

#### 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 "<u>Embedded -</u> <u>Microcontrollers</u>"

## Details

-XF

Product Status	Obsolete
Core Processor	ARM® Cortex®-M3
Core Size	32-Bit Single-Core
Speed	67MHz
Connectivity	I <sup>2</sup> C, LINbus, SPI, UART/USART, USB
Peripherals	CapSense, DMA, LCD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 1x20b, 2x12b; D/A 4x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	68-VFQFN Exposed Pad
Supplier Device Package	68-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy8c5568lti-114

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong





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

# 3. Pin Descriptions

**IDAC0**, **IDAC1**, **IDAC2**, **IDAC3**. Low-resistance output pin for high-current DACs (IDAC).

**OpAmp0out, OpAmp1out, OpAmp2out, OpAmp3out.** High current output of uncommitted opamp.<sup>[6]</sup>

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

**OpAmp0-, OpAmp1-, OpAmp2-, OpAmp3-.** Inverting input to uncommitted opamp.

**OpAmp0+, OpAmp1+, OpAmp2+, OpAmp3+.** Noninverting input to uncommitted opamp.

SAR0ref, SAR1ref. External references for SAR ADCs.

**GPIO.** Provides interfaces to the CPU, digital peripherals, analog peripherals, interrupts, LCD segment drive, and CapSense.<sup>[6]</sup>

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

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

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

**SWDIO.** SWD Input and Output programming and debug port connection.

## SWV. SWV output.

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

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

**VCCA.** Output of analog core regulator and input to analog core. Requires a 1  $\mu$ F capacitor to VSSA (10  $\mu$ F is required for sleep mode. See Table 11-3). 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  $\mu$ F capacitor to VSSD (10  $\mu$ F is required for sleep mode. See Table 11-3); see Power System on page 21. Regulator output not for external use.

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.<sup>[7]</sup>

**VDDD.** Supply for all digital peripherals and digital core regulator. VDDD must be less than or equal to VDDA.<sup>[7]</sup>

VSSA. Ground for all analog peripherals.

VSSD. Ground for all digital logic and I/O pins.

**VDDIO<sub>0</sub>, VDDIO1, VDDIO2, VDDIO3.** Supply for I/O pins. Each VDDIO must be tied to a valid operating voltage (2.7 V to 5.5 V), and must be less than or equal to VDDA.

XRES. External reset pin. Active low with internal pull-up.

**RSVD.** Reserved pins

#### Notes

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

7. VDDD and VDDA must be brought up in synchronization with each other, that is, at the same rates and levels. VDDA must be greater than or equal to all other supplies.



# 4. CPU

# 4.1 ARM Cortex-M3 CPU

The CY8C55 family of devices has an ARM Cortex-M3 CPU core. The Cortex-M3 is a low-power 32-bit three-stage pipelined Harvard-architecture CPU that delivers 1.25 DMIPS/MHz. It is intended for deeply embedded applications that require fast interrupt handling features.

Figure 4-1. ARM Cortex-M3 Block Diagram



The Cortex-M3 CPU subsystem includes these features:

- ARM Cortex-M3 CPU
- Programmable nested vectored interrupt controller (NVIC), tightly integrated with the CPU core
- Full featured debug and trace module, tightly integrated with the CPU core
- Up to 256 KB of flash memory, 2 KB of EEPROM, and 64 KB of SRAM
- Cache controller with 128 bytes of memory
- Peripheral HUB (PHUB)
- DMA controller

## 4.1.1 Cortex-M3 Features

The Cortex-M3 CPU features include:

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





The Cortex-M3 does not support ARM instructions.

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

#### 4.1.2 Cortex-M3 Operating Modes

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

#### Table 4-1. Operational Level

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

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

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

## 4.1.3 CPU Registers

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

## Table 4-2. Cortex M3 CPU Registers

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

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

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



# 4.2 Cache Controller

The CY8C55 family has 128 bytes of direct mapped instruction cache between the CPU and the flash memory. This allows the CPU to access instructions much faster. The cache is enabled by default but user have the option to disable it.

# 4.3 DMA and PHUB

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

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

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

#### 4.3.1 PHUB Features

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

## Table 4-3. PHUB Spokes and Peripherals

PHUB Spokes	Peripherals
0	SRAM
1	IOs, PICU
2	PHUB local configuration, Power manager, Clocks, IC, EEPROM, Flash programming interface
3	Analog interface and trim, Decimator
4	USB, I <sup>2</sup> C, Timers, Counters, and PWMs
5	DFB
6	UDBs group 1
7	UDBs group 2

#### 4.3.2 DMA Features

- 24 DMA channels
- Each channel has one or more transaction descriptors (TDs) to configure channel behavior. Up to 127 total TDs can be defined
- TDs can be dynamically updated
- Eight levels of priority per channel

- Any digitally routable signal, the CPU, or another DMA channel, can trigger a transaction
- Each channel can generate up to two interrupts per transfer
- Transactions can be stalled or canceled
- Supports transaction size of infinite or 1 to 64k bytes
- Large transactions may be broken into smaller bursts of 1 to 127 bytes
- TDs may be nested and/or chained for complex transactions

#### 4.3.3 Priority Levels

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

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

#### Table 4-4. Priority Levels

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

#### 4.3.4 Transaction Modes Supported

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

#### 4.3.4.1 Simple DMA

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



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

#### 6.1.2 External Oscillators

#### 6.1.2.1 MHz External Crystal Oscillator

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

### Figure 6-2. MHzECO Block Diagram



#### 6.1.2.2 32.768 kHz ECO

The 32.768 kHz external crystal oscillator (32kHzECO) provides precision timing with minimal power consumption using an external 32.768 kHz watch crystal (see Figure 6-3). The RTC uses a 1 second interrupt to implement the RTC functionality in firmware. The oscillator works in two distinct power modes. This allows users to trade off power consumption with noise immunity from neighboring circuits. The GPIO pins connected to the external crystal and capacitors are fixed.

#### Figure 6-3. 32kHzECO Block Diagram



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

#### 6.1.2.3 Digital System Interconnect

The DSI provides routing for clocks taken from external clock oscillators connected to I/O. The oscillators can also be generated within the device in the digital system and UDBs. While the primary DSI clock input provides access to all clocking resources, up to eight other DSI clocks (internally or externally generated) may be routed directly to the eight digital clock dividers. This is only possible if there are multiple precision clock sources.

#### 6.1.3 Clock Distribution

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

- The system clock is used to select and supply the fastest clock in the system for general system clock requirements and clock synchronization of the PSoC device.
- Bus clock 16-bit divider uses the system clock to generate the system's bus clock used for data transfers and the CPU. The CPU clock is directly derived from the bus clock.





- Eight fully programmable 16-bit clock dividers generate digital system clocks for general use in the digital system, as configured by the design's requirements. Digital system clocks can generate custom clocks derived from any of the seven clock sources for any purpose. Examples include baud rate generators, accurate PWM periods, and timer clocks, and many others. If more than eight digital clock dividers are required, the UDBs and fixed function timer/counter/PWMs can also generate clocks.
- Four 16-bit clock dividers generate clocks for the analog system components that require clocking, such as ADCs 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 the MHzECO or DSI signal.

#### 6.2 Power System

The power system consists of separate analog, digital, and I/O supply pins, labeled VDDA, VDDD, and VDDIO<sub>X</sub>, respectively. It also includes two internal 1.8 V 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 (10  $\mu$ F is required for sleep mode. See Table 11-3). The two VCCD pins must be shorted together, with as short a trace as possible. The power system also contains a hibernate regulator.





#### Note

8. 10  $\mu$ F is required for sleep mode. See Table 11-3.



## 6.2.1 Power Modes

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

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

- Active
- Alternate active

Table 6-2. Power Modes

- Sleep
- Hibernate

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

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

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

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

Notes

9. Bus clock off. Execute from CPU instruction buffer at 6 MHz. See Table 11-2 on page 58.

During sleep mode, the CTW generates periodic interrupts to wake up the device. This affects the average current, which is a composite of the sleep mode current and active mode current, and the time spent in each mode. With the maximum wakeup interval of 128 ms, and at wakeup the CPU executes only the standard PSoC Creator sleep API (for a duty cycle of 0.2%), the average current draw is typically 35 µA.



# 7.1.2 Example Analog Components

The following is a sample of the analog components available in PSoC Creator for the CY8C55 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

#### ADCs

- Delta-Sigma
- □ Successive Approximation (SAR)
- DACs
- Current
- □ Voltage
- 🗆 PWM
- Comparators
- Mixers

#### 7.1.3 Example System Function Components

The following is a sample of the system function components available in PSoC Creator for the CY8C55 family. The exact amount of hardware resources (UDBs, DFB taps, SC/CT blocks, routing, RAM, flash) used by a component varies with the features selected in PSoC Creator for the component.

- CapSense
- LCD Drive
- LCD Control
- Filters

## 7.1.4 Designing with PSoC Creator

7.1.4.1 More Than a Typical IDE

A successful design tool allows for the rapid development and deployment of both simple and complex designs. It reduces or eliminates any learning curve. It makes the integration of a new design into the production stream straightforward.

PSoC Creator is that design tool.

PSoC Creator is a full featured Integrated Development Environment (IDE) for hardware and software design. It is optimized specifically for PSoC devices and combines a modern, powerful software development platform with a sophisticated graphical design tool. This unique combination of tools makes PSoC Creator the most flexible embedded design platform available.

Graphical design entry simplifies the task of configuring a particular part. You can select the required functionality from an extensive catalog of components and place it in your design. All components are parameterized and have an editor dialog that allows you to tailor functionality to your needs.

PSoC Creator automatically configures clocks and routes the I/O to the selected pins and then generates APIs to give the application complete control over the hardware. Changing the PSoC device configuration is as simple as adding a new component, setting its parameters, and rebuilding the project.

At any stage of development you are free to change the hardware configuration and even the target processor. To retarget your application (hardware and software) to new devices, even from 8- to 32-bit families, just select the new device and rebuild.

You also have the ability to change the C compiler and evaluate an alternative. Components are designed for portability and are validated against all devices, from all families, and against all supported tool chains. Switching compilers is as easy as editing from the project options and rebuilding the application with no errors from the generated APIs or boot code.



## 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<sup>®</sup> Limited, Keil<sup>™</sup>, 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<sup>™</sup> compiler.

#### 7.1.4.5 Nonintrusive Debugging

#### Figure 7-5. PSoC Creator Debugger



With SWD 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. PSoC Creator contains all the tools necessary to complete a design, and then to maintain and extend that design for years to come. All steps of the design flow are carefully integrated and optimized for ease-of-use and to maximize productivity.

# 7.2 Universal Digital Block

The Universal Digital Block (UDB) represents an evolutionary step to the next generation of PSoC embedded digital peripheral functionality. The architecture in first generation PSoC digital blocks provides coarse programmability in which a few fixed functions with a small number of options are available. The new UDB architecture is the optimal balance between configuration granularity and efficient implementation. A cornerstone of this approach is to provide the ability to customize the devices digital operation to match application requirements.

To achieve this, UDBs consist of a combination of uncommitted logic (PLD), structured logic (Datapath), and a flexible routing scheme to provide interconnect between these elements, I/O connections, and other peripherals. UDB functionality ranges from simple self contained functions that are implemented in one UDB, or even a portion of a UDB (unused resources are available for other functions), to more complex functions that require multiple UDBs. Examples of basic functions are timers, counters, CRC generators, PWMs, dead band generators, and communications functions, such as UARTs, SPI, and I<sup>2</sup>C. Also, the PLD blocks and connectivity provide full featured general purpose programmable logic within the limits of the available resources.

#### Figure 7-6. UDB Block Diagram



The main component blocks of the UDB are:

- PLD blocks There are two small PLDs per UDB. These blocks take inputs from the routing array and form registered or combinational sum-of-products logic. PLDs are used to implement state machines, state bits, and combinational logic equations. PLD configuration is automatically generated from graphical primitives.
- Datapath Module This 8-bit wide datapath contains structured logic to implement a dynamically configurable ALU, a variety of compare configurations and condition generation. This block also contains input/output FIFOs, which are the primary parallel data interface between the CPU/DMA system and the UDB.
- Status and Control Module The primary role of this block is to provide a way for CPU firmware to interact and synchronize with UDB operation.
- Clock and Reset Module This block provides the UDB clocks and reset selection and control.



## 7.2.2.3 Conditionals

Each datapath has two compares, with bit masking options. Compare operands include the two accumulators and the two data registers in a variety of configurations. Other conditions include zero detect, all ones detect, and overflow. These conditions are the primary datapath outputs, a selection of which can be driven out to the UDB routing matrix. Conditional computation can use the built in chaining to neighboring UDBs to operate on wider data widths without the need to use routing resources.

#### 7.2.2.4 Variable MSB

The most significant bit of an arithmetic and shift function can be programmatically specified. This supports variable width CRC and PRS functions, and in conjunction with ALU output masking, can implement arbitrary width timers, counters and shift blocks.

## 7.2.2.5 Built in CRC/PRS

The datapath has built in support for single cycle Cyclic Redundancy Check (CRC) computation and Pseudo Random Sequence (PRS) generation of arbitrary width and arbitrary polynomial. CRC/PRS functions longer than 8 bits may be implemented in conjunction with PLD logic, or built in chaining may be use to extend the function into neighboring UDBs.

## 7.2.2.6 Input/Output FIFOs

Each datapath contains two four-byte deep FIFOs, which can be independently configured as an input buffer (system bus writes to the FIFO, datapath internal reads the FIFO), or an output buffer (datapath internal writes to the FIFO, the system bus reads from the FIFO). The FIFOs generate status that are selectable as datapath outputs and can therefore be driven to the routing, to interact with sequencers, interrupts, or DMA.

## Figure 7-2. Example FIFO Configurations



## 7.2.2.7 Chaining

The datapath can be configured to chain conditions and signals such as carries and shift data with neighboring datapaths to create higher precision arithmetic, shift, CRC/PRS functions.

## 7.2.2.8 Time Multiplexing

In applications that are over sampled, or do not need high clock rates, the single ALU block in the datapath can be efficiently shared with two sets of registers and condition generators. Carry and shift out data from the ALU are registered and can be selected as inputs in subsequent cycles. This provides support for 16-bit functions in one (8-bit) datapath.

#### 7.2.2.9 Datapath I/O

There are six inputs and six outputs that connect the datapath to the routing matrix. Inputs from the routing provide the configuration for the datapath operation to perform in each cycle, and the serial data inputs. Inputs can be routed from other UDB blocks, other device peripherals, device I/O pins, and so on. The outputs to the routing can be selected from the generated conditions, and the serial data outputs. Outputs can be routed to other UDB blocks, device peripherals, interrupt and DMA controller, I/O pins, and so on.

## 7.2.3 Status and Control Module

The primary purpose of this circuitry is to coordinate CPU firmware interaction with internal UDB operation.

#### Figure 7-3. Status and Control Registers



Routing Channel

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.



# 7.8 Digital Filter Block

Some devices in the CY8C55 family of devices have a dedicated HW accelerator block used for digital filtering. The DFB has a dedicated multiplier and accumulator that calculates a 24-bit by 24-bit multiply accumulate in one system clock cycle. This enables the mapping of a direct form FIR filter that approaches a computation rate of one FIR tap for each clock cycle. The MCU can implement any of the functions performed by this block, but at a slower rate that consumes significant MCU bandwidth.

The PSoC Creator interface provides a wizard to implement FIR and IIR digital filters with coefficients for LPF, BPF, HPF, Notch and arbitrary shape filters. 64 pairs of data and coefficients are stored. This enables a 64 tap FIR filter or up to 4 16 tap filters of either FIR or IIR formulation.

## Figure 7-14. DFB Application Diagram (pwr/gnd not shown)



The typical use model is for data to be supplied to the DFB over the system bus from another on-chip system data source such as an ADC. The data typically passes through main memory or is directly transferred from another chip resource through DMA. The DFB processes this data and passes the result to another on chip resource such as a DAC or main memory through DMA on the system bus.

Data movement in or out of the DFB is typically controlled by the system DMA controller but can be moved directly by the MCU.

# 8. Analog Subsystem

The analog programmable system creates application specific combinations of both standard and advanced analog signal processing blocks. These blocks are then interconnected to each other and also to any pin on the device, providing a high level of design flexibility and IP security. The features of the analog subsystem are outlined here to provide an overview of capabilities and architecture.

- Flexible, configurable analog routing architecture provided by analog globals, analog mux bus, and analog local buses
- High resolution Delta-Sigma ADC
- Two successive approximation (SAR) ADCs
- Four 8-bit DACs that provide either voltage or current output
- Four comparators with optional connection to configurable LUT outputs
- Four configurable switched capacitor/continuos time (SC/CT) blocks for functions that include opamp, unity gain buffer, programmable gain amplifier, transimpedance amplifier, and mixer
- Four opamps for internal use and connection to GPIO that can be used as high current output buffers
- CapSense subsystem to enable capacitive touch sensing
- Precision reference for generating an accurate analog voltage for internal analog blocks





## Figure 8-2. CY8C55 Analog Interconnect

Analog local buses (abus) are routing resources located within the analog subsystem and are used to route signals between different analog blocks. There are eight abus routes in PSoC 5, four in the left half (abusl [0:3]) and four in the right half (abusr [0:3]) as shown in Figure 8-2. Using the abus saves the analog globals and analog mux buses from being used for interconnecting the analog blocks. Multiplexers and switches exist on the various buses to direct signals into and out of the analog blocks. A multiplexer can have only one connection on at a time, whereas a switch can have multiple connections on simultaneously. In Figure 8-2, multiplexers are indicated by grayed ovals and switches are indicated by transparent ovals.





## Figure 8-12. DAC Block Diagram

## 8.10.1 Current DAC

The current DAC (IDAC) can be configured for the ranges 0 to 31.875  $\mu$ A, 0 to 255  $\mu$ A, and 0 to 2.04 mA. The IDAC can be configured to source or sink current.

## 8.10.2 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 1.02 V and 0 to 4.08 V. In voltage mode any load connected to the output of a DAC should be purely capacitive (the output of the VDAC is not buffered).

# 8.11 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.

## Figure 8-13. Mixer Configuration



# 8.12 Sample and Hold

The main application for a sample and hold, is to hold a value stable while an ADC is performing a conversion. Some applications require multiple signals to be sampled simultaneously, such as for power calculations (V and I).

# Figure 8-14. Sample and Hold Topology ( $\Phi$ 1 and $\Phi$ 2 are opposite phases of a clock)



#### 8.12.1 Down Mixer

The S+H can be used as a mixer to down convert an input signal. This circuit is a high bandwidth passive sample network that can sample input signals up to 14 MHz. This sampled value is then held using the opamp with a maximum clock rate of 4 MHz. The output frequency is at the difference between the input frequency and the highest integer multiple of the Local Oscillator that is less than the input.

#### 8.12.2 First Order Modulator - SC Mode

A first order modulator is constructed by placing the switched capacitor block in an integrator mode and using a comparator to provide a 1-bit feedback to the input. Depending on this bit, a reference voltage is either subtracted or added to the input signal. The block output is the output of the comparator and not the integrator in the modulator case. The signal is downshifted and buffered and then processed by a decimator to make a delta-sigma converter or a counter to make an incremental converter. The accuracy of the sampled data from the first-order modulator is determined from several factors. The main application for this modulator is for a low frequency ADC with high accuracy. Applications include strain gauges, thermocouples, precision voltage, and current measurement.



# **10. Development Support**

The CY8C55 family has a rich set of documentation, development tools, and online resources to assist you during your development process. Visit

psoc.cypress.com/getting-started to find out more.

# 10.1 Documentation

A suite of documentation, to ensure that you can find answers to your questions quickly, supports the CY8C55 family. This section contains a list of some of the key documents.

**Software User Guide**: A step-by-step guide for using PSoC Creator. The software user guide shows you how the PSoC Creator build process works in detail, how to use source control with PSoC Creator, and much more.

**Component Datasheets**: The flexibility of PSoC allows the creation of new peripherals (components) long after the device has gone into production. Component datasheets provide all of the information needed to select and use a particular component, including a functional description, API documentation, example code, and AC/DC specifications.

Application Notes: PSoC application notes discuss a particular application of PSoC in depth; examples include brushless DC motor control and on-chip filtering. Application notes often include example projects in addition to the application note document.

**Technical Reference Manual**: PSoC Creator makes designing with PSoC as easy as dragging a peripheral onto a schematic, but, when low level details of the PSoC device are required, use the technical reference manual (TRM) as your guide.

**Note** Visit www.arm.com for detailed documentation about the Cortex-M3 CPU.

# 10.2 Online

In addition to print documentation, the Cypress PSoC forums connect you with fellow PSoC users and experts in PSoC from around the world, 24 hours a day, 7 days a week.

## 10.3 Tools

With industry standard cores, programming, and debugging interfaces, the CY8C55 family is part of a development tool ecosystem. Visit us at www.cypress.com/go/psoccreator for the latest information on the revolutionary, easy to use PSoC Creator IDE, supported third party compilers, programmers, debuggers, and development kits.



# Table 11-11. USBIO AC Specifications

Parameter	Description	Conditions	Min	Тур	Max	Units
Tdrate	Full-speed data rate average bit rate	Using external 24 MHz crystal	12 – 0.25%	12	12 + 0.25%	MHz
Tjr1	Receiver data jitter tolerance to next transition		-8	_	8	ns
Tjr2	Receiver data jitter tolerance to pair transition		-5	_	5	ns
Tdj1	Driver differential jitter to next transition		-3.5	-	3.5	ns
Tdj2	Driver differential jitter to pair transition		-4	-	4	ns
Tfdeop	Source jitter for differential transition to SE0 transition		-2	_	5	ns
Tfeopt	Source SE0 interval of EOP		160	-	175	ns
Tfeopr	Receiver SE0 interval of EOP		82	-	-	ns
Tfst	Width of SE0 interval during differential transition		-	_	14	ns
Fgpio_out	GPIO mode output operating frequency	$3~V \leq V_{DDD} \leq 5.5~V$	-	_	20	MHz
		$V_{DDD} = 2.7 V$	_	-	6	MHz
Tr_gpio	Rise time, GPIO mode, 10%/90% $V_{DDD}$	V <sub>DDD</sub> > 3 V, 25 pF load	-	-	12	ns
		V <sub>DDD</sub> = 2.7 V, 25 pF load	-	-	40	ns
Tf_gpio	Fall time, GPIO mode, 90%/10% V <sub>DDD</sub>	V <sub>DDD</sub> > 3 V, 25 pF load	-	_	12	ns
		V <sub>DDD</sub> = 2.7 V, 25 pF load	-	_	40	ns

# Figure 11-16. USBIO Output Rise and Fall Times, GPIO Mode, $V_{DDD}$ = 3.3 V, 25 pF Load



Table 11	-12. USE	Driver AC	Specifications
----------	----------	-----------	----------------

Parameter	Description	Conditions	Min	Тур	Max	Units
Tr	Transition rise time		-	-	20	ns
Tf	Transition fall time		-	-	20	ns
TR	Rise/fall time matching	V <sub>USB_5</sub> , V <sub>USB_3.3</sub> , see USB DC Specifications on page 92	80%	_	135%	
Vcrs	Output signal crossover voltage		1.1	-	2.3	V



## 11.5.7 Current Digital-to-analog Converter (IDAC)

All specifications are based on use of the low-resistance IDAC output pins (see Pin Descriptions on page 9 for details). See the IDAC component data sheet in PSoC Creator for full electrical specifications and APIs.

Unless otherwise specified, all charts and graphs show typical values.

# Table 11-30. IDAC DC Specifications

Parameter	Description	Conditions		Тур	Max	Units
	Resolution		-	-	8	bits
I <sub>OUT</sub>	Output current at code = 255	Range = 2.04 mA, code = 255, Rload = $600 \Omega$	-	2.04	-	mA
		Range = 255 $\mu$ A, code = 255, Rload = 600 $\Omega$	_	255	-	μA
		Range = 31.875 $\mu$ A, code = 255, Rload = 600 $\Omega$	-	31.875	-	μA
	Monotonicity		-	-	Yes	
Ezs	Zero scale error		-	0	±2.5	LSB
Eg	Gain error		-	-	±5	%
TC_Eg	Temperature coefficient of gain	Range = 2.04 mA	_	-	0.04	% / °C
	error	Range = 255 μΑ	_	-	0.04	% / °C
		Range = 31.875 µA	_	-	0.05	% / °C
INL	Integral nonlinearity	Range = 255 μA, Codes 8 – 255, Rload = 600 Ω, Cload = 15 pF	-	-	±3	LSB
DNL	Differential nonlinearity, non-monotonic	Range = 255 $\mu$ A, Rload = 600 $\Omega$ , Cload = 15 pF	-	-	±1.6	LSB
Vcompliance	Dropout voltage, source or sink mode	Voltage headroom at max current, Rload to $V_{DDA}$ or Rload to $V_{SSA}$ , $V_{DIFF}$ from $V_{DDA}$	1	-	-	V
I <sub>DD</sub>	Operating current, code = 0	Slow mode, source mode, range = 31.875 µA	-	44	100	μA
		Slow mode, source mode, range = 255 µA,	-	33	100	μA
		Slow mode, source mode, range = 2.04 mA	-	33	100	μA
		Slow mode, sink mode, range = 31.875 µA	_	36	100	μA
		Slow mode, sink mode, range = 255 µA	_	33	100	μA
		Slow mode, sink mode, range = 2.04 mA	_	33	100	μA
		Fast mode, source mode, range = 31.875 μΑ	-	310	500	μA
		Fast mode, source mode, range = 255 µA	_	305	500	μA
		Fast mode, source mode, range = 2.04 mA	_	305	500	μA
		Fast mode, sink mode, range = 31.875 µA	_	310	500	μA
		Fast mode, sink mode, range = 255 µA	_	300	500	μA
		Fast mode, sink mode, range = 2.04 mA	_	300	500	μA





## Figure 11-58. Bandwidth vs. Temperature, Gain = 1, Power Mode = High

Figure 11-59. Noise vs. Frequency,  $V_{DDA} = 5 V$ , Power Mode = High

# 11.5.12 LCD Direct Drive

# Table 11-40. LCD Direct Drive DC Specifications

Parameter	Description	Conditions	Min	Тур	Max	Units
I <sub>CC</sub>	LCD system operating current	Bus clock = 3 MHz, $V_{DDIO} = V_{DDA} =$ 3 V, 4 commons, 16 segments, 1/4 duty cycle, 50 Hz frame rate, no glass connected	-	63	_	μΑ
I <sub>CC_SEG</sub>	Current per segment driver		-	148	-	μA
V <sub>BIAS</sub>	LCD bias range (V <sub>BIAS</sub> refers to the main output voltage(V0) of LCD DAC)	3 V $\leq$ V_{BIAS} $\leq$ V_{DDIO} for the drive pin	2.09	-	5.2	V
	LCD bias step size	3 V $\leq$ V_{BIAS} $\leq$ V_{DDIO} for the drive pin	-	25.8	-	mV
	LCD capacitance per segment/common driver	Drivers may be combined	-	500	5000	pF
	Long term segment offset	$V_{BIAS} \le V_{DDA} - 0.5 V$	-	-	20	mV
I <sub>OUT</sub>	Output drive current per segment driver	V <sub>DDIO</sub> = 5.5 V	90	-	165	μA

Table 11-41. LCD Direct Drive AC Specifications

Parameter	Description	Conditions	Min	Тур	Max	Units
f <sub>LCD</sub>	LCD frame rate		10	50	150	Hz

1000



# 11.8 PSoC System Resources

Specifications are valid for –40 °C  $\leq$  T<sub>A</sub>  $\leq$  85 °C and T<sub>J</sub>  $\leq$  100 °C, except where noted. Specifications are valid for 2.7 V to 5.5 V, except where noted.

11.8.1 Voltage Monitors

## Table 11-61. Voltage Monitors DC Specifications

Parameter	Description	Conditions	Min	Тур	Max	Units
LVI	Trip voltage					
	LVI_A/D_SEL[3:0] = 0011b		2.38	2.45	2.53	V
	LVI_A/D_SEL[3:0] = 0100b		2.62	2.71	2.79	V
	LVI_A/D_SEL[3:0] = 0101b		2.87	2.95	3.04	V
	LVI_A/D_SEL[3:0] = 0110b		3.11	3.21	3.31	V
	LVI_A/D_SEL[3:0] = 0111b		3.35	3.46	3.56	V
	LVI_A/D_SEL[3:0] = 1000b		3.59	3.70	3.81	V
	LVI_A/D_SEL[3:0] = 1001b		3.84	3.95	4.07	V
	LVI_A/D_SEL[3:0] = 1010b		4.08	4.20	4.33	V
	LVI_A/D_SEL[3:0] = 1011b		4.32	4.45	4.59	V
	LVI_A/D_SEL[3:0] = 1100b		4.56	4.70	4.84	V
	LVI_A/D_SEL[3:0] = 1101b		4.83	4.98	5.13	V
	LVI_A/D_SEL[3:0] = 1110b		5.05	5.21	5.37	V
	LVI_A/D_SEL[3:0] = 1111b		5.30	5.47	5.63	V
HVI	Trip voltage		5.57	5.75	5.92	V

# Table 11-62. Voltage Monitors AC Specifications

Parameter	Description	Conditions	Min	Тур	Max	Units
Response time <sup>[43]</sup>			-	1	-	μs

## 11.8.2 Interrupt Controller

## Table 11-63. Interrupt Controller AC Specifications

Parameter	Description	Conditions	Min	Тур	Max	Units
	Delay from interrupt signal input to ISR code execution from main line code <sup>[44]</sup>		-	_	12	Tcy CPU
	Delay from interrupt signal input to ISR code execution from ISR code (tail-chaining) <sup>[44]</sup>		_	-	6	Tcy CPU

Notes

43. Based on device characterization (Not production tested).

44. ARM Cortex-M3 NVIC spec. Visit www.arm.com for detailed documentation about the Cortex-M3 CPU.



## 11.8.3 SWD Interface



# Figure 11-61. SWD Interface Timing

# Table 11-64. SWD Interface AC Specifications<sup>[45]</sup>

Parameter	Description	Conditions	Min	Тур	Max	Units
f_SWDCK	SWDCLK frequency	$3.3~V \le V_{DDD} \le 5~V$	-	-	12 <sup>[46]</sup>	MHz
		$2.7 \text{ V} \leq \text{V}_{\text{DDD}} < 3.3 \text{ V}$	_	_	6.5 <sup>[46]</sup>	MHz
		$2.7 \text{ V} \leq \text{V}_{DDD}$ < $3.3 \text{ V}$ , SWD over USBIO pins	-	_	5 <sup>[46]</sup>	MHz
T_SWDI_setup	SWDIO input setup before SWDCK high	T = 1/f_SWDCK max	T/4	-	-	
T_SWDI_hold	SWDIO input hold after SWDCK high	T = 1/f_SWDCK max	T/4	_	_	
T_SWDO_valid	SWDCK high to SWDIO output	T = 1/f_SWDCK max	-	_	2T/5	

# 11.8.4 TPIU Interface

# Table 11-65. TPIU Interface AC Specifications<sup>[45]</sup>

Parameter	Description	Conditions	Min	Тур	Max	Units
	SWV bit rate		-	_	33 <sup>[47]</sup>	Mbit

 Notes

 45. Based on device characterization (Not production tested).

 46. f\_SWDCK must also be no more than 1/3 CPU clock frequency.

 47. SWV signal frequency and bit rate are limited by GPIO output frequency, see "GPIO AC Specifications" on page 62.



# 14. Acronyms

# Table 14-1. Acronyms Used in this Document

Acronym	Description
abus	analog local bus
ADC	analog-to-digital converter
AG	analog global
АНВ	AMBA (advanced microcontroller bus archi- tecture) high-performance bus, an ARM data transfer bus
ALU	arithmetic logic unit
AMUXBUS	analog multiplexer bus
API	application programming interface
APSR	application program status register
ARM®	advanced RISC machine, a CPU architecture
ATM	automatic thump mode
BW	bandwidth
CMRR	common-mode rejection ratio
CPU	central processing unit
CRC	cyclic redundancy check, an error-checking protocol
DAC	digital-to-analog converter, see also IDAC, VDAC
DFB	digital filter block
DIO	digital input/output, GPIO with only digital capabilities, no analog. See GPIO.
DMA	direct memory access, see also TD
DNL	differential nonlinearity, see also INL
DNU	do not use
DR	port write data registers
DSI	digital system interconnect
DWT	data watchpoint and trace
ECO	external crystal oscillator
EEPROM	electrically erasable programmable read-only memory
EMI	electromagnetic interference
EOC	end of conversion
EOF	end of frame
EPSR	execution program status register
ESD	electrostatic discharge
FIR	finite impulse response, see also IIR
FPB	flash patch and breakpoint
FS	full-speed
GPIO	general-purpose input/output, applies to a PSoC pin

# Table 14-1. Acronyms Used in this Document (continued)

Acronym	Description
HVI	high-voltage interrupt, see also LVI, LVD
IC	integrated circuit
IDAC	current DAC, see also DAC, VDAC
IDE	integrated development environment
I <sup>2</sup> C, or IIC	Inter-Integrated Circuit, a communications protocol
lir	infinite impulse response, see also FIR
ILO	internal low-speed oscillator, see also IMO
IMO	internal main oscillator, see also ILO
INL	integral nonlinearity, see also DNL
I/O	input/output, see also GPIO, DIO, SIO, USBIO
IPOR	initial power-on reset
IPSR	interrupt program status register
IRQ	interrupt request
ITM	instrumentation trace macrocell
LCD	liquid crystal display
LIN	Local Interconnect Network, a communications protocol.
LR	link register
LUT	lookup table
LVD	low-voltage detect, see also LVI
LVI	low-voltage interrupt, see also HVI
LVTTL	low-voltage transistor-transistor logic
MAC	multiply-accumulate
MCU	microcontroller unit
MISO	master-in slave-out
NC	no connect
NMI	nonmaskable interrupt
NRZ	non-return-to-zero
NVIC	nested vectored interrupt controller
NVL	nonvolatile latch, see also WOL
opamp	operational amplifier
PAL	programmable array logic, see also PLD
PC	program counter
PCB	printed circuit board
PGA	programmable gain amplifier
PHUB	peripheral hub
PHY	physical layer
PICU	port interrupt control unit
PLA	programmable logic array