 CPU

The CY8C53 family of devices has an ARM Cortex-M3 CPU core. The Cortex-M3 is a low power 32-bit three-stage pipelined Harvard architecture CPU that delivers 1.25 DMIPS/MHz. It is intended for deeply embedded applications that require fast interrupt handling features.

Figure 4-1. ARM Cortex-M3 Block Diagram



The Cortex-M3 CPU subsystem includes these features:

- ARM Cortex-M3 CPU
- Programmable NVIC, tightly integrated with the CPU core
- Full-featured debug and trace module, tightly integrated with the CPU core
- Up to 128 KB of flash memory, 2 KB of EEPROM, and 32 KB of SRAM
- Cache controller with 128 bytes of memory
- Peripheral HUB (PHUB)
- DMA controller

4.1.1 Cortex-M3 Features

The Cortex-M3 CPU features include:

- 4-GB address space. Predefined address regions for code, data, and peripherals. Multiple buses for efficient and simultaneous accesses of instructions, data, and peripherals.
- The Thumb®-2 instruction set, which offers ARM-level performance at Thumb-level code density. This includes 16-bit and 32-bit instructions. Advanced instructions include:
  - Bit-field control
  - Hardware multiply and divide
  - Saturation
  - If-Then
  - Wait for events and interrupts
  - Exclusive access and barrier
  - Special register access

  The Cortex-M3 does not support ARM instructions.

- Bit-band support for the SRAM region. Atomic bit-level write and read operations for SRAM addresses.
- Unaligned data storage and access. Contiguous storage of data of different byte lengths.
- Operation at two privilege levels (privileged and user) and in two modes (thread and handler). Some instructions can only be executed at the privileged level. There are also two stack pointers: Main (MSP) and Process (PSP). These features support a multitasking operating system running one or more user-level processes.
- Extensive interrupt and system exception support.

4.1.2 Cortex-M3 Operating Modes

The Cortex-M3 operates at either the privileged level or the user level, and in either the thread mode or the handler mode. Because the handler mode is only enabled at the privileged level, there are actually only three states, as shown in Table 4-1.

Table 4-1. Operational Level

Condition	Privileged	User
Running an exception	Handler mode	Not used
Running main program	Thread mode	Thread mode

At the user level, access to certain instructions, special registers, configuration registers, and debugging components is blocked. Attempts to access them cause a fault exception. At the privileged level, access to all instructions and registers is allowed.

The processor runs in the handler mode (always at the privileged level) when handling an exception, and in the thread mode when not.

4.1.3 CPU Registers

The Cortex-M3 CPU registers are listed in Table 4-2. Registers R0-R15 are all 32 bits wide.

Table 4-2. Cortex M3 CPU Registers

Register	Description
R0-R12	General purpose registers R0-R12 have no special architecturally defined uses. Most instructions that specify a general purpose register specify R0-R12. ■ Low Registers: Registers R0-R7 are accessible by all instructions that specify a general purpose register. ■ High Registers: Registers R8-R12 are accessible by all 32-bit instructions that specify a general purpose register; they are not accessible by all 16-bit instructions.

Table 4-2. Cortex M3 CPU Registers (continued)

Register	Description
R13	R13 is the stack pointer register. It is a banked register that switches between two 32-bit stack pointers: the Main Stack Pointer (MSP) and the Process Stack Pointer (PSP). The PSP is used only when the CPU operates at the user level in thread mode. The MSP is used in all other privilege levels and modes. Bits[0:1] of the SP are ignored and considered to be 0, so the SP is always aligned to a word (4 byte) boundary.
R14	R14 is the Link Register (LR). The LR stores the return address when a subroutine is called.
R15	R15 is the Program Counter (PC). Bit 0 of the PC is ignored and considered to be 0, so instructions are always aligned to a half word (2 byte) boundary.
xPSR	The Program status registers are divided into three status registers, which are accessed either together or separately: ■ Application Program Status Register (APSR) holds program execution status bits such as zero, carry, negative, in bits[27:31]. ■ Interrupt Program Status Register (IPSR) holds the current exception number in bits[0:8]. ■ Execution Program Status Register (EPSR) holds control bits for interrupt continuable and IF-THEN instructions in bits[10:15] and [25:26]. Bit 24 is always set to 1 to indicate Thumb mode. Trying to clear it causes a fault exception.
PRIMASK	A 1-bit interrupt mask register. When set, it allows only the nonmaskable interrupt (NMI) and hard fault exception. All other exceptions and interrupts are masked.
FAULTMASK	A 1-bit interrupt mask register. When set, it allows only the NMI. All other exceptions and interrupts are masked.
BASEPRI	A register of up to nine bits that define the masking priority level. When set, it disables all interrupts of the same or higher priority value. If set to 0 then the masking function is disabled.
CONTROL	A 2-bit register for controlling the operating mode. Bit 0: 0 = privileged level in thread mode, 1 = user level in thread mode. Bit 1: 0 = default stack (MSP) is used, 1 = alternate stack is used. If in thread mode or user level then the alternate stack is the PSP. There is no alternate stack for handler mode; the bit must be 0 while in handler mode.

4.2 Cache Controller

The CY8C53 family has 128 bytes of direct mapped instruction cache between the CPU and the flash memory. This allows the CPU to access instructions much faster. The cache is enabled by default but user have the option to disable it.

4.3 DMA and PHUB

The PHUB and the DMA controller are responsible for data transfer between the CPU and peripherals, and also data transfers between peripherals. The PHUB and DMA also control device configuration during boot. The PHUB consists of:

- A central hub that includes the DMA controller, arbiter, and router
- Multiple spokes that radiate outward from the hub to most peripherals

There are two PHUB masters: the CPU and the DMA controller. Both masters may initiate transactions on the bus. The DMA channels can handle peripheral communication without CPU intervention. The arbiter in the central hub determines which DMA channel is the highest priority if there are multiple requests.

4.3.1 PHUB Features

- CPU and DMA controller are both bus masters to the PHUB
- Eight Multi-layer AHB Bus parallel access paths (spokes) for peripheral access
- Simultaneous CPU and DMA access to peripherals located on different spokes
- Simultaneous DMA source and destination burst transactions on different spokes
- Supports 8-, 16-, 24-, and 32-bit addressing and data

Table 4-3. PHUB Spokes and Peripherals

PHUB Spokes	Peripherals
0	SRAM
1	IOs, PICU
2	PHUB local configuration, Power manager, Clocks, IC, EEPROM, Flash programming interface
3	Analog interface and trim, Decimator
4	USB, I²C, Timers, Counters, and PWMs
5	Reserved
6	UDBs group 1
7	UDBs group 2

4.3.2 DMA Features

- 24 DMA channels
- Each channel has one or more transaction descriptors (TDs) to configure channel behavior. Up to 127 total TDs can be defined
- TDs can be dynamically updated
- Eight levels of priority per channel
- Any digitally routable signal, the CPU, or another DMA channel, can trigger a transaction
- Each channel can generate up to two interrupts per transfer
- Transactions can be stalled or canceled
- Supports transaction size of infinite or 1 to 64k bytes
- Large transactions may be broken into smaller bursts of 1 to 127 bytes
- TDs may be nested and/or chained for complex transactions

4.3.3 Priority Levels

The CPU always has higher priority than the DMA controller when their accesses require the same bus resources. Due to the system architecture, the CPU can never starve the DMA. DMA channels of higher priority (lower priority number) may interrupt current DMA transfers. In the case of an interrupt, the current transfer is allowed to complete its current transaction. To ensure latency limits when multiple DMA accesses are requested simultaneously, a fairness algorithm guarantees an interleaved minimum percentage of bus bandwidth for priority levels 2 through 7. Priority levels 0 and 1 do not take part in the fairness algorithm and may use 100% of the bus bandwidth. If a tie occurs on two DMA requests of the same priority level, a simple round robin method is used to evenly share the allocated bandwidth. The round robin allocation can be disabled for each DMA channel, allowing it to always be at the head of the line. Priority levels 2 to 7 are guaranteed the minimum bus bandwidth shown in Table 4-4 after the CPU and DMA priority levels 0 and 1 have satisfied their requirements.

Table 4-4. Priority Levels

Priority Level	% Bus Bandwidth
0	100.0
1	100.0
2	50.0
3	25.0
4	12.5
5	6.2
6	3.1
7	1.5

When the fairness algorithm is disabled, DMA access is granted based solely on the priority level; no bus bandwidth guarantees are made.

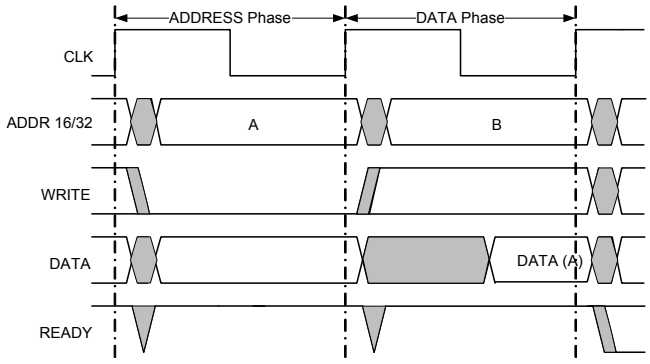
4.3.4 Transaction Modes Supported

The flexible configuration of each DMA channel and the ability to chain multiple channels allow the creation of both simple and complex use cases. General use cases include, but are not limited to:

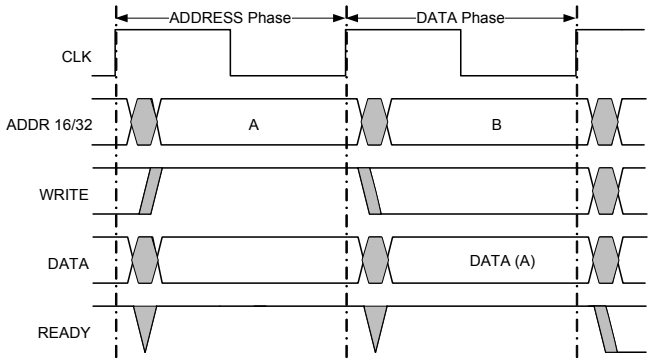
4.3.4.1 Simple DMA

In a simple DMA case, a single TD transfers data between a source and sink (peripherals or memory location). The basic timing diagrams of DMA read and write cycles are shown in Figure 4-2. For more description on other transfer modes, refer to the Technical Reference Manual.

Figure 4-2. DMA Timing Diagram



Basic DMA Read Transfer without wait states



Basic DMA Write Transfer without wait states

4.3.4.2 Auto Repeat DMA

Auto repeat DMA is typically used when a static pattern is repetitively read from system memory and written to a peripheral. This is done with a single TD that chains to itself.

4.3.4.3 Ping Pong DMA

A ping pong DMA case uses double buffering to allow one buffer to be filled by one client while another client is consuming the data previously received in the other buffer. In its simplest form, this is done by chaining two TDs together so that each TD calls the opposite TD when complete.

4.3.4.4 Circular DMA

Circular DMA is similar to ping pong DMA except it contains more than two buffers. In this case there are multiple TDs; after the last TD is complete it chains back to the first TD.

4.3.4.5 Indexed DMA

In an indexed DMA case, an external master requires access to locations on the system bus as if those locations were shared memory. As an example, a peripheral may be configured as an SPI or I^2C slave where an address is received by the external master. That address becomes an index or offset into the internal system bus memory space. This is accomplished with an initial “address fetch” TD that reads the target address location from the peripheral and writes that value into a subsequent TD in the chain. This modifies the TD chain on the fly. When the “address fetch” TD completes it moves on to the next TD, which has the new address information embedded in it. This TD then carries out the data transfer with the address location required by the external master.

4.3.4.6 Scatter Gather DMA

In the case of scatter gather DMA, there are multiple noncontiguous sources or destinations that are required to effectively carry out an overall DMA transaction. For example, a packet may need to be transmitted off of the device and the packet elements, including the header, payload, and trailer, exist in various noncontiguous locations in memory. Scatter gather DMA allows the segments to be concatenated together by using multiple TDs in a chain. The chain gathers the data from the multiple locations. A similar concept applies for the reception of data onto the device. Certain parts of the received data may need to be scattered to various locations in memory for software processing convenience. Each TD in the chain specifies the location for each discrete element in the chain.

4.3.4.7 Packet Queuing DMA

Packet queuing DMA is similar to scatter gather DMA but specifically refers to packet protocols. With these protocols, there may be separate configuration, data, and status phases associated with sending or receiving a packet.

For instance, to transmit a packet, a memory mapped configuration register can be written inside a peripheral, specifying the overall length of the ensuing data phase. The CPU can set up this configuration information anywhere in system memory and copy it with a simple TD to the peripheral. After the configuration phase, a data phase TD (or a series of data phase TDs) can begin (potentially using scatter gather). When the data phase TD(s) finish, a status phase TD can be invoked that reads some memory mapped status information from the peripheral and copies it to a location in system memory specified by the CPU for later inspection. Multiple sets of configuration, data, and status phase “subchains” can be strung together to create larger chains that transmit multiple packets in this way. A similar concept exists in the opposite direction to receive the packets.

4.3.4.8 Nested DMA

One TD may modify another TD, as the TD configuration space is memory mapped similar to any other peripheral. For example, a first TD loads a second TD’s configuration and then calls the second TD. The second TD moves data as required by the application. When complete, the second TD calls the first TD, which again updates the second TD’s configuration. This process repeats as often as necessary.

4.4 Interrupt Controller

The Cortex-M3 NVIC supports 16 system exceptions and 32 interrupts from peripherals, as shown in Table 4-5.

Table 4-5. Cortex-M3 Exceptions and Interrupts

Exception Number	Exception Type	Priority	Exception Table Address Offset	Function
			0x00	Starting value of R13 / MSP
1	Reset	–3 (highest)	0x04	Reset
2	NMI	–2	0x08	Non maskable interrupt
3	Hard fault	–1	0x0C	All classes of fault, when the corresponding fault handler cannot be activated because it is currently disabled or masked
4	MemManage	Programmable	0x10	Memory management fault, for example, instruction fetch from a nonexecutable region
5	Bus fault	Programmable	0x14	Error response received from the bus system; caused by an instruction prefetch abort or data access error
6	Usage fault	Programmable	0x18	Typically caused by invalid instructions or trying to switch to ARM mode
7 – 10	–	–	0x1C – 0x28	Reserved
11	SVC	Programmable	0x2C	System service call via SVC instruction
12	Debug monitor	Programmable	0x30	Debug monitor
13	–	–	0x34	Reserved
14	PendSV	Programmable	0x38	Deferred request for system service
15	SYSTICK	Programmable	0x3C	System tick timer
16 – 47	IRQ	Programmable	0x40 – 0x3FC	Peripheral interrupt request #0 - #31

Bit 0 of each exception vector indicates whether the exception is executed using ARM or Thumb instructions. Because the Cortex-M3 only supports Thumb instructions, this bit must always be 1. The Cortex-M3 non maskable interrupt (NMI) input can be routed to any pin, via the DSI, or disconnected from all pins. See "DSI Routing Interface Description" section on page 39.

The NVIC handles interrupts from the peripherals, and passes the interrupt vectors to the CPU. It is closely integrated with the CPU for low latency interrupt handling. Features include:

- 32 interrupts. Multiple sources for each interrupt.
- Configurable number of priority levels: from 3 to 8.
- Dynamic reprioritization of interrupts.
- Priority grouping. This allows selection of preempting and non preempting interrupt levels.
- Support for tail-chaining, and late arrival, of interrupts. This enables back-to-back interrupt processing without the overhead of state saving and restoration between interrupts.
- Processor state automatically saved on interrupt entry, and restored on interrupt exit, with no instruction overhead.

If the same priority level is assigned to two or more interrupts, the interrupt with the lower vector number is executed first. Each interrupt vector may choose from three interrupt sources: Fixed Function, DMA, and UDB. The fixed function interrupts are direct connections to the most common interrupt sources and provide the lowest resource cost connection. The DMA interrupt sources provide direct connections to the two DMA interrupt sources provided per DMA channel. The third interrupt source for vectors is from the UDB digital routing array. This allows any digital signal available to the UDB array to be used as an interrupt source. All interrupt sources may be routed to any interrupt vector using the UDB interrupt source connections.

Table 4-6. Interrupt Vector Table

Interrupt #	Cortex-M3 Exception #	Fixed Function	DMA	UDB
0	16	Low voltage detect (LVD)	phub_termout0[0]	udb_intr[0]
1	17	Cache	phub_termout0[1]	udb_intr[1]
2	18	Reserved	phub_termout0[2]	udb_intr[2]
3	19	Pwr Mgr	phub_termout0[3]	udb_intr[3]
4	20	PICU[0]	phub_termout0[4]	udb_intr[4]
5	21	PICU[1]	phub_termout0[5]	udb_intr[5]
6	22	PICU[2]	phub_termout0[6]	udb_intr[6]
7	23	PICU[3]	phub_termout0[7]	udb_intr[7]
8	24	PICU[4]	phub_termout0[8]	udb_intr[8]
9	25	PICU[5]	phub_termout0[9]	udb_intr[9]
10	26	PICU[6]	phub_termout0[10]	udb_intr[10]
11	27	PICU[12]	phub_termout0[11]	udb_intr[11]
12	28	PICU[15]	phub_termout0[12]	udb_intr[12]
13	29	Comparators Combined	phub_termout0[13]	udb_intr[13]
14	30	Switched Caps Combined	phub_termout0[14]	udb_intr[14]
15	31	I^2C	phub_termout0[15]	udb_intr[15]
16	32	Reserved	phub_termout1[0]	udb_intr[16]
17	33	Reserved	phub_termout1[1]	udb_intr[17]
18	34	Reserved	phub_termout1[2]	udb_intr[18]
19	35	Reserved	phub_termout1[3]	udb_intr[19]
20	36	Reserved	phub_termout1[4]	udb_intr[20]
21	37	USB SOF Int	phub_termout1[5]	udb_intr[21]
22	38	USB Arb Int	phub_termout1[6]	udb_intr[22]
23	39	USB Bus Int	phub_termout1[7]	udb_intr[23]
24	40	USB Endpoint[0]	phub_termout1[8]	udb_intr[24]
25	41	USB Endpoint Data	phub_termout1[9]	udb_intr[25]
26	42	Reserved	phub_termout1[10]	udb_intr[26]
27	43	Reserved	phub_termout1[11]	udb_intr[27]
28	44	Reserved	phub_termout1[12]	udb_intr[28]
29	45	Decimator Int	phub_termout1[13]	udb_intr[29]
30	46	phub_err_int	phub_termout1[14]	udb_intr[30]
31	47	eeprom_fault_int	phub_termout1[15]	udb_intr[31]

5. Memory

5.1 Static RAM

CY8C53 static RAM (SRAM) is used for temporary data storage. Code can be executed at full speed from the portion of SRAM that is located in the code space. This process is slower from SRAM above 0x20000000. The device provides up to 64 KB of SRAM. The CPU or the DMA controller can access all of SRAM. The SRAM can be accessed simultaneously by the Cortex-M3 CPU and the DMA controller if accessing different 32-KB blocks.

5.2 Flash Program Memory

Flash memory in PSoC devices provides nonvolatile storage for user firmware, user configuration data and bulk data storage. The main flash memory area contains up to 256 KB of user program space.

Up to an additional 32 KB of flash space is available for storing device configuration data and bulk user data. User code may not be run out of this flash memory section. The flash output is 9 bytes wide with 8 bytes of data and 1 additional byte.

The flash programming interface performs flash erasing, programming and setting code protection levels. Flash In System Serial Programming (ISSP), typically used for production programming, is possible through the SWD interface. In-system programming, typically used for bootloaders, is also possible using serial interfaces such as I^2C, USB, UART, and SPI, or any communications protocol.

5.3 Flash Security

All PSoC devices include a flexible flash protection model that prevents access and visibility to on-chip flash memory. This prevents duplication or reverse engineering of proprietary code. Flash memory is organized in blocks, where each block contains 256 bytes of program or data and 32 bytes of configuration or general-purpose data.

The device offers the ability to assign one of four protection levels to each row of flash. Table 5-1 lists the protection modes available. Flash protection levels can only be changed by performing a complete flash erase. The Full Protection and Field Upgrade settings disable external access (through a debugging tool such as PSoC Creator, for example). If your application requires code update through a boot loader, then use the Field Upgrade setting. Use the Unprotected setting only when no security is needed in your application. The PSoC device also offers an advanced security feature called Device Security which permanently disables all test, programming, and debug ports, protecting your application from external access (see the "Device Security" section on page 55). For more information on how to take full advantage of the security features in PSoC, see the PSoC 5 TRM.

Table 5-1. Flash Protection

Protection Setting	Allowed	Not Allowed
Unprotected	External read and write + internal read and write	–
Factory Upgrade	External write + internal read and write	External read
Field Upgrade	Internal read and write	External read and write
Full Protection	Internal read	External read and write + internal write

Disclaimer

Note the following details of the flash code protection features on Cypress devices.

Cypress products meet the specifications contained in their particular Cypress data sheets. Cypress believes that its family of products is one of the most secure families of its kind on the market today, regardless of how they are used. There may be methods, unknown to Cypress, that can breach the code protection features. Any of these methods, to our knowledge, would be dishonest and possibly illegal. Neither Cypress nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Cypress is willing to work with the customer who is concerned about the integrity of their code. Code protection is constantly evolving. We at Cypress are committed to continuously improving the code protection features of our products.

5.4 EEPROM

PSoC EEPROM memory is a byte addressable nonvolatile memory. The CY8C53 has 2 KB of EEPROM memory to store user data. Reads from EEPROM are random access at the byte level. Reads are done directly; writes are done by sending write commands to an EEPROM programming interface. CPU code execution can continue from flash during EEPROM writes. EEPROM is erasable and writeable at the row level. The EEPROM is divided into two sections, each containing 64 rows of 16 bytes each.

The CPU cannot execute out of EEPROM.

5.5 Memory Map

The Cortex-M3 has a fixed address map, which allows peripherals to be accessed by simple memory access instructions.

5.5.1 Address Map

The 4-GB address space is divided into the ranges shown in Table 5-2:

Table 5-2. Address Map

Address Range	Size	Use
0x00000000 – 0x1FFFFFFF	0.5 GB	Program code. This includes the exception vector table at power up, which starts at address 0.
0x20000000 – 0x3FFFFFFF	0.5 GB	Static RAM. This includes a 1 MByte bit-band region starting at 0x20000000 and a 32 Mbyte bit-band alias region starting at 0x22000000.
0x40000000 – 0x5FFFFFFF	0.5 GB	Peripherals.
0x60000000 – 0x9FFFFFFF	1 GB	External RAM.
0xA0000000 – 0xDFFFFFFF	1 GB	External peripherals.
0xE0000000 – 0xFFFFFFFF	0.5 GB	Internal peripherals, including the NVIC and debug and trace modules.

Table 5-3. Peripheral Data Address Map

Address Range	Purpose
0x00000000 – 0x0003FFFF	256K Flash
0x1FFF8000 – 0x1FFFFFFF	32K SRAM in Code region
0x20000000 – 0x20007FFF	32K SRAM in SRAM region
0x40004000 – 0x400042FF	Clocking, PLLs, and oscillators
0x40004300 – 0x400043FF	Power management
0x40004500 – 0x400045FF	Ports interrupt control
0x40004700 – 0x400047FF	Flash programming interface

Table 5-3. Peripheral Data Address Map (continued)

Address Range	Purpose
0x40004800 – 0x400048FF	Cache controller
0x40004900 – 0x400049FF	I^2C controller
0x40004E00 – 0x40004EFF	Decimator
0x40004F00 – 0x40004FFF	Fixed timer/counter/PWMs
0x40005000 – 0x400051FF	I/O ports control
0x40005800 – 0x40005FFF	Analog Subsystem Interface
0x40006000 – 0x400060FF	USB Controller
0x40006400 – 0x40006FFF	UDB Configuration
0x40007000 – 0x40007FFF	PHUB Configuration
0x40008000 – 0x400087FF	EEPROM
0x40010000 – 0x4001FFFF	Digital Interconnect Configuration
0xE0000000 – 0xE00FFFFF	Cortex-M3 PPB Registers, including NVIC, debug, and trace

The bit-band feature allows individual bits in SRAM to be read or written as atomic operations. This is done by reading or writing bit 0 of corresponding words in the bit-band alias region. For example, to set bit 3 in the word at address 0x20000000, write a 1 to address 0x2200000C. To test the value of that bit, read address 0x2200000C and the result is either 0 or 1 depending on the value of the bit.

Most memory accesses done by the Cortex-M3 are aligned, that is, done on word (4-byte) boundary addresses. Unaligned accesses of words and 16-bit half-words on nonword boundary addresses can also be done, although they are less efficient.

5.5.2 Address Map and Cortex-M3 Buses

The ICode and DCode buses are used only for accesses within the Code address range, 0 - 0x1FFFFFFF.

The system bus is used for data accesses and debug accesses within the ranges 0x20000000 – 0xDFFFFFFF and 0xE0100000 – 0xFFFFFFFF. Instruction fetches can also be done within the range 0x20000000 – 0x3FFFFFFF, although these can be slower than instruction fetches via the ICode bus.

The private peripheral bus (PPB) is used within the Cortex-M3 to access system control registers and debug and trace module registers.

6. System Integration

6.1 Clocking System

The clocking system generates, divides, and distributes clocks throughout the PSoC system. The IMO and PLL together can generate up to a 67 MHz clock, accurate to ±5% over voltage and temperature. Additional internal and external clock sources allow each design to optimize accuracy, power, and cost. All of the system clock sources can be used to generate other clock frequencies in the 16-bit clock dividers and UDBs for anything the user wants, for example a UART baud rate generator.

Clock generation and distribution is automatically configured through the PSoC Creator IDE graphical interface. This is based on the complete system's requirements. It greatly speeds the design process. PSoC Creator allows designers to build clocking systems with minimal input. The designer can specify desired clock frequencies and accuracies, and the software locates or builds a clock that meets the required specifications. This is possible because of the programmability inherent PSoC.

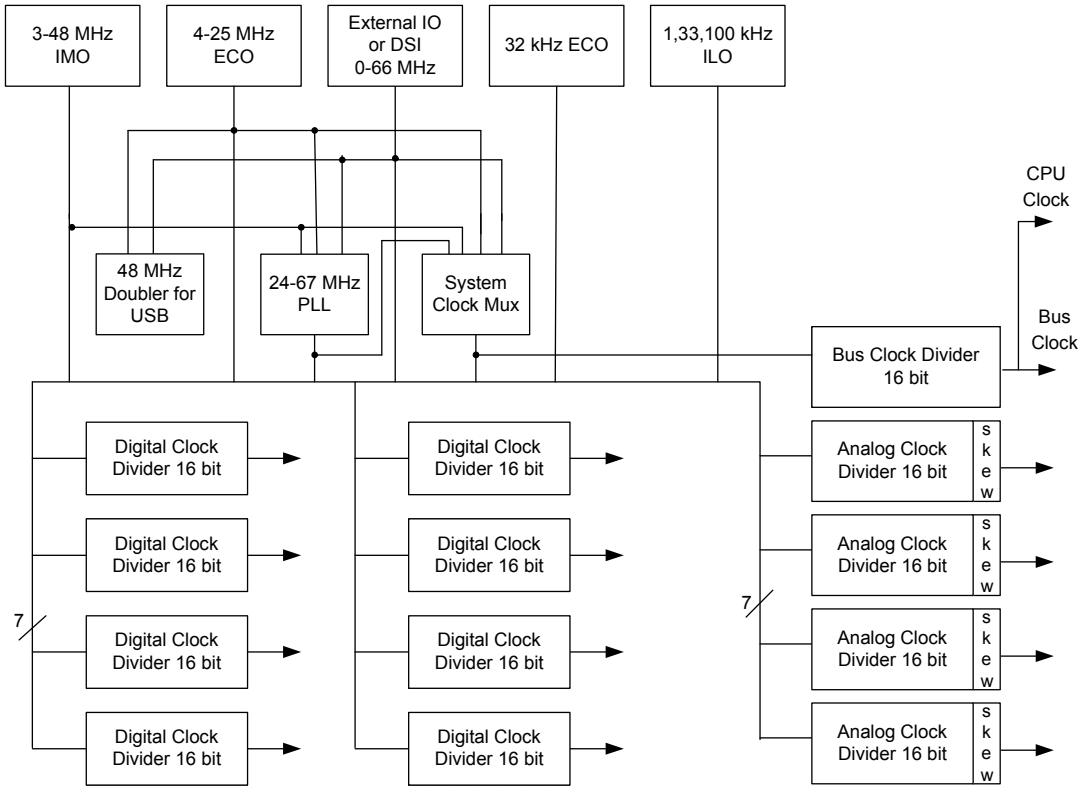
Key features of the clocking system include:

- Seven general purpose clock sources
  - 3 to 48 MHz IMO, ±5% at 3 MHz
  - 4 to 25 MHz external crystal oscillator (MHzECO)
  - Clock doubler provides a doubled clock frequency output for the USB block, see USB Clock Domain on page 21
  - DSI signal from an external I/O pin or other logic
  - 24 to 67 MHz fractional PLL sourced from IMO, MHzECO, or DSI
  - 1 kHz, 33 kHz, 100 kHz ILO for watchdog timer (WDT) and sleep timer
  - 32.768 kHz external crystal oscillator (kHzECO) for RTC
- Independently sourced clock dividers in all clocks
- Eight 16-bit clock dividers for the digital system
- Four 16-bit clock dividers for the analog system
- Dedicated 16-bit divider for the CPU bus and CPU clock
- Automatic clock configuration in PSoC Creator

Table 6-1. Oscillator Summary

Source	Fmin	Tolerance at Fmin	Fmax	Tolerance at Fmax	Startup Time
IMO	3 MHz	±5% over voltage and temperature	48 MHz	±10%	12 µs max
MHzECO	4 MHz	Crystal dependent	25 MHz	Crystal dependent	5 ms typ, max is crystal dependent
DSI	0 MHz	Input dependent	66 MHz	Input dependent	Input dependent
PLL	24 MHz	Input dependent	67 MHz	Input dependent	250 µs max
Doubler	48 MHz	Input dependent	48 MHz	Input dependent	1 µs max
ILO	1 kHz	–50%, +100%	100 kHz	-55%, +100%	15 ms max in lowest power mode
kHzECO	32 kHz	Crystal dependent	32 kHz	Crystal dependent	500 ms typ, max is crystal dependent

Figure 6-1. Clocking Subsystem



6.1.1 Internal Oscillators

6.1.1.1 Internal Main Oscillator

The IMO operates with no external components and outputs a stable clock. A factory trim for each frequency range is stored in the device. With the factory trim, tolerance varies from ±5% at 3 MHz, up to ±10% at 48 MHz. The IMO, in conjunction with the PLL, allows generation of CPU and system clocks up to the device's maximum frequency. The IMO provides clock outputs at 3, 6, 12, 24, and 48 MHz.

6.1.1.2 Clock Doubler

The clock doubler outputs a clock at twice the frequency of the input clock. The doubler works at input frequency of 24 MHz, providing 48 MHz for the USB. It can be configured to use a clock from the MHzECO or the DSI (external pin). The doubler is typically used to clock the USB.

6.1.1.3 Phase-Locked Loop

The PLL allows low frequency, high accuracy clocks to be multiplied to higher frequencies. This is a tradeoff between higher clock frequency and accuracy and, higher power consumption and increased startup time.

The PLL block provides a mechanism for generating clock frequencies based upon a variety of input sources. The PLL outputs clock frequencies in the range of 24 to 67 MHz. Its input and feedback dividers supply 4032 discrete ratios to create almost any desired system clock frequency. The accuracy of the PLL output depends on the accuracy of the PLL input source. The most common PLL use is to multiply the IMO clock at 3 MHz, where it is most accurate, to generate the CPU and system clocks up to the device's maximum frequency.

The PLL achieves phase lock within 250 µs (verified by bit setting). It can be configured to use a clock from the IMO, MHzECO, or DSI (external pin). The PLL clock source can be used until lock is complete and signaled with a lock bit. The lock signal can be routed through the DSI to generate an interrupt. Disable the PLL before entering low power modes.

6.1.1.4 Internal Low Speed Oscillator

The ILO provides clock frequencies for low power consumption, including the sleep timer. The ILO generates up to three different clocks: 1 kHz, 33 kHz, and 100 kHz.

The 1 kHz clock (CLK1K) is typically used for a background 'heartbeat' timer. This clock inherently lends itself to long sleep intervals using the central timewheel (CTW). The central timewheel is a free running counter clocked by the ILO 1 kHz output. The central timewheel is always enabled except in hibernate mode and when the CPU is stopped during debug on chip mode. It can be used to generate periodic interrupts for timing purposes or to wake the system from a low power mode. Firmware can reset the central timewheel.

The central timewheel can be programmed to wake the system periodically and optionally issue an interrupt. This enables flexible, periodic wakeups from low power modes or coarse timing applications. Systems that require accurate timing should use the RTC capability instead of the central timewheel. The 100 kHz clock (CLK100K) works as a low-power system clock to

run the CPU. It can also generate fast time intervals using the fast timewheel.

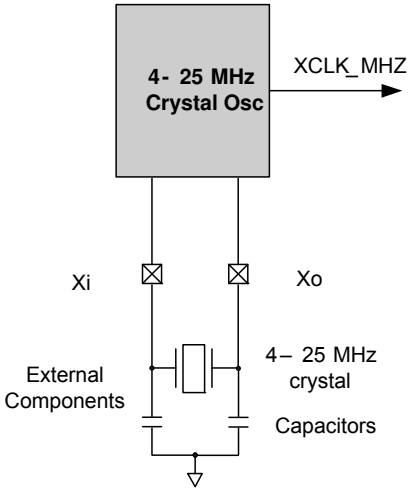
The fast timewheel is a 100 kHz, 5-bit counter clocked by the ILO that can also be used to generate periodic interrupts. The fast timewheel settings are programmable, and the counter automatically resets when the terminal count is reached. This enables flexible, periodic interrupts to the CPU at a higher rate than is allowed using the central timewheel. The fast timewheel can generate an optional interrupt each time the terminal count is reached. The 33 kHz clock (CLK33K) comes from a divide-by-3 operation on CLK100K. This output can be used as a reduced accuracy version of the 32.768 kHz ECO clock with no need for a crystal. The fast timewheel cannot be used as a wakeup source and must be turned off before entering sleep or hibernate mode.

6.1.2 External Oscillators

6.1.2.1 MHz External Crystal Oscillator

The MHzECO provides high frequency, high precision clocking using an external crystal (see Figure 6-2). It supports crystals in the range of 4 to 25 MHz. When used in conjunction with the PLL, it can generate CPU and system clocks up to the device's maximum frequency (see Phase-Locked Loop on page 19). The MHzECO with a 24 MHz crystal can be used with the clock doubler to generate a 48 MHz clock for the USB. If a crystal is not used then Xi must be shorted to ground and Xo must be left floating. MHzECO accuracy depends on the crystal chosen.

Figure 6-2. MHzECO Block Diagram

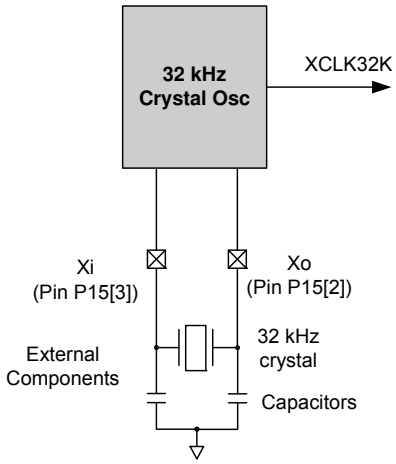


6.1.2.2 32.768 kHz ECO

The 32.768 kHz external crystal oscillator (32kHzECO) provides precision timing with minimal power consumption using an external 32.768 kHz watch crystal (see Figure 6-3). The RTC uses a 1-second interrupt to implement the RTC functionality in firmware.

The oscillator works in two distinct power modes. This allows users to trade off power consumption with noise immunity from neighboring circuits. The GPIO pins connected to the external crystal and capacitors are fixed.

Figure 6-3. 32kHzECO Block Diagram



It is recommended that the external 32.768-kHz watch crystal have a load capacitance (CL) of 6 pF or 12.5 pF. Check the crystal manufacturer's datasheet. The two external capacitors, CL1 and CL2, are typically of the same value, and their total capacitance, CL1CL2 / (CL1 + CL2), including pin and trace capacitance, should equal the crystal CL value. For more information, refer to application note AN54439: PSoC 3 and PSoC 5 External Oscillators. See also pin capacitance specifications in the "GPIO" section on page 61.

6.1.2.3 Digital System Interconnect

The DSI provides routing for clocks taken from external clock oscillators connected to I/O. The oscillators can also be generated within the device in the digital system and Universal Digital Blocks.

While the primary DSI clock input provides access to all clocking resources, up to eight other DSI clocks (internally or externally generated) may be routed directly to the eight digital clock dividers. This is only possible if there are multiple precision clock sources.

6.1.3 Clock Distribution

All seven clock sources are inputs to the central clock distribution system. The distribution system is designed to create multiple high precision clocks. These clocks are customized for the design's requirements and eliminate the common problems found with limited resolution prescalers attached to peripherals. The clock distribution system generates several types of clock trees.

- The system clock is used to select and supply the fastest clock in the system for general system clock requirements and clock synchronization of the PSoC device.
- Bus clock 16-bit divider uses the system clock to generate the system's bus clock used for data transfers and the CPU. The CPU clock is directly derived from the bus clock.
- Eight fully programmable 16-bit clock dividers generate digital system clocks for general use in the digital system, as configured by the design's requirements. Digital system clocks can generate custom clocks derived from any of the seven clock sources for any purpose. Examples include baud rate generators, accurate PWM periods, and timer clocks, and many others. If more than eight digital clock dividers are required, the UDBs and fixed function timer/counter/PWMs can also generate clocks.
- Four 16-bit clock dividers generate clocks for the analog system components that require clocking, such as the ADC. The analog clock dividers include skew control to ensure that critical analog events do not occur simultaneously with digital switching events. This is done to reduce analog system noise.

Each clock divider consists of an 8-input multiplexer, a 16-bit clock divider (divide by 2 and higher) that generates ~50% duty cycle clocks, system clock resynchronization logic, and deglitch logic. The outputs from each digital clock tree can be routed into the digital system interconnect and then brought back into the clock system as an input, allowing clock chaining of up to 32 bits.

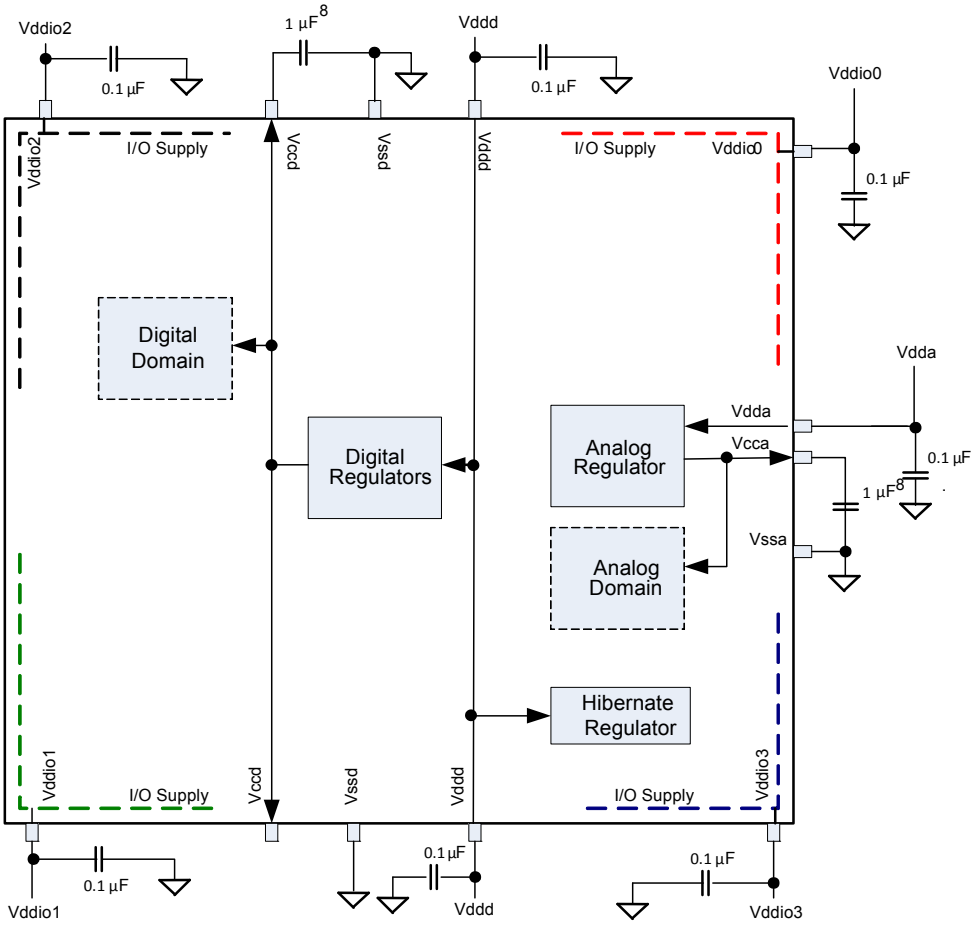
6.1.4 USB Clock Domain

The USB clock domain is unique in that it operates largely asynchronously from the main clock network. The USB logic contains a synchronous bus interface to the chip, while running on an asynchronous clock to process USB data. The USB logic requires a 48 MHz frequency. This frequency can be generated from different sources, including DSI clock at 48 MHz or doubled value of 24 MHz from the MHzECO or DSI signal.

6.2 Power System

The power system consists of separate analog, digital, and I/O supply pins, labeled V_{DDA}, V_{DDD}, and Vddiox, respectively. It also includes two internal 1.8 V regulators that provide the digital (V_{CCD}) and analog (V_{CCA}) supplies for the internal core logic. The output pins of the regulators (V_{CCD} and V_{CCA}) and the V_{DDIO} pins must have capacitors connected as shown in Figure 6-4 (10 µF is required for sleep mode. See Table 11-3). The two V_{CCD} pins must be shorted together, with as short a trace as possible. The power system also contains a hibernate regulator.

Figure 6-4. PSoC Power System



Note The two V_{CCD} pins must be connected together with as short a trace as possible. A trace under the device is recommended, as shown in Figure 2-4.

Note
8. 10 µF is required for sleep mode. See Table 11-3.

6.2.1 Power Modes

PSoC 5 devices have four different power modes, as shown in Table 6-2 and Table 6-3. The power modes allow a design to easily provide required functionality and processing power while simultaneously minimizing power consumption and maximizing battery life in low power and portable devices.

PSoC 5 power modes, in order of decreasing power consumption are:

- Active
- Alternate Active
- Sleep
- Hibernate

Active is the main processing mode. Its functionality is configurable. Each power controllable subsystem is enabled or disabled by using separate power configuration template registers. In alternate active mode, fewer subsystems are enabled, reducing power. In sleep mode most resources are disabled regardless of the template settings. Sleep mode is optimized to provide timed sleep intervals. The lowest power mode is hibernate, which retains register and SRAM state, but no clocks, and allows wakeup only from reset Figure 6-5 illustrates the allowable transitions between power modes. Sleep and hibernate modes should not be entered until all V_{DDIO} supplies are at valid voltage levels and interrupts are enabled.

Table 6-2. Power Modes

Power Modes	Description	Entry Condition	Wakeup Source	Active Clocks	Regulator
Active	Primary mode of operation, all peripherals available (programmable)	Wakeup, reset, manual register entry	Any interrupt	Any (programmable)	All regulators available.
Alternate Active	Similar to Active mode, and is typically configured to have fewer peripherals active to reduce power. One possible configuration is to use the UDBs for processing, with the CPU turned off	Manual register entry	Any interrupt	Any (programmable)	All regulators available.
Sleep	All subsystems automatically disabled	Manual register entry	CTW[10]	ILO	All regulators available.
Hibernate	All subsystems automatically disabled Lowest power consuming mode with all peripherals and internal regulators disabled, except hibernate regulator is enabled Configuration and memory contents retained	Manual register entry			Only hibernate regulator active.

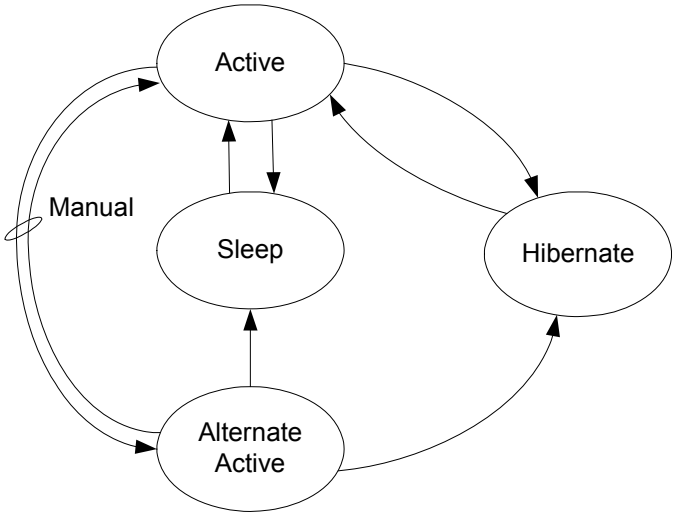
Table 6-3. Power Modes Wakeup Time and Power Consumption

Sleep Modes	Wakeup Time	Current (Typ)	Code Execution	Digital Resources	Analog Resources	Clock Sources Available	Wakeup Sources	Reset Sources
Active	–	6 mA[9]	Yes	All	All	All	–	All
Alternate Active	–	–	User defined	All	All	All	–	All
Sleep	125 µs typ	2 µA[10]	No	None	None	ILO	CTW	XRES
Hibernate	–	300 nA	No	None	None	None	–	XRES

Notes

9. Bus clock off. Execute from CPU instruction buffer at 6 MHz. See Table 11-2 on page 58.
10. During sleep mode, the CTW generates periodic interrupts to wake up the device. This affects the average current, which is a composite of the sleep mode current and active mode current, and the time spent in each mode. With the maximum wakeup interval of 128 ms, and at wakeup the CPU executes only the standard PSoC Creator sleep API (for a duty cycle of 0.2%), the average current draw is typically 35 µA.

Figure 6-5. Power Mode Transitions



6.2.1.1 Active Mode

Active mode is the primary operating mode of the device. When in active mode, the active configuration template bits control which available resources are enabled or disabled. When a resource is disabled, the digital clocks are gated, analog bias currents are disabled, and leakage currents are reduced as appropriate. User firmware can dynamically control subsystem power by setting and clearing bits in the active configuration template. The CPU can disable itself, in which case the CPU is automatically reenabled at the next wakeup event.

When a wakeup event occurs, the global mode is always returned to active, and the CPU is automatically enabled, regardless of its template settings. Active mode is the default global power mode upon boot.

6.2.1.2 Alternate Active Mode

Alternate Active mode is very similar to active mode. In alternate active mode, fewer subsystems are enabled, to reduce power consumption. One possible configuration is to turn off the CPU and flash, and run peripherals at full speed.

6.2.1.3 Sleep Mode

Sleep mode powers down the CPU and other internal circuitry to reduce power consumption. However, supervisory services such as the central timewheel (CTW) remain available in this mode. The device can wake up using CTW or system reset. The wake up time from sleep mode is 125 µs (typical).

6.2.1.4 Hibernate Mode

In hibernate mode nearly all of the internal functions are disabled. Internal voltages are reduced to the minimal level to keep vital systems alive. Configuration state is preserved in hibernate mode and SRAM memory is retained. GPIOs configured as digital outputs maintain their previous values and external GPIO pin interrupt settings are preserved. The device can only return from hibernate mode in response to an external reset (XRES).

6.2.1.5 Wakeup Events

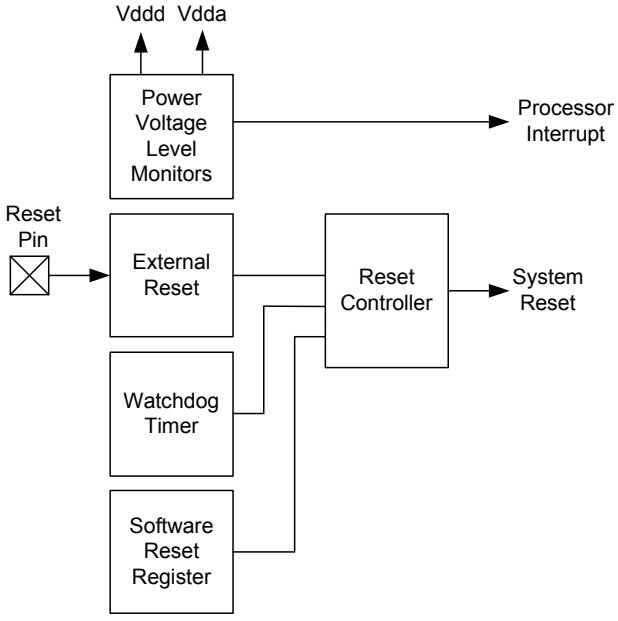
Wakeup events can come from the central timewheel or device reset. A wakeup event restores the system to active mode. The central timewheel allows the system to periodically wake up, poll peripherals, do voltage monitoring, or perform real-time functions. Reset event sources include the external reset pin (XRES).

6.3 Reset

CY8C53 has multiple internal and external reset sources available. The reset sources are:

- Power source monitoring - The analog and digital power voltages, V_{DDA}, V_{DDD}, V_{CCA}, and V_{CCD} are monitored in several different modes during power up and active mode. The monitors are programmable to generate an interrupt to the processor under certain conditions.
- External - The device can be reset from an external source by pulling the reset pin (XRES) low. The XRES pin includes an internal pull-up to Vddio1. V_{DDD}, V_{DDA}, and Vddio1 must all have voltage applied before the part comes out of reset.
- Watchdog timer - A watchdog timer monitors the execution of instructions by the processor. If the watchdog timer is not reset by firmware within a certain period of time, the watchdog timer generates a reset. The watchdog timer can be used only when the part remains in active mode.
- Software - The device can be reset under program control.

Figure 6-6. Resets



The term **system reset** indicates that the processor as well as analog and digital peripherals and registers are reset.

A reset status register shows some of the resets or power voltage monitoring interrupts. The program may examine this register to detect and report certain exception conditions. This register is cleared after a power-on reset. For details see the Technical Reference Manual.

6.3.1 Power Voltage Level Monitors

- IPOR - Initial Power on Reset

  At initial power on, IPOR monitors the power voltages V_{DDD} and V_{DDA}, both directly at the pins and at the outputs of the corresponding internal regulators. The trip level is not precise. It is set to approximately 1 volt, which is below the lowest specified operating voltage but high enough for the internal circuits to be reset and to hold their reset state. The monitor generates a reset pulse that is at least 100 ns wide. It may be much wider if one or more of the voltages ramps up slowly.

  To save power the IPOR circuit is disabled when the internal digital supply is stable. When the voltage is high enough, the IMO starts.

- ALVI, DLVI, AHVI - Analog/Digital Low Voltage Interrupt, Analog High Voltage Interrupt

  Interrupt circuits are available to detect when V_{DDA} and V_{DDD} go outside a voltage range. For AHVI, V_{DDA} is compared to a fixed trip level. For ALVI and DLVI, V_{DDA} and V_{DDD} are compared to trip levels that are programmable, as listed in Table 6-4.

Table 6-4. Analog/Digital Low Voltage Interrupt, Analog High Voltage Interrupt

Interrupt	Supply	Normal Voltage Range	Available Trip Settings
DLVI	V_{DDD}	2.7 V-5.5 V	2.45 V-5.45 V in 250 mV increments. The 2.45 V setting is used for LVD.
ALVI	V_{DDA}	2.7 V-5.5 V	2.45 V-5.45 V in 250 mV increments. The 2.45 V setting is used for LVD
AHVI	V_{DDA}	2.7 V-5.5 V	5.75 V

The monitors are disabled until after IPOR. The monitors are not available in low-power modes. To monitor voltages in sleep mode, wake up periodically using the CTW. After wakeup, the 2.45 V LVI interrupt may trigger. Voltage monitoring is not available in hibernate mode.

6.3.2 Other Reset Sources

- XRES - External Reset

  CY8C53 has a dedicated XRES pin which, holds the part in reset while held active (low). The response to an XRES is the same as to an IPOR reset.

  The external reset is active low. It includes an internal pull-up resistor. XRES is active during sleep and hibernate modes.

- SRES - Software Reset

  A reset can be commanded under program control by setting a bit in the software reset register. This is done either directly by the program or indirectly by DMA access. The response to a SRES is the same as after an IPOR reset.

  Another register bit exists to disable this function.

- WRES - Watchdog Timer Reset

  The watchdog reset detects when the software program is no longer being executed correctly. To indicate to the watchdog timer that it is running correctly, the program must periodically reset the timer. If the timer is not reset before a user-specified amount of time, then a reset is generated.

  Note IPOR disables the watchdog function. The program must enable the watchdog function at an appropriate point in the code by setting a register bit. When this bit is set, it cannot be cleared again except by an IPOR power on reset event. The watchdog timer can be used only when the part remains in active mode.

6.4 I/O System and Routing

PSoC I/Os are extremely flexible. Every GPIO has analog and digital I/O capability. All I/Os have a large number of drive modes, which are set at POR. PSoC also provides up to four individual I/O voltage domains through the V_{DDIO} pins.

There are two types of I/O pins on every device; those with USB provide a third type. Both General Purpose I/O (GPIO) and Special I/O (SIO) provide similar digital functionality. The primary differences are their analog capability and drive strength. Devices that include USB also provide two USBIO pins that support specific USB functionality as well as limited GPIO capability.

All I/O pins are available for use as digital inputs and outputs for both the CPU and digital peripherals. In addition, all I/O pins can generate an interrupt. The flexible and advanced capabilities of the PSoC I/O, combined with any signal to any pin routability, greatly simplify circuit design and board layout. All GPIO pins can be used for analog input, CapSense[6], and LCD segment drive, while SIO pins are used for voltages in excess of V_{DDA} and for programmable output voltages.

- Features supported by both GPIO and SIO:
  - Separate I/O supplies and voltages for up to four groups of I/O
  - Digital peripherals use DSI to connect the pins
  - Input or output or both for CPU and DMA
  - Eight drive modes
  - Every pin can be an interrupt source configured as rising edge, falling edge or both edges. If required, level sensitive interrupts are supported through the DSI
  - Dedicated port interrupt vector for each port
  - Slew rate controlled digital output drive mode
  - Access port control and configuration registers on either port basis or pin basis
  - Separate port read (PS) and write (DR) data registers to avoid