



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	67MHz
Connectivity	EBI/EMI, I ² C, LINbus, SPI, UART/USART, USB
Peripherals	CapSense, DMA, POR, PWM, WDT
Number of I/O	25
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.71V ~ 5.5V
Data Converters	A/D 16x20b; D/A 4x8b
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/cy8c3865pvi-053

1. Architectural Overview

Introducing the CY8C38 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 CY8C38 family provides configurable blocks of analog, digital, and interconnect circuitry around a CPU subsystem. The combination of a CPU with a flexible analog subsystem, digital subsystem, routing, and I/O enables a high level of integration in a wide variety of consumer, industrial, and medical applications.

Figure 1-1. Simplified Block Diagram

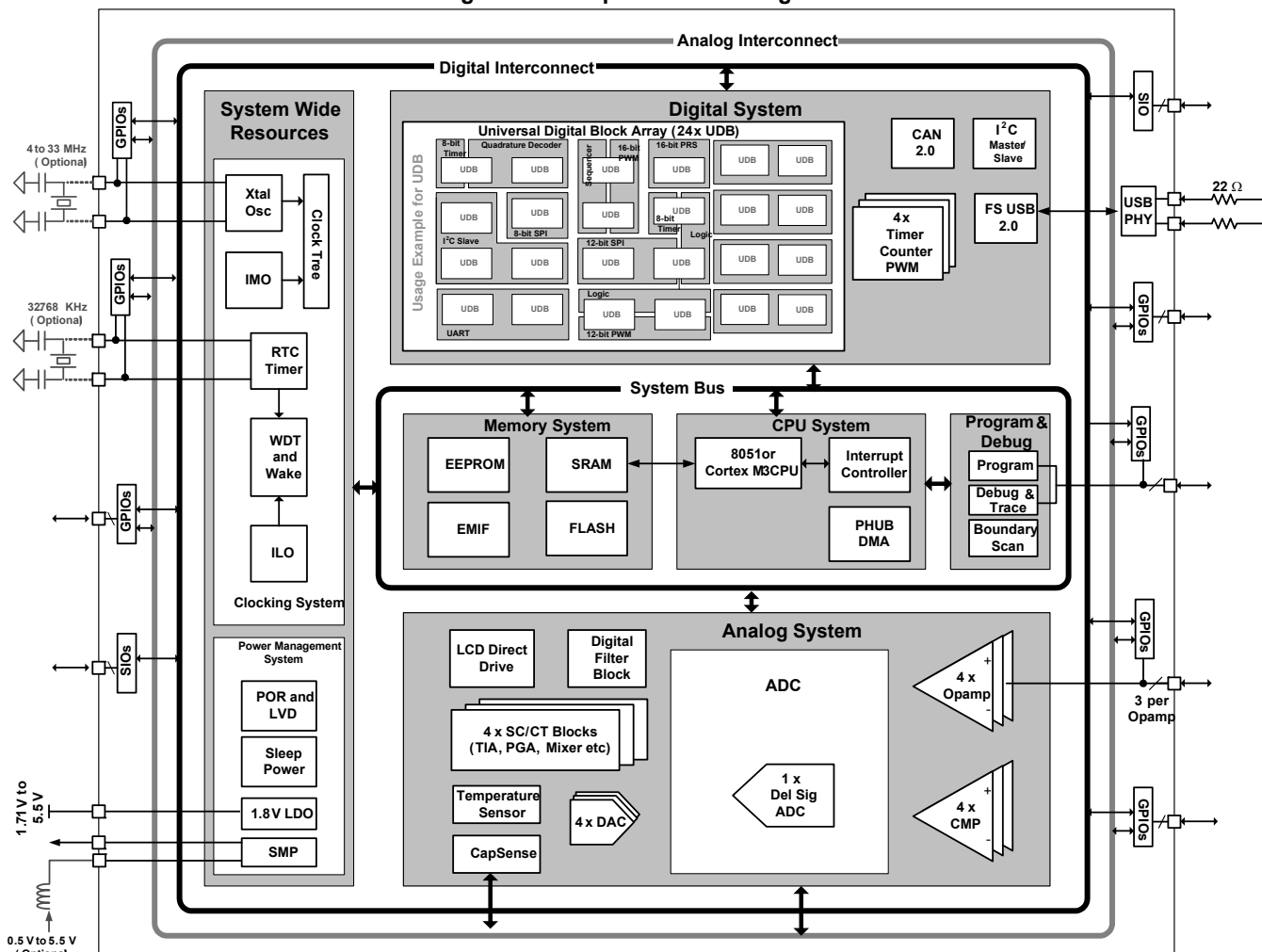


Figure 1-1 illustrates the major components of the CY8C38 family. They are:

- 8051 CPU subsystem
- Nonvolatile subsystem
- Programming, debug, and test subsystem
- Inputs and outputs
- Clocking
- Power
- Digital subsystem
- Analog subsystem

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

In addition to the flexibility of the UDB array, PSoC also provides configurable digital blocks targeted at specific functions. For the CY8C38 family these blocks can include four 16-bit timers, counters, and PWM blocks; I²C slave, master, and multimaster; FS USB; and Full CAN 2.0b.

4.3.1 Instruction Set Summary

4.3.1.1 Arithmetic Instructions

Arithmetic instructions support the direct, indirect, register, immediate constant, and register-specific instructions. Arithmetic modes are used for addition, subtraction, multiplication, division, increment, and decrement operations. [Table 4-1](#) lists the different arithmetic instructions.

Table 4-1. Arithmetic Instructions

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

4.3.1.2 Logical Instructions

The logical instructions perform Boolean operations such as AND, OR, XOR on bytes, rotate of accumulator contents, and swap of nibbles in an accumulator. The Boolean operations on the bytes are performed on the bit-by-bit basis. [Table 4-2](#) 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

Table 4-3. Data Transfer Instructions *(continued)*

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

Table 4-4. Boolean Instructions

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

4.3.1.5 Program Branching Instructions

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

Table 4-5. Jump Instructions

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

4.4 DMA and PHUB

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

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

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

4.4.1 PHUB Features

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

Table 4-6. PHUB Spokes and Peripherals

PHUB Spokes	Peripherals
0	SRAM
1	IOs, PICU, EMIF
2	PHUB local configuration, Power manager, Clocks, IC, SWV, EEPROM, Flash programming interface
3	Analog interface and trim, Decimator
4	USB, CAN, I ² C, Timers, Counters, and PWMs
5	DFB
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:

- Thirty-two 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 ² C	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	DFB Int	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]

6.2.1 Power Modes

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

PSoC 3 power modes, in order of decreasing power consumption are:

- Active
- Alternate Active
- Sleep
- Hibernate

Active is the main processing mode. Its functionality is configurable. Each power controllable subsystem is enabled or disabled by using separate power configuration template registers. In alternate active mode, fewer subsystems are enabled, reducing power. In sleep mode most resources are disabled regardless of the template settings. Sleep mode is optimized to provide timed sleep intervals and Real Time Clock functionality. The lowest power mode is hibernate, which retains register and SRAM state, but no clocks, and allows wakeup only from I/O pins. [Figure 6-5](#) on page 26 illustrates the allowable transitions between power modes

Table 6-2. Power Modes

Power Modes	Description	Entry Condition	Wakeup Source	Active Clocks	Regulator
Active	Primary mode of operation, all peripherals available (programmable)	Wakeup, reset, manual register entry	Any interrupt	Any (programmable)	All regulators available. Digital and analog regulators can be disabled if external regulation used.
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. Digital and analog regulators can be disabled if external regulation used.
Sleep	All subsystems automatically disabled	Manual register entry	Comparator, PICU, I ² C, RTC, CTW, LVD	ILO/kHzECO	Both digital and analog regulators buzzed. Digital and analog regulators can be disabled if external regulation used.
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	PICU		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	–	1.2 mA ^[13]	Yes	All	All	All	–	All
Alternate Active	–	–	User defined	All	All	All	–	All
Sleep	<15 µs	1 µA	No	I ² C	Comparator	ILO/kHzECO	Comparator, PICU, I ² C, RTC, CTW, LVD	XRES, LVD, WDR
Hibernate	<100 µs	200 nA	No	None	None	None	PICU	XRES

Note

13. Bus clock off. Execute from CPU instruction buffer at 6 MHz. See [Table 11-2](#) on page 60.

7.1 Example Peripherals

The flexibility of the CY8C38 family's 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 datasheet, and the list is always growing. An example of a component available for use in CY8C38 family, but, not explicitly called out in this datasheet 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 CY8C38 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
 - I²C
 - 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 CY8C38 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

7.1.3 Example System Function Components

The following is a sample of the system function components available in PSoC Creator for the CY8C38 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 the from the project options and rebuilding the application with no errors from the generated APIs or boot code.

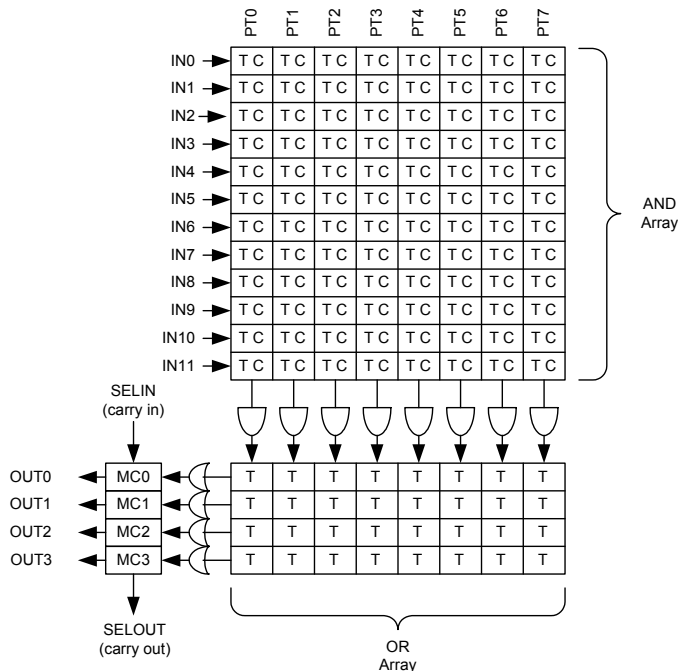
The main component blocks of the UDB are:

- **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.

7.2.1 PLD Module

The primary purpose of the PLD blocks is to implement logic expressions, state machines, sequencers, lookup tables, and decoders. In the simplest use model, consider the PLD blocks as a standalone resource onto which general purpose RTL is synthesized and mapped. The more common and efficient use model is to create digital functions from a combination of PLD and datapath blocks, where the PLD implements only the random logic and state portion of the function while the datapath (ALU) implements the more structured elements.

Figure 7-7. PLD 12C4 Structure

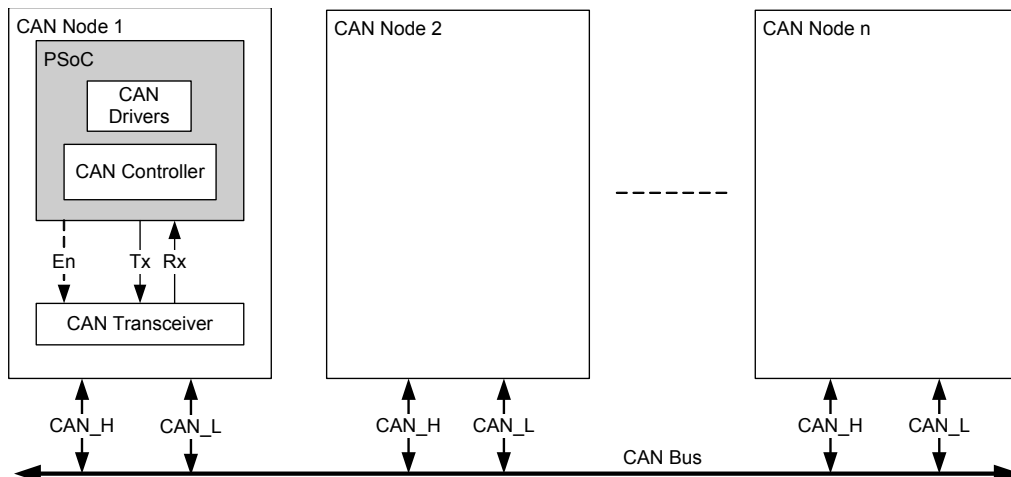


One 12C4 PLD block is shown in Figure 7-7. This PLD has 12 inputs, which feed across eight product terms. Each product term (AND function) can be from 1 to 12 inputs wide, and in a given product term, the true (T) or complement (C) of each input can be selected. The product terms are summed (OR function) to create the PLD outputs. A sum can be from 1 to 8 product terms wide. The 'C' in 12C4 indicates that the width of the OR gate (in this case 8) is constant across all outputs (rather than variable as in a 22V10 device). This PLA like structure gives maximum flexibility and insures that all inputs and outputs are permutable for ease of allocation by the software tools. There are two 12C4 PLDs in each UDB.

7.5 CAN

The CAN peripheral is a fully functional controller area network (CAN) supporting communication baud rates up to 1 Mbps. The CAN controller implements the CAN2.0A and CAN2.0B specifications as defined in the Bosch specification and conforms to the ISO-11898-1 standard. The CAN protocol was originally designed for automotive applications with a focus on a high level of fault detection. This ensures high communication reliability at a low cost. Because of its success in automotive applications, CAN is used as a standard communication protocol for motion oriented machine control networks (CANOpen) and factory automation applications (DeviceNet). The CAN controller features allow the efficient implementation of higher level protocols without affecting the performance of the microcontroller CPU. Full configuration support is provided in PSoC Creator.

Figure 7-18. CAN Bus System Implementation



7.5.1 CAN Features

- CAN2.0A/B protocol implementation – ISO 11898 compliant
 - Standard and extended frames with up to 8 bytes of data per frame
 - Message filter capabilities
 - Remote Transmission Request (RTR) support
 - Programmable bit rate up to 1 Mbps
- Listen Only mode
- SW readable error counter and indicator
- Sleep mode: Wake the device from sleep with activity on the Rx pin
- Supports two or three wire interface to external transceiver (Tx, Rx, and Enable). The three-wire interface is compatible with the Philips PHY; the PHY is not included on-chip. The three wires can be routed to any I/O
- Enhanced interrupt controller
 - CAN receive and transmit buffers status
 - CAN controller error status including BusOff

- Receive path
 - 16 receive buffers each with its own message filter
 - Enhanced hardware message filter implementation that covers the ID, IDE, and RTR
 - DeviceNet addressing support
 - Multiple receive buffers linkable to build a larger receive message array
 - Automatic transmission request (RTR) response handler
 - Lost received message notification
- Transmit path
 - Eight transmit buffers
 - Programmable transmit priority
 - Round robin
 - Fixed priority
 - Message transmissions abort capability

7.5.2 Software Tools Support

CAN Controller configuration integrated into PSoC Creator:

- CAN Configuration walkthrough with bit timing analyzer
- Receive filter setup

7.7 Timers, Counters, and PWMs

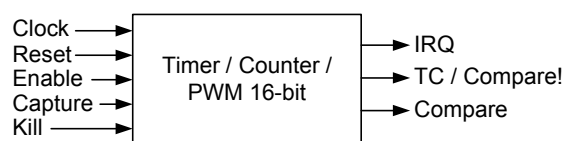
The timer/counter/PWM peripheral is a 16-bit dedicated peripheral providing three of the most common embedded peripheral features. As almost all embedded systems use some combination of timers, counters, and PWMs. Four of them have been included on this PSoC device family. Additional and more advanced functionality timers, counters, and PWMs can also be instantiated in UDBs as required. PSoC Creator allows you to choose the timer, counter, and PWM features that they require. The tool set utilizes the most optimal resources available.

The timer/counter/PWM peripheral can select from multiple clock sources, with input and output signals connected through the DSI routing. DSI routing allows input and output connections to any device pin and any internal digital signal accessible through the DSI. Each of the four instances has a compare output, terminal count output (optional complementary compare output), and programmable interrupt request line. The Timer/Counter/PWMs are configurable as free running, one shot, or Enable input controlled. The peripheral has timer reset and capture inputs, and a kill input for control of the comparator outputs. The peripheral supports full 16-bit capture.

Timer/Counter/PWM features include:

- 16-bit Timer/Counter/PWM (down count only)
- Selectable clock source
- PWM comparator (configurable for LT, LTE, EQ, GTE, GT)
- Period reload on start, reset, and terminal count
- Interrupt on terminal count, compare true, or capture
- Dynamic counter reads
- Timer capture mode
- Count while enable signal is asserted mode
- Free run mode
- One Shot mode (stop at end of period)
- Complementary PWM outputs with deadband
- PWM output kill

Figure 7-21. Timer/Counter/PWM



7.8 I²C

The I²C peripheral provides a synchronous two wire interface designed to interface the PSoC device with a two wire I²C serial communication bus. The bus is compliant with Philips 'The I²C Specification' version 2.1. Additional I²C interfaces can be instantiated using Universal Digital Blocks (UDBs) in PSoC Creator, as required.

To eliminate the need for excessive CPU intervention and overhead, I²C specific support is provided for status detection and generation of framing bits. I²C operates as a slave, a master, or multimaster (Slave and Master). In slave mode, the unit always listens for a start condition to begin sending or receiving data. Master mode supplies the ability to generate the Start and Stop conditions and initiate transactions. Multimaster mode provides clock synchronization and arbitration to allow multiple masters on the same bus. If Master mode is enabled and Slave mode is not enabled, the block does not generate interrupts on externally generated Start conditions. I²C interfaces through the DSI routing and allows direct connections to any GPIO or SIO pins.

I²C provides hardware address detect of a 7-bit address without CPU intervention. Additionally the device can wake from low-power modes on a 7-bit hardware address match. If wakeup functionality is required, I²C pin connections are limited to the two special sets of SIO pins.

I²C features include:

- Slave and master, transmitter, and receiver operation
- Byte processing for low CPU overhead
- Interrupt or polling CPU interface
- Support for bus speeds up to 1 Mbps (3.4 Mbps in UDBs)
- 7 or 10-bit addressing (10-bit addressing requires firmware support)
- SMBus operation (through firmware support - SMBus supported in hardware in UDBs)
- 7-bit hardware address compare
- Wake from low-power modes on address match

7.9 Digital Filter Block

Some devices in the CY8C38 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 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.

Table 11-2. DC Specifications (continued)

Parameter	Description	Conditions		Min	Typ	Max	Units
	Sleep Mode ^[22]						
	CPU = OFF RTC = ON (= ECO32K ON, in low-power mode) Sleep timer = ON (= ILO ON at 1 kHz) ^[23] WDT = OFF I ² C Wake = OFF Comparator = OFF POR = ON Boost = OFF SIO pins in single ended input, unregulated output mode	V _{DD} = V _{DDIO} = 4.5–5.5 V	T = –40 °C	–	–	–	μA
			T = 25 °C	–	–	–	μA
			T = 85 °C	–	–	–	μA
		V _{DD} = V _{DDIO} = 2.7–3.6 V	T = –40 °C	–	–	–	μA
			T = 25 °C	–	1	–	μA
			T = 85 °C	–	–	–	μA
		V _{DD} = V _{DDIO} = 1.71–1.95 V	T = –40 °C	–	–	–	μA
			T = 25 °C	–	–	–	μA
			T = 85 °C	–	–	–	μA
	Comparator = ON CPU = OFF RTC = OFF Sleep timer = OFF WDT = OFF I2C Wake = OFF POR = ON Boost = OFF SIO pins in single ended input, unregulated output mode	V _{DD} = V _{DDIO} = 2.7–3.6V	T = 25 °C	–	–	–	μA
	I2C Wake = ON CPU = OFF RTC = OFF Sleep timer = OFF WDT = OFF Comparator = OFF POR = ON Boost = OFF SIO pins in single ended input, unregulated output mode	V _{DD} = V _{DDIO} = 2.7–3.6V	T = 25 °C	–	–	–	μA
	Hibernate Mode ^[22]						
	Hibernate mode current All regulators and oscillators off. SRAM retention GPIO interrupts are active Boost = OFF SIO pins in single ended input, unregulated output mode	V _{DD} = V _{DDIO} = 4.5–5.5 V	T = –40 °C	–	–	–	nA
			T = 25 °C	–	–	–	nA
			T = 85 °C	–	–	–	nA
		V _{DD} = V _{DDIO} = 2.7–3.6 V	T = –40 °C	–	–	–	nA
			T = 25 °C	–	200	–	nA
			T = 85 °C	–	–	–	nA
		V _{DD} = V _{DDIO} = 1.71–1.95 V	T = –40 °C	–	–	–	nA
T = 25 °C			–	–	–	nA	
T = 85 °C			–	–	–	nA	

Notes

 22. If V_{CCD} and V_{CCA} are externally regulated, the voltage difference between V_{CCD} and V_{CCA} must be less than 50 mV.

23. Sleep timer generates periodic interrupts to wake up the CPU. This specification applies only to those times that the CPU is off.

11.4 Inputs and Outputs

Specifications are valid for $-40\text{ }^{\circ}\text{C} \leq T_A \leq 85\text{ }^{\circ}\text{C}$ and $T_J \leq 100\text{ }^{\circ}\text{C}$, except where noted. Specifications are valid for 1.71 V to 5.5 V, except where noted.

11.4.1 GPIO

Table 11-9. GPIO DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
V_{IH}	Input voltage high threshold	CMOS Input, PRT[x]CTL = 0	$0.7 \times V_{DDIO}$	—	—	V
V_{IL}	Input voltage low threshold	CMOS Input, PRT[x]CTL = 0	—	—	$0.3 \times V_{DDIO}$	V
V_{IH}	Input voltage high threshold	LVTTL Input, PRT[x]CTL = 1, $V_{DDIO} < 2.7\text{ V}$	$0.7 \times V_{DDIO}$	—	—	V
V_{IH}	Input voltage high threshold	LVTTL Input, PRT[x]CTL = 1, $V_{DDIO} \geq 2.7\text{ V}$	2.0	—	—	V
V_{IL}	Input voltage low threshold	LVTTL Input, PRT[x]CTL = 1, $V_{DDIO} < 2.7\text{ V}$	—	—	$0.3 \times V_{DDIO}$	V
V_{IL}	Input voltage low threshold	LVTTL Input, PRT[x]CTL = 1, $V_{DDIO} \geq 2.7\text{ V}$	—	—	0.8	V
V_{OH}	Output voltage high	$I_{OH} = 4\text{ mA}$ at 3.3 V_{DDIO}	$V_{DDIO} - 0.6$	—	—	V
		$I_{OH} = 1\text{ mA}$ at 1.8 V_{DDIO}	$V_{DDIO} - 0.5$	—	—	V
V_{OL}	Output voltage low	$I_{OL} = 8\text{ mA}$ at 3.3 V_{DDIO}	—	—	0.6	V
		$I_{OL} = 4\text{ mA}$ at 1.8 V_{DDIO}	—	—	0.6	V
Rpullup	Pull-up resistor		3.5	5.6	8.5	k Ω
Rpulldown	Pull-down resistor		3.5	5.6	8.5	k Ω
I_{IL}	Input leakage current (absolute value) ^[29]	25 $^{\circ}\text{C}$, $V_{DDIO} = 3.0\text{ V}$	—	—	2	nA
C_{IN}	Input capacitance ^[29]	GPIOs without opamp outputs	—	—	7	pF
		GPIOs with opamp outputs	—	—	18	pF
V_H	Input voltage hysteresis (Schmitt-Trigger) ^[29]		—	40	—	mV
Idiode	Current through protection diode to V_{DDIO} and V_{SSIO}		—	—	100	μA
Rglobal	Resistance pin to analog global bus	25 $^{\circ}\text{C}$, $V_{DDIO} = 3.0\text{ V}$	—	320	—	Ω
Rmux	Resistance pin to analog mux bus	25 $^{\circ}\text{C}$, $V_{DDIO} = 3.0\text{ V}$	—	220	—	Ω

Table 11-10. GPIO AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
TriseF	Rise time in Fast Strong Mode ^[29]	3.3 V V_{DDIO} Cload = 25 pF	—	—	12	ns
TfallF	Fall time in Fast Strong Mode ^[29]	3.3 V V_{DDIO} Cload = 25 pF	—	—	12	ns
TriseS	Rise time in Slow Strong Mode ^[29]	3.3 V V_{DDIO} Cload = 25 pF	—	—	60	ns
TfallS	Fall time in Slow Strong Mode ^[29]	3.3 V V_{DDIO} Cload = 25 pF	—	—	60	ns
Fgpioout	GPIO output operating frequency					
	3.3 V $\leq V_{DDIO} \leq 5.5\text{ V}$, fast strong drive mode	90/10% V_{DDIO} into 25 pF	—	—	33	MHz
	1.71 V $\leq V_{DDIO} < 3.3\text{ V}$, fast strong drive mode	90/10% V_{DDIO} into 25 pF	—	—	20	MHz
	3.3 V $\leq V_{DDIO} \leq 5.5\text{ V}$, slow strong drive mode	90/10% V_{DDIO} into 25 pF	—	—	7	MHz
	1.71 V $\leq V_{DDIO} < 3.3\text{ V}$, slow strong drive mode	90/10% V_{DDIO} into 25 pF	—	—	3.5	MHz
Fgpioin	GPIO input operating frequency					
	1.71 V $\leq V_{DDIO} \leq 5.5\text{ V}$	90/10% V_{DDIO}	—	—	66	MHz

Note

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

Table 11-14. USBIO AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
Tdrate	Full-speed data rate average bit rate		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	$3\text{ V} \leq V_{\text{DDD}} \leq 5.5\text{ V}$	–	–	20	MHz
		$V_{\text{DDD}} = 1.71\text{ V}$	–	–	6	MHz
Tr_gpio	Rise time, GPIO mode, 10%/90% V_{DDD}	$V_{\text{DDD}} > 3\text{ V}$, 25 pF load	–	–	12	ns
		$V_{\text{DDD}} = 1.71\text{ V}$, 25 pF load	–	–	40	ns
Tf_gpio	Fall time, GPIO mode, 90%/10% V_{DDD}	$V_{\text{DDD}} > 3\text{ V}$, 25 pF load	–	–	12	ns
		$V_{\text{DDD}} = 1.71\text{ V}$, 25 pF load	–	–	40	ns

Table 11-15. USB Driver AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
Tr	Transition rise time		–	–	20	ns
Tf	Transition fall time		–	–	20	ns
TR	Rise/fall time matching	V_{USB_5} , $V_{\text{USB}_3.3}$, see USB DC Specifications on page 93	90%	–	111%	
Vcrs	Output signal crossover voltage		1.3	–	2	V

11.4.4 XRES

Table 11-16. XRES DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
V_{IH}	Input voltage high threshold		$0.7 \times V_{\text{DDIO}}$	–	–	V
V_{IL}	Input voltage low threshold		–	–	$0.3 \times V_{\text{DDIO}}$	V
Rpullup	Pull-up resistor		3.5	5.6	8.5	k Ω
C_{IN}	Input capacitance ^[32]		–	3	–	pF
V_{H}	Input voltage hysteresis (Schmitt-Trigger) ^[32]		–	100	–	mV
Idiode	Current through protection diode to V_{DDIO} and V_{SSIO}		–	–	100	μA

Table 11-17. XRES AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
T _{RESET}	Reset pulse width		1	–	–	μs

Note

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

Figure 11-12. Opamp Closed Loop Frequency Response, Gain = 10, Vdda = 5V

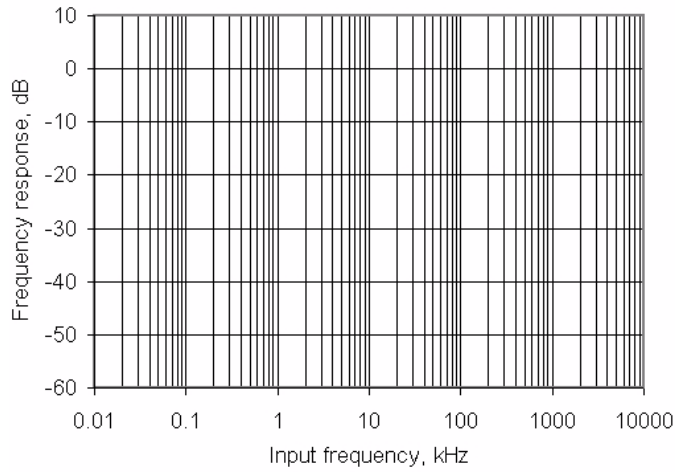


Figure 11-13. Opamp test Circuit for Gain = 10

Figure 11-14. Opamp Noise vs Frequency, Power Mode = High, Vdda = 5V

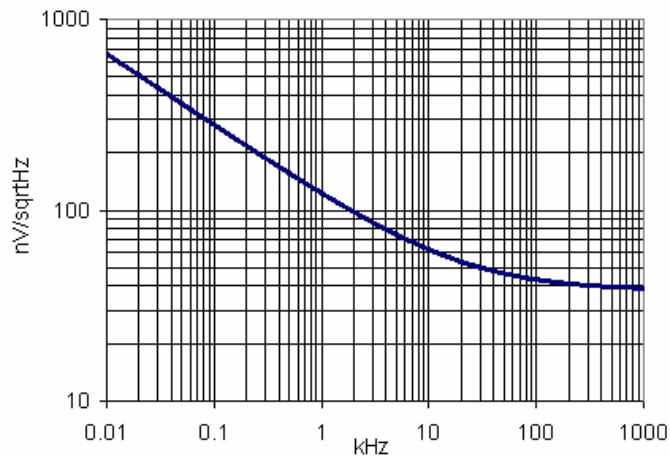


Figure 11-15. Opamp CMRR vs Frequency

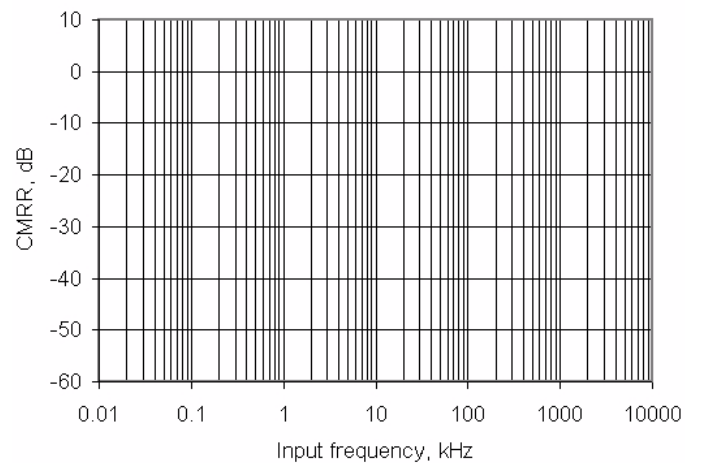


Table 11-24. Delta-sigma ADC RMS Noise vs. Input Range and Sample Rate, 16-bit, Internal Reference

RMS Noise, Counts	Input Voltage Range								
	Single-Ended				Differential				
Sample rate, sps	0 to V_{REF}	0 to $V_{REF} \times 2$	V_{SSA} to V_{DDA}	0 to $V_{REF} \times 6$	$\pm V_{REF}$	$\pm V_{REF}/2$	$\pm V_{REF}/4$	$\pm V_{REF}/8$	$\pm V_{REF}/16$
750									
1500									
3000									
6000									
12000									
24000									
48000									

Figure 11-23. Delta-sigma ADC DNL vs Output Code, 16-bit, 48 ksps, 25 °C $V_{DDA} = 3.3$ V

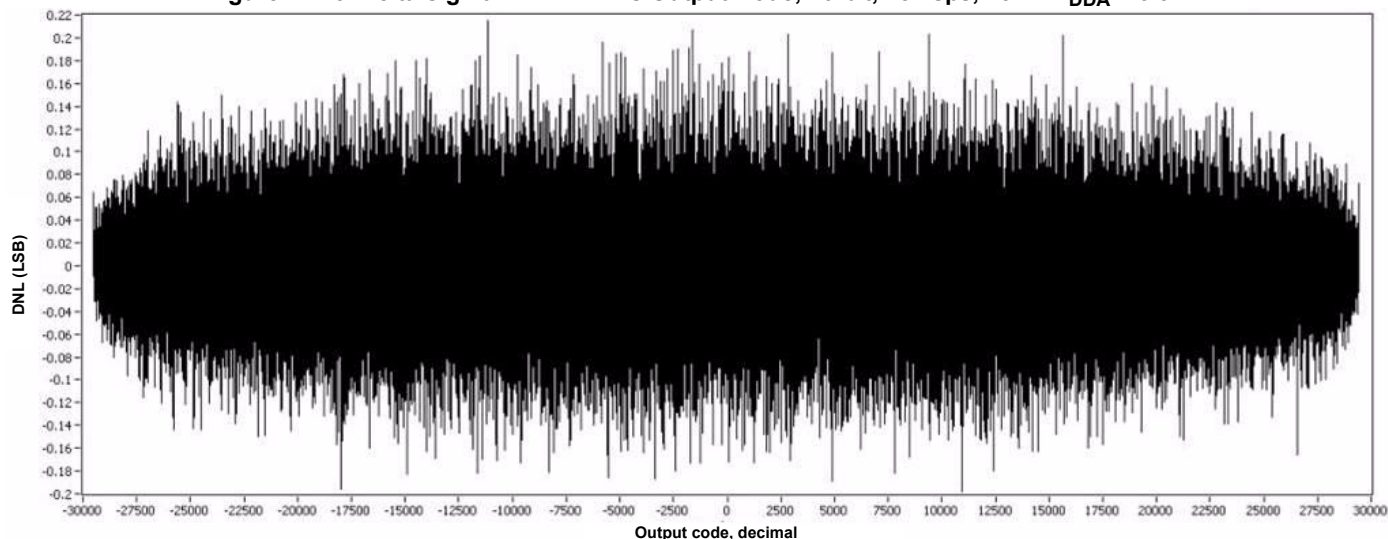
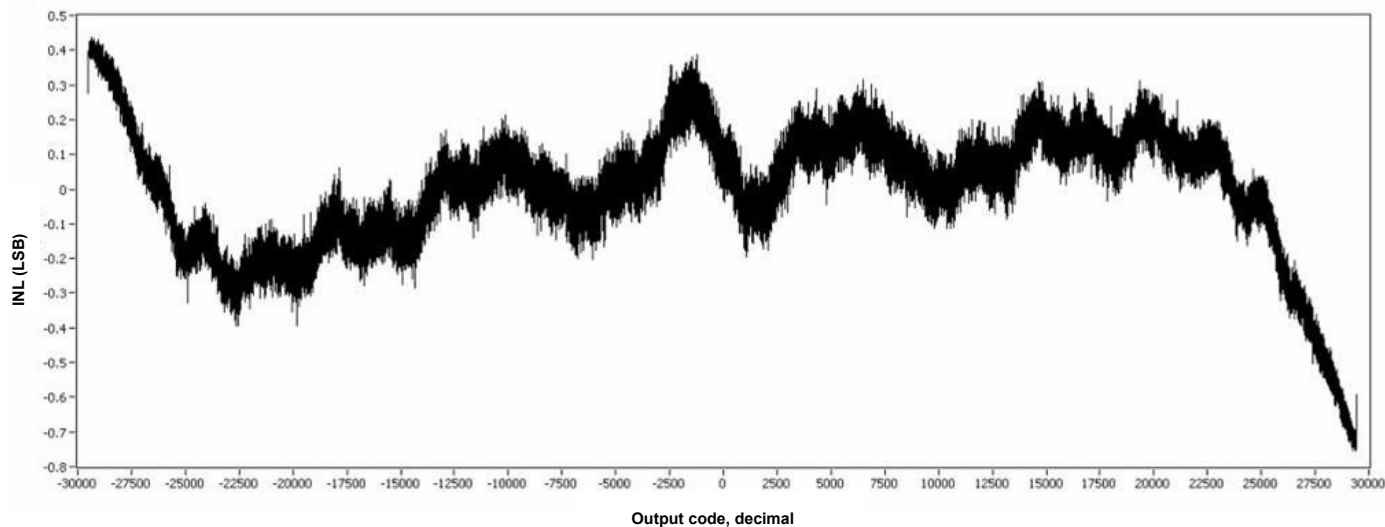


Figure 11-24. Delta-sigma ADC INL vs Output Code, 16-bit, 48 ksps, 25 °C $V_{DDA} = 3.3$ V



11.5.10 Programmable Gain Amplifier

The PGA is created using a SC/CT analog block; see the PGA component datasheet in PSoC Creator for full electrical specifications and APIs.

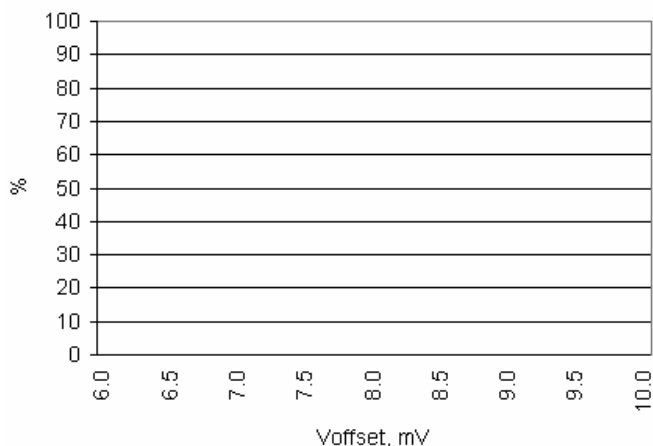
Unless otherwise specified, operating conditions are:

- Operating temperature = 25 °C for typical values
- Unless otherwise specified, all charts and graphs show typical values

Table 11-37. PGA DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
V _{in}	Input voltage range	Power mode = minimum	V _{ssa}	–	V _{dda}	V
V _{os}	Input offset voltage	Power mode = high, gain = 1	–	–	10	mV
TCV _{os}	Input offset voltage drift with temperature	Power mode = high, gain = 1	–	–	±30	µV/°C
Ge ₁	Gain error, gain = 1		–	–	±0.15	%
Ge ₁₆	Gain error, gain = 16		–	–	±2.5	%
Ge ₅₀	Gain error, gain = 50		–	–	±5	%
V _{onl}	DC output nonlinearity	Gain = 1	–	–	±0.01	% of FSR
C _{in}	Input capacitance		–	–	7	pF
V _{oh}	Output voltage swing	Power mode = high, gain = 1, R _{load} = 100 kΩ to V _{DDA} / 2	V _{DDA} – 0.15	–	–	V
V _{ol}	Output voltage swing	Power mode = high, gain = 1, R _{load} = 100 kΩ to V _{DDA} / 2	–	–	V _{SSA} + 0.15	V
V _{src}	Output voltage under load	I _{load} = 250 µA, V _{dda} ≥ 2.7V, power mode = high	–	–	300	mV
I _{dd}	Operating current	Power mode = high	–	1.5	1.65	mA
PSRR	Power supply rejection ratio		48	–	–	dB

Figure 11-43. V_{offset} Histogram, 1000 Samples, V_{dda} = 5 V



11.7 Memory

Specifications are valid for $-40\text{ }^{\circ}\text{C} \leq T_A \leq 85\text{ }^{\circ}\text{C}$ and $T_J \leq 100\text{ }^{\circ}\text{C}$, except where noted. Specifications are valid for 1.71 V to 5.5 V, except where noted.

11.7.1 Flash

Table 11-55. Flash DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	Erase and program voltage	V_{DD} pin	1.71	–	5.5	V

Table 11-56. Flash AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
T_{WRITE}	Row write time (erase + program)		–	–	15	ms
T_{ERASE}	Row erase time		–	–	10	ms
	Row program time		–	–	5	ms
T_{BULK}	Bulk erase time (16 KB to 64 KB)		–	–	35	ms
	Sector erase time (8 KB to 16 KB)		–	–	15	ms
	Total device program time (including JTAG, and so on)		–	–	5	seconds
	Flash endurance		100 k	–	–	program/erase cycles
	Flash data retention time, retention period measured from last erase cycle	Average ambient temp. $T_A \leq 55\text{ }^{\circ}\text{C}$, 100 K erase/program cycles	20	–	–	years

11.7.2 EEPROM

Table 11-57. EEPROM DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	Erase and program voltage		1.71	–	5.5	V

Table 11-58. EEPROM AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
T_{WRITE}	Single row erase/write cycle time		–	2	15	ms
	EEPROM data retention time, retention period measured from last erase cycle (up to 100 K cycles)	Average ambient temp, $T_A \leq 25\text{ }^{\circ}\text{C}$, 1M erase/program cycles	20	–	–	years
		Average ambient temp, $T_A \leq 55\text{ }^{\circ}\text{C}$, 100 K erase/program cycles	20	–	–	

11.7.3 Nonvolatile Latches (NVL)

Table 11-59. NVL DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	Erase and program voltage	V_{DD} pin	1.71	–	5.5	V

Table 11-60. NVL AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	NVL endurance	Programmed at $25\text{ }^{\circ}\text{C}$	1K	–	–	program/erase cycles
		Programmed at $0\text{ }^{\circ}\text{C}$ to $70\text{ }^{\circ}\text{C}$	100	–	–	program/erase cycles
	NVL data retention time	Programmed at $25\text{ }^{\circ}\text{C}$	20	–	–	years
		Programmed at $0\text{ }^{\circ}\text{C}$ to $70\text{ }^{\circ}\text{C}$	20	–	–	years

Figure 11-51. Synchronous Write Cycle Timing

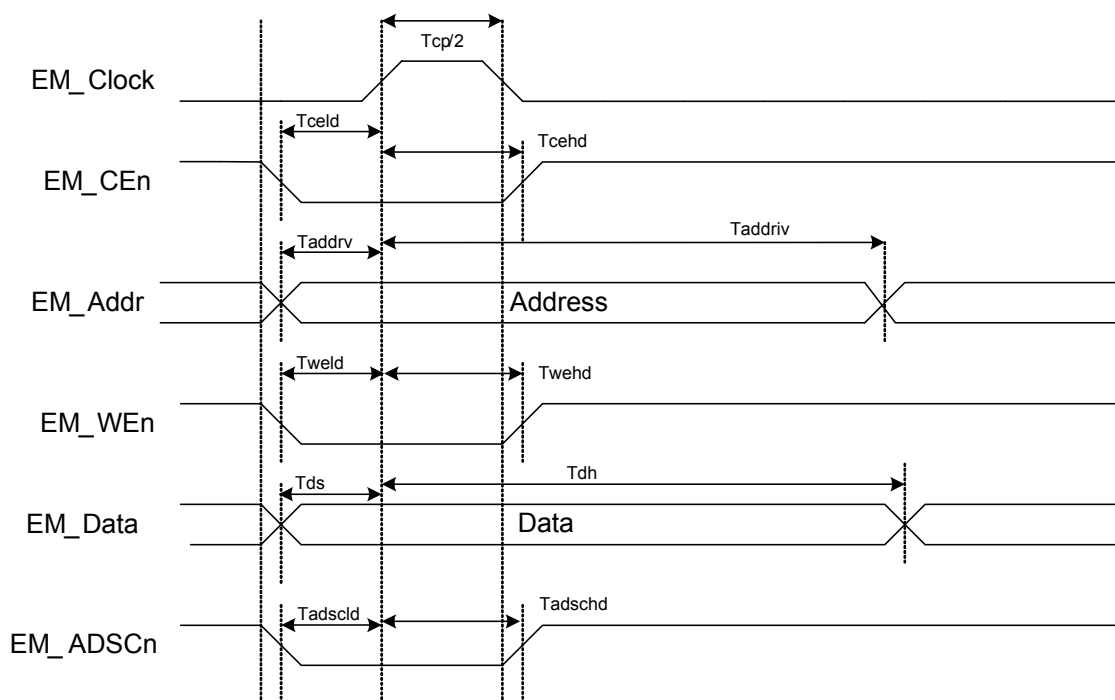


Table 11-66. Synchronous Write Cycle Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
T	EMIF clock Period ^[45]	Vdda ≥ 3.3 V	30.3	–	–	nS
Tcp/2	EM_Clock pulse high		T/2	–	–	nS
Tceld	EM_CEn low to EM_Clock high		5	–	–	nS
Tcehd	EM_Clock high to EM_CEn high		T/2 – 5	–	–	nS
Taddrv	EM_Addr valid to EM_Clock high		5	–	–	nS
Taddriv	EM_Clock high to EM_Addr invalid		T/2 – 5	–	–	nS
Tweld	EM_WEn low to EM_Clock high		5	–	–	nS
Twehd	EM_Clock high to EM_WEn high		T/2 – 5	–	–	nS
Tds	Data valid before EM_Clock high		5	–	–	nS
Tdh	Data invalid after EM_Clock high		T	–	–	nS
Tadscl	EM_ADSCn low to EM_Clock high		5	–	–	nS
Tadschd	EM_Clock high to EM_ADSCn high		T/2 – 5	–	–	nS

Note

45. Limited by GPIO output frequency, see [Table 11-10](#) on page 65.

Table 11-78. IMO AC Specifications (continued)

Parameter	Description	Conditions	Min	Typ	Max	Units
Jp-p	Jitter (peak to peak) ^[50]					
	F = 24 MHz		–	0.9	–	ns
	F = 3 MHz		–	1.6	–	ns
Jperiod	Jitter (long term) ^[50]					
	F = 24 MHz		–	0.9	–	ns
	F = 3 MHz		–	12	–	ns

11.9.3 Internal Low-Speed Oscillator

Table 11-79. ILO DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
I _{CC}	Operating current	F _{OUT} = 1 kHz	–	0.3	1.7	μA
		F _{OUT} = 33 kHz	–	1.0	2.6	μA
		F _{OUT} = 100 kHz	–	1.0	2.6	μA
	Leakage current	Power down mode	–	2.0	15	nA

Table 11-80. ILO AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
	Startup time, all frequencies	Turbo mode	–	–	2	ms
F _{ILO}	ILO frequencies (trimmed)					
	100 kHz		45	100	200	kHz
	1 kHz		0.5	1	2	kHz
	ILO frequencies (untrimmed)					
	100 kHz		30	100	300	kHz
	1 kHz		0.3	1	3.5	kHz

11.9.4 External Crystal Oscillator

Table 11-81. ECO AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
F	Crystal frequency range		4	–	33	MHz

11.9.5 External Clock Reference

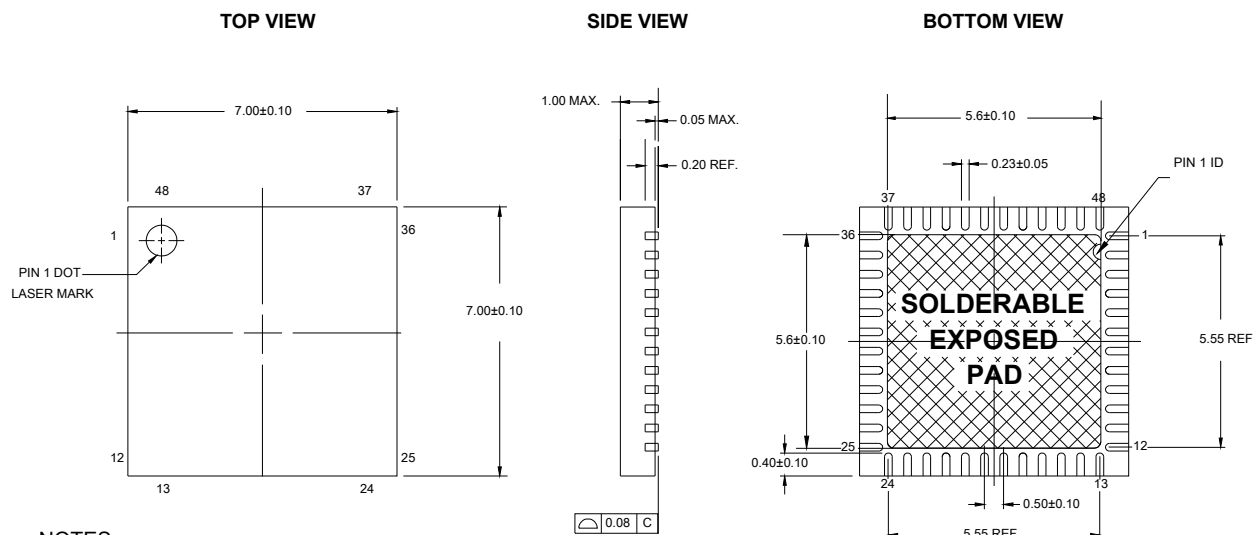
Table 11-82. External Clock Reference AC Specifications^[50]

Parameter	Description	Conditions	Min	Typ	Max	Units
	External frequency range		0	–	33	MHz
	Input duty cycle range	Measured at V _{DDIO} /2	30	50	70	%
	Input edge rate	V _{IL} to V _{IH}	0.1	–	–	V/ns


Note
Note

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

Figure 13-2. 48-pin QFN Package Outline



NOTES:

- 1.  HATCH AREA IS SOLDERABLE EXPOSED METAL.
- 2. REFERENCE JEDEC#: MO-220
- 3. PACKAGE WEIGHT: 0.13g
- 4. ALL DIMENSIONS ARE IN MM [MIN/MAX]
- 5. PACKAGE CODE

PART #	DESCRIPTION
LT48D	LEAD FREE

001- 45616 *B