Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	EBI/EMI, I²C, LINbus, SPI, UART/USART
Peripherals	CapSense, DMA, LCD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	1.71V ~ 5.5V
Data Converters	A/D 16x12b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-BSSOP (0.295", 7.50mm Width)
Supplier Device Package	48-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy8c3446pvi-091

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

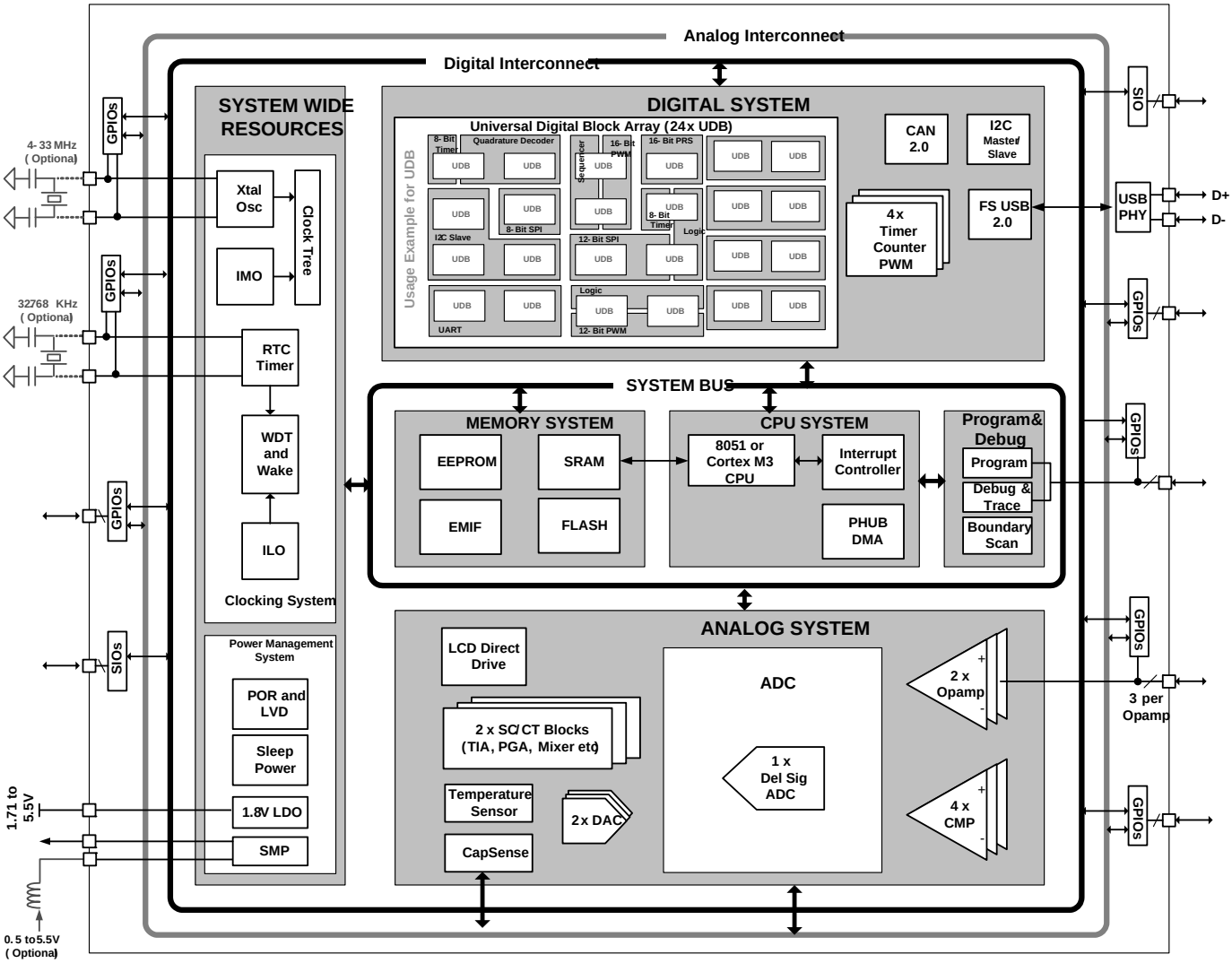


Table 4-1. Arithmetic Instructions

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

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

4.4.4.7 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	I^2C	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]

5.6.3.1 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

7. Digital Subsystem

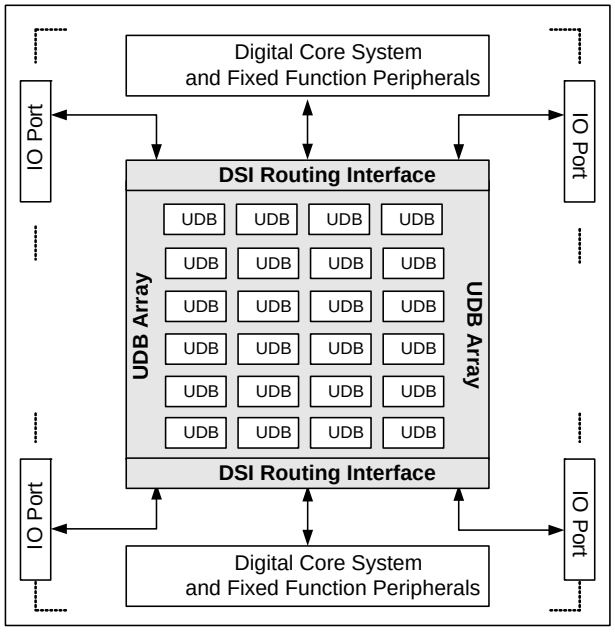
The digital programmable system creates application specific combinations of both standard and advanced digital peripherals and custom logic functions. These peripherals and logic are then interconnected to each other and to any pin on the device, providing a high level of design flexibility and IP security.

The features of the digital programmable system are outlined here to provide an overview of capabilities and architecture. Designers do not need to interact directly with the programmable digital system at the hardware and register level. PSoC Creator provides a high level schematic capture graphical interface to automatically place and route resources similar to PLDs.

The main components of the digital programmable system are:

- Universal Digital Blocks (UDB) - These form the core functionality of the digital programmable system. UDBs are a collection of uncommitted logic (PLD) and structural logic (Datapath) optimized to create all common embedded peripherals and customized functionality that are application or design specific.
- Universal Digital Block Array - UDB blocks are arrayed within a matrix of programmable interconnect. The UDB array structure is homogeneous and allows for flexible mapping of digital functions onto the array. The array supports extensive and flexible routing interconnects between UDBs and the Digital System Interconnect.
- Digital System Interconnect (DSI) - Digital signals from Universal Digital Blocks (UDBs), fixed function peripherals, I/O pins, interrupts, DMA, and other system core signals are attached to the Digital System Interconnect to implement full featured device connectivity. The DSI allows any digital function to any pin or other feature routability when used with the Universal Digital Block Array.

Figure 7-1. CY8C34 Digital Programmable Architecture



7.1 Example Peripherals

The flexibility of the CY8C34 family's Universal Digital Blocks (UDBs) and Analog Blocks allow the user to create a wide range of components (peripherals). The most common peripherals were built and characterized by Cypress and are shown in the PSoC Creator component catalog, however, users may also create their own custom components using PSoC Creator. Using PSoC Creator, users may also create their own components for reuse within their organization, for example sensor interfaces, proprietary algorithms, and display interfaces.

The number of components available through PSoC Creator is too numerous to list in the data sheet, and the list is always growing. An example of a component available for use in CY8C34 family, but, not explicitly called out in this data sheet is the UART component.

7.1.1 Example Digital Components

The following is a sample of the digital components available in PSoC Creator for the CY8C34 family. The exact amount of hardware resources (UDBs, routing, RAM, flash) used by a component varies with the features selected in PSoC Creator for the component.

- Communications
  - I^2C
  - UART
  - SPI
- Functions
  - EMIF
  - PWMs
  - Timers
  - Counters
- Logic
  - NOT
  - OR
  - XOR
  - AND

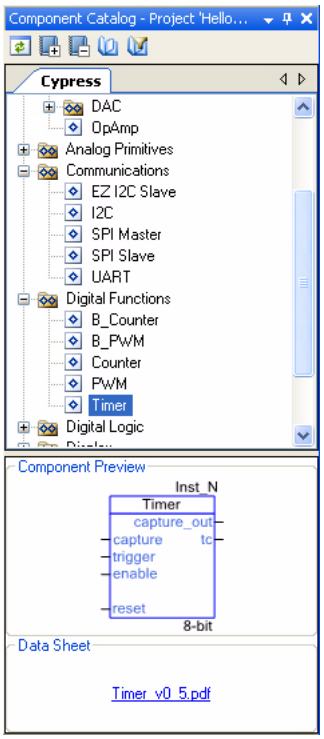
7.1.2 Example Analog Components

The following is a sample of the analog components available in PSoC Creator for the CY8C34 family. The exact amount of hardware resources (SC/CT blocks, routing, RAM, flash) used by a component varies with the features selected in PSoC Creator for the component.

- Amplifiers
  - TIA
  - PGA
  - opamp
- ADC
  - Delta-Sigma
- DACs
  - Current
  - Voltage
  - PWM
- Comparators
- Mixers

7.1.4.2 Component Catalog

Figure 7-3. Component Catalog



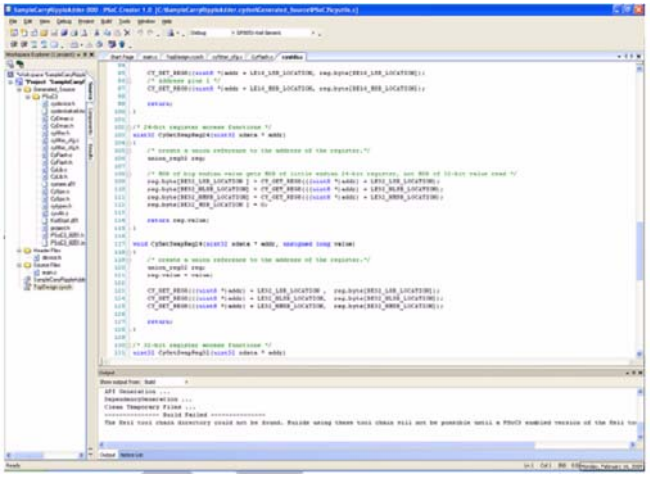
The component catalog is a repository of reusable design elements that select device functionality and customize your PSoC device. It is populated with an impressive selection of content; from simple primitives such as logic gates and device registers, through the digital timers, counters and PWMs, plus analog components such as ADC, DACs, and filters, and communication protocols, such as I^2C, USB, and CAN. See Example Peripherals on page 35 for more details about available peripherals. All content is fully characterized and carefully documented in datasheets with code examples, AC/DC specifications, and user code ready APIs.

7.1.4.3 Design Reuse

The symbol editor gives you the ability to develop reusable components that can significantly reduce future design time. Just draw a symbol and associate that symbol with your proven design. PSoC Creator allows for the placement of the new symbol anywhere in the component catalog along with the content provided by Cypress. You can then reuse your content as many times as you want, and in any number of projects, without ever having to revisit the details of the implementation.

7.1.4.4 Software Development

Figure 7-4. Code Editor

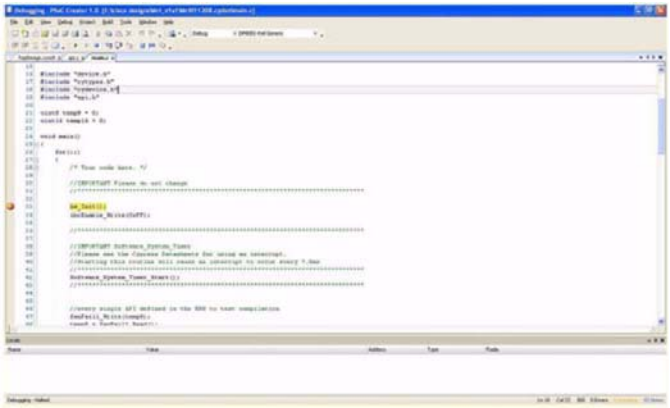


Anchoring the tool is a modern, highly customizable user interface. It includes project management and integrated editors for C and assembler source code, as well the design entry tools.

Project build control leverages compiler technology from top commercial vendors such as ARM® Limited, Keil™, and CodeSourcery (GNU). Free versions of Keil C51 and GNU C Compiler (GCC) for ARM, with no restrictions on code size or end product distribution, are included with the tool distribution. Upgrading to more optimizing compilers is a snap with support for the professional Keil C51 product and ARM RealView™ compiler.

7.1.4.5 Nonintrusive Debugging

Figure 7-5. PSoC Creator Debugger



With JTAG (4-wire) and SWD (2-wire) debug connectivity available on all devices, the PSoC Creator debugger offers full control over the target device with minimum intrusion. Breakpoints and code execution commands are all readily available from toolbar buttons and an impressive lineup of windows—register, locals, watch, call stack, memory and peripherals—make for an unparalleled level of visibility into the system.

The bits of the control register, which may be written to by the system bus, are used to drive into the routing matrix, and thus provide firmware with the opportunity to control the state of UDB processing. The status register is read-only and it allows internal UDB state to be read out onto the system bus directly from internal routing. This allows firmware to monitor the state of UDB processing. Each bit of these registers has programmable connections to the routing matrix and routing connections are made depending on the requirements of the application.

7.2.3.1 Usage Examples

As an example of control input, a bit in the control register can be allocated as a function enable bit. There are multiple ways to enable a function. In one method the control bit output would be routed to the clock control block in one or more UDBs and serve as a clock enable for the selected UDB blocks. A status example is a case where a PLD or datapath block generated a condition, such as a “compare true” condition that is captured and latched by the status register and then read (and cleared) by CPU firmware.

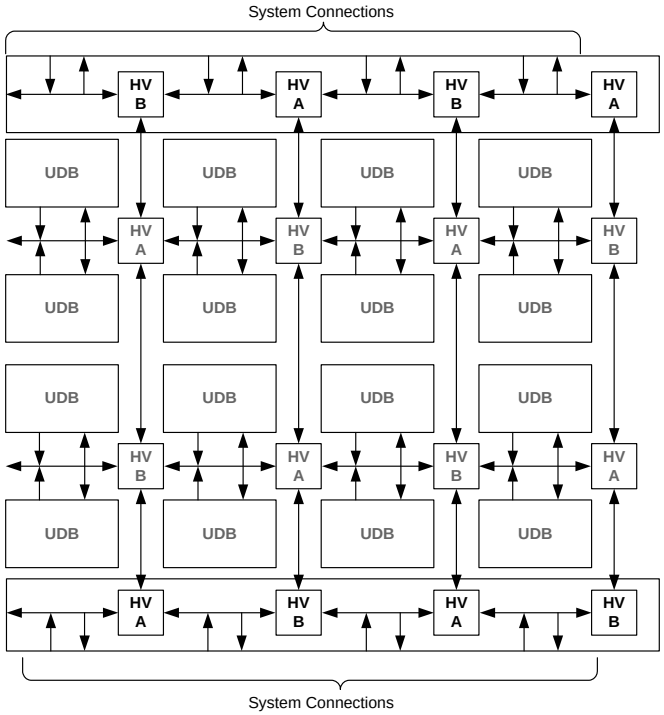
7.2.3.2 Clock Generation

Each subcomponent block of a UDB including the two PLDs, the datapath, and Status and Control, has a clock selection and control block. This promotes a fine granularity with respect to allocating clocking resources to UDB component blocks and allows unused UDB resources to be used by other functions for maximum system efficiency.

7.3 UDB Array Description

Figure 7-11 shows an example of a 16 UDB array. In addition to the array core, there are a DSI routing interfaces at the top and bottom of the array. Other interfaces that are not explicitly shown include the system interfaces for bus and clock distribution. The UDB array includes multiple horizontal and vertical routing channels each comprised of 96 wires. The wire connections to UDBs, at horizontal/vertical intersection and at the DSI interface are highly permutable providing efficient automatic routing in PSoC Creator. Additionally the routing allows wire by wire segmentation along the vertical and horizontal routing to further increase routing flexibility and capability.

Figure 7-11. Digital System Interface Structure



7.3.1 UDB Array Programmable Resources

Figure 7-12 shows an example of how functions are mapped into a bank of 16 UDBs. The primary programmable resources of the UDB are two PLDs, one datapath and one status/control register. These resources are allocated independently, because they have independently selectable clocks, and therefore unused blocks are allocated to other unrelated functions.

An example of this is the 8-bit Timer in the upper left corner of the array. This function only requires one datapath in the UDB, and therefore the PLD resources may be allocated to another function. A function such as a Quadrature Decoder may require more PLD logic than one UDB can supply and in this case can utilize the unused PLD blocks in the 8-bit Timer UDB. Programmable resources in the UDB array are generally homogeneous so functions can be mapped to arbitrary boundaries in the array.

The TIA configuration is used for applications where an external sensor's output is current as a function of some type of stimulus such as temperature, light, magnetic flux etc. In a common application, the voltage DAC output can be connected to the VREF TIA input to allow calibration of the external sensor bias current by adjusting the voltage DAC output voltage.

8.6 LCD Direct Drive

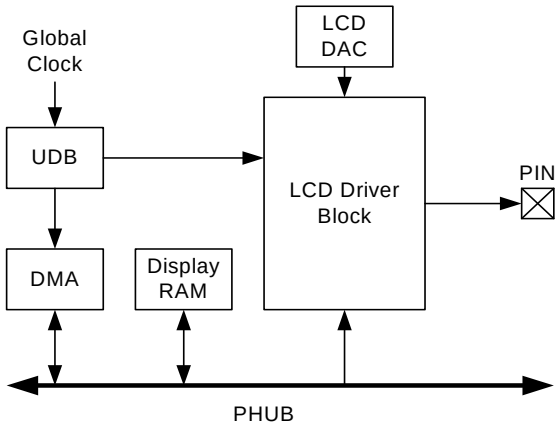
The PSoC Liquid Crystal Display (LCD) driver system is a highly configurable peripheral designed to allow PSoC to directly drive a broad range of LCD glass. All voltages are generated on chip, eliminating the need for external components. With a high multiplex ratio of up to 1/16, the CY8C34 family LCD driver system can drive a maximum of 736 segments. The PSoC LCD driver module was also designed with the conservative power budget of portable devices in mind, enabling different LCD drive modes and power down modes to conserve power.

PSoC Creator provides an LCD segment drive component. The component wizard provides easy and flexible configuration of LCD resources. You can specify pins for segments and commons along with other options. The software configures the device to meet the required specifications. This is possible because of the programmability inherent to PSoC devices.

Key features of the PSoC LCD segment system are:

- LCD panel direct driving
- Type A (standard) and Type B (low power) waveform support
- Wide operating voltage range support (2V to 5V) for LCD panels
- Static, 1/2, 1/3, 1/4, 1/5 bias voltage levels
- Internal bias voltage generation through internal resistor ladder
- Up to 62 total common and segment outputs
- Up to 1/16 multiplex for a maximum of 16 backplane/common outputs
- Up to 62 front plane/segment outputs for direct drive
- Drives up to 736 total segments (16 backplane x 46 front plane)
- Up to 64 levels of software controlled contrast
- Ability to move display data from memory buffer to LCD driver through DMA (without CPU intervention)
- Adjustable LCD refresh rate from 10 Hz to 150 Hz
- Ability to invert LCD display for negative image
- Three LCD driver drive modes, allowing power optimization

Figure 8-9. LCD System



8.6.1 LCD Segment Pin Driver

Each GPIO pin contains an LCD driver circuit. The LCD driver buffers the appropriate output of the LCD DAC to directly drive the glass of the LCD. A register setting determines whether the pin is a common or segment. The pin's LCD driver then selects one of the six bias voltages to drive the I/O pin, as appropriate for the display data.

8.6.2 Display Data Flow

The LCD segment driver system reads display data and generates the proper output voltages to the LCD glass to produce the desired image. Display data resides in a memory buffer in the system SRAM. Each time you need to change the common and segment driver voltages, the next set of pixel data moves from the memory buffer into the Port Data Registers via DMA.

8.6.3 UDB and LCD Segment Control

A UDB is configured to generate the global LCD control signals and clocking. This set of signals is routed to each LCD pin driver through a set of dedicated LCD global routing channels. In addition to generating the global LCD control signals, the UDB also produces a DMA request to initiate the transfer of the next frame of LCD data.

8.6.4 LCD DAC

The LCD DAC generates the contrast control and bias voltage for the LCD system. The LCD DAC produces up to five LCD drive voltages plus ground, based on the selected bias ratio. The bias voltages are driven out to GPIO pins on a dedicated LCD bias bus, as required.

8.7 CapSense

The CapSense system provides a versatile and efficient means for measuring capacitance in applications such as touch sense buttons, sliders, proximity detection, etc. The CapSense system uses a configuration of system resources, including a few hardware functions primarily targeted for CapSense. Specific resource usage is detailed in the CapSense component in PSoC Creator.

A capacitive sensing method using a Delta-Sigma Modulator (CSD) is used. It provides capacitance sensing using a switched capacitor technique with a delta-sigma modulator to convert the sensing current to a digital code.

8.8 Temp Sensor

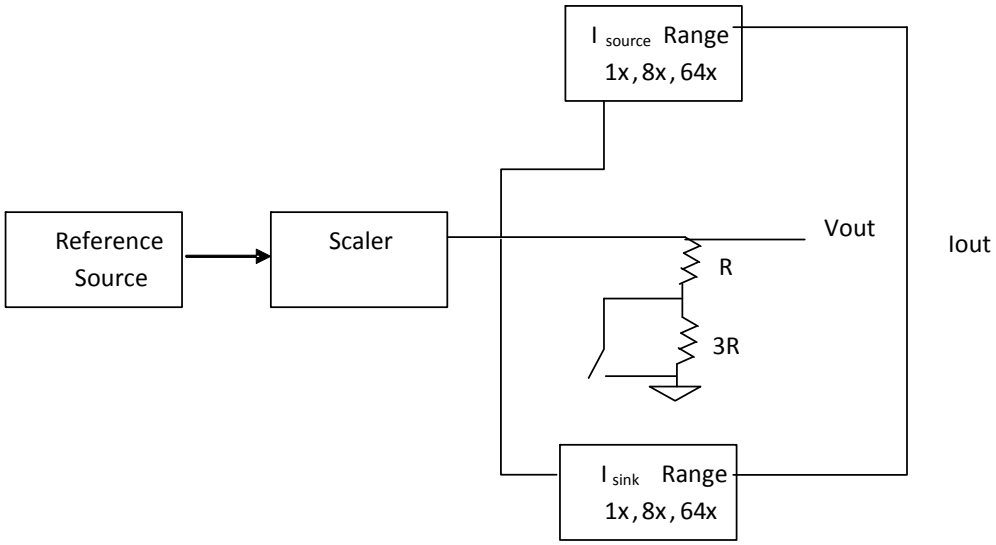
Die temperature is used to establish programming parameters for writing flash. Die temperature is measured using a dedicated sensor based on a forward biased transistor. The temperature sensor has its own auxiliary ADC.

8.9 DAC

The CY8C34 parts contain two Digital to Analog Convertors (DACs). Each DAC is 8-bit and can be configured for either voltage or current output. The DACs support CapSense, power supply regulation, and waveform generation. Each DAC has the following features:

- Adjustable voltage or current output in 255 steps
- Programmable step size (range selection)
- Eight bits of calibration to correct ± 25% of gain error
- Source and sink option for current output
- 8 Msps conversion rate for current output
- 1 Msps conversion rate for voltage output
- Monotonic in nature

Figure 8-10. DAC Block Diagram



8.9.1 Current DAC

The current DAC (IDAC) can be configured for the ranges 0 to 32 µA, 0 to 256 µA, and 0 to 2.048 mA. The IDAC can be configured to source or sink current.

8.9.2 Voltage DAC

For the voltage DAC (VDAC), the current DAC output is routed through resistors. The two ranges available for the VDAC are 0 to 1.024V and 0 to 4.096V. In voltage mode any load connected to the output of a DAC should be purely capacitive (the output of the VDAC is not buffered).

8.10 Up/Down Mixer

In continuous time mode, the SC/CT block components are used to build an up or down mixer. Any mixing application contains an input signal frequency and a local oscillator frequency. The polarity of the clock, Fclk, switches the amplifier between inverting or noninverting gain. The output is the product of the input and the switching function from the local oscillator, with frequency components at the local oscillator plus and minus the signal frequency (Fclk + Fin and Fclk - Fin) and reduced-level frequency components at odd integer multiples of the local oscillator frequency. The local oscillator frequency is provided by the selected clock source for the mixer.

Continuous time up and down mixing works for applications with input signals and local oscillator frequencies up to 1 MHz.

9.1 JTAG Interface

The IEEE 1149.1 compliant JTAG interface exists on four or five pins (the nTRST pin is optional). The JTAG clock frequency can be up to 8 MHz, or 1/3 of the CPU clock frequency for 8 and 16-bit transfers, or 1/5 of the CPU clock frequency for 32-bit transfers, whichever is least. By default, the JTAG pins are enabled on new devices but the JTAG interface can be disabled, allowing these pins to be used as General Purpose I/O (GPIO) instead. The JTAG interface is used for programming the flash memory, debugging, I/O scan chains, and JTAG device chaining.

9.2 Serial Wire Debug Interface

The SWD interface is the preferred alternative to the JTAG interface. It requires only two pins instead of the four or five needed by JTAG. SWD provides all of the programming and debugging features of JTAG at the same speed. SWD does not provide access to scan chains or device chaining. The SWD clock frequency can be up to 1/3 of the CPU clock frequency.

SWD uses two pins, either two of the JTAG pins (TMS and TCK) or the USBIO D+ and D- pins. The USBIO pins are useful for in system programming of USB solutions that would otherwise require a separate programming connector. One pin is used for the data clock and the other is used for data input and output. SWD can be enabled on only one of the pin pairs at a time. This only happens if, within 8 µs (key window) after reset, that pin pair (JTAG or USB) receives a predetermined sequence of 1s and 0s. SWD is used for debugging or for programming the flash memory.

The SWD interface can be enabled from the JTAG interface or disabled, allowing its pins to be used as GPIO. Unlike JTAG, the SWD interface can always be reacquired on any device during the key window. It can then be used to reenable the JTAG interface, if desired. When using SWD or JTAG pins as standard GPIO, make sure that the GPIO functionality and PCB circuits do not interfere with SWD or JTAG use.

9.3 Debug Features

Using the JTAG or SWD interface, the CY8C34 supports the following debug features:

- Halt and single-step the CPU
- View and change CPU and peripheral registers, and RAM addresses
- Eight program address breakpoints
- One memory access breakpoint—break on reading or writing any memory address and data value
- Break on a sequence of breakpoints (non recursive)
- Debugging at the full speed of the CPU
- Debug operations are possible while the device is reset, or in low power modes
- Compatible with PSoC Creator and MiniProg3 programmer and debugger
- Standard JTAG programming and debugging interfaces make CY8C34 compatible with other popular third-party tools (for example, ARM / Keil)

9.4 Trace Features

The CY8C34 supports the following trace features when using JTAG or SWD:

- Trace the 8051 program counter (PC), accumulator register (ACC), and one SFR / 8051 core RAM register
- Trace depth up to 1000 instructions if all registers are traced, or 2000 instructions if only the PC is traced (on devices that include trace memory)
- Program address trigger to start tracing
- Trace windowing, that is, only trace when the PC is within a given range
- Two modes for handling trace buffer full: continuous (overwriting the oldest trace data) or break when trace buffer is full

9.5 Single Wire Viewer Interface

The SWV interface is closely associated with SWD but can also be used independently. SWV data is output on the JTAG interface's TDO pin. If using SWV, the designer must configure the device for SWD, not JTAG. SWV is not supported with the JTAG interface.

SWV is ideal for application debug where it is helpful for the firmware to output data similar to 'printf' debugging on PCs. The SWV is ideal for data monitoring, because it requires only a single pin and can output data in standard UART format or Manchester encoded format. For example, it can be used to tune a PID control loop in which the output and graphing of the three error terms greatly simplifies coefficient tuning.

The following features are supported in SWV:

- 32 virtual channels, each 32 bits long
- Simple, efficient packing and serializing protocol
- Supports standard UART format (N81)

9.6 Programming Features

The JTAG and SWD interfaces provide full programming support. The entire device can be erased, programmed, and verified. Designers can increase flash protection levels to protect firmware IP. Flash protection can only be reset after a full device erase. Individual flash blocks can be erased, programmed, and verified, if block security settings permit.

9.7 Device Security

PSoC 3 offers an advanced security feature called device security, which permanently disables all test, programming, and debug ports, protecting your application from external access. The device security is activated by programming a 32-bit key (0x50536F43) to a Write Once Latch (WOL).

The Write Once Latch is a type of nonvolatile latch (NVL). The cell itself is an NVL with additional logic wrapped around it. Each WOL device contains four bytes (32 bits) of data. The wrapper outputs a '1' if a super-majority (28 of 32) of its bits match a pre-determined pattern (0x50536F43); it outputs a '0' if this majority is not reached. When the output is 1, the Write Once NV latch locks the part out of Debug and Test modes; it also permanently gates off the ability to erase or alter the contents of the latch. Matching all bits is intentionally not required, so that single (or few) bit failures do not deassert the WOL output. The state of

11. Electrical Specifications

Specifications are valid for -40°C ≤ T_A ≤ 85°C and T_J ≤ 100°C, except where noted. Specifications are valid for 1.71V to 5.5V, except where noted. The unique flexibility of the PSoC UDBs and analog blocks enable many functions to be implemented in PSoC Creator components, see the component data sheets for full AC/DC specifications of individual functions. See the "Example Peripherals" section on page 35 for further explanation of PSoC Creator components.

11.1 Absolute Maximum Ratings

Table 11-1. Absolute Maximum Ratings DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
T_{STG}	Storage temperature	Higher storage temperatures reduce NVL data retention time. Recommended storage temperature is +25°C ±25°C. Extended duration storage temperatures above 85°C degrade reliability.	-55	25	100	°C
V_{DDA}	Analog supply voltage relative to Vssa		-0.5	-	6	V
V_{DDD}	Digital supply voltage relative to Vssd		-0.5	-	6	V
Vddio	I/O supply voltage relative to Vssd		-0.5	-	6	V
Vcca	Direct analog core voltage input		-0.5	-	1.95	V
Vccd	Direct digital core voltage input		-0.5	-	1.95	V
Vssa	Analog ground voltage		Vssd -0.5	-	Vssd + 0.5	V
Vgpio[5]	DC input voltage on GPIO	Includes signals sourced by V_{DDA} and routed internal to the pin	Vssd -0.5	-	Vddio + 0.5	V
Vsio	DC input voltage on SIO	Output disabled	Vssd -0.5	-	7	V
		Output enabled	Vssd -0.5	-	6	V
Vind	Voltage at boost converter input		0.5	-	5.5	V
Vbat	Boost converter supply		Vssd -0.5	-	5.5	V
Ivddio	Current per Vddio supply pin		-	-	100	mA
ESD_{HBM}	Electro-static discharge voltage	Human Body Model	2200	-	-	V
ESD_{CDM}	Electro-static discharge voltage	Charge Device Model	500	-	-	V

Note Usage above the absolute maximum conditions listed in Table 11-1 may cause permanent damage to the device. Exposure to maximum conditions for extended periods of time may affect device reliability. When used below maximum conditions but above normal operating conditions the device may not operate to specification.

Note
5. The Vddio supply voltage must be greater than the maximum analog voltage on the associated GPIO pins. Maximum analog voltage on GPIO pin ≤ Vddio ≤ V_{DDA}.

Table 11-6. Inductive Boost Regulator DC Specifications (continued)

Parameter	Description	Conditions	Min	Typ	Max	Units
Vboost	Boost output voltage range[9]					
	1.8V		1.71	1.80	1.89	V
	1.9V		1.81	1.90	2.00	V
	2.0V		1.90	2.00	2.10	V
	2.4V		2.28	2.40	2.52	V
	2.7V		2.57	2.70	2.84	V
	3.0V		2.85	3.00	3.15	V
	3.3V		3.14	3.30	3.47	V
	3.6V		3.42	3.60	3.78	V
	5.0V	External diode required	4.75	5.00	5.25	V
	Load regulation		-	-	TBD	%
	Line regulation		-	-	TBD	%
	Efficiency	Vbat = 2.4 V, Vout = 2.7 V, Iout = 10 mA, Fsw = 400 kHz	90	-	-	%

Table 11-7. Inductive Boost Regulator AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
Vripple	Ripple voltage (peak-to-peak)	Vout = 1.8V, Fsw = 400 kHz, Iout = 10 mA	-	-	100	mV
Fsw	Switching frequency		-	0.1, 0.4, or 2	-	MHz
	Duty cycle		20	-	80	%

11.4 Inputs and Outputs

Specifications are valid for -40°C ≤ T_A ≤ 85°C and T_J ≤ 100°C, except where noted. Specifications are valid for 1.71V to 5.5V, except where noted.

11.4.1 GPIO

Table 11-8. GPIO DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
Vih	Input voltage high threshold	CMOS Input, PRT[x]CTL = 0	0.7 × Vddio	-	-	V
Vil	Input voltage low threshold	CMOS Input, PRT[x]CTL = 0	-	-	0.3 × Vddio	V
Vih	Input voltage high threshold	LVTTL Input, PRT[x]CTL = 1,Vddio < 2.7V	0.7 x Vddio	-	-	V
Vih	Input voltage high threshold	LVTTL Input, PRT[x]CTL = 1, Vddio ≥ 2.7V	2.0	-	-	V
Vil	Input voltage low threshold	LVTTL Input, PRT[x]CTL = 1,Vddio < 2.7V	-	-	0.3 x Vddio	V
Vil	Input voltage low threshold	LVTTL Input, PRT[x]CTL = 1, Vddio ≥ 2.7V	-	-	0.8	V
Voh	Output voltage high	Ioh = 4 mA at 3.3 Vddio	Vddio - 0.6	-	-	V
		Ioh = 1 mA at 1.8 Vddio	Vddio - 0.5	-	-	V
Vol	Output voltage low	Iol = 8 mA at 3.3 Vddio	-	-	0.6	V
		Iol = 4 mA at 1.8 Vddio	-	-	0.6	V
Rpullup	Pull up resistor		4	5.6	8	kΩ
Rpulldown	Pull down resistor		4	5.6	8	kΩ
Iil	Input leakage current (absolute value)[9]	25°C, Vddio = 3.0V	-	-	2	nA
Cin	Input capacitance[9]	GPIOs without OpAmp outputs	-	-	7	pF
		GPIOs with OpAmp outputs	-	-	18	pF
Vh	Input voltage hysteresis (Schmitt-Trigger)[9]		-	40	-	mV
Idiode	Current through protection diode to Vddio and Vssio		-	-	100	µA
Rglobal	Resistance pin to analog global bus	25°C, Vddio = 3.0V	-	320	-	Ω
Rmux	Resistance pin to analog mux bus	25°C, Vddio = 3.0V	-	220	-	Ω

Table 11-9. GPIO AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
TriseF	Rise time in Fast Strong Mode[9]	3.3V Vddio Cload = 25 pF	2	-	12	ns
TfallF	Fall time in Fast Strong Mode[9]	3.3V Vddio Cload = 25 pF	2	-	12	ns
TriseS	Rise time in Slow Strong Mode[9]	3.3V Vddio Cload = 25 pF	10	-	60	ns
TfallS	Fall time in Slow Strong Mode[9]	3.3V Vddio Cload = 25 pF	10	-	60	ns
Fgpioout	GPIO output operating frequency					
	3.3V ≤ Vddio ≤ 5.5V, fast strong drive mode	90/10% Vddio into 25 pF	-	-	33	MHz
	1.71V ≤ Vddio < 3.3V, fast strong drive mode	90/10% Vddio into 25 pF	-	-	20	MHz
	3.3V ≤ Vddio ≤ 5.5V, slow strong drive mode	90/10% Vddio into 25 pF	-	-	7	MHz
	1.71V ≤ Vddio < 3.3V, slow strong drive mode	90/10% Vddio into 25 pF	-	-	3.5	MHz
Fgpioin	GPIO input operating frequency					
	1.71V ≤ Vddio ≤ 5.5V	90/10% Vddio	-	-	50	MHz

11.5.6 IDAC

Table 11-25. IDAC (Current Digital-to-Analog Converter) DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
Resolution			-	8	-	
Iout	Output current					
	High[9]	Code = 255, $V_{DDA} \geq 2.7V$, RL 600Ω	-	2.040	-	mA
		Code = 255, $V_{DDA} \leq 2.7V$, RL 300Ω	-	2.040	-	mA
	Medium[9]	Code = 255, RL 600Ω	-	255	-	µA
	Low[9]	Code = 255, RL 600Ω	-	31.875	-	µA
INL	Integral non linearity	RL 600Ω, CL=15 pF	-	-	±1	LSB
DNL	Differential non linearity	RL 600Ω, CL=15 pF	-	-	±0.5	LSB
Ezs	Zero scale error		-	0	±1	LSB
Eg	Gain error	Uncompensated	-	-	3.5	%
		Temperature compensated	-	-	TBD	%
IDAC_ICC	DAC current low speed mode[9]	Code = 0	-	-	100	µA
IDAC_ICC	DAC current high speed mode[9]	Code = 0	-	-	500	µA

Table 11-26. IDAC (Current Digital-to-Analog Converter) AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
Fdac	Update rate		-	-	8	Msps
Tsettle	Settling time to 0.5LSB	Full scale transition, 600Ω load, CL = 15 pF				
	Fast mode	Independent of IDAC range setting (Iout)	-	-	100	ns
	Slow mode	Independent of IDAC range setting (Iout)	-	-	1000	ns

11.5.7 VDAC

Table 11-27. VDAC (Voltage Digital-to-Analog Converter) DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
Resolution			-	8	-	
Rout	Output resistance[9]					
	High	Vout = 4V	-	16	-	kΩ
	Low	Vout = 1V	-	4	-	kΩ
Vout	Output voltage range[9]					
	High	Code = 255, $V_{DDA} \geq 5V$	-	4	-	V
	Low	Code = 255	-	1	-	V
INL	Integral non linearity	CL=15 pF	-	-	±1	LSB
DNL	Differential non linearity	CL=15 pF	-	-	±1	LSB
Ezs	Zero scale error		-	-	±1	LSB
Eg	Gain error	Uncompensated	-	-	3	%
		Temperature compensated	-	-	TBD	%
VDAC_ICC	DAC current low speed mode[9]	Code = 0	-	-	100	µA
VDAC_ICC	DAC current high speed mode[9]	Code = 0	-	-	500	µA

11.5.10 Transimpedance Amplifier

The TIA is created using a SC/CT Analog Block, see the TIA component data sheet in PSoC Creator for full AC/DC specifications, and APIs and example code.

Table 11-33. Transimpedance Amplifier (TIA) DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
Vioff	Input offset voltage		-	-	10	mV
Rconv	Conversion resistance[16]					
	R = 20K	40 pF load	-20	-	+30	%
	R = 30K	40 pF load	-20	-	+30	%
	R = 40K	40 pF load	-20	-	+30	%
	R = 80K	40 pF load	-20	-	+30	%
	R = 120K	40 pF load	-20	-	+30	%
	R = 250K	40 pF load	-20	-	+30	%
	R= 500K	40 pF load	-20	-	+30	%
	R = 1M	40 pF load	-20	-	+30	%
	Quiescent current		-	900	-	µA

Table 11-34. Transimpedance Amplifier (TIA) AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	Input bandwidth (-3 dB) - 20 pF load[15]					
	R = 20K		1800	-	-	kHz
	R = 120K		330	-	-	kHz
	R = 1M		47	-	-	kHz
	Input bandwidth (3 dB) - 40 pF load					
	R = 20K		1500	-	-	kHz
	R = 120K		300	-	-	kHz
	R = 1M		46	-	-	kHz

Note

16. Conversion resistance values are not calibrated. Calibrated values and details about calibration are provided in PSoC Creator component data sheets. External precision resistors can also be used.

11.6.3 Pulse Width Modulation

Table 11-46. PWM DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	Block current consumption	16-bit PWM, at listed input clock frequency	-	-	-	µA
	3 MHz		-	8	-	µA
	12 MHz		-	30	-	µA
	50 MHz		-	120	-	µA

Table 11-47. Pulse Width Modulation (PWM) AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	Operating frequency		DC	-	50	MHz
	Pulse width		21	-	-	ns
	Pulse width (external)		42	-	-	ns
	Kill pulse width		21	-	-	ns
	Kill pulse width (external)		42			ns
	Enable pulse width		21	-	-	ns
	Enable pulse width (external)		42	-	-	ns
	Reset pulse width		21	-	-	ns
	Reset pulse width (external)		42	-	-	ns

11.6.4 I^2C

Table 11-48. Fixed I^2C DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	Block current consumption	Enabled, configured for 100 kbps	-	-	64	µA
		Enabled, configured for 400 kbps	-	-	74	µA
		Wake from sleep mode	-	-	TBD	µA

Table 11-49. Fixed I^2C AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	Bit rate		-	-	1	Mbps

11.6.5 Controller Area Network[19]

Table 11-50. CAN DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	Block current consumption	500 kbps	-	-	285	µA
		1 Mbps	-	-	330	µA

Table 11-51. CAN AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	Bit rate	Minimum 8 MHz clock	-	-	1	Mbit

Note
19. Refer to ISO 11898 specification for details.

Figure 11-5. Synchronous Read Cycle Timing

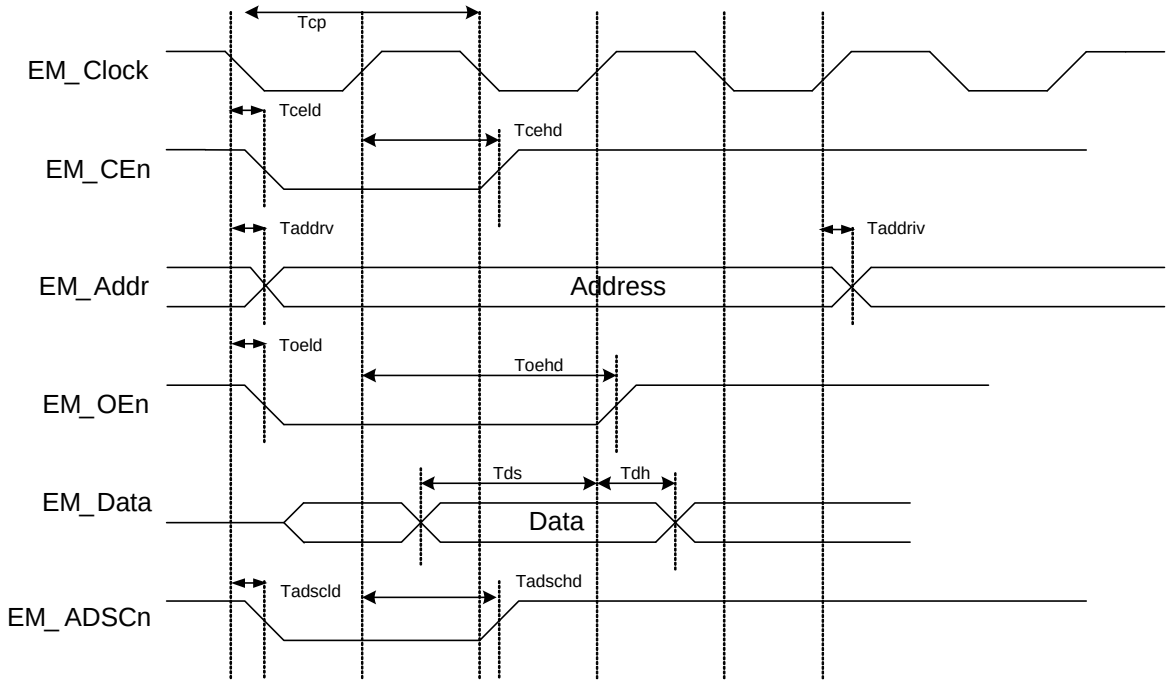


Table 11-64. Synchronous Read Cycle Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
T	EMIF Clock period		30.3	-	-	ns
Tcp	EM_Clock period		30.3	-	-	ns
Tceld	EM_Clock low to EM_CEn low			-	5	ns
Tcehd	EM_Clock high to EM_CEn high		T/2 - 2	-	-	ns
Taddrv	EM_Clock low to EM_Addr valid		-	-	5	ns
Taddriv	EM_Clock high to EM_Addr invalid		T/2 - 2	-	-	ns
Toeld	EM_Clock low to EM_OEn low		-	-	5	ns
Toehd	EM_Clock high to EM_OEn high		T+2	-	-	ns
Tds	Data valid before EM_Clock high		20	-	-	ns
Tdh	Data valid after EM_Clock high		2	-	-	ns
Tadscld	EM_clock low to EM_ADSCn low		-	-	5	ns
Tadschd	EM_clock high to EM_ADSCn high		T/2 - 2	-	-	ns

Figure 11-6. Synchronous Write Cycle Timing

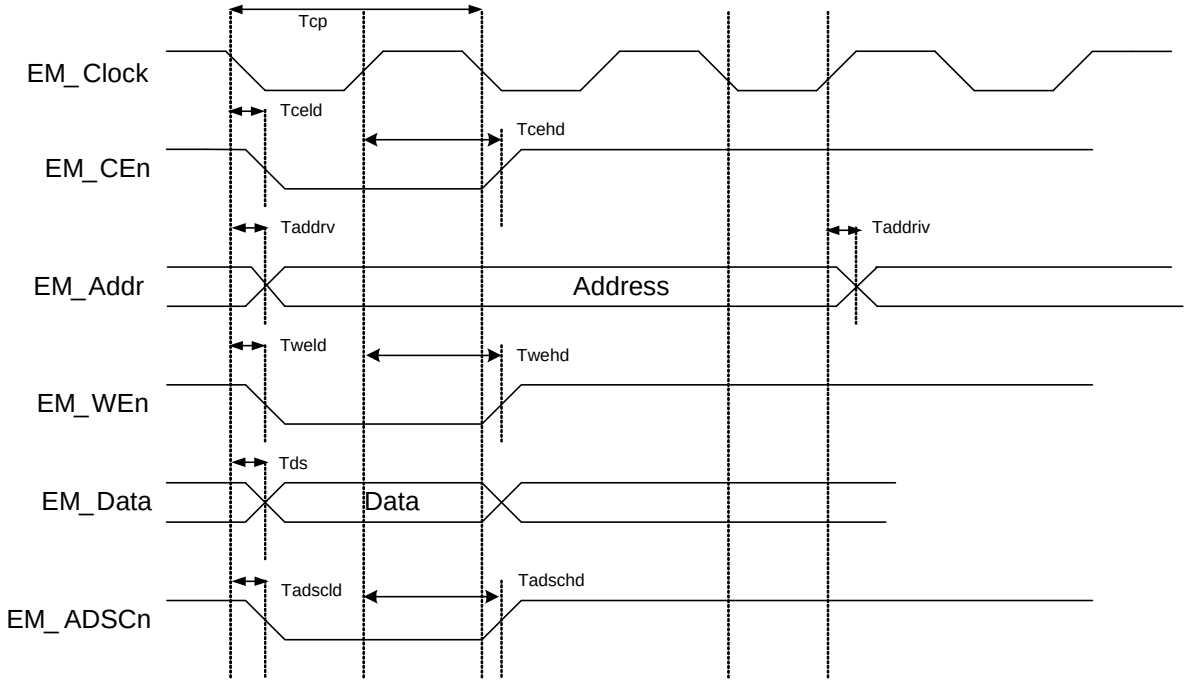


Table 11-65. Synchronous Write Cycle Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
T	EMIF Clock period		30.3	-	-	ns
Tcp	EM_Clock Period		30.3	-	-	ns
Tceld	EM_Clock low to EM_CEn low		-	-	5	ns
Tcehd	EM_Clock high to EM_CEn high		T/2 - 2	-	-	ns
Taddrv	EM_Clock low to EM_Addr valid		-	-	5	ns
Taddriv	EM_Clock high to EM_Addr invalid		T/2 - 2	-	-	ns
Tweld	EM_Clock low to EM_WEn low		-	-	5	ns
Twehd	EM_Clock high to EM_WEn high		T/2 - 2	-	-	ns
Tds	Data valid after EM_Clock low		-	-	5	ns
Tadscld	EM_clock low to EM_ADSCn low		-	-	5	ns
Tadschd	EM_clock high to EM_ADSCn high		T/2 - 2	-	-	ns

Table 11-66. External Memory Interface (EMIF) AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	Bus frequency		-	-	33	MHz

11.9 Clocking

Specifications are valid for -40°C ≤ T_A ≤ 85°C and T_J ≤ 100°C, except where noted. Specifications are valid for 1.71V to 5.5V, except where noted.

11.9.1 32 kHz External Crystal

Table 11-75. 32 kHz External Crystal DC Specifications[9]

Parameter	Description	Conditions	Min	Typ	Max	Units
Icc	Operating current	Low power mode	-	-	0.25	µA
CL	External crystal capacitance		-	6	-	pF
DL	Drive level		-	-	1	µW

Table 11-76. 32 kHz External Crystal AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
F	Frequency		-	32.768	-	kHz
DC	Output duty cycle[9]		20	50	80	%
Ton	Startup time	High power mode	-	1	-	s

11.9.2 Internal Main Oscillator

Table 11-77. IMO DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	Supply current					
	24 MHz - USB mode	With oscillator locking to USB bus	-	-	500	µA
	24 MHz - non USB mode		-	-	300	µA
	12 MHz		-	-	200	µA
	6 MHz		-	-	180	µA
	3 MHz		-	-	150	µA

Table 11-78. IMO AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
F_{IMO}	IMO frequency stability (with factory trim)					
	24 MHz - Non USB mode		-4	-	4	%
	24 MHz - USB mode	With oscillator locking to USB bus	-0.25	-	0.25	%
	12 MHz		-3	-	3	%
	6 MHz		-2	-	2	%
	3 MHz		-1	-	1	%
	Startup time[9]	From enable (during normal system operation) or wakeup from low power state	-	-	12	µs
Jp-p	Jitter (peak to peak)[9]					
	F = 24 MHz		-	0.9	-	ns
	F = 3 MHz		-	1.6	-	ns
Jperiod	Jitter (long term)[9]					
	F = 24 MHz		-	0.9	-	ns
	F = 3 MHz		-	12	-	ns