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® 5LP: CY8C56LP Family Datasheet

Programmable System-on-Chip (PSoC®)

General Description

PSoC® 5LP is a true programmable embedded system-on-chip, integrating configurable analog and digital peripherals, memory, and a microcontroller on a single chip. The PSoC 5LP architecture boosts performance through:

- 32-bit ARM® Cortex®-M3 core plus DMA controller and digital filter processor, at up to 80 MHz
- Ultra low power with industry's widest voltage range
- Programmable digital and analog peripherals enable custom functions
- Flexible routing of any analog or digital peripheral function to any pin

PSoC devices employ a highly configurable system-on-chip architecture for embedded control design. They integrate configurable analog and digital circuits, controlled by an on-chip microcontroller. A single PSoC device can integrate as many as 100 digital and analog peripheral functions, reducing design time, board space, power consumption, and system cost while improving system quality.

Features

- Operating characteristics
  - Voltage range: 1.71 to 5.5 V, up to 6 power domains
  - Temperature range (ambient): –40 to 85 °C[1] Extended temperature parts: –40 to 105 °C
  - DC to 80-MHz operation
  - Power modes
    - Active mode 3.1 mA at 6 MHz, and 15.4 mA at 48 MHz
    - 2-µA sleep mode
    - 300-nA hibernate mode with RAM retention
  - Boost regulator from 0.5-V input up to 5-V output
- Performance
  - 32-bit ARM Cortex-M3 CPU, 32 interrupt inputs
  - 24-channel direct memory access (DMA) controller
  - 24-bit 64-tap fixed-point digital filter processor (DFB)
- Memories
  - Up to 256 KB program flash, with cache and security features
  - Up to 32 KB additional flash for error correcting code (ECC)
  - Up to 64 KB RAM
  - 2 KB EEPROM
- Digital peripherals
  - Four 16-bit timer, counter, and PWM (TCPWM) blocks
  - I^2C, 1 Mbps bus speed
  - USB 2.0 certified Full-Speed (FS) 12 Mbps peripheral interface (TID#10840032) using internal oscillator[2]
  - Full CAN 2.0b, 16 Rx, 8 Tx buffers
  - 20 to 24 universal digital blocks (UDB), programmable to create any number of functions:
    - 8-, 16-, 24-, and 32-bit timers, counters, and PWMs
    - I^2C, UART, SPI, I2S, LIN 2.0 interfaces
    - Cyclic redundancy check (CRC)
    - Pseudo random sequence (PRS) generators
    - Quadrature decoders
    - Gate-level logic functions
- Programmable clocking
  - 3- to 74-MHz internal oscillator, 1% accuracy at 3 MHz
  - 4- to 25-MHz external crystal oscillator
  - Internal PLL clock generation up to 80 MHz
  - Low-power internal oscillator at 1, 33, and 100 kHz
  - 32.768-kHz external watch crystal oscillator
  - 12 clock dividers routable to any peripheral or I/O
- Analog peripherals
  - Configurable 8- to 12-bit delta-sigma ADC
  - Up to two 12-bit SAR ADCs
  - Four 8-bit DACs
  - Four comparators
  - Four opamps
  - Four programmable analog blocks, to create:
    - Programmable gain amplifier (PGA)
    - Transimpedance amplifier (TIA)
    - Mixer
    - Sample and hold circuit
  - CapSense® support, up to 62 sensors
  - 1.024 V ±0.1% internal voltage reference
- Versatile I/O system
  - 48 to 72 I/O pins – up to 62 general-purpose I/Os (GPIOs)
  - Up to eight performance I/O (SIO) pins
    - 25 mA current sink
    - Programmable input threshold and output high voltages
    - Can act as a general-purpose comparator
    - Hot swap capability and overvoltage tolerance
  - Two USBIO pins that can be used as GPIOs
  - Route any digital or analog peripheral to any GPIO
  - LCD direct drive from any GPIO, up to 46 × 16 segments
  - CapSense support from any GPIO
  - 1.2-V to 5.5-V interface voltages, up to four power domains
- Programming, debug, and trace
  - JTAG (4-wire), serial wire debug (SWD) (2-wire), single wire viewer (SWV), and Traceport (5-wire) interfaces
  - ARM debug and trace modules embedded in the CPU core
  - Bootloader programming through I^2C, SPI, UART, USB, and other interfaces
- Package options: 68-pin QFN,100-pin TQFP, and 99-pin CSP
- Development support with free PSoC Creator™ tool
  - Schematic and firmware design support
  - Over 100 PSoC Components™ integrate multiple ICs and system interfaces into one PSoC. Components are free embedded ICs represented by icons. Drag and drop component icons to design systems in PSoC Creator.
  - Includes free GCC compiler, supports Keil/ARM MDK compiler
  - Supports device programming and debugging

Notes
1. The maximum storage temperature is 150 °C in compliance with JEDEC Standard JESD22-A103, High Temperature Storage Life.
2. This feature on select devices only. See Ordering Information on page 119 for details.

More Information

Cypress provides a wealth of data at www.cypress.com to help you to select the right PSoC device for your design, and to help you to quickly and effectively integrate the device into your design. For a comprehensive list of resources, see the knowledge base article KBA86521, How to Design with PSoC 3, PSoC 4, and PSoC 5LP. Following is an abbreviated list for PSoC 5LP:

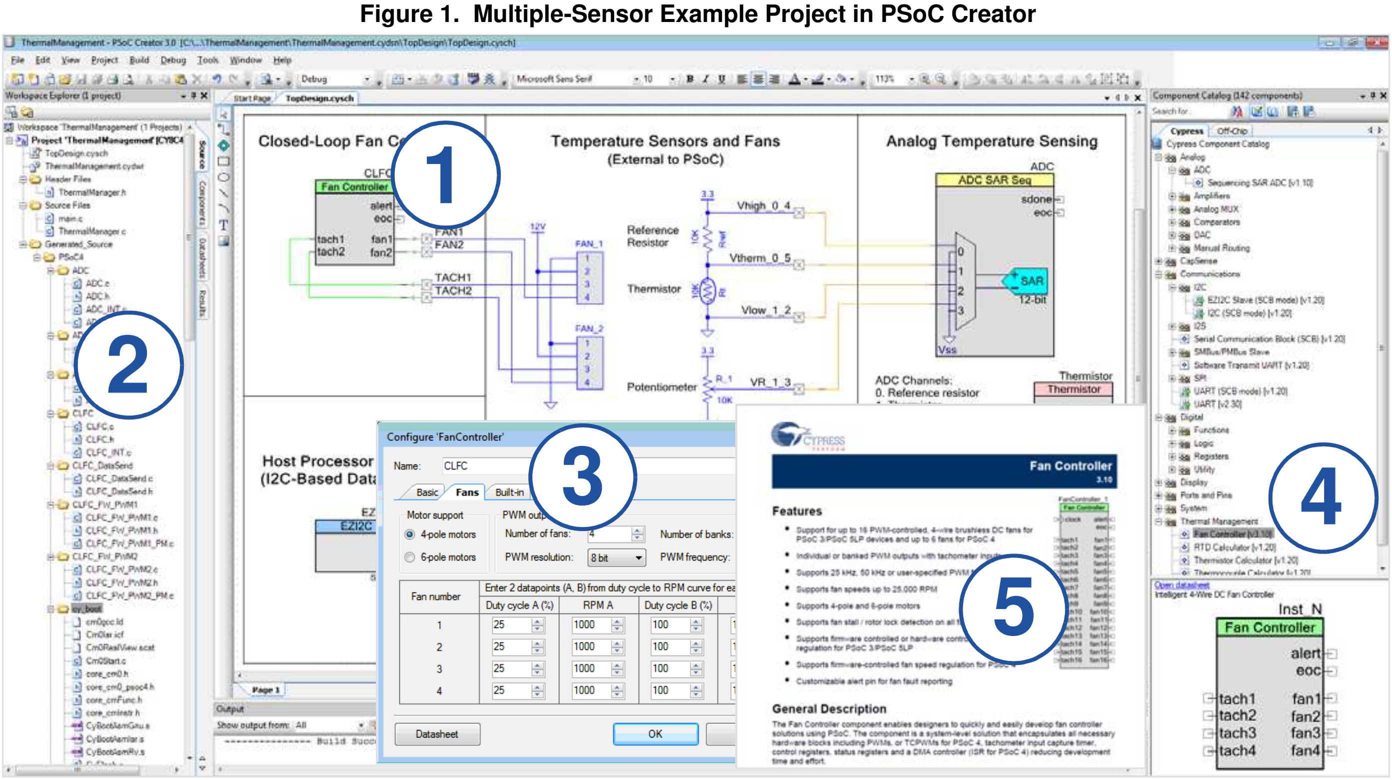
- Overview: PSoC Portfolio, PSoC Roadmap
- Product Selectors: PSoC 1, PSoC 3, PSoC 4, PSoC 5LP
  In addition, PSoC Creator includes a device selection tool.
- Application notes: Cypress offers a large number of PSoC application notes and code examples covering a broad range of topics, from basic to advanced level. Recommended application notes for getting started with PSoC 5LP are:
  - AN77759: Getting Started With PSoC 5LP
  - AN77835: PSoC 3 to PSoC 5LP Migration Guide
  - AN61290: Hardware Design Considerations
  - AN57821: Mixed Signal Circuit Board Layout
  - AN58304: Pin Selection for Analog Designs
  - AN81623: Digital Design Best Practices
  - AN73854: Introduction To Bootloaders
- Development Kits:
  - CY8CKIT-059 is a low-cost platform for prototyping, with a unique snap-away programmer and debugger on the USB connector.
  - CY8CKIT-050 is designed for analog performance, for developing high-precision analog, low-power, and low-voltage applications.
  - CY8CKIT-001 provides a common development platform for any one of the PSoC 1, PSoC 3, PSoC 4, or PSoC 5LP families of devices.
  - The MiniProg3 device provides an interface for flash programming and debug.
- Technical Reference Manuals (TRM)
  - Architecture TRM
  - Registers TRM
- Programming Specification

PSoC Creator

PSoC Creator is a free Windows-based Integrated Design Environment (IDE). It enables concurrent hardware and firmware design of PSoC 3, PSoC 4, and PSoC 5LP based systems. Create designs using classic, familiar schematic capture supported by over 100 pre-verified, production-ready PSoC Components; see the list of component datasheets. With PSoC Creator, you can:

1. Drag and drop component icons to build your hardware system design in the main design workspace
2. Codesign your application firmware with the PSoC hardware, using the PSoC Creator IDE C compiler
3. Configure components using the configuration tools
4. Explore the library of 100+ components
5. Review component datasheets

Figure 1. Multiple-Sensor Example Project in PSoC Creator



Contents

1. Architectural Overview

Introducing the CY8C56LP family of ultra low power, flash Programmable System-on-Chip (PSoC) devices, part of a scalable 8-bit PSoC 3 and 32-bit PSoC 5LP platform. The CY8C56LP 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

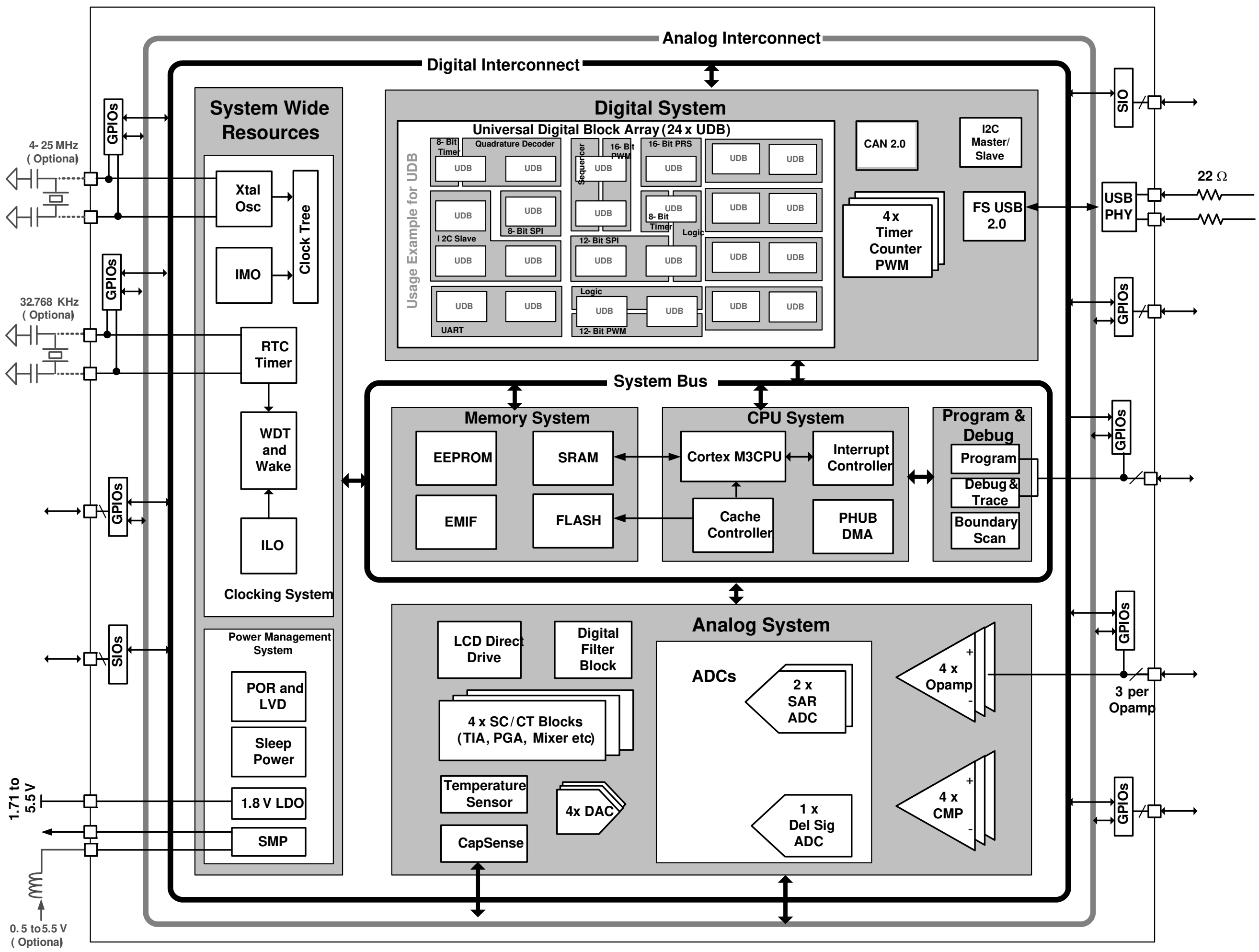


Figure 1-1 illustrates the major components of the CY8C56LP family. They are:

- 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. You 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 CY8C56LP family these blocks can include four 16-bit timer, counter, and PWM blocks; I^2C slave, master, and multi-master; Full-Speed USB; and Full CAN 2.0.

For more details on the peripherals see the "Example Peripherals" section on page 39 of this datasheet. For information on UDBs, DSI, and other digital blocks, see the "Digital Subsystem" section on page 38 of this datasheet.

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 0.1% error over temperature and voltage. The configurable analog subsystem includes:

- Analog muxes
- Comparators
- Analog mixers
- Voltage references
- ADCs
- DACs
- DFB

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.

Some CY8C56LP devices offer a fast, accurate, configurable delta-sigma ADC with these features:

- Less than 100 µV offset
- A gain error of 0.2 percent
- INL less than ±1 LSB
- DNL less than ±1 LSB
- SINAD better than 66 dB

The CY8C56LP family also offers one or two successive approximation register (SAR) ADCs, depending on device selected. Featuring 12-bit conversions at up to 1 M samples per second, they also offer low nonlinearity and offset errors and SNR better than 70 dB. They are well suited for a variety of higher speed analog applications.

The output of either ADC can optionally feed the programmable DFB via DMA without CPU intervention. You can configure the DFB to perform IIR and FIR digital filters and several user defined custom functions. The DFB can implement filters with up to 64 taps. It can perform a 48-bit multiply-accumulate (MAC) operation in one clock cycle.

Four high speed voltage or current DACs support 8-bit output signals at an update rate of up to 8 Msps. They can be routed out of any GPIO pin. You can create higher resolution voltage PWM 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 ADCs, DACs, and DFB, 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 50 of this datasheet for more details.

PSoC's CPU subsystem is built around a 32-bit three-stage pipelined ARM Cortex-M3 processor running at up to 80 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, flash cache, and RAM. 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 flash cache also reduces system power consumption by allowing less frequent flash access.

PSoC's nonvolatile subsystem consists of flash, byte-writeable EEPROM, and nonvolatile configuration options. It provides up to 256 KB of on-chip flash. The CPU can reprogram individual blocks of flash, enabling boot loaders. You can enable an ECC for high reliability applications. 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. Additionally, selected configuration options such as boot speed and pin drive mode are stored in nonvolatile memory. This allows settings to activate immediately after power on reset (POR).

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 VDDIO pins. Every GPIO has analog I/O, LCD drive, CapSense, flexible interrupt generation, slew rate control, and digital I/O capability. The SIOs on PSoC allow V_{OH} to be set independently of VDDIO 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 FS 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 32 of this datasheet.

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 1% accuracy at 3 MHz. The IMO can be configured to run from 3 MHz up to 74 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 80 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 RTC applications. The clocks, together with programmable clock dividers, provide the flexibility to integrate most timing requirements.

The CY8C56LP family supports a wide supply operating range from 1.71 to 5.5 V. This allows operation from regulated supplies such as 1.8 ± 5%, 2.5 V ±10%, 3.3 V ± 10%, or 5.0 V ± 10%, or directly from a wide range of battery types. In addition, it provides an integrated high efficiency synchronous boost converter that can power the device from supply voltages as low as 0.5 V. This enables the device to be powered directly from a single battery. In addition, you can use the boost converter to generate other voltages required by the device, such as a 3.3 V supply for LCD glass drive. The boost's output is available on the VBOOST pin, allowing other devices in the application to be powered from the PSoC.

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 with RTC. In the second mode the optional 32.768-kHz watch crystal runs continuously and maintains an accurate RTC.

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 3.1 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 26 of this datasheet.

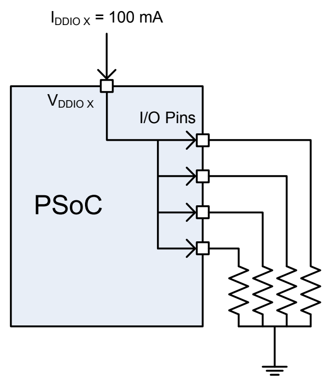
PSoC uses JTAG (4 wire) or SWD (2 wire) interfaces for programming, debug, and test. Using these standard interfaces you can debug or program the PSoC with a variety of hardware solutions from Cypress or third party vendors. The Cortex-M3 debug and trace modules include Flash Patch and Breakpoint (FPB), Data Watchpoint and Trace (DWT), Embedded Trace Macrocell (ETM), and Instrumentation Trace Macrocell (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 61 of this datasheet.

2. Pinouts

Each VDDIO pin powers a specific set of I/O pins. (The USBIOs are powered from VDDD.) Using the VDDIO pins, a single PSoC can support multiple voltage levels, reducing the need for off-chip level shifters. The black lines drawn on the pinout diagrams in Figure 2-3 and Figure 2-4, as well as Table 2-1, show the pins that are powered by each VDDIO.

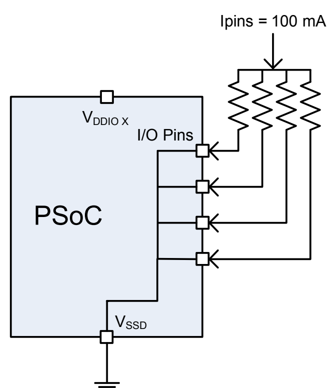
Each VDDIO may source up to 100 mA total to its associated I/O pins, as shown in Figure 2-1.

Figure 2-1. VDDIO Current Limit



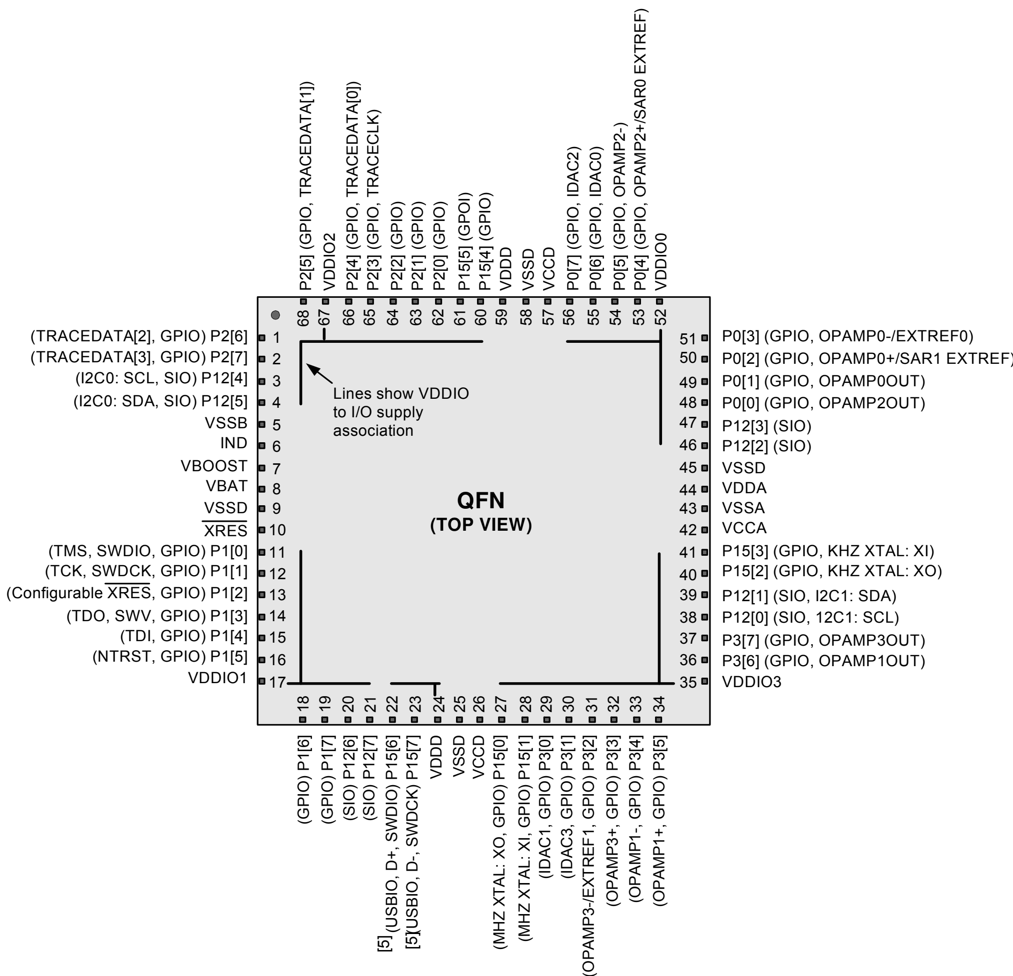
Conversely, for the 100-pin and 68-pin devices, the set of I/O pins associated with any VDDIO may sink up to 100 mA total, as shown in Figure 2-2.

Figure 2-2. I/O Pins Current Limit



Note
3. Pins are Do Not Use (DNU) on devices without USB. The pin must be left floating.

Figure 2-3. 68-pin QFN Part Pinout[4]



Notes

4. The center pad on the QFN package should be connected to digital ground (VSSD) for best mechanical, thermal, and electrical performance. If not connected to ground, it should be electrically floated and not connected to any other signal. For more information, see AN72845, Design Guidelines for QFN Devices.
5. Pins are Do Not Use (DNU) on devices without USB. The pin must be left floating.

Figure 2-4. 100-pin TQFP Part Pinout

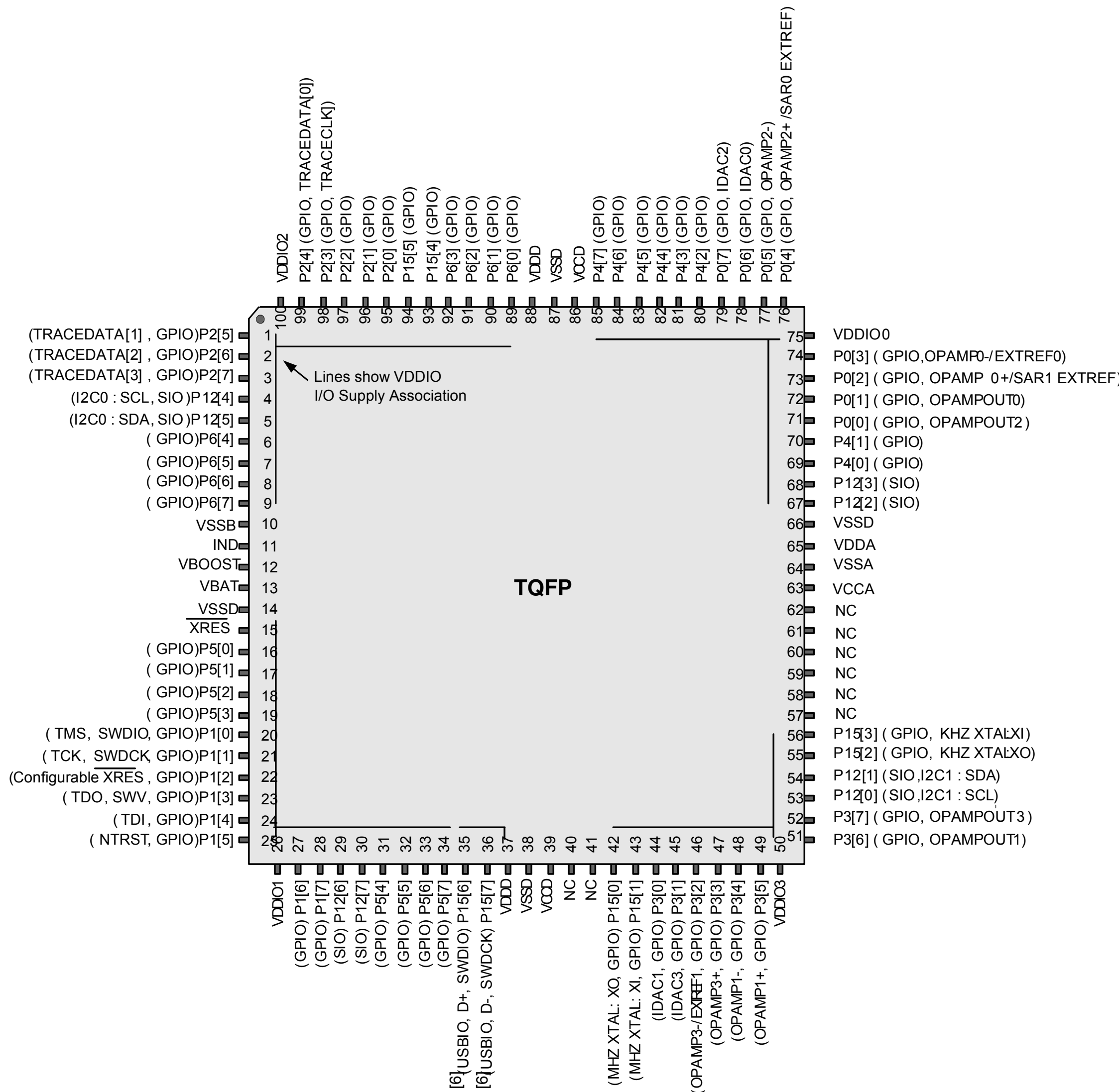


Table 2-1. V_{DDIO} and Port Pin Associations

VDDIO	Port Pins
VDDIO0	P0[7:0], P4[7:0], P12[3:2]
VDDIO1	P1[7:0], P5[7:0], P12[7:6]
VDDIO2	P2[7:0], P6[7:0], P12[5:4], P15[5:4]
VDDIO3	P3[7:0], P12[1:0], P15[3:0]
VDDD	P15[7:6] (USB D+, D-)

Note
6. Pins are Do Not Use (DNU) on devices without USB. The pin must be left floating.

Table 2-2 shows the pinout for the 99-pin CSP package. Since there are four V_{DDIO} pins, the set of I/O pins associated with any V_{DDIO} may sink up to 100 mA total, same as for the 100-pin and 68-pin devices.

Table 2-2. CSP Pinout

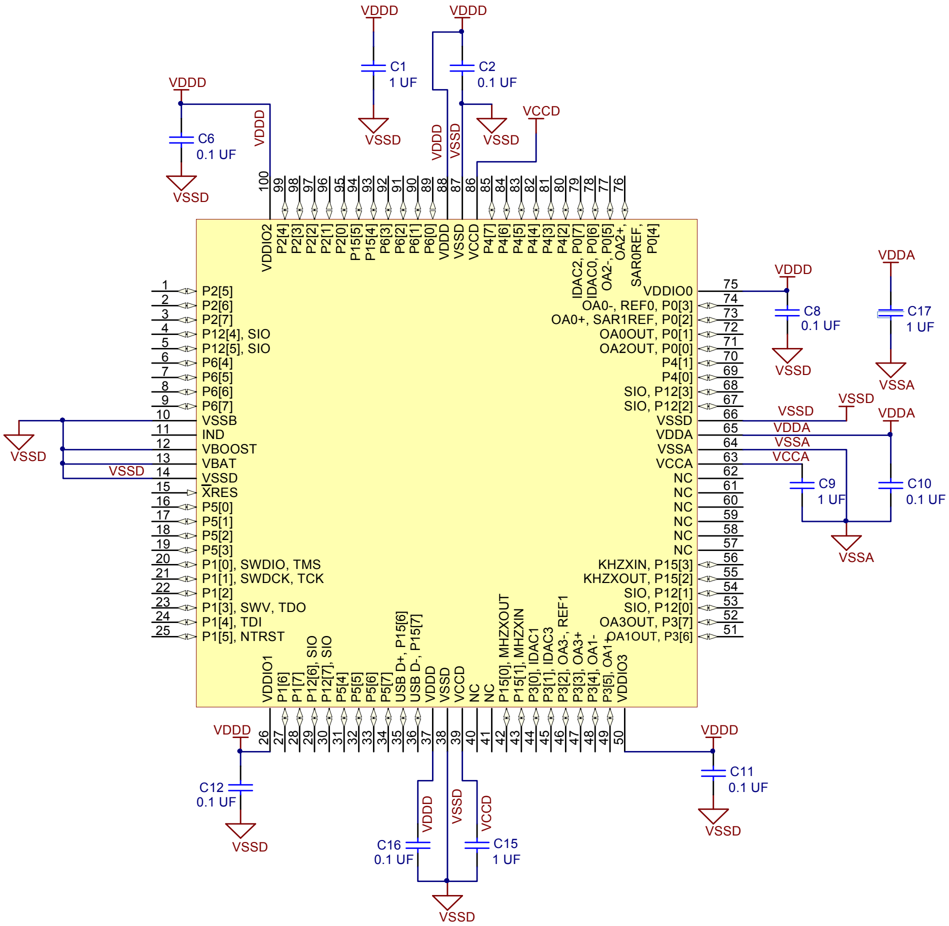
Ball	Name	Ball	Name	Ball	Name	Ball	Name
E5	P2[5]	L2	VIO1	B2	P3[6]	C8	VIO0
G6	P2[6]	K2	P1[6]	B3	P3[7]	D7	P0[4]
G5	P2[7]	C9	P4[2]	C3	P12[0]	E7	P0[5]
H6	P12[4]	E8	P4[3]	C4	P12[1]	B9	P0[6]
K7	P12[5]	K1	P1[7]	E3	P15[2]	D8	P0[7]
L8	P6[4]	H2	P12[6]	E4	P15[3]	D9	P4[4]
J6	P6[5]	F4	P12[7]	A1	NC	F8	P4[5]
H5	P6[6]	J1	P5[4]	A9	NC	F7	P4[6]
J5	P6[7]	H1	P5[5]	L1	NC	E6	P4[7]
L7	VSSB	F3	P5[6]	L9	NC	E9	VCCD
K6	Ind	G1	P5[7]	A3	VCCA	F9	VSSD
L6	VBOOST	G2	P15[6]	A4	VSSA	G9	VDDD
K5	VBAT	F2	P15[7]	B7	VSSA	H9	P6[0]
L5	VSSD	E2	VDDD	B8	VSSA	G8	P6[1]
L4	XRES	F1	VSSD	C7	VSSA	H8	P6[2]
J4	P5[0]	E1	VCCD	A5	VDDA	J9	P6[3]
K4	P5[1]	D1	P15[0]	A6	VSSD	G7	P15[4]
K3	P5[2]	D2	P15[1]	B5	P12[2]	F6	P15[5]
L3	P5[3]	C1	P3[0]	A7	P12[3]	F5	P2[0]
H4	P1[0]	C2	P3[1]	C5	P4[0]	J7	P2[1]
J3	P1[1]	D3	P3[2]	D5	P4[1]	J8	P2[2]
H3	P1[2]	D4	P3[3]	B6	P0[0]	K9	P2[3]
J2	P1[3]	B4	P3[4]	C6	P0[1]	H7	P2[4]
G4	P1[4]	A2	P3[5]	A8	P0[2]	K8	VIO2
G3	P1[5]	B1	VIO3	D6	P0[3]		

Figure 2-5 and Figure 2-6 on page 11 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 VDDD must be connected together.
- The two pins labeled VCCD must be connected together, with capacitance added, as shown in Figure 2-5 on page 10 and Power System on page 26. The trace between the two VCCD pins should be as short as possible.
- The two pins labeled VSSD must be connected together.

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.

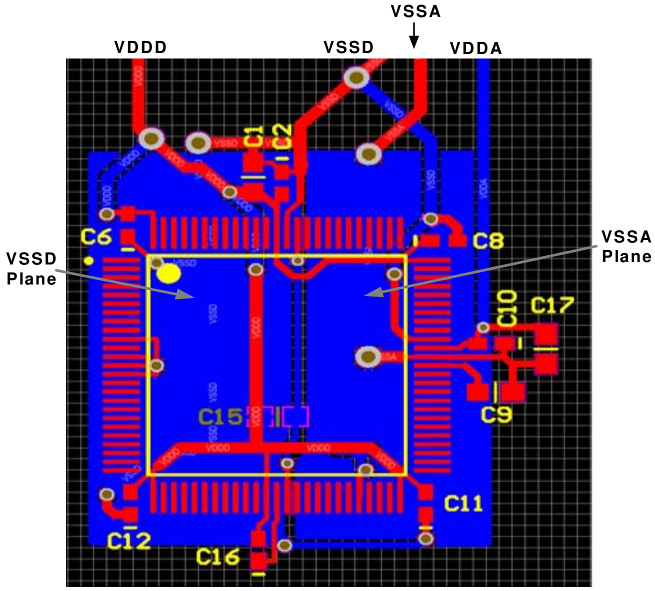
Figure 2-5. Example Schematic for 100-pin TQFP Part with Power Connections



Note The two VCCD pins must be connected together with as short a trace as possible. A trace under the device is recommended, as shown in Figure 2-6.

For more information on pad layout, refer to http://www.cypress.com/cad-resources/psoc-5lp-cad-libraries.

Figure 2-6. Example PCB Layout for 100-pin TQFP Part for Optimal Analog Performance



3. Pin Descriptions

IDAC0, IDAC1, IDAC2, IDAC3

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

Opamp0out, Opamp1out, Opamp2out, Opamp3out

High current output of uncommitted opamp[7].

Extref0, Extref1

External reference input to the analog system.

SAR0 EXTREF, SAR1 EXTREF

External references for SAR ADCs

Opamp0–, Opamp1–, Opamp2–, Opamp3–

Inverting input to uncommitted opamp.

Opamp0+, Opamp1+, Opamp2+, Opamp3+

Noninverting input to uncommitted opamp.

GPIO

General purpose I/O pin provides interfaces to the CPU, digital peripherals, analog peripherals, interrupts, LCD segment drive, and CapSense[7].

I2C0: SCL, I2C1: SCL

I^2C SCL line providing wake from sleep on an address match. Any I/O pin can be used for I^2C SCL if wake from sleep is not required.

I2C0: SDA, I2C1: SDA

I^2C SDA line providing wake from sleep on an address match. Any I/O pin can be used for I^2C SDA if wake from sleep is not required.

Ind

Inductor connection to boost pump.

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.

nTRST

Optional JTAG Test Reset programming and debug port connection to reset the JTAG connection.

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.

SWDIO

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

TCK

JTAG Test Clock programming and debug port connection.

TDI

JTAG Test Data In programming and debug port connection.

TDO

JTAG Test Data Out programming and debug port connection.

TMS

JTAG Test Mode Select programming and debug port connection.

TRACECLK

Cortex-M3 TRACEPORT connection, clocks TRACEDATA pins.

TRACEDATA[3:0].

Cortex-M3 TRACEPORT connections, output data.

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 VDDD instead of from a VDDIO. Pins are Do Not Use (DNU) on devices without USB.

Note

7. GPIOs with opamp outputs are not recommended for use with CapSense.

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 VDDD instead of from a VDDIO. Pins are Do Not Use (DNU) on devices without USB.

VBOOST

Power sense connection to boost pump.

VBAT

Battery supply to boost pump.

VCCA

Output of the analog core regulator or the input to the analog core. Requires a 1uF capacitor to VSSA. The regulator output is not designed to drive external circuits. **Note that if you use the device with an external core regulator (externally regulated mode), the voltage applied to this pin must not exceed the allowable range of 1.71 V to 1.89 V.** When using the internal core regulator, (internally regulated mode, the default), do not tie any power to this pin. For details see Power System on page 26.

VCCD.

Output of the digital core regulator or the input to the digital core. The two VCCD pins must be shorted together, with the trace between them as short as possible, and a 1uF capacitor to VSSD. The regulator output is not designed to drive external circuits. **Note that if you use the device with an external core regulator (externally regulated mode), the voltage applied to this pin must not exceed the allowable range of 1.71 V to 1.89 V.** When using the internal core regulator (internally regulated mode, the default), do not tie any power to this pin. For details see Power System on page 26.

VDDA

Supply for all analog peripherals and analog core regulator. **VDDA must be the highest voltage present on the device. All other supply pins must be less than or equal to VDDA.**

VDDD

Supply for all digital peripherals and digital core regulator. VDDD must be less than or equal to VDDA.

VSSA

Ground for all analog peripherals.

VSSB

Ground connection for boost pump.

VSSD

Ground for all digital logic and I/O pins.

VDDIO0, VDDIO1, VDDIO2, VDDIO3

Supply for I/O pins. Each VDDIO must be tied to a valid operating voltage (1.71 V to 5.5 V), and must be less than or equal to VDDA.

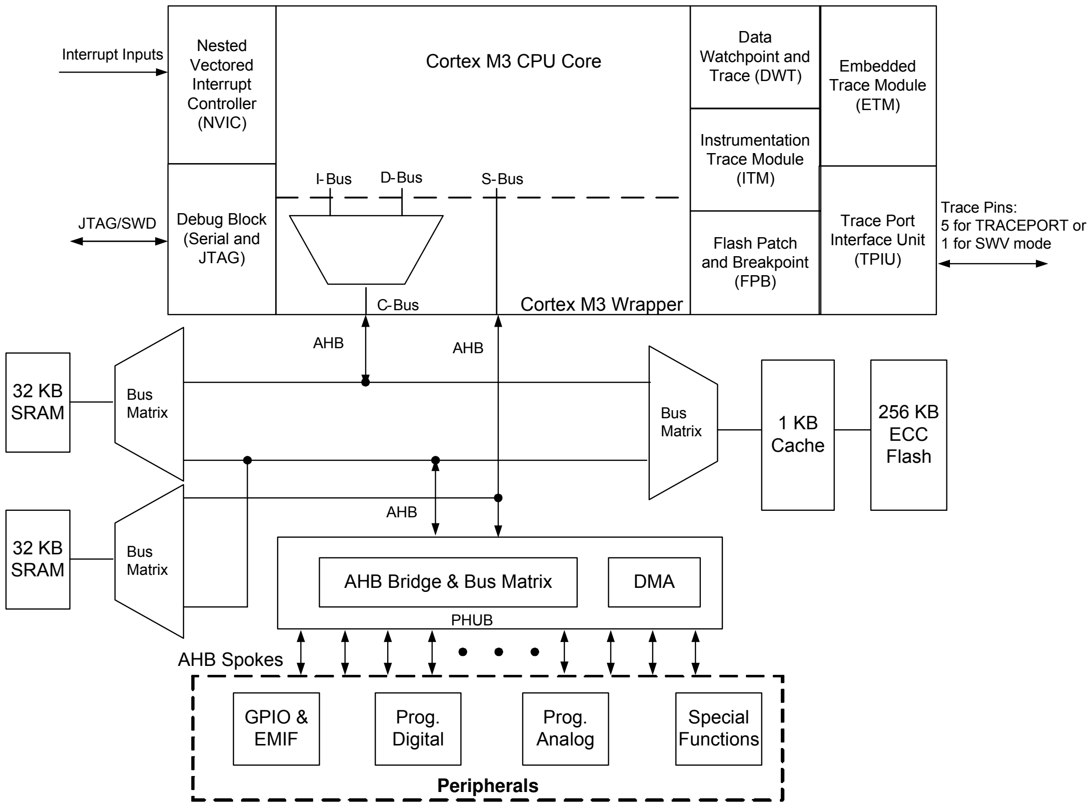
$\overline{\text{XRES}}$. External reset pin. Active low with internal pull-up.

4. CPU

4.1 ARM Cortex-M3 CPU

The CY8C56LP 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 Nested Vectored Interrupt Controller (NVIC), tightly integrated with the CPU core
- Full featured debug and trace modules, tightly integrated with the CPU core
- Up to 256 KB of flash memory, 2 KB of EEPROM, and 64 KB of SRAM
- Cache controller
- Peripheral HUB (PHUB)
- DMA controller
- External Memory Interface (EMIF)

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.
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 CY8C56LP family has 1 KB, 4-way set-associative instruction cache between the CPU and the flash memory. This improves instruction execution rate and reduces system power consumption by requiring less frequent flash access.

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, EMIF
2	PHUB local configuration, Power manager, Clocks, IC, SWV, EEPROM, Flash programming interface
3	Analog interface and trim, Decimator
4	USB, I^{2}C, CAN, Timers, Counters, and PWMs
5	DFB
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 128 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.

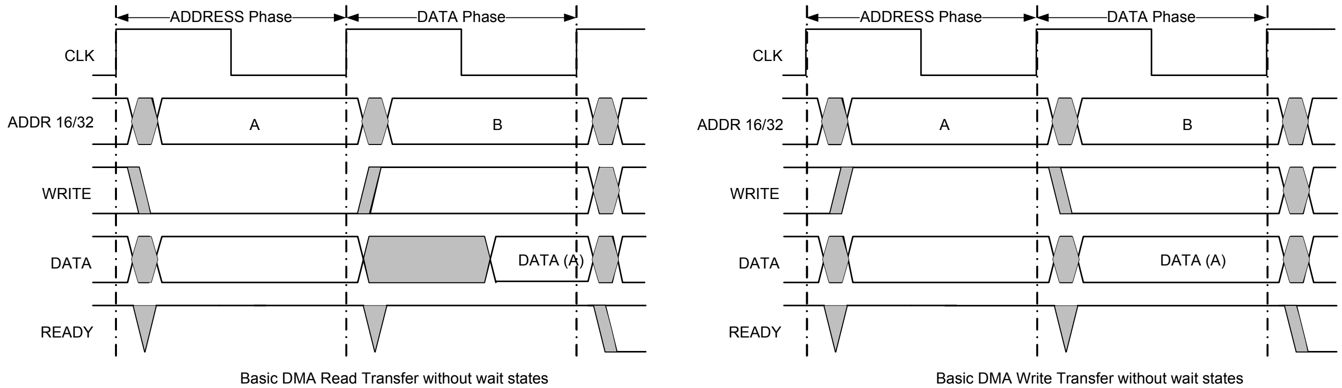
4.3.4 Transaction Modes Supported

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



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^{2}C 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	Pcrogrammable	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 44.

The Nested Vectored Interrupt Controller (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.
- Eight priority levels, with dynamic priority control.
- 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/ECC	phub_termout0[1]	udb_intr[1]
2	18	Reserved	phub_termout0[2]	udb_intr[2]
3	19	Sleep (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^{2}C	phub_termout0[15]	udb_intr[15]
16	32	CAN	phub_termout1[0]	udb_intr[16]
17	33	Timer/Counter0	phub_termout1[1]	udb_intr[17]
18	34	Timer/Counter1	phub_termout1[2]	udb_intr[18]
19	35	Timer/Counter2	phub_termout1[3]	udb_intr[19]
20	36	Timer/Counter3	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	LCD	phub_termout1[11]	udb_intr[27]
28	44	DFB Int	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

CY8C56LP 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, bulk data storage, and optional ECC data. 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 Error Correcting Codes (ECC). If ECC is not used this space can store device configuration data and bulk user data. User code may not be run out of the ECC flash memory section. ECC can correct one bit error and detect two bit errors per 8 bytes of firmware memory; an interrupt can be generated when an error is detected. The flash output is 9 bytes wide with 8 bytes of data and 1 byte of ECC data.

The CPU or DMA controller read both user code and bulk data located in flash through the cache controller. This provides higher CPU performance. If ECC is enabled, the cache controller also performs error checking and correction.

Flash programming is performed through a special interface and preempts code execution out of flash. Code execution may be done out of SRAM during flash programming.

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 both the SWD and JTAG interfaces. 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 ECC or configuration 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 64). 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 datasheets. 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 CY8C56LP 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 128 rows of 16 bytes each. The factory default values of all EEPROM bytes are 0.

Because the EEPROM is mapped to the Cortex-M3 Peripheral region, the CPU cannot execute out of EEPROM. There is no ECC hardware associated with EEPROM. If ECC is required it must be handled in firmware.

It can take as much as 20 milliseconds to write to EEPROM or flash. During this time the device should not be reset, or unexpected changes may be made to portions of EEPROM or flash. Reset sources (see *Section 6.3.1*) include XRES pin, software reset, and watchdog; care should be taken to make sure that these are not inadvertently activated. In addition, the low voltage detect circuits should be configured to generate an interrupt instead of a reset.

5.5 Nonvolatile Latches (NVLs)

PSoC has a 4-byte array of nonvolatile latches (NVLs) that are used to configure the device at reset. The NVL register map is shown in Table 5-3.

Table 5-2. Device Configuration NVL Register Map

Register Address	7	6	5	4	3	2	1	0
0x00	PRT3RDM[1:0]		PRT2RDM[1:0]		PRT1RDM[1:0]		PRT0RDM[1:0]	
0x01	PRT12RDM[1:0]		PRT6RDM[1:0]		PRT5RDM[1:0]		PRT4RDM[1:0]	
0x02	XRESMEN	DBGEN					PRT15RDM[1:0]	
0x03	DIG_PHS_DLY[3:0]				ECCEN	DPS[1:0]		CFGSPEED

The details for individual fields and their factory default settings are shown in Table 5-3:.

Table 5-3. Fields and Factory Default Settings

Field	Description	Settings
PRTxRDM[1:0]	Controls reset drive mode of the corresponding I/O port. See "Reset Configuration" on page 38. All pins of the port are set to the same mode.	00b (default) - high impedance analog 01b - high impedance digital 10b - resistive pull up 11b - resistive pull down
XRESMEN	Controls whether pin P1[2] is used as a GPIO or as an external reset. P1[2] is generally used as a GPIO, and not as an external reset.	0 (default) - GPIO 1 - external reset
DBGEN	Debug Enable allows access to the debug system, for third-party programmers.	0 - access disabled 1 (default) - access enabled
CFGSPEED	Controls the speed of the IMO-based clock during the device boot process, for faster boot or low-power operation	0 (default) - 12 MHz IMO 1 - 48 MHz IMO
DPS[1:0]	Controls the usage of various P1 pins as a debug port. See "Programming, Debug Interfaces, Resources" on page 61.	00b - 5-wire JTAG 01b (default) - 4-wire JTAG 10b - SWD 11b - debug ports disabled
ECCEN	Controls whether ECC flash is used for ECC or for general configuration and data storage. See "Flash Program Memory" on page 19.	0 - ECC disabled 1 (default) - ECC enabled
DIG_PHS_DLY[3:0]	Selects the digital clock phase delay.	See the TRM for details.

Although PSoC Creator provides support for modifying the device configuration NVLs, the number of NVL erase/write cycles is limited – see "Nonvolatile Latches (NVL)" on page 109.

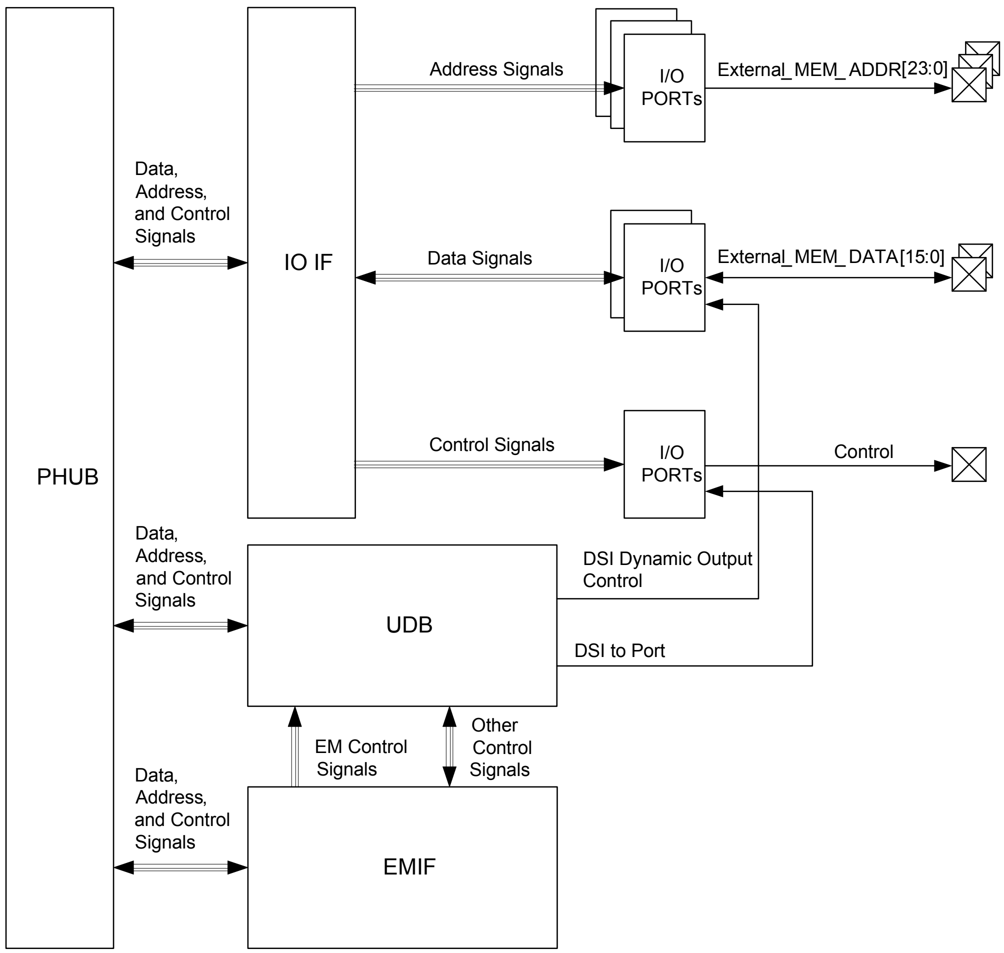
5.6 External Memory Interface

CY8C56LP provides an External Memory Interface (EMIF) for connecting to external memory devices. The connection allows read and write accesses to external memories. The EMIF operates in conjunction with UDBs, I/O ports, and other hardware to generate external memory address and control signals. At 33 MHz, each memory access cycle takes four bus clock cycles. Figure 5-1 is the EMIF block diagram. The EMIF supports synchronous and asynchronous memories. The CY8C56LP only supports one type of external memory device at a time.

External memory is located in the Cortex-M3 external RAM space; it can use up to 24 address bits. See Table 5-4 on page 22Memory Map on page 22. The memory can be 8 or 16 bits wide.

Cortex-M3 instructions can be fetched from external memory if it is 16-bit. Other limitations apply; for details, see application note AN89610, PSoC® 4 and PSoC 5LP ARM Cortex Code Optimization. There is no provision for code security in external memory. If code must be kept secure, then it should be placed in internal flash. See Flash Security on page 19 and Device Security on page 64.

Figure 5-1. EMIF Block Diagram



5.7 Memory Map

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

5.7.1 Address Map

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

Table 5-4. 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 5-5. 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
0x40004800–0x400048FF	Cache controller
0x40004900–0x400049FF	I^2C controller
0x40004E00–0x40004EFF	Decimator
0x40004F00–0x40004FFF	Fixed timer/counter/PWMs
0x40005000–0x400051FF	I/O ports control
0x40005400–0x400054FF	External Memory Interface (EMIF) control registers
0x40005800–0x40005FFF	Analog Subsystem Interface
0x40006000–0x400060FF	USB Controller
0x40006400–0x40006FFF	UDB Working Registers
0x40007000–0x40007FFF	PHUB Configuration
0x40008000–0x400087FF	EEPROM
0x4000A000–0x4000A400	CAN
0x4000C000–0x4000C800	Digital Filter Block
0x40010000–0x4001FFFF	Digital Interconnect Configuration
0x48000000–0x48007FFF	Flash ECC Bytes
0x60000000–0x60FFFFFF	External Memory Interface (EMIF)
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.

5.7.2 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. For the majority of systems, no external crystal is required. The IMO and PLL together can generate up to a 80 MHz clock, accurate to ±1% 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 in PSoC.

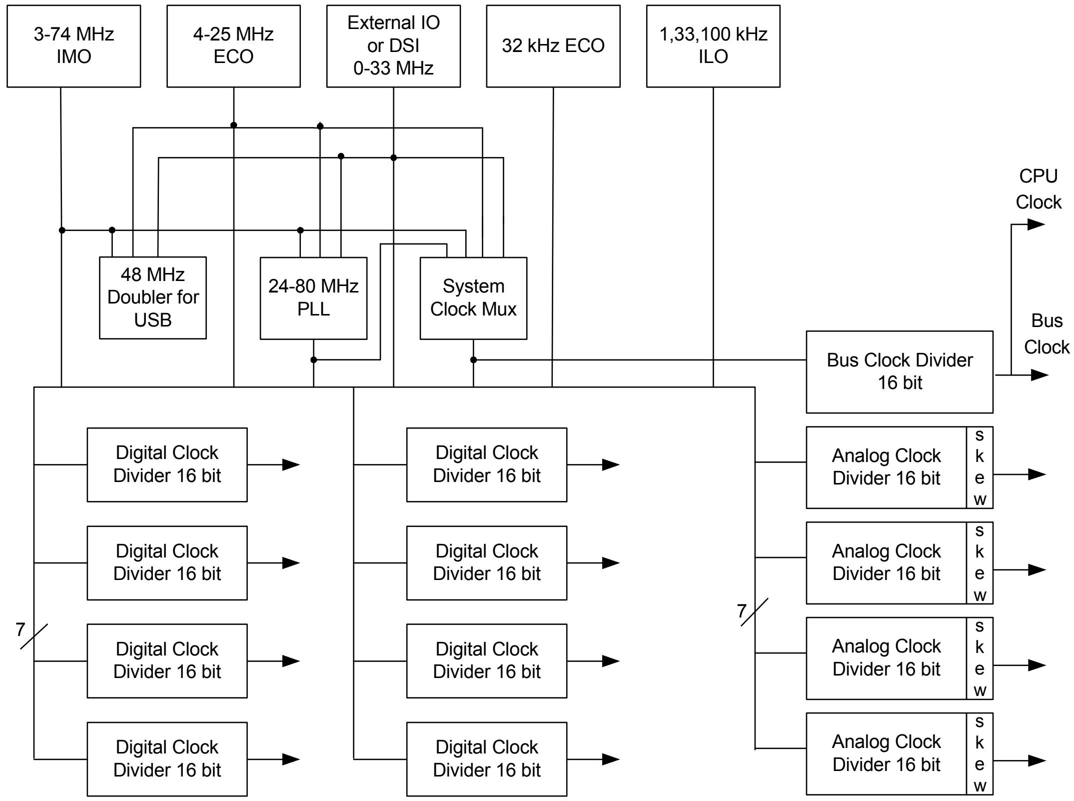
Key features of the clocking system include:

- Seven general purpose clock sources
  - 3- to 74-MHz IMO, ±1% 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 26
  - DSI signal from an external I/O pin or other logic
  - 24- to 80-MHz fractional Phase-Locked Loop (PLL) sourced from IMO, MHzECO, or DSI
  - Clock Doubler
  - 1-kHz, 33-kHz, 100-kHz ILO for Watch Dog Timer (WDT) and Sleep Timer
  - 32.768-kHz External Crystal Oscillator (kHzECO) for Real Time Clock (RTC)
- IMO has a USB mode that auto locks to USB bus clock requiring no external crystal for USB. (USB equipped parts only)
- Independently sourced clock in all clock dividers
- 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	±1% over voltage and temperature	74 MHz	±7%	13 µs max
MHzECO	4 MHz	Crystal dependent	25 MHz	Crystal dependent	5 ms typ, max is crystal dependent
DSI	0 MHz	Input dependent	33 MHz	Input dependent	Input dependent
PLL	24 MHz	Input dependent	80 MHz	Input dependent	250 µs max
Doubler	12 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

Figure 6-1 shows that there are two internal oscillators. They can be routed directly or divided. The direct routes may not have a 50% duty cycle. Divided clocks have a 50% duty cycle.

6.1.1.1 Internal Main Oscillator

In most designs the IMO is the only clock source required, due to its ±1% accuracy. 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 ±1% at 3 MHz, up to ±7% at 74 MHz. The IMO, in conjunction with the PLL, allows generation of CPU and system clocks up to the device's maximum frequency (see Phase-Locked Loop)

The IMO provides clock outputs at 3, 6, 12, 24, 48, and 74 MHz.

6.1.1.2 Clock Doubler

The clock doubler outputs a clock at twice the frequency of the input clock. The doubler works for input frequency ranges of 6 to 24 MHz (providing 12 to 48 MHz at the output). It can be configured to use a clock from the IMO, 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 80 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 watchdog timer, and 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 low power supervisory operations such as the watchdog timer and long sleep intervals using the central timewheel (CTW).

The central timewheel is a 1 kHz, free running, 13-bit counter clocked by the ILO. 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 Real Time Clock capability instead of the central timewheel.

The 100-kHz clock (CLK100K) can be used as a low power system clock to run the CPU. It can also generate time intervals using the fast timewheel.

The fast timewheel is a 5-bit counter, clocked by the 100-kHz clock. It features programmable settings and automatically resets when the terminal count is reached. An optional interrupt can be generated each time the terminal count is reached. This enables flexible, periodic interrupts of the CPU at a higher rate than is allowed using the central timewheel.

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.

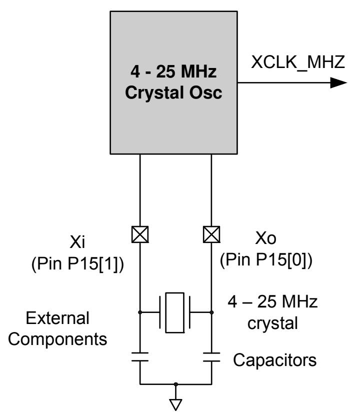
6.1.2 External Oscillators

Figure 6-1 shows that there are two external oscillators. They can be routed directly or divided. The direct routes may not have a 50% duty cycle. Divided clocks have a 50% duty cycle.

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 a wide variety of crystal types, 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 24). The GPIO pins connecting to the external crystal and capacitors are fixed. MHzECO accuracy depends on the crystal chosen.

Figure 6-2. MHzECO Block Diagram

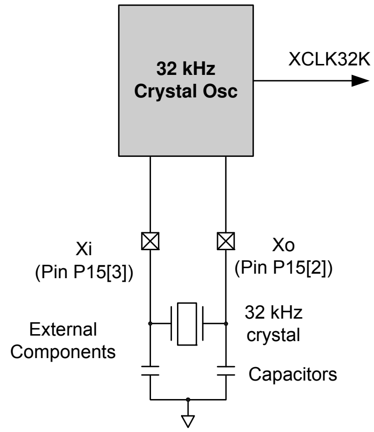


6.1.2.2 32.768 kHz ECO

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 32kHzECO also connects directly to the sleep timer and provides the source for the Real Time Clock (RTC). 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 75.

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 UDBs.

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.