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

PRELIMINARY

PSoC® 3: CY8C34 Family Data Sheet

Programmable System-on-Chip (PSoC®)

General Description

With its unique array of configurable blocks, PSoC® 3 is a true system level solution providing MCU, memory, analog, and digital peripheral functions in a single chip. The CY8C34 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 CY8C34 family can handle dozens of data acquisition channels and analog inputs on every GPIO pin. The CY8C34 family is also a high performance configurable digital system with some part numbers including interfaces such as USB, multi-master I^2C, and CAN. In addition to communication interfaces, the CY8C34 family has an easy to configure logic array, flexible routing to all I/O pins, and a high performance single cycle 8051 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 CY8C34 family provides unparalleled opportunities for analog and digital bill of materials integration while easily accommodating last minute design changes through simple firmware updates.

Features

- Single cycle 8051 CPU core
  - DC to 50 MHz operation
  - Multiply and divide instructions
  - Flash program memory, up to 64 KB, 100,000 write cycles, 20 years retention, multiple security features
  - Up to 8 KB flash ECC or configuration storage
  - Up to 8 KB SRAM memory
  - Up to 2 KB EEPROM memory, 1M cycles, 20 years retention
  - 24 channel DMA with multilayer AHB bus access
    - Programmable chained descriptors and priorities
    - High bandwidth 32-bit transfer support
- Low voltage, ultra low power
  - Wide operating voltage range: 0.5V to 5.5V
  - High efficiency boost regulator from 0.5V input to 1.8V-5.0V output
  - 0.8 mA at 3 MHz, 1.2 mA at 6 MHz, 6.6 mA at 50 MHz
  - Low power modes including:
    - 1 µA sleep mode with real time clock and low voltage detect (LVD) interrupt
    - 200 nA hibernate mode with RAM retention
- Versatile I/O system
  - 28 to 72 I/O (62 GPIO, 8 SIO, 2 USBIO[1])
  - Any GPIO to any digital or analog peripheral routability
  - LCD direct drive from any GPIO, up to 46x16 segments[1]
  - CapSense® support from any GPIO[4]
  - 1.2V to 5.5V I/O interface voltages, up to 4 domains
  - Maskable, independent IRQ on any pin or port
  - Schmitt trigger TTL inputs
  - All GPIO configurable as open drain high/low, pull up/down, High-Z, or strong output
  - Configurable GPIO pin state at power on reset (POR)
  - 25 mA sink on SIO
- Digital peripherals
  - 16 to 24 programmable PLD based Universal Digital Blocks
  - Full CAN 2.0b 16 RX, 8 TX buffers[1]
  - Full-speed (FS) USB 2.0 12 Mbps using internal oscillator[1]
  - 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, I^2C
    - Many others available in catalog
  - Library of advanced peripherals
    - Cyclic Redundancy Check (CRC)
    - Pseudo Random Sequence (PRS) generator
    - LIN Bus 2.0
    - Quadrature decoder
- Analog peripherals (1.71V ≤ Vdda ≤ 5.5V)
  - 1.024V±0.9% internal voltage reference across -40°C to +85°C (14 ppm/°C)
  - Configurable Delta-Sigma ADC with 12-bit resolution
    - Programmable gain stage: x0.25 to x16
    - 12-bit mode, 192 ksps, 70 dB SNR, 1 bit INL/DNL
  - Two 8-bit, 8 Msps IDACs or 1 Msps VDACs
  - Four comparators with 75 ns response time
  - Two uncommitted opamps with 25 mA drive capability
  - Two configurable multifunction analog blocks. Example configurations are PGA, TIA, Mixer, and Sample and Hold
  - CapSense support
- Programming, debug, and trace
  - JTAG (4 wire), Serial Wire Debug (SWD) (2 wire), and Single Wire Viewer (SWV) interfaces
  - 8 address and 1 data breakpoint
  - 4 KB instruction trace buffer
  - Bootloader programming supportable through I^2C, SPI, UART, USB, and other interfaces
- Precision, programmable clocking
  - 3 to 24 MHz internal oscillator over full temperature and voltage range
  - 4 to 33 MHz crystal oscillator for crystal PPM accuracy
  - Internal PLL clock generation up to 50 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
  - 48-pin SSOP, 48-pin QFN, 68-pin QFN, and 100-pin TQFP package options

Note
1. This feature on select devices only. See Ordering Information on page 92 for details.

Content Overview

1. Architectural Overview

Introducing the CY8C34 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 CY8C34 family provides configurable blocks of analog, digital, and interconnect circuitry around a CPU subsystem. The combination of a CPU with a very 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

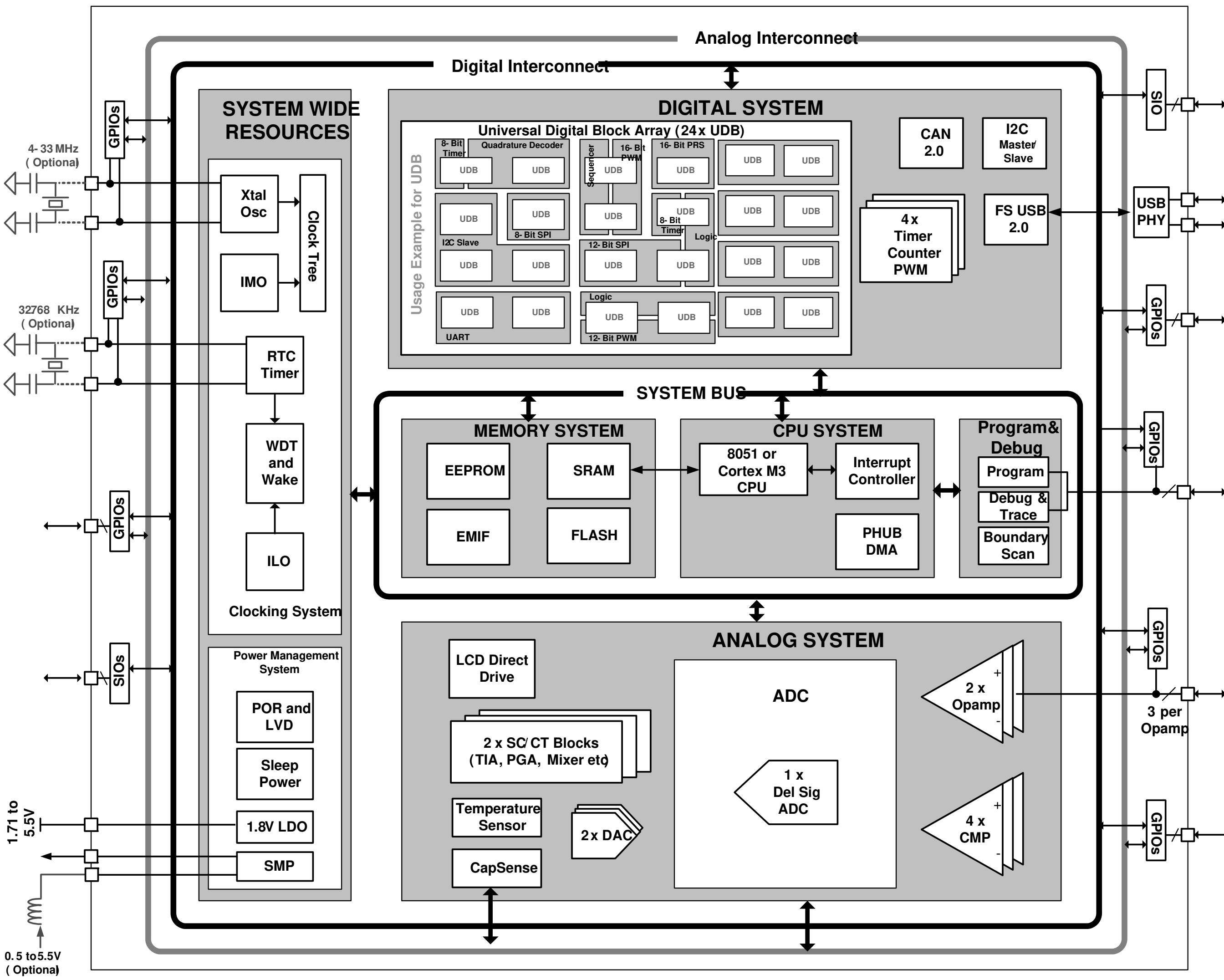


Figure 1-1 illustrates the major components of the CY8C34 family. They are:

- 8051 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 Universal Digital Blocks (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 CY8C34 family these blocks can include four 16-bit timer, counter, and PWM blocks; I^{2}C slave, master, and multi-master; Full-Speed USB; and Full CAN 2.0b.

For more details on the peripherals see the "Example Peripherals" section on page 35 of this data sheet. For information on UDBs, DSI, and other digital blocks, see the "Digital Subsystem" section on page 35 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 0.9% error over temperature and voltage. The configurable analog subsystem includes:

- Analog muxes
- Comparators
- Voltage references
- Analog-to-Digital Converter (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 heart of the analog subsystem is a fast, accurate, configurable Delta-Sigma ADC with these features:

- Less than 100 µV offset
- A gain error of 0.2%
- Integral Non Linearity (INL) less than 1 LSB
- Differential Non Linearity (DNL) less than 1 LSB
- Signal-to-noise ratio (SNR) better than 70 dB (Delta-Sigma) in 12-bit mode

This converter addresses a wide variety of precision analog applications including some of the most demanding sensors.

Two high speed voltage or current DACs support 8-bit output signals at update rate of 8 Msps in current DAC (IDAC) and 1 Msps in voltage DAC (VDAC). 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 ADC and DACs, the analog subsystem provides multiple:

- 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 48 of this data sheet for more details.

PSoC's 8051 CPU subsystem is built around a single cycle pipelined 8051 8-bit processor running at up to 50 MHz. The CPU subsystem includes a programmable nested vector interrupt controller, DMA controller, and RAM. PSoC's nested vector interrupt controller provides low latency by allowing the CPU to vector directly to the first address of the interrupt service routine, bypassing the jump instruction required by other architectures. 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 single cycle 8051 CPU runs ten times faster than a standard 8051 processor. The processor speed itself is configurable allowing active power consumption to be tuned for specific applications.

PSoC's nonvolatile subsystem consists of flash, byte-writeable EEPROM, and nonvolatile configuration options. It provides up to 64 KB of on-chip flash. The CPU can reprogram individual blocks of flash, enabling boot loaders. The designer can enable an Error Correcting Code (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. Up to 2 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[1], CapSense[4], flexible interrupt generation, slew rate control, and digital I/O capability. The SIOs on PSoC allow Voh 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 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 29 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 1% accuracy at 3 MHz. The IMO can be configured to run from 3 MHz up to 24 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 50 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 CY8C34 family supports a wide supply operating range from 1.71 to 5.5V. This allows operation from regulated supplies such as 1.8 ± 5%, 2.5V ±10%, 3.3V ± 10%, or 5.0V ± 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 0.5V. This enables the device to be powered directly from a single battery or solar cell. In addition, the designer can use the boost converter to generate other voltages required by the device, such as a 3.3V 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 200 nA hibernate mode with RAM retention and a 1 µA sleep mode with real time clock (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 1.2 mA when the CPU is running at 6 MHz or 0.8 mA running at 3 MHz.

The details of the PSoC power modes are covered in the "Power System" section on page 25 of this data sheet.

PSoC uses JTAG (4 wire) or Serial Wire Debug (SWD) (2 wire) interfaces for programming, debug, and test. The 1-wire Single Wire Viewer (SWV) may also be used for "printf" style debugging. By combining SWD and SWV, the designer can implement a full debugging interface with just three pins. Using these standard interfaces enables the designer to debug or program the PSoC with a variety of hardware solutions from Cypress or third party vendors. PSoC supports on-chip break points and 4 KB instruction and data race memory for debug. Details of the programming, test, and debugging interfaces are discussed in the "Programming, Debug Interfaces, Resources" section on page 57 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 through Figure 2-4. 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 100 mA total to its associated I/O pins and opamps. On the 68 pin and 100 pin devices each set of Vddio associated pins may sink up to 100 mA. The 48 pin device may sink up to 100 mA total for all Vddio0 plus Vddio2 associated I/O pins and 100 mA total for all Vddio1 plus Vddio3 associated I/O pins.

Figure 2-1. 48-Pin SSOP Part Pinout

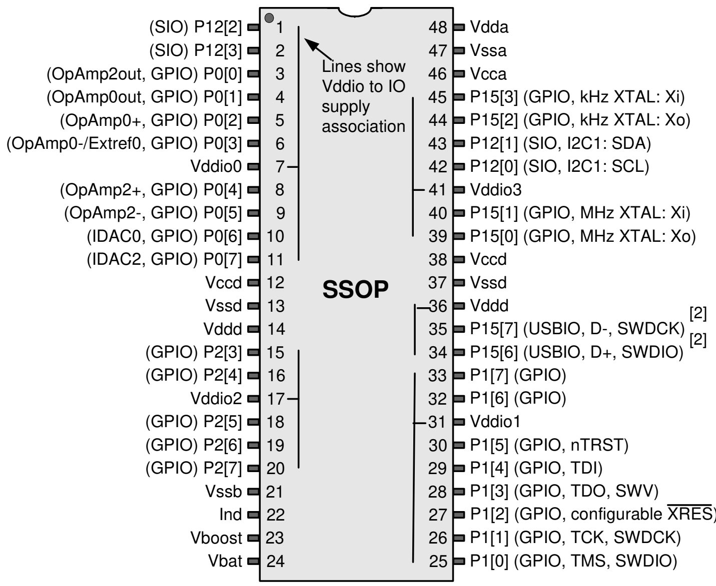
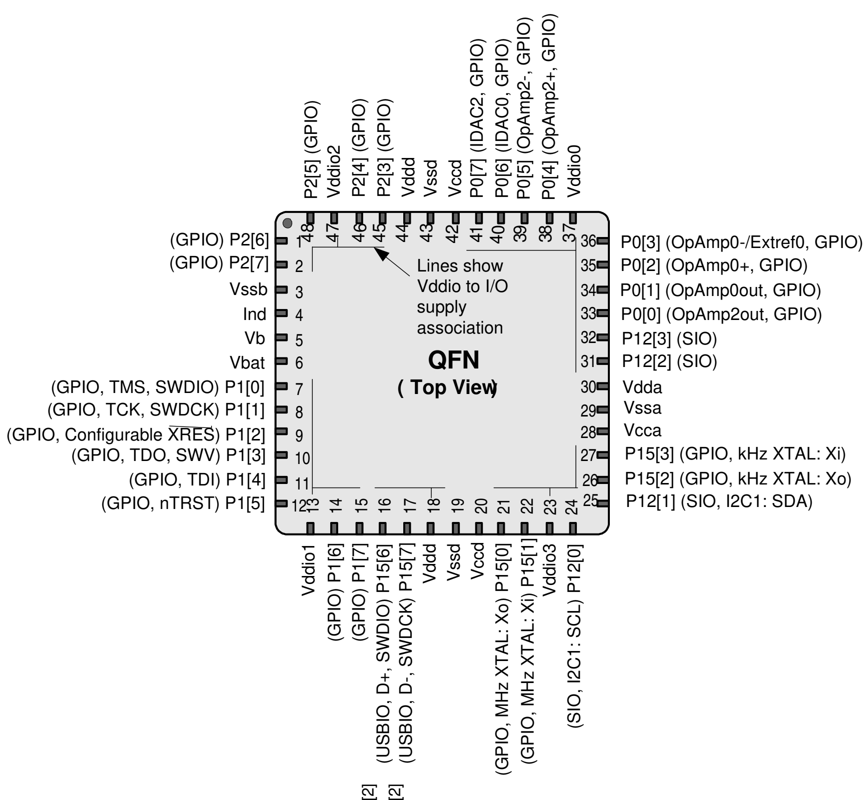


Figure 2-2. 48-Pin QFN Part Pinout[3]



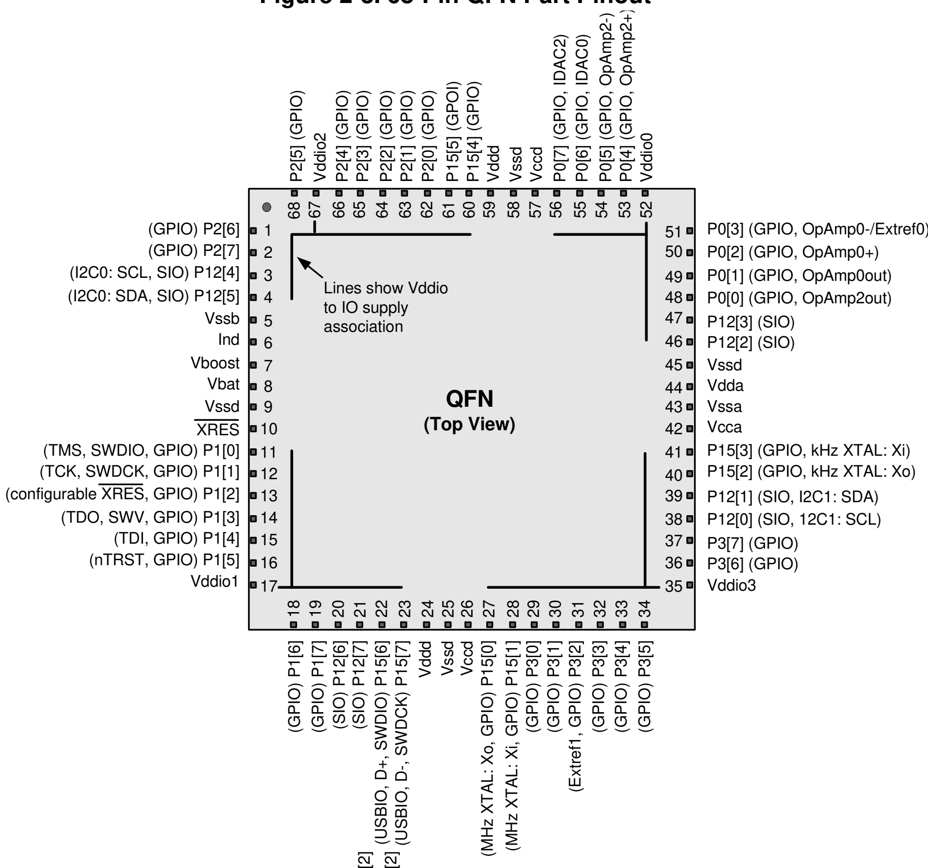


Figure 2-3. 68-Pin QFN Part Pinout[3]

Notes

2. Pins are No Connect (NC) on devices without USB. NC means that the pin has no electrical connection. The pin can be left floating or tied to a supply voltage or ground.
3. 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.

Figure 2-4. 100-Pin TQFP Part Pinout

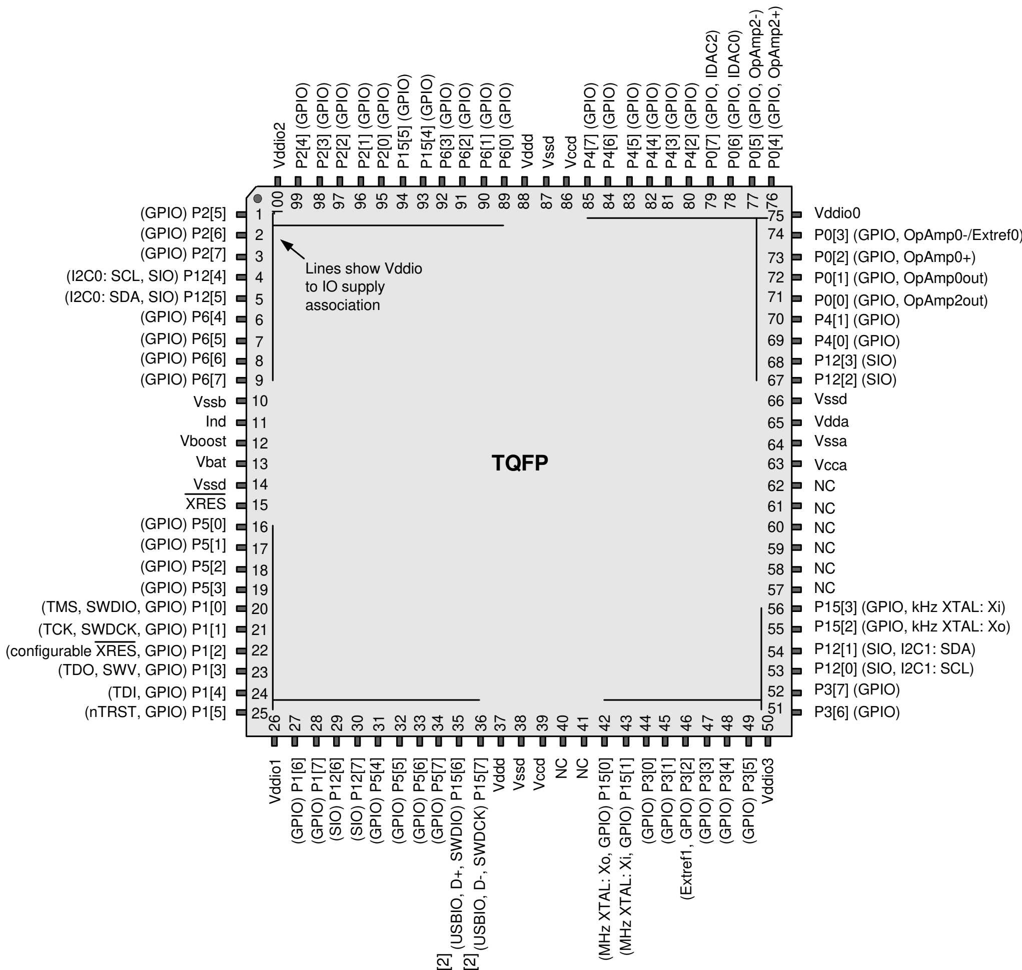
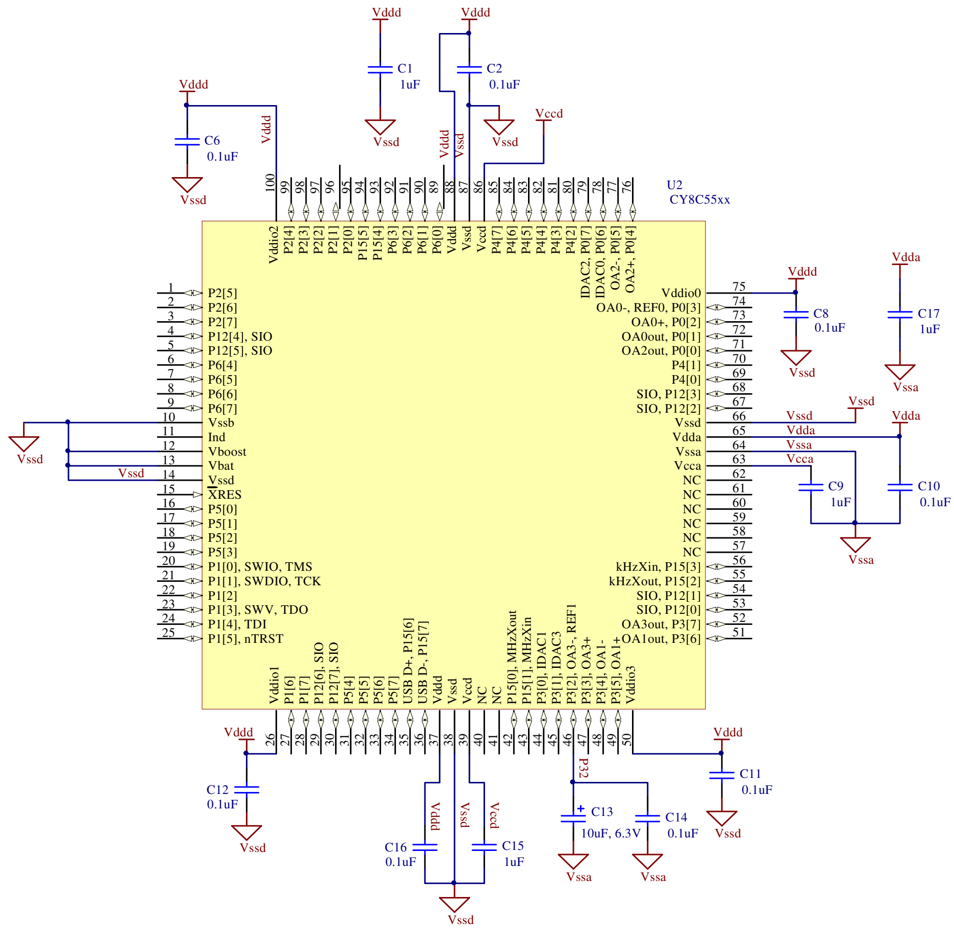


Figure 2-5 and Figure 2-6 show an example schematic and an example PCB layout, for the 100-pin TQFP part, for optimal analog performance on a 2-layer board.

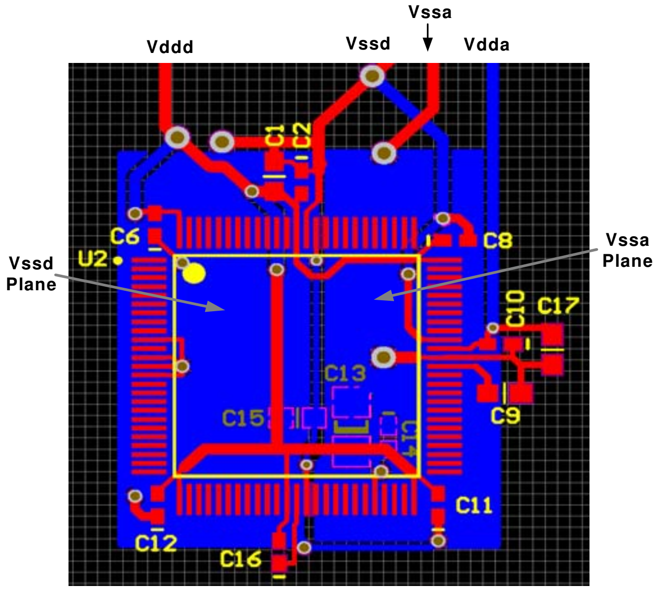
- The two pins labeled Vddd must be connected together.
- The two pins labeled Vccd must be connected together, with capacitance added, as shown in Figure 2-5 and Power System on page 25. The trace between the two Vccd pins should be as short as possible.
- The two pins labeled Vssd must be connected together.

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.

Figure 2-6. 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[4].

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

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

OpAmp0+, OpAmp2+. 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[4].

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 33 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.

SWV. Single Wire Viewer debug output.

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.

USBIO, D+. Provides D+ connection directly to a USB 2.0 bus. May be used as a digital I/O pin. Pins are No Connect (NC) on devices without USB.[2]

USBIO, D-. Provides D- connection directly to a USB 2.0 bus. May be used as a digital I/O pin. Pins are No Connect (NC) on devices without USB.[2]

Vboost. Power sense connection to boost pump.

Vbat. Battery supply to boost pump.

Vcca. Output of analog core regulator and input to analog core. Requires a 1 µF capacitor to Vssa. Regulator output not for external use.

Vccd. Output of digital core regulator and input to digital core. The two Vccd pins must be shorted together, with the trace between them as short as possible, and a 1 µF capacitor to Vssd; see Power System on page 25. Regulator output not for external use.

Note

4. GPIOs with OpAmp outputs are not recommended for use with CapSense.

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. See pinouts for specific I/O pin to Vddio mapping. Each Vddio must be tied to a valid operating voltage (1.71V to 5.5V), and must be less than or equal to V_{DDA}. If the I/O pins associated with Vddio0, Vddio2 or Vddio3 are not used then that Vddio should be tied to ground (Vssd or Vssa).

XRES (and configurable XRES). External reset pin. Active low with internal pullup. In 48-pin SSOP parts, P1[2] is configured as XRES. In all other parts the pin is configured as a GPIO.

4. CPU

4.1 8051 CPU

The CY8C34 devices use a single cycle 8051 CPU, which is fully compatible with the original MCS-51 instruction set. The CY8C34 family uses a pipelined RISC architecture, which executes most instructions in 1 to 2 cycles to provide peak performance of up to 24 MIPS with an average of 2 cycles per instruction. The single cycle 8051 CPU runs ten times faster than a standard 8051 processor.

The 8051 CPU subsystem includes these features:

- Single cycle 8051 CPU
- Up to 64 kB of flash memory, up to 2 kB of EEPROM, and up to 8 kB of SRAM
- Programmable nested vector interrupt controller
- Direct Memory Access (DMA) controller
- Peripheral HUB (PHUB)
- External Memory Interface (EMIF)

4.2 Addressing Modes

The following addressing modes are supported by the 8051:

- Direct Addressing: The operand is specified by a direct 8-bit address field. Only the internal RAM and the SFRs can be accessed using this mode.
- Indirect Addressing: The instruction specifies the register which contains the address of the operand. The registers R0 or R1 are used to specify the 8-bit address, while the Data Pointer (DPTR) register is used to specify the 16-bit address.
- Register Addressing: Certain instructions access one of the registers (R0-R7) in the specified register bank. These instructions are more efficient because there is no need for an address field.
- Register Specific Instructions: Some instructions are specific to certain registers. For example, some instructions always act on the accumulator. In this case, there is no need to specify the operand.
- Immediate Constants: Some instructions carry the value of the constants directly instead of an address.
- Indexed Addressing: This type of addressing can be used only for a read of the program memory. This mode uses the Data Pointer as the base and the accumulator value as an offset to read a program memory.
- Bit Addressing: In this mode, the operand is one of 256 bits.

4.3 Instruction Set

The 8051 instruction set is highly optimized for 8-bit handling and Boolean operations. The types of instructions supported include:

- Arithmetic instructions
- Logical instructions
- Data transfer instructions
- Boolean instructions
- Program branching instructions

4.3.1 Instruction Set Summary

4.3.1.1 Arithmetic Instructions

Arithmetic instructions support the direct, indirect, register, immediate constant, and register specific instructions. Arithmetic modes are used for addition, subtraction, multiplication, division, increment, and decrement operations. lists the different arithmetic instructions.

Table 4-1. Arithmetic Instructions

Mnemonic	Description	Bytes	Cycles
ADD A,Rn	Add register to accumulator	1	1
ADD A,Direct	Add direct byte to accumulator	2	2
ADD A,@Ri	Add indirect RAM to accumulator	1	2
ADD A,#data	Add immediate data to accumulator	2	2
ADDC A,Rn	Add register to accumulator with carry	1	1
ADDC A,Direct	Add direct byte to accumulator with carry	2	2
ADDC A,@Ri	Add indirect RAM to accumulator with carry	1	2
ADDC A,#data	Add immediate data to accumulator with carry	2	2
SUBB A,Rn	Subtract register from accumulator with borrow	1	1
SUBB A,Direct	Subtract direct byte from accumulator with borrow	2	2
SUBB A,@Ri	Subtract indirect RAM from accumulator with borrow	1	2
SUBB A,#data	Subtract immediate data from accumulator with borrow	2	2
INC A	Increment accumulator	1	1
INC Rn	Increment register	1	2
INC Direct	Increment direct byte	2	3
INC @Ri	Increment indirect RAM	1	3
DEC A	Decrement accumulator	1	1
DEC Rn	Decrement register	1	2
DEC Direct	Decrement direct byte	2	3
DEC @Ri	Decrement indirect RAM	1	3
INC DPTR	Increment data pointer	1	1
MUL	Multiply accumulator and B	1	2
DIV	Divide accumulator by B	1	6
DAA	Decimal adjust accumulator	1	3

4.3.1.2 Logical Instructions

The logical instructions perform Boolean operations such as AND, OR, XOR on bytes, rotate of accumulator contents, and swap of nibbles in an accumulator. The Boolean operations on the bytes are performed on the bit-by-bit basis. shows the list of logical instructions and their description.

Table 4-2. Logical Instructions

Mnemonic	Description	Bytes	Cycles
ANL A,Rn	AND register to accumulator	1	1
ANL A,Direct	AND direct byte to accumulator	2	2
ANL A,@Ri	AND indirect RAM to accumulator	1	2
ANL A,#data	AND immediate data to accumulator	2	2
ANL Direct, A	AND accumulator to direct byte	2	3
ANL Direct, #data	AND immediate data to direct byte	3	3
ORL A,Rn	OR register to accumulator	1	1
ORL A,Direct	OR direct byte to accumulator	2	2
ORL A,@Ri	OR indirect RAM to accumulator	1	2

Table 4-2. Logical Instructions (continued)

Mnemonic	Description	Bytes	Cycles
ORL A,#data	OR immediate data to accumulator	2	2
ORL Direct, A	OR accumulator to direct byte	2	3
ORL Direct, #data	OR immediate data to direct byte	3	3
XRL A,Rn	XOR register to accumulator	1	1
XRL A,Direct	XOR direct byte to accumulator	2	2
XRL A,@Ri	XOR indirect RAM to accumulator	1	2
XRL A,#data	XOR immediate data to accumulator	2	2
XRL Direct, A	XOR accumulator to direct byte	2	3
XRL Direct, #data	XOR immediate data to direct byte	3	3
CLR A	Clear accumulator	1	1
CPL A	Complement accumulator	1	1
RL A	Rotate accumulator left	1	1
RLC A	Rotate accumulator left through carry	1	1
RR A	Rotate accumulator right	1	1
RRC A	Rotate accumulator right though carry	1	1
SWAP A	Swap nibbles within accumulator	1	1

4.3.1.3 Data Transfer Instructions

The data transfer instructions are of three types: the core RAM, xdata RAM, and the look up tables. The core RAM transfer includes transfer between any two core RAM locations or SFRs. These instructions can use direct, indirect, register, and immediate addressing. The xdata RAM transfer includes only the transfer between the accumulator and the xdata RAM location. It can use only indirect addressing. The look up tables involve nothing but the read of program memory using the Indexed addressing mode. Table 4-3 lists the various data transfer instructions available.

4.3.1.4 Boolean Instructions

The 8051 core has a separate bit addressable memory location. It has 128 bits of bit addressable RAM and a set of SFRs that are bit addressable. The instruction set includes the whole menu of bit operations such as move, set, clear, toggle, OR, and AND instructions and the conditional jump instructions. Table 4-4 lists the available Boolean instructions.

Table 4-3. Data Transfer Instructions

Mnemonic	Description	Bytes	Cycles
MOV A,Rn	Move register to accumulator	1	1
MOV A,Direct	Move direct byte to accumulator	2	2
MOV A,@Ri	Move indirect RAM to accumulator	1	2
MOV A,#data	Move immediate data to accumulator	2	2
MOV Rn,A	Move accumulator to register	1	1
MOV Rn,Direct	Move direct byte to register	2	3
MOV Rn, #data	Move immediate data to register	2	2
MOV Direct, A	Move accumulator to direct byte	2	2
MOV Direct, Rn	Move register to direct byte	2	2
MOV Direct, Direct	Move direct byte to direct byte	3	3
MOV Direct, @Ri	Move indirect RAM to direct byte	2	3
MOV Direct, #data	Move immediate data to direct byte	3	3
MOV @Ri, A	Move accumulator to indirect RAM	1	2

Table 4-3. Data Transfer Instructions (continued)

Mnemonic	Description	Bytes	Cycles
MOV @Ri, Direct	Move direct byte to indirect RAM	2	3
MOV @Ri, #data	Move immediate data to indirect RAM	2	2
MOV DPTR, #data16	Load data pointer with 16 bit constant	3	3
MOVC A, @A+DPTR	Move code byte relative to DPTR to accumulator	1	5
MOVC A, @A + PC	Move code byte relative to PC to accumulator	1	4
MOVX A,@Ri	Move external RAM (8 bit) to accumulator	1	3
MOVX A, @DPTR	Move external RAM (16 bit) to accumulator	1	2
MOVX @Ri, A	Move accumulator to external RAM (8 bit)	1	4
MOVX @DPTR, A	Move accumulator to external RAM (16 bit)	1	3
PUSH Direct	Push direct byte onto stack	2	3
POP Direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange register with accumulator	1	2
XCH A, Direct	Exchange direct byte with accumulator	2	3
XCH A, @Ri	Exchange indirect RAM with accumulator	1	3
XCHD A, @Ri	Exchange low order indirect digit RAM with accumulator	1	3

Table 4-4. Boolean Instructions

Mnemonic	Description	Bytes	Cycles
CLR C	Clear carry	1	1
CLR bit	Clear direct bit	2	3
SETB C	Set carry	1	1
SETB bit	Set direct bit	2	3
CPL C	Complement carry	1	1
CPL bit	Complement direct bit	2	3
ANL C, bit	AND direct bit to carry	2	2
ANL C, /bit	AND complement of direct bit to carry	2	2
ORL C, bit	OR direct bit to carry	2	2
ORL C, /bit	OR complement of direct bit to carry	2	2
MOV C, bit	Move direct bit to carry	2	2
MOV bit, C	Move carry to direct bit	2	3
JC rel	Jump if carry is set	2	3
JNC rel	Jump if no carry is set	2	3
JB bit, rel	Jump if direct bit is set	3	5
JNB bit, rel	Jump if direct bit is not set	3	5
JBC bit, rel	Jump if direct bit is set and clear bit	3	5

4.3.1.5 Program Branching Instructions

The 8051 supports a set of conditional and unconditional jump instructions that help to modify the program execution flow. Table 4-5 shows the list of jump instructions.

Table 4-5. Jump Instructions

Mnemonic	Description	Bytes	Cycles
ACALL addr11	Absolute subroutine call	2	4
LCALL addr16	Long subroutine call	3	4
RET	Return from subroutine	1	4
RETI	Return from interrupt	1	4
AJMP addr11	Absolute jump	2	3
LJMP addr16	Long jump	3	4
SJMP rel	Short jump (relative address)	2	3
JMP @A + DPTR	Jump indirect relative to DPTR	1	5
JZ rel	Jump if accumulator is zero	2	4
JNZ rel	Jump if accumulator is nonzero	2	4
CJNE A,Direct, rel	Compare direct byte to accumulator and jump if not equal	3	5
CJNE A, #data, rel	Compare immediate data to accumulator and jump if not equal	3	4
CJNE Rn, #data, rel	Compare immediate data to register and jump if not equal	3	4
CJNE @Ri, #data, rel	Compare immediate data to indirect RAM and jump if not equal	3	5
DJNZ Rn,rel	Decrement register and jump if not zero	2	4
DJNZ Direct, rel	Decrement direct byte and jump if not zero	3	5
NOP	No operation	1	1

4.4 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.

4.4.1 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 4-6. 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, CAN, I^2C, Timers, Counters, and PWMs
5	Reserved
6	UDBs group 1
7	UDBs group 2

4.4.2 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
- TDs may be nested and/or chained for complex transactions

4.4.3 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 4-7 after the CPU and DMA priority levels 0 and 1 have satisfied their requirements.

Table 4-7. 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.

4.4.4 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:

4.4.4.1 Simple DMA

In a simple DMA case, a single TD transfers data between a source and sink (peripherals or memory location).

4.4.4.2 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.

4.4.4.3 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.

4.4.4.4 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.

4.4.4.5 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.

4.4.4.6 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.

4.4.4.7 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.

4.5 Interrupt Controller

The interrupt controller provides a mechanism for hardware resources to change program execution to a new address, independent of the current task being executed by the main code. The interrupt controller provides enhanced features not found on original 8051 interrupt controllers:

- 32 interrupt vectors
- Jumps directly to ISR anywhere in code space with dynamic vector addresses
- Multiple sources for each vector
- Flexible interrupt to vector matching
- Each interrupt vector is independently enabled or disabled
- Each interrupt can be dynamically assigned one of eight priorities
- Eight level nestable interrupts
- Multiple I/O interrupt vectors
- Software can send interrupts
- Software can clear pending interrupts

When an interrupt is pending, the current instruction is completed and the program counter is pushed onto the stack. Code execution then jumps to the program address provided by the vector. After the ISR is completed, a RETI instruction is executed and returns execution to the instruction following the previously interrupted instruction. To do this the RETI instruction pops the program counter from the stack.

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. Fixed function interrupts and all interrupt sources may be routed to any interrupt vector using the UDB interrupt source connections.

Table 4-8. Interrupt Vector Table

#	Fixed Function	DMA	UDB
0	LVD	phub_termout0[0]	udb_intr[0]
1	ECC	phub_termout0[1]	udb_intr[1]
2	Reserved	phub_termout0[2]	udb_intr[2]
3	Sleep (Pwr Mgr)	phub_termout0[3]	udb_intr[3]
4	PICU[0]	phub_termout0[4]	udb_intr[4]
5	PICU[1]	phub_termout0[5]	udb_intr[5]
6	PICU[2]	phub_termout0[6]	udb_intr[6]
7	PICU[3]	phub_termout0[7]	udb_intr[7]
8	PICU[4]	phub_termout0[8]	udb_intr[8]
9	PICU[5]	phub_termout0[9]	udb_intr[9]
10	PICU[6]	phub_termout0[10]	udb_intr[10]
11	PICU[12]	phub_termout0[11]	udb_intr[11]
12	PICU[15]	phub_termout0[12]	udb_intr[12]
13	Comparators Combined	phub_termout0[13]	udb_intr[13]
14	Switched Caps Combined	phub_termout0[14]	udb_intr[14]
15	I²C	phub_termout0[15]	udb_intr[15]
16	CAN	phub_termout1[0]	udb_intr[16]
17	Timer/Counter0	phub_termout1[1]	udb_intr[17]
18	Timer/Counter1	phub_termout1[2]	udb_intr[18]
19	Timer/Counter2	phub_termout1[3]	udb_intr[19]
20	Timer/Counter3	phub_termout1[4]	udb_intr[20]
21	USB SOF Int	phub_termout1[5]	udb_intr[21]
22	USB Arb Int	phub_termout1[6]	udb_intr[22]
23	USB Bus Int	phub_termout1[7]	udb_intr[23]
24	USB Endpoint[0]	phub_termout1[8]	udb_intr[24]
25	USB Endpoint Data	phub_termout1[9]	udb_intr[25]
26	Reserved	phub_termout1[10]	udb_intr[26]
27	Reserved	phub_termout1[11]	udb_intr[27]
28	Reserved	phub_termout1[12]	udb_intr[28]
29	Decimator Int	phub_termout1[13]	udb_intr[29]
30	PHUB Error Int	phub_termout1[14]	udb_intr[30]
31	EEPROM Fault Int	phub_termout1[15]	udb_intr[31]

5. Memory

5.1 Static RAM

CY8C34 Static RAM (SRAM) is used for temporary data storage. Up to 8 KB of SRAM is provided and can be accessed by the 8051 or the DMA controller. See the "Memory Map" section on page 19. Simultaneous access of SRAM by the 8051 and the DMA controller is possible if different 4 KB blocks are accessed.

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 64 KB of user program space.

Up to an additional 8 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.

Flash is read in units of rows; each row is 9 bytes wide with 8 bytes of data and 1 byte of ECC data. When a row is read, the data bytes are copied into an 8-byte instruction buffer. The CPU fetches its instructions from this buffer, for improved CPU performance.

Flash programming is performed through a special interface and preempts code execution out of flash. 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. A total of up to 256 blocks are provided on 64 KB flash devices.

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 58). For more information on how to take full advantage of the security features in PSoC, see the PSoC 3 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 CY8C34 has up to 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 CPU can not execute out of EEPROM. There is no ECC hardware associated with EEPROM. If ECC is required it must be handled in firmware.

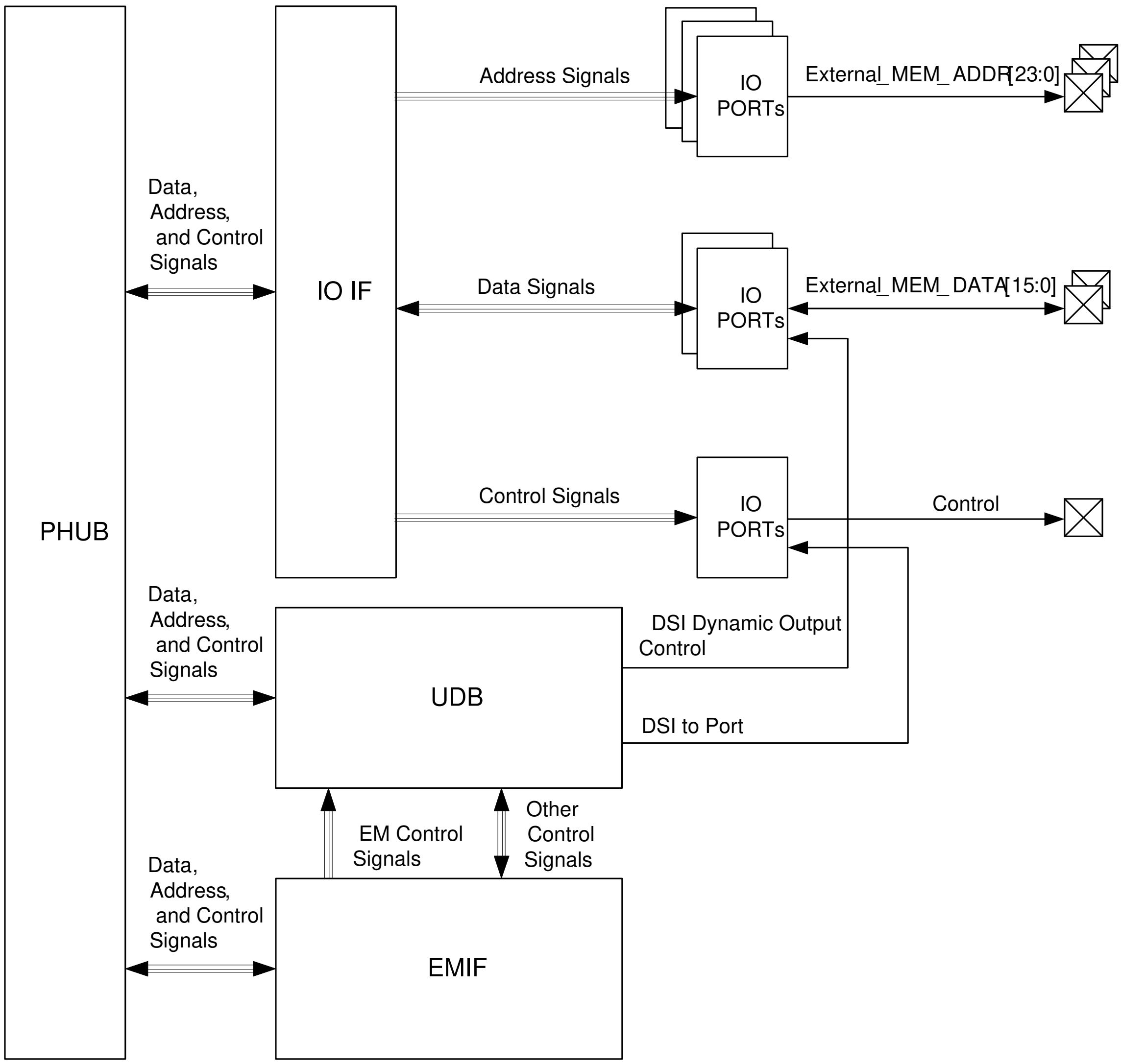
5.5 External Memory Interface

CY8C34 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.

Figure 5-1 is the EMIF block diagram. The EMIF supports synchronous and asynchronous memories. The CY8C34 supports only one type of external memory device at a time.

External memory can be accessed via the 8051 xdata space; up to 24 address bits can be used. See "xdata Space" section on page 21. The memory can be 8 or 16 bits wide.

Figure 5-1. EMIF Block Diagram



5.6 Memory Map

The CY8C34 8051 memory map is very similar to the MCS-51 memory map.

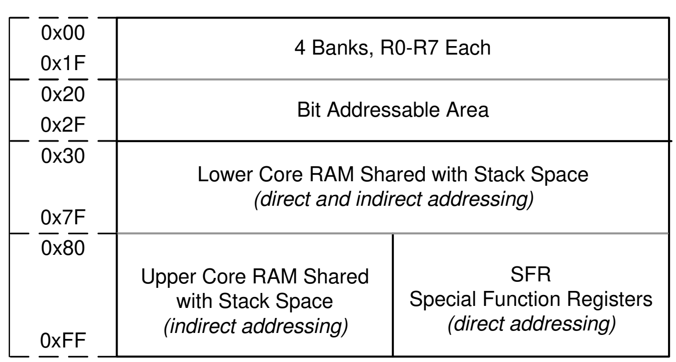
5.6.1 Code Space

The CY8C34 8051 code space is 64 KB. Only main flash exists in this space. See the "Flash Program Memory" section on page 18.

5.6.2 Internal Data Space

The CY8C34 8051 internal data space is 384 bytes, compressed within a 256-byte space. This space consists of 256 bytes of RAM (in addition to the SRAM mentioned in "Static RAM" on page 18) and a 128-byte space for Special Function Registers (SFRs). See Figure 5-2. The lowest 32 bytes are used for 4 banks of registers R0-R7. The next 16 bytes are bit-addressable.

Figure 5-2. 8051 Internal Data Space



In addition to the register or bit address modes used with the lower 48 bytes, the lower 128 bytes can be accessed with direct or indirect addressing. With direct addressing mode, the upper 128 bytes map to the SFRs. With indirect addressing mode, the upper 128 bytes map to RAM. Stack operations use indirect addressing; the 8051 stack space is 256 bytes. See the "Addressing Modes" section on page 11

5.6.3 SFRs

The Special Function Register (SFR) space provides access to frequently accessed registers. The memory map for the SFR memory space is shown in Table 5-2.

Table 5-2. SFR Map

Address	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0xF8	SFRPRT15DR	SFRPRT15PS	SFRPRT15SEL					
0xF0	B		SFRPRT12SEL					
0xE8	SFRPRT12DR	SFRPRT12PS	MXAX					
0xE0	ACC							
0xD8	SFRPRT6DR	SFRPRT6PS	SFRPRT6SEL					
0xD0	PSW							
0xC8	SFRPRT5DR	SFRPRT5PS	SFRPRT5SEL					
0xC0	SFRPRT4DR	SFRPRT4PS	SFRPRT4SEL					
0xB8								
0xB0	SFRPRT3DR	SFRPRT3PS	SFRPRT3SEL					
0xA8	IE							
0xA0	P2AX		SFRPRT1SEL					
0x98	SFRPRT2DR	SFRPRT2PS	SFRPRT2SEL					
0x90	SFRPRT1DR	SFRPRT1PS		DPX0		DPX1		
0x88		SFRPRT0PS	SFRPRT0SEL					
0x80	SFRPRT0DR	SP	DPL0	DPH0	DPL1	DPH1	DPS	

The CY8C34 family provides the standard set of registers found on industry standard 8051 devices. In addition, the CY8C34 devices add SFRs to provide direct access to the I/O ports on the device. The following sections describe the SFRs added to the CY8C34 family.

XData Space Access SFRs

The 8051 core features dual DPTR registers for faster data transfer operations. The data pointer select SFR, DPS, selects which data pointer register, DPTR0 or DPTR1, is used for the following instructions:

- MOVX @DPTR, A
- MOVX A, @DPTR
- MOVC A, @A+DPTR
- JMP @A+DPTR
- INC DPTR
- MOV DPTR, #data16

The extended data pointer SFRs, DPX0, DPX1, MXAX, and P2AX, hold the most significant parts of memory addresses during access to the xdata space. These SFRs are used only with the MOVX instructions.

During a MOVX instruction using the DPTR0/DPTR1 register, the most significant byte of the address is always equal to the contents of DPX0/DPX1.

During a MOVX instruction using the R0 or R1 register, the most significant byte of the address is always equal to the contents of MXAX, and the next most significant byte is always equal to the contents of P2AX.

I/O Port SFRs

The I/O ports provide digital input sensing, output drive, pin interrupts, connectivity for analog inputs and outputs, LCD, and access to peripherals through the DSI. Full information on I/O ports is found in "I/O System and Routing" on page 29.

I/O ports are linked to the CPU through the PHUB and are also available in the SFRs. Using the SFRs allows faster access to a limited set of I/O port registers, while using the PHUB allows boot configuration and access to all I/O port registers.

Each SFR supported I/O port provides three SFRs:

- SFRPRTxDR sets the output data state of the port (where x is port number and includes ports 0-6, 12 and 15).
- The SFRPRTxSEL selects whether the PHUB PRTxDR register or the SFRPRTxDR controls each pin's output buffer within the port. If a SFRPRTxSEL[y] bit is high, the corresponding SFRPRTxDR[y] bit sets the output state for that pin. If a SFRPRTxSEL[y] bit is low, the corresponding PRTxDR[y] bit sets the output state of the pin (where y varies from 0 to 7).
- The SFRPRTxPS is a read only register that contains pin state values of the port pins.

5.6.3.1 xdata Space

The 8051 xdata space is 24-bit, or 16 MB in size. The majority of this space is not "external"—it is used by on-chip components. See Table 5-3. External, that is, off-chip, memory can be accessed using the EMIF. See External Memory Interface.

Table 5-3. XDATA Data Address Map

Address Range	Purpose
0x00 0000 - 0x00 1FFF	SRAM
0x00 4000 - 0x00 42FF	Clocking, PLLs, and oscillators
0x00 4300 - 0x00 43FF	Power management
0x00 4400 - 0x00 44FF	Interrupt controller
0x00 4500 - 0x00 45FF	Ports interrupt control
0x00 4700 - 0x00 47FF	System performance controller
0x00 4900 - 0x00 49FF	I^2C controller
0x00 4E00 - 0x00 4EFF	Decimator
0x00 4F00 - 0x00 4FFF	Fixed timer/counter/PWMs
0x00 5000 - 0x00 51FF	General purpose I/Os
0x00 5300 - 0x00 530F	Output port select register
0x00 5400 - 0x00 54FF	External Memory Interface control registers
0x00 5800 - 0x00 5FFF	Analog Subsystem interface
0x00 6000 - 0x00 60FF	USB controller
0x00 6400 - 0x00 6FFF	UDB configuration
0x00 7000 - 0x00 7FFF	PHUB configuration
0x00 8000 - 0x00 8FFF	EEPROM
0x00 A000 - 0x00 A400	CAN
0x01 0000 - 0x01 FFFF	Digital Interconnect configuration
0x03 0000 - 0x03 01FF	Reserved
0x05 0220 - 0x05 02F0	Debug controller
0x08 0000 - 0x08 1FFF	Flash ECC bytes
0x80 0000 - 0xFF FFFF	External Memory Interface

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 50 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 PSoC.

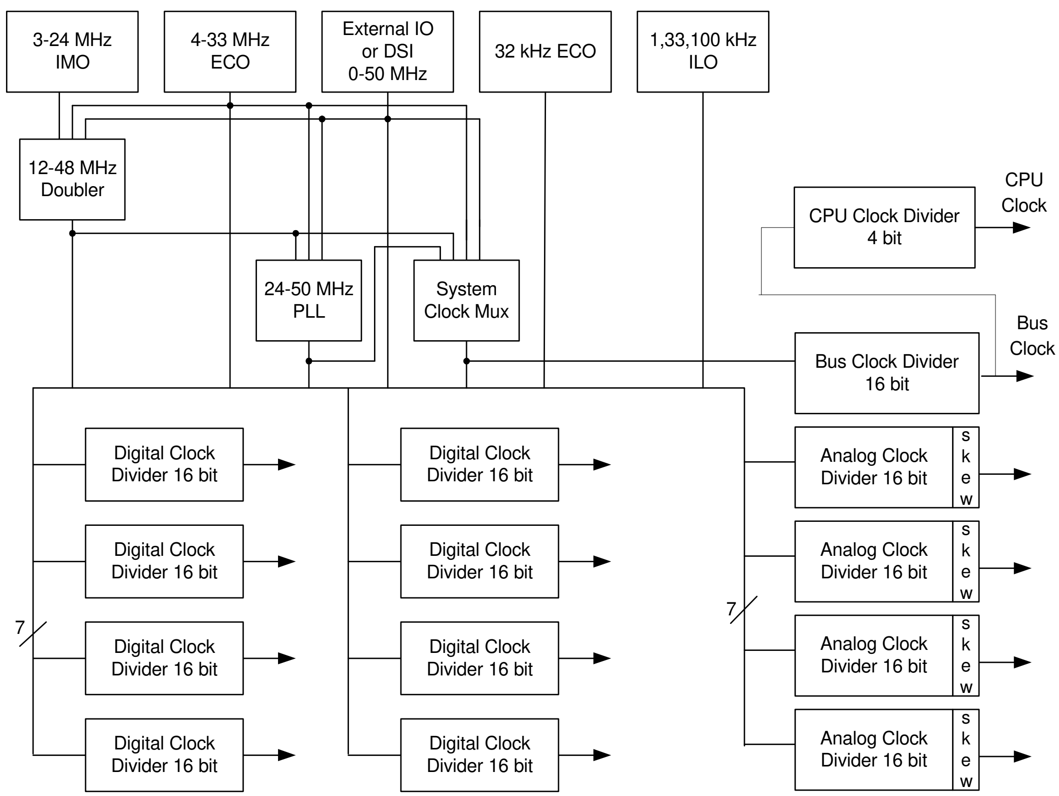
Key features of the clocking system include:

- Seven general purpose clock sources
  - 3 to 24 MHz IMO, ±1% at 3 MHz
  - 4 to 33 MHz External Crystal Oscillator (MHzECO)
  - Clock doubler provides a doubled clock frequency output for the USB block, see USB Clock Domain on page 24
  - DSI signal from an external I/O pin or other logic
  - 24 to 50 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 the 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 bus clock
- Dedicated 4-bit divider for the 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	24 MHz	±4%	10 µs max
MHzECO	4 MHz	Crystal dependent	33 MHz	Crystal dependent	5 ms typ, max is crystal dependent
DSI	0 MHz	Input dependent	50 MHz	Input dependent	Input dependent
PLL	24 MHz	Input dependent	50 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

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 ±4% at 24 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, and 24 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 50 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. Systems that require accurate timing should use the Real Time Clock 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 time intervals such as fast sleep 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 wake the system. The fast timewheel settings are programmable, and the counter automatically resets when the terminal count is reached. This enables flexible, periodic wakeups of 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.

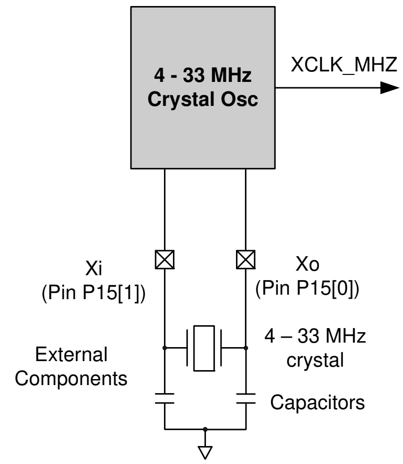
The 33 kHz clock (CLK33K) comes from a divide-by-3 operation on CLK100K. This output can be used as a reduced accuracy version of the 32.768 kHz ECO clock with no need for a crystal.

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 a wide variety of crystal types, in the range of 4 to 33 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" section on page 22). 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

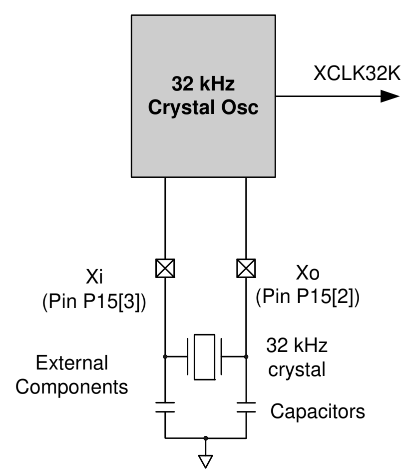


6.1.2.2 32.768 kHz ECO

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



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. Bus clock is the source clock for the CPU clock divider.
- 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 Universal Digital Blocks (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 ADC and mixers. 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.

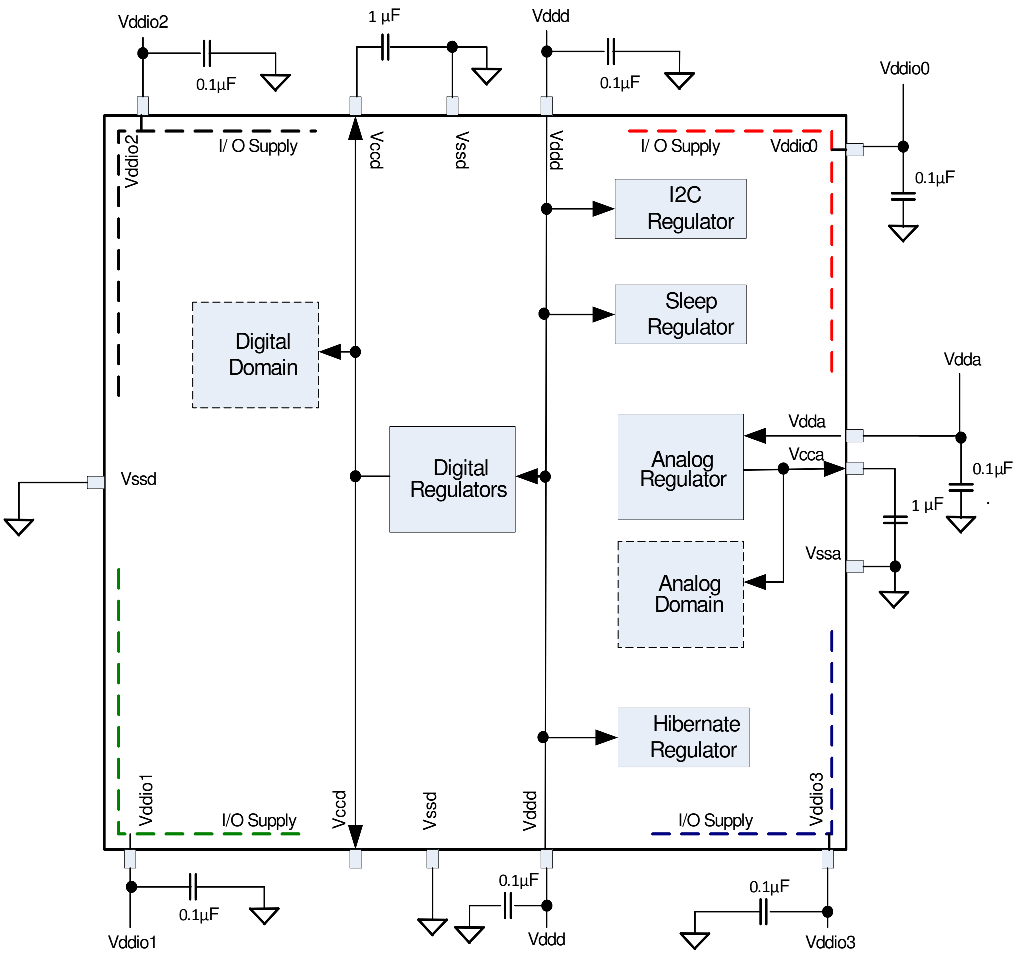
6.1.4 USB Clock Domain

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 internal oscillator, DSI signal, or crystal oscillator.

6.2 Power System

The power system consists of separate analog, digital, and I/O supply pins, labeled Vdda, Vddd, and Vddiox, respectively. It also includes two internal 1.8V regulators that provide the digital (Vccd) and analog (Vcca) supplies for the internal core logic. The output pins of the regulators (Vccd and Vcca) and the Vddio pins must have capacitors connected as shown in Figure 6-4. The two Vccd pins must be shorted together, with as short a trace as possible, and connected to a 1 µF ±10% X5R capacitor. The power system also contains a sleep regulator, an I²C regulator, and a hibernate regulator.

Figure 6-4. PSoC Power System



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.

6.2.1 Power Modes

PSoC 3 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 3 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 and Real Time Clock functionality. The lowest power mode is hibernate, which retains register and SRAM state, but no clocks, and allows wakeup only from I/O pins. Figure 6-5 illustrates the allowable transitions between power modes.