



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
Peripherals	CapSense, DMA, LCD, POR, PWM, WDT
Number of I/O	38
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	68-VFQFN Exposed Pad
Supplier Device Package	68-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy8c3865lti-024

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

4.4.2 DMA Features

- 24 DMA channels
- Each channel has one or more transaction descriptors (TD) to configure channel behavior. Up to 128 total TDs can be defined
- TDs can be dynamically updated
- Eight levels of priority per channel
- Any digitally routable signal, the CPU, or another DMA channel, can trigger a transaction
- Each channel can generate up to two interrupts per transfer
- Transactions can be stalled or canceled
- Supports transaction size of infinite or 1 to 64 KB
- TDs may be nested and/or chained for complex transactions

4.4.3 Priority Levels

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

Table 4-7. Priority Levels

Priority Level	% Bus Bandwidth
0	100.0
1	100.0
2	50.0
3	25.0
4	12.5
5	6.2
6	3.1
7	1.5

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

4.4.4 Transaction Modes Supported

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

4.4.4.1 Simple DMA

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

4.4.4.2 Auto Repeat DMA

Auto repeat DMA is typically used when a static pattern is repetitively read from system memory and written to a peripheral. This is done with a single TD that chains to itself.

4.4.4.3 Ping Pong DMA

A ping pong DMA case uses double buffering to allow one buffer to be filled by one client while another client is consuming the data previously received in the other buffer. In its simplest form, this is done by chaining two TDs together so that each TD calls the opposite TD when complete.

4.4.4.4 Circular DMA

Circular DMA is similar to ping pong DMA except it contains more than two buffers. In this case there are multiple TDs; after the last TD is complete it chains back to the first TD.

4.4.4.5 Scatter Gather DMA

In the case of scatter gather DMA, there are multiple noncontiguous sources or destinations that are required to effectively carry out an overall DMA transaction. For example, a packet may need to be transmitted off of the device and the packet elements, including the header, payload, and trailer, exist in various noncontiguous locations in memory. Scatter gather DMA allows the segments to be concatenated together by using multiple TDs in a chain. The chain gathers the data from the multiple locations. A similar concept applies for the reception of data onto the device. Certain parts of the received data may need to be scattered to various locations in memory for software processing convenience. Each TD in the chain specifies the location for each discrete element in the chain.

4.4.4.6 Packet Queuing DMA

Packet queuing DMA is similar to scatter gather DMA but specifically refers to packet protocols. With these protocols, there may be separate configuration, data, and status phases associated with sending or receiving a packet.

For instance, to transmit a packet, a memory mapped configuration register can be written inside a peripheral, specifying the overall length of the ensuing data phase. The CPU can set up this configuration information anywhere in system memory and copy it with a simple TD to the peripheral. After the configuration phase, a data phase TD (or a series of data phase TDs) can begin (potentially using scatter gather). When the data phase TD(s) finish, a status phase TD can be invoked that reads some memory mapped status information from the peripheral and copies it to a location in system memory specified by the CPU for later inspection. Multiple sets of configuration, data, and status phase 'subchains' can be strung together to create larger chains that transmit multiple packets in this way. A similar concept exists in the opposite direction to receive the packets.

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

5.6.3 SFRs

The SFR space provides access to frequently accessed registers. The memory map for the SFR memory space is shown in [Table 5-2](#).

Table 5-2. SFR Map

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

The CY8C38 family provides the standard set of registers found on industry standard 8051 devices. In addition, the CY8C38 devices add SFRs to provide direct access to the I/O ports on the device. The following sections describe the SFRs added to the CY8C38 family.

XData Space Access SFRs

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

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

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

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

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

I/O Port SFRs

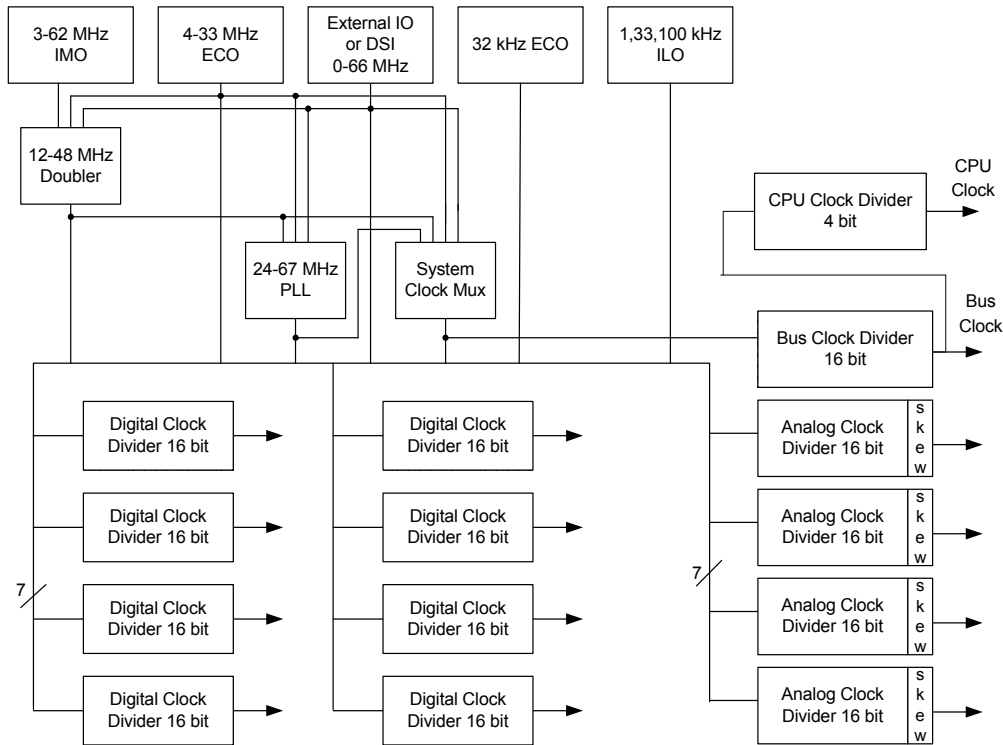
The I/O ports provide digital input sensing, output drive, pin interrupts, connectivity for analog inputs and outputs, LCD, and access to peripherals through the DSI. Full information on I/O ports is found in [I/O System and Routing](#) on page 28.

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

Each SFR supported I/O port provides three SFRs:

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

Figure 6-1. Clocking Subsystem



6.1.1 Internal Oscillators

6.1.1.1 Internal Main Oscillator

In most designs the IMO is the only clock source required, due to its ± 1 -percent accuracy. The IMO operates with no external components and outputs a stable clock. A factory trim for each frequency range is stored in the device. With the factory trim, tolerance varies from ± 1 percent at 3 MHz, up to ± 7 percent at 62 MHz. The IMO, in conjunction with the PLL, allows generation of CPU and system clocks up to the device's maximum frequency (see PLL). The IMO provides clock outputs at 3, 6, 12, 24, 48, and 62 MHz.

6.1.1.2 Clock Doubler

The clock doubler outputs a clock at twice the frequency of the input clock. The doubler works for input frequency ranges of 6 to 24 MHz (providing 12 to 48 MHz at the output). It can be configured to use a clock from the IMO, MHzECO, or the DSI (external pin). The doubler is typically used to clock the USB.

6.1.1.3 PLL

The PLL allows low-frequency, high-accuracy clocks to be multiplied to higher frequencies. This is a trade off between higher clock frequency and accuracy and, higher power consumption and increased startup time.

The PLL block provides a mechanism for generating clock frequencies based upon a variety of input sources. The PLL outputs clock frequencies in the range of 24 to 67 MHz. Its input and feedback dividers supply 4032 discrete ratios to create almost any desired system clock frequency. The accuracy of the PLL output depends on the accuracy of the PLL input source. The most common PLL use is to multiply the IMO clock at 3 MHz, where it is most accurate, to generate the CPU and system clocks up to the device's maximum frequency.

The PLL achieves phase lock within 250 μ s (verified by bit setting). It can be configured to use a clock from the IMO, MHzECO or DSI (external pin). The PLL clock source can be used until lock is complete and signaled with a lock bit. The lock signal can be routed through the DSI to generate an interrupt. Disable the PLL before entering low-power modes.

6.1.1.4 Internal Low-Speed Oscillator

The ILO provides clock frequencies for low-power consumption, including the watchdog timer, and sleep timer. The ILO generates up to three different clocks: 1 kHz, 33 kHz, and 100 kHz. The 1-kHz clock (CLK1K) is typically used for a background 'heartbeat' timer. This clock inherently lends itself to low-power supervisory operations such as the watchdog timer and long sleep intervals using the central timewheel (CTW).

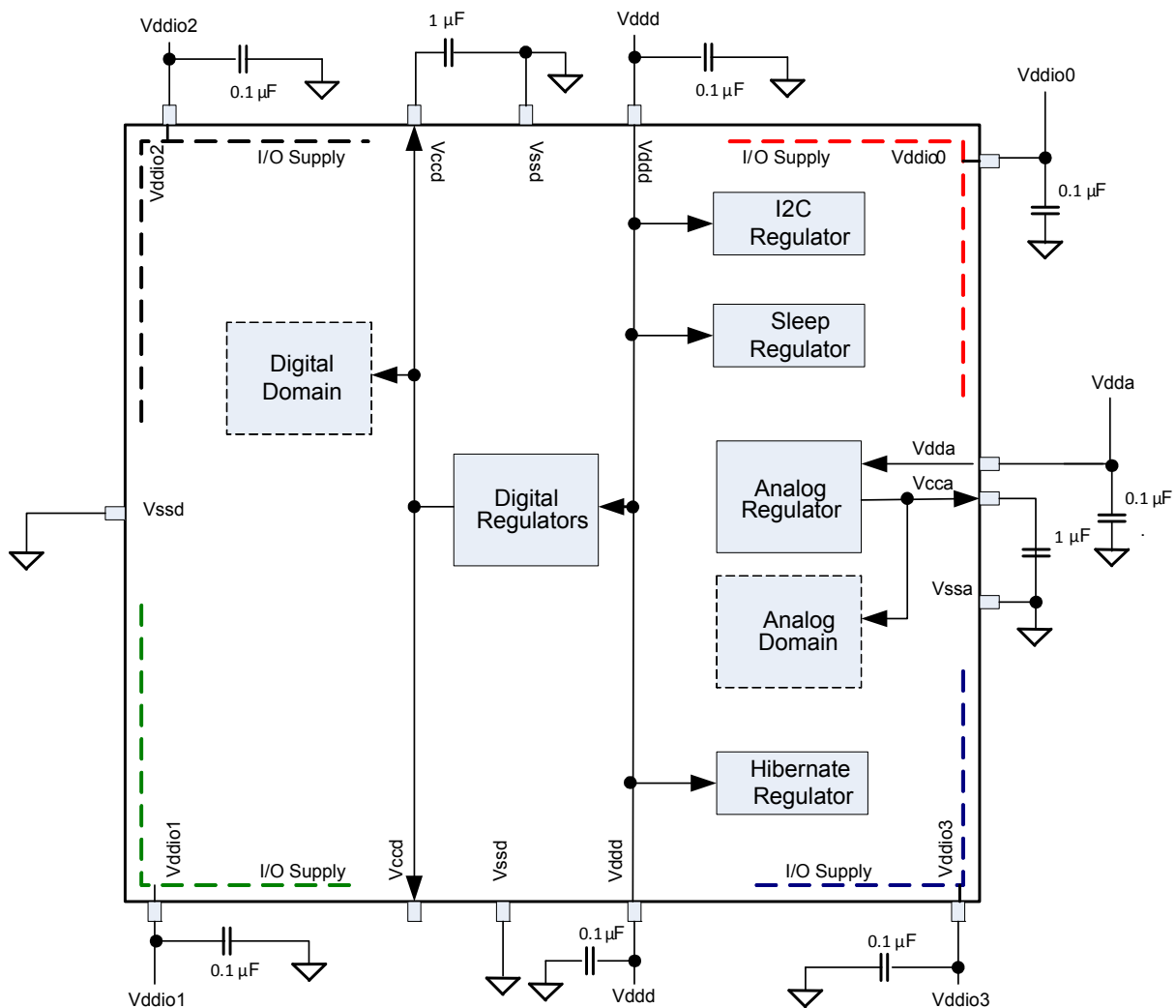
The central timewheel is a 1-kHz, free running, 13-bit counter clocked by the ILO. The central timewheel is always enabled, except in hibernate mode and when the CPU is stopped during debug on chip mode. It can be used to generate periodic interrupts for timing purposes or to wake the system from a low-power mode. Firmware can reset the central timewheel. Systems that require accurate timing should use the RTC capability instead of the central timewheel.

The 100-kHz clock (CLK100K) works as a low-power system clock to run the CPU. It can also generate time intervals such as fast sleep intervals using the fast timewheel. The fast timewheel is a 100-kHz, 5-bit counter clocked by the ILO that can also be used to wake the system. The fast timewheel settings are programmable, and the counter automatically resets when the terminal count is reached. This enables flexible, periodic wakeups of the CPU at a higher rate than is allowed using the central timewheel. The fast timewheel can generate an optional interrupt each time the terminal count is reached.

6.2 Power System

The power system consists of separate analog, digital, and I/O supply pins, labeled Vdda, Vddd, and Vddiox, respectively. It also includes two internal 1.8-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](#). The two Vccd pins must be shorted together, with as short a trace as possible, and connected to a 1-μF ±10-percent X5R capacitor. The power system also contains a sleep regulator, an I²C regulator, and a hibernate regulator.

Figure 6-4. PSoC Power System



Note The two Vccd pins must be connected together with as short a trace as possible. A trace under the device is recommended, as shown in [Figure 2-6](#) on page 10.

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.

voltage. In standby mode, most boost functions are disabled, thus reducing power consumption of the boost circuit. The converter can be configured to provide low-power, low-current regulation in the standby mode. The external 32-kHz crystal can be used to generate inductor boost pulses on the rising and falling edge of the clock when the output voltage is less than the programmed value. This is called automatic thump mode (ATM).

The boost typically draws 200 μ A in active mode and 12 μ A in standby mode. The boost operating modes must be used in conjunction with chip power modes to minimize the total chip power consumption. Table 6-4 lists the boost power modes available in different chip power modes.

Table 6-4. Chip and Boost Power Modes Compatibility

Chip Power Modes	Boost Power Modes
Chip – Active mode	Boost can be operated in either active or standby mode.
Chip – Sleep mode	Boost can be operated in either active or standby mode. However, it is recommended to operate boost in standby mode for low-power consumption
Chip – Hibernate mode	Boost can only be operated in active mode. However, it is recommended not to use boost in chip hibernate mode due to high current consumption in boost active mode

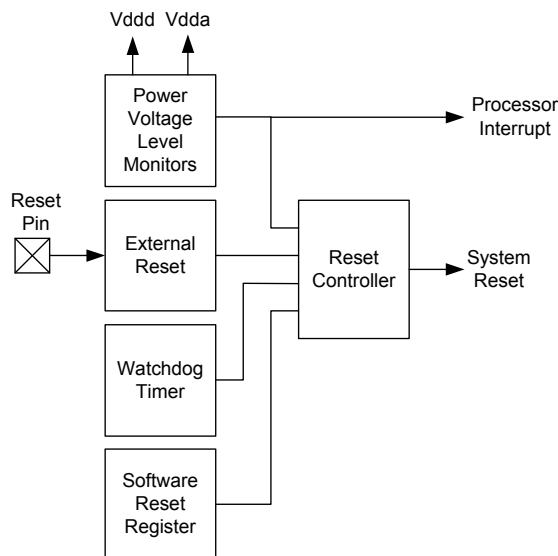
If the boost converter is not used in a given application, tie the Vbat, Vssb, and Vboost pins to ground and leave the Ind pin unconnected.

6.3 Reset

CY8C38 has multiple internal and external reset sources available. The reset sources are:

- **Power source monitoring** – The analog and digital power voltages, Vdda, Vddd, Vcca, and Vccd are monitored in several different modes during power up, active mode, and sleep mode (buzzing). If any of the voltages goes outside predetermined ranges then a reset is generated. The monitors are programmable to generate an interrupt to the processor under certain conditions before reaching the reset thresholds.
- **External** – The device can be reset from an external source by pulling the reset pin (XRES) low. The XRES pin includes an internal pull-up to Vddio1. Vddd, Vdda, and Vddio1 must all have voltage applied before the part comes out of reset.
- **Watchdog timer** – A watchdog timer monitors the execution of instructions by the processor. If the watchdog timer is not reset by firmware within a certain period of time, the watchdog timer generates a reset.
- **Software** – The device can be reset under program control.

Figure 6-7. Resets



The term **device reset** indicates that the processor as well as analog and digital peripherals and registers are reset.

A reset status register holds the source of the most recent reset or power voltage monitoring interrupt. The program may examine this register to detect and report exception conditions. This register is cleared after a power-on reset.

6.3.1 Reset Sources

6.3.1.1 Power Voltage Level Monitors

■ IPOR – Initial power-on reset

At initial power on, IPOR monitors the power voltages V_{DDP} and V_{DDA}, both directly at the pins and at the outputs of the corresponding internal regulators. The trip level is not precise. It is set to approximately 1 volt, which is below the lowest specified operating voltage but high enough for the internal circuits to be reset and to hold their reset state. The monitor generates a reset pulse that is at least 100 ns wide. It may be much wider if one or more of the voltages ramps up slowly.

To save power the IPOR circuit is disabled when the internal digital supply is stable. Voltage supervision is then handed off to the precise low voltage reset (PRES) circuit. When the voltage is high enough for PRES to release, the IMO starts.

■ PRES – Precise low voltage reset

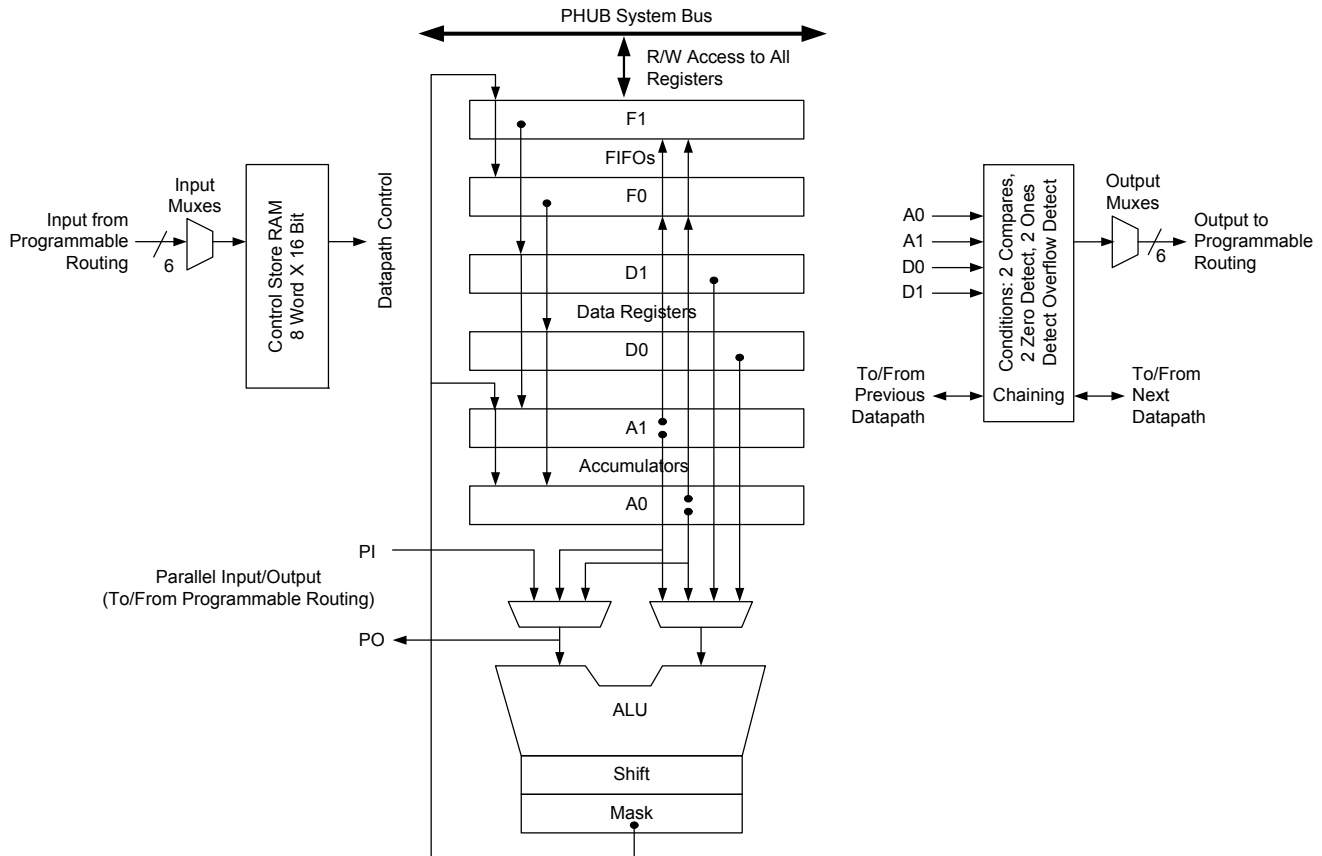
This circuit monitors the outputs of the analog and digital internal regulators after power up. The regulator outputs are compared to a precise reference voltage. The response to a PRES trip is identical to an IPOR reset.

In normal operating mode, the program cannot disable the digital PRES circuit. The analog regulator can be disabled, which also disables the analog portion of the PRES. The PRES circuit is disabled automatically during sleep and hibernate modes, with one exception: During sleep mode the regulators are periodically activated (buzzed) to provide supervisory services and to reduce wakeup time. At these times the PRES circuit is also buzzed to allow periodic voltage monitoring.

7.2.2 Datapath Module

The datapath contains an 8-bit single cycle ALU, with associated compare and condition generation logic. This datapath block is optimized to implement embedded functions, such as timers, counters, integrators, PWMs, PRS, CRC, shifters and dead band generators and many others.

Figure 7-8. Datapath Top Level



7.2.2.1 Working Registers

The datapath contains six primary working registers, which are accessed by CPU firmware or DMA during normal operation.

Table 7-1. Working Datapath Registers

Name	Function	Description
A0 and A1	Accumulators	These are sources and sinks for the ALU and also sources for the compares.
D0 and D1	Data Registers	These are sources for the ALU and sources for the compares.
F0 and F1	FIFOs	These are the primary interface to the system bus. They can be a data source for the data registers and accumulators or they can capture data from the accumulators or ALU. Each FIFO is four bytes deep.

7.2.2.2 Dynamic Datapath Configuration RAM

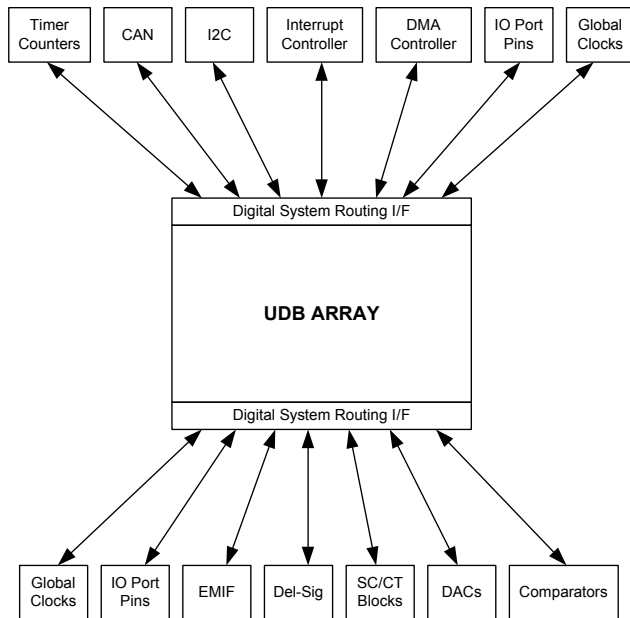
Dynamic configuration is the ability to change the datapath function and internal configuration on a cycle-by-cycle basis, under sequencer control. This is implemented using the 8-word × 16-bit configuration RAM, which stores eight unique 16-bit wide configurations. The address input to this RAM controls the sequence, and can be routed from any block connected to the UDB routing matrix, most typically PLD logic, I/O pins, or from the outputs of this or other datapath blocks.

ALU

The ALU performs eight general purpose functions. They are:

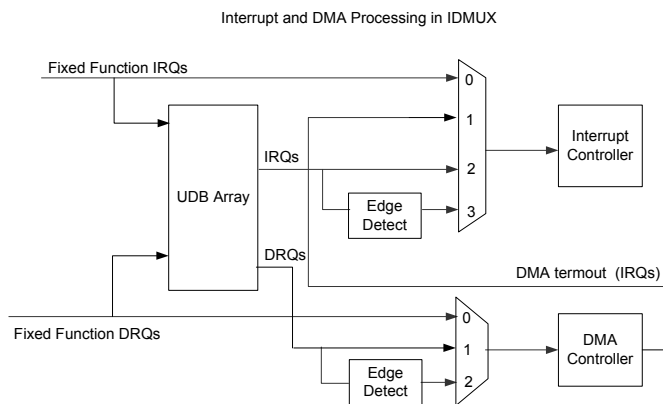
- Increment
- Decrement
- Add
- Subtract
- Logical AND
- Logical OR
- Logical XOR
- Pass, used to pass a value through the ALU to the shift register, mask, or another UDB register.

Figure 7-13. Digital System Interconnect



Interrupt and DMA routing is very flexible in the CY8C38 programmable architecture. In addition to the numerous fixed function peripherals that can generate interrupt requests, any data signal in the UDB array routing can also be used to generate a request. A single peripheral may generate multiple independent interrupt requests simplifying system and firmware design. Figure 7-14 shows the structure of the IDMUX (Interrupt/DMA Multiplexer).

Figure 7-14. Interrupt and DMA Processing in the IDMUX



7.4.1 I/O Port Routing

There are a total of 20 DSI routes to a typical 8-bit I/O port, 16 for data and four for drive strength control.

When an I/O pin is connected to the routing, there are two primary connections available, an input and an output. In conjunction with drive strength control, this can implement a bidirectional I/O pin. A data output signal has the option to be

single synchronized (pipelined) and a data input signal has the option to be double synchronized. The synchronization clock is the system clock (see Figure 6-1). Normally all inputs from pins are synchronized as this is required if the CPU interacts with the signal or any signal derived from it. Asynchronous inputs have rare uses. An example of this is a feed through of combinational PLD logic from input pins to output pins.

Figure 7-15. I/O Pin Synchronization Routing

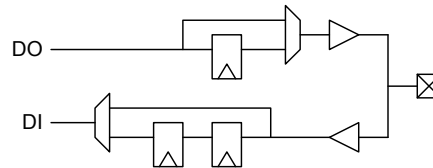
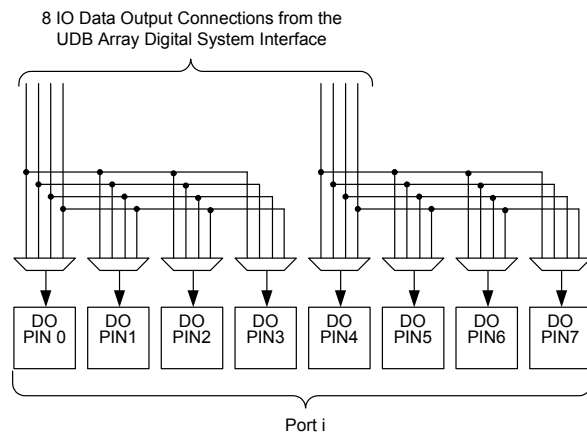


Figure 7-16. I/O Pin Output Connectivity



There are four more DSI connections to a given I/O port to implement dynamic output enable control of pins. This connectivity gives a range of options, from fully ganged 8-bits controlled by one signal, to up to four individually controlled pins. The output enable signal is useful for creating tri-state bidirectional pins and buses.

Figure 7-17. I/O Pin Output Enable Connectivity

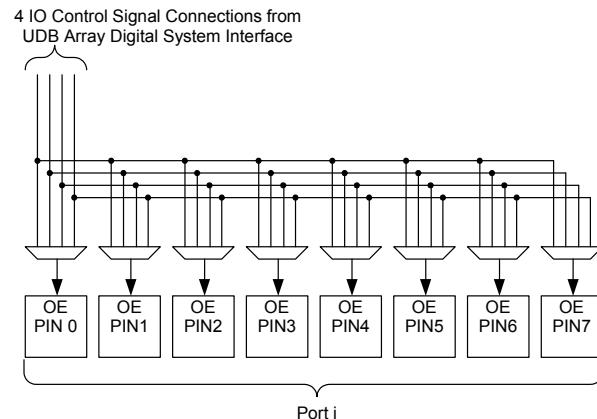
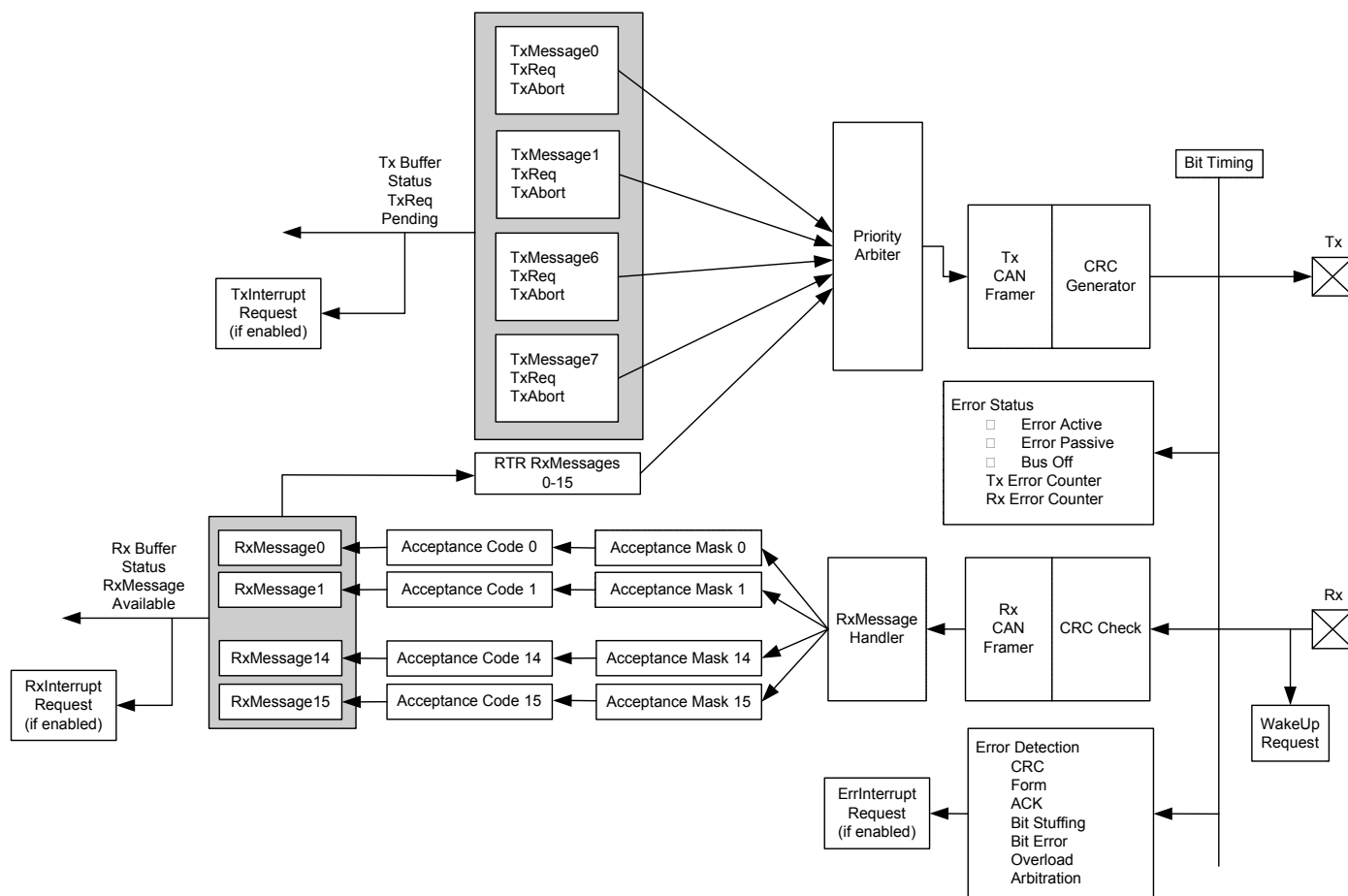


Figure 7-19. CAN Controller Block Diagram



7.6 USB

PSoC includes a dedicated Full-Speed (12 Mbps) USB 2.0 transceiver supporting all four USB transfer types: control, interrupt, bulk, and isochronous. PSoC Creator provides full configuration support. USB interfaces to hosts through two dedicated USBIO pins, which are detailed in the “[I/O System and Routing](#)” section on page 28.

USB includes the following features:

- Eight unidirectional data endpoints
- One bidirectional control endpoint 0 (EP0)
- Shared 512-byte buffer for the eight data endpoints
- Dedicated 8-byte buffer for EP0
- Three memory modes
 - Manual memory management with no DMA access
 - Manual memory management with manual DMA access
 - Automatic memory management with automatic DMA access
- Internal 3.3-V regulator for transceiver

- Internal 48-MHz main oscillator mode that auto locks to USB bus clock, requiring no external crystal for USB (USB equipped parts only)
- Interrupts on bus and each endpoint event, with device wakeup
- USB reset, suspend, and resume operations
- Bus-powered and self-powered modes

Figure 7-20. USB

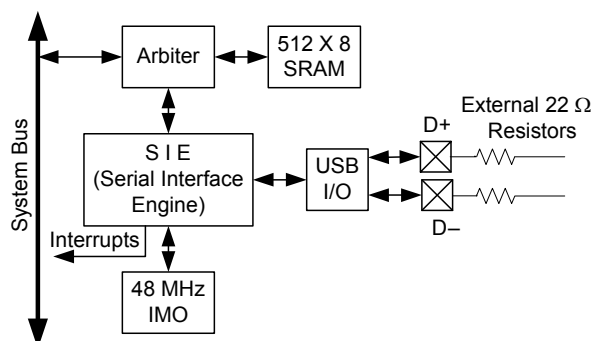
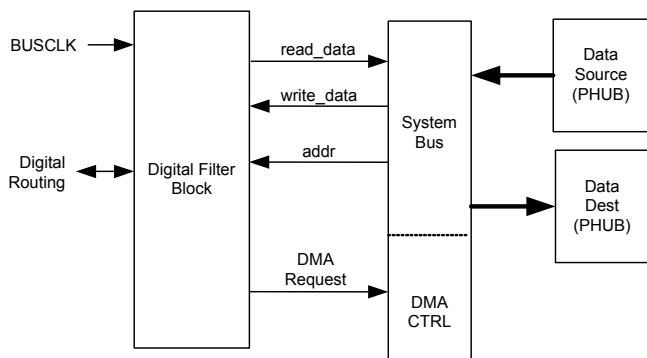


Figure 7-22. DFB Application Diagram (pwr/gnd not shown)



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.
- Up to four 8-bit DACs that provide either voltage or current output.
- Four comparators with optional connection to configurable LUT outputs.
- Up to four configurable switched capacitor/continuous time (SC/CT) blocks for functions that include opamp, unity gain buffer, programmable gain amplifier, transimpedance amplifier, and mixer.
- Up to 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 8-1. Analog Subsystem Block Diagram

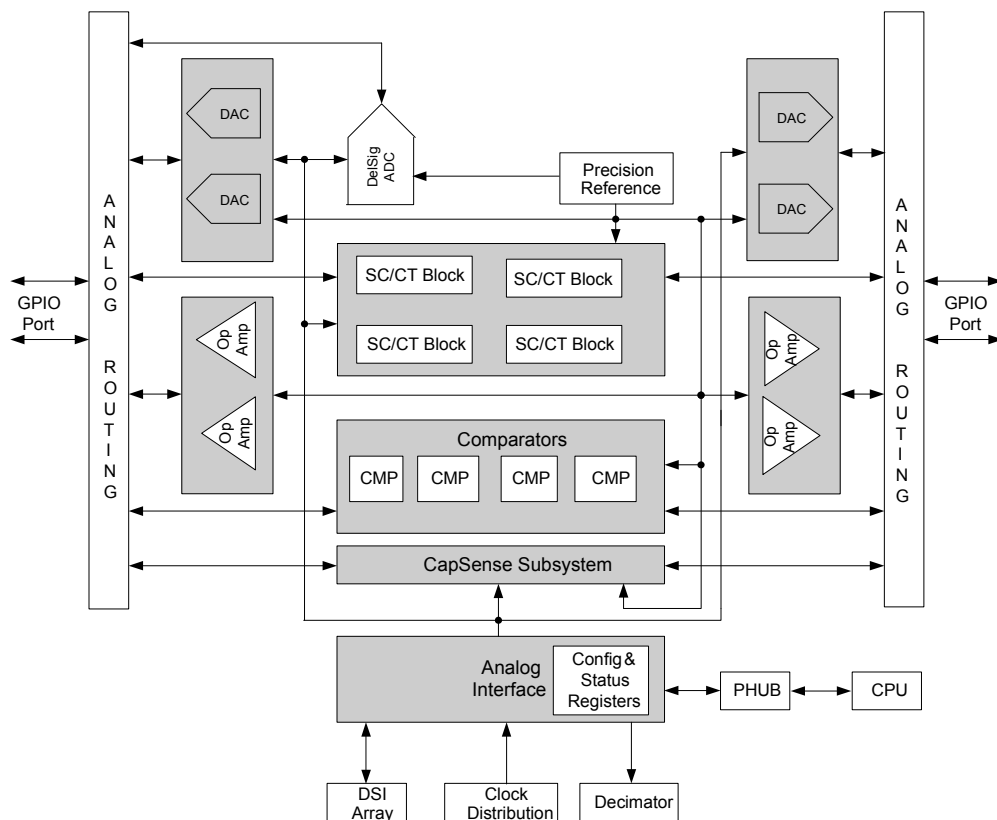
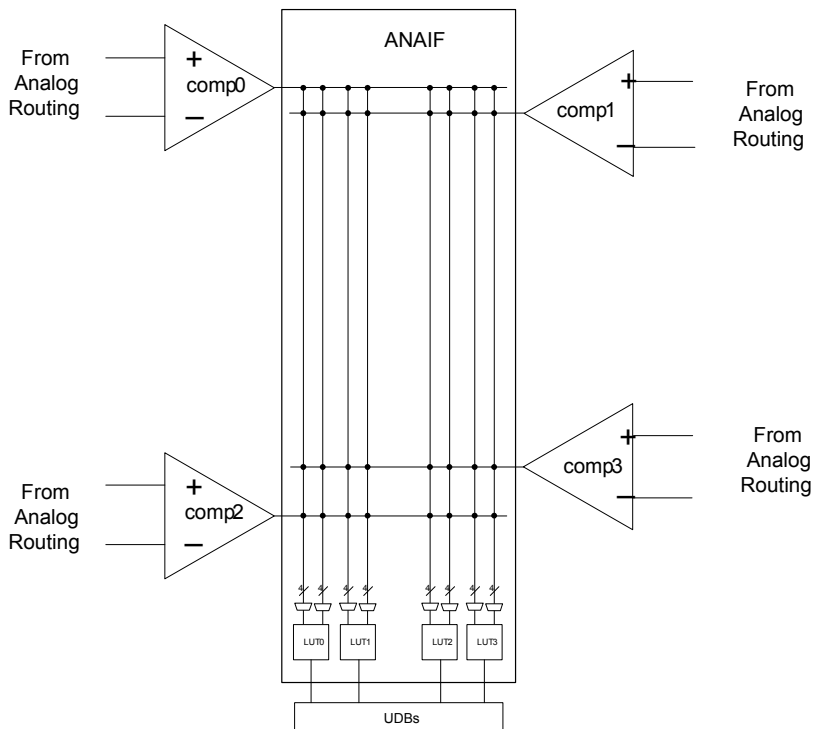


Figure 8-6. Analog Comparator



8.3.2 LUT

The CY8C38 family of devices contains four LUTs. The LUT is a two input, one output lookup table that is driven by any one or two of the comparators in the chip. The output of any LUT is routed to the digital system interface of the UDB array. From the digital system interface of the UDB array, these signals can be connected to UDBs, DMA controller, I/O, or the interrupt controller.

The LUT control word written to a register sets the logic function on the output. The available LUT functions and the associated control word is shown in [Table 8-2](#).

Table 8-2. LUT Function vs. Program Word and Inputs

Control Word	Output (A and B are LUT inputs)
0000b	FALSE ('0')
0001b	A AND B
0010b	A AND (NOT B)
0011b	A
0100b	(NOT A) AND B
0101b	B
0110b	A XOR B
0111b	A OR B
1000b	A NOR B
1001b	A XNOR B
1010b	NOT B
1011b	A OR (NOT B)
1100b	NOT A
1101b	(NOT A) OR B
1110b	A NAND B
1111b	TRUE ('1')

8.4 Opamps

The CY8C38 family of devices contain up to four general purpose opamps in a device.

11. Electrical Specifications

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. The unique flexibility of the PSoC UDBs and analog blocks enable many functions to be implemented in PSoC Creator components, see the component datasheets for full AC/DC specifications of individual functions. See the [“Example Peripherals”](#) section on page 35 for further explanation of PSoC Creator components.

11.1 Absolute Maximum Ratings

Table 11-1. Absolute Maximum Ratings DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
T_{STG}	Storage temperature	Higher storage temperatures reduce NVL data retention time. Recommended storage temperature is $+25\text{ }^{\circ}\text{C} \pm 25\text{ }^{\circ}\text{C}$. Extended duration storage temperatures above $85\text{ }^{\circ}\text{C}$ degrade reliability.	-55	25	100	$^{\circ}\text{C}$
V_{DDA}	Analog supply voltage relative to V_{SSA}		-0.5	–	6	V
V_{DDD}	Digital supply voltage relative to V_{SSD}		-0.5	–	6	V
V_{DDIO}	I/O supply voltage relative to V_{SSD}		-0.5	–	6	V
V_{CCA}	Direct analog core voltage input		-0.5	–	1.95	V
V_{CCD}	Direct digital core voltage input		-0.5	–	1.95	V
V_{SSA}	Analog ground voltage		$V_{SSD} - 0.5$	–	$V_{SSD} + 0.5$	V
$V_{GPIO}^{[17]}$	DC input voltage on GPIO	Includes signals sourced by V_{DDA} and routed internal to the pin	$V_{SSD} - 0.5$	–	$V_{DDIO} + 0.5$	V
V_{SIO}	DC input voltage on SIO	Output disabled	$V_{SSD} - 0.5$	–	7	V
		Output enabled	$V_{SSD} - 0.5$	–	6	V
V_{IND}	Voltage at boost converter input		0.5	–	5.5	V
V_{BAT}	Boost converter supply		$V_{SSD} - 0.5$	–	5.5	V
I_{VDDIO}	Current per V_{DDIO} supply pin		–	–	100	mA
V_{extref}	ADC external reference inputs	Pins P0[3], P3[2]	–	–	2	V
I_{LU}	Latch up current ^[18]		-140	–	140	mA
ESD_{HBM}	Electrostatic discharge voltage	Human body model	750	–	–	V
ESD_{CDM}	Electrostatic discharge voltage	Charge device model	500	–	–	V

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

Notes

17. The V_{DDIO} supply voltage must be greater than the maximum analog voltage on the associated GPIO pins. Maximum analog voltage on GPIO pin $\leq V_{DDIO} \leq V_{DDA}$.

18. Meets or exceeds JEDEC Spec EIA/JESD78 IC Latch-up Test.

11.4.2 SIO

Table 11-11. SIO DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
Vinmax	Maximum input voltage	All allowed values of Vddio and Vddd, see Section 11.2.1	–	–	5.5	V
Vinref	Input voltage reference (Differential input mode)		0.5	–	$0.52 \times V_{DDIO}$	V
Voutref	Output voltage reference (Regulated output mode)					
		$V_{DDIO} > 3.7$	1	–	$V_{DDIO} - 1$	V
		$V_{DDIO} < 3.7$	1	–	$V_{DDIO} - 0.5$	V
V _{IH}	Input voltage high threshold					
	GPIO mode	CMOS input	$0.7 \times V_{DDIO}$	–	–	V
	Differential input mode ^[30]	Hysteresis disabled	SIO_ref + 0.2	–	–	V
V _{IL}	Input voltage low threshold					
	GPIO mode	CMOS input	–	–	$0.3 \times V_{DDIO}$	V
	Differential input mode ^[30]	Hysteresis disabled	–	–	SIO_ref – 0.2	V
V _{OH}	Output voltage high					
	Unregulated mode	$I_{OH} = 4 \text{ mA}$, $V_{DDIO} = 3.3 \text{ V}$	$V_{DDIO} - 0.4$	–	–	V
	Regulated mode ^[30]	$I_{OH} = 1 \text{ mA}$	SIO_ref – 0.65	–	SIO_ref + 0.2	V
	Regulated mode ^[30]	$I_{OH} = 0.1 \text{ mA}$	SIO_ref – 0.3	–	SIO_ref + 0.2	V
V _{OL}	Output voltage low					
		$V_{DDIO} = 3.30 \text{ V}$, $I_{OL} = 25 \text{ mA}$	–	–	0.8	V
		$V_{DDIO} = 1.80 \text{ V}$, $I_{OL} = 4 \text{ mA}$	–	–	0.4	V
Rpullup	Pull-up resistor		3.5	5.6	8.5	kΩ
Rpulldown	Pull-down resistor		3.5	5.6	8.5	kΩ
I _{IL}	Input leakage current (Absolute value) ^[31]					
	$V_{IH} \leq V_{DDIO}$	25 °C, Vddsio = 3.0 V, $V_{IH} = 3.0 \text{ V}$	–	–	14	nA
	$V_{IH} > V_{DDIO}$	25 °C, Vddsio = 0 V, $V_{IH} = 3.0 \text{ V}$	–	–	10	μA
C _{IN}	Input Capacitance ^[31]		–	–	7	pF
V _H	Input voltage hysteresis (Schmitt-Trigger) ^[31]	Single ended mode (GPIO mode)	–	40	–	mV
		Differential mode	–	35	–	mV
Idiode	Current through protection diode to V _{SSIO}		–	–	100	μA

Table 11-12. SIO AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
TriseF	Rise time in fast strong mode (90/10%) ^[31]	Clod = 25 pF, $V_{DDIO} = 3.3 \text{ V}$	–	–	12	ns
TfallF	Fall time in fast strong mode (90/10%) ^[31]	Clod = 25 pF, $V_{DDIO} = 3.3 \text{ V}$	–	–	12	ns
TriseS	Rise time in slow strong mode (90/10%) ^[31]	Clod = 25 pF, $V_{DDIO} = 3.0 \text{ V}$	–	–	75	ns
TfallS	Fall time in slow strong mode (90/10%) ^[31]	Clod = 25 pF, $V_{DDIO} = 3.0 \text{ V}$	–	–	60	ns

Notes

30. See Figure 6-9 on page 30 and Figure 6-12 on page 33 for more information on SIO reference.

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

Figure 11-6. Opamp Operating Current vs Vdda, Power Mode = Minimum

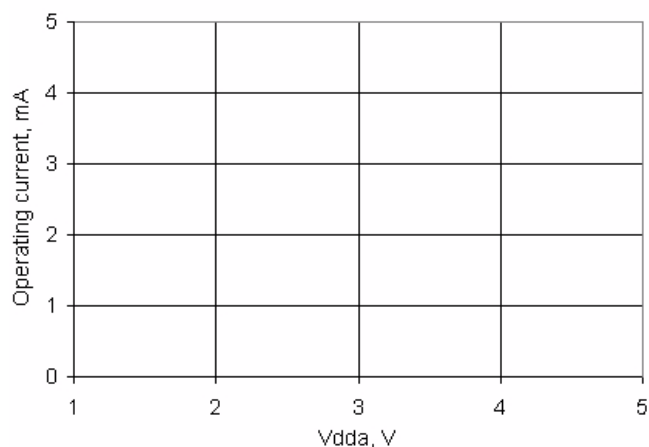


Figure 11-7. Opamp Operating Current vs Vdda, Power Mode = Low

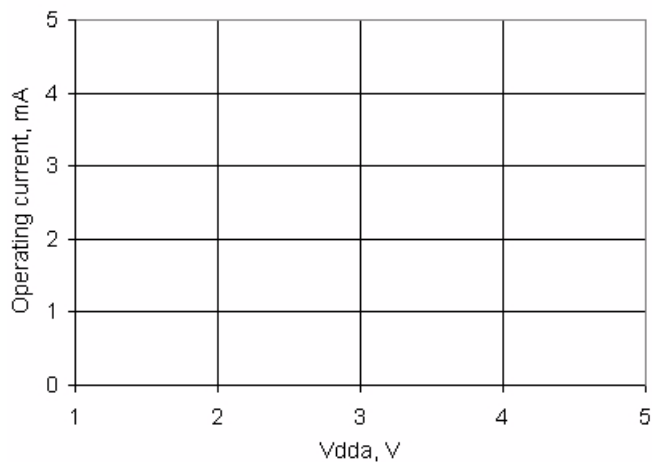


Figure 11-8. . Opamp Operating Current vs Vdda, Power Mode = Medium

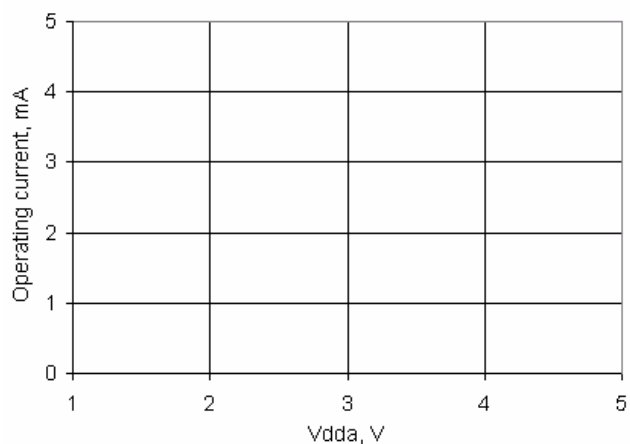


Figure 11-9. Opamp Operating Current vs Vdda, Power Mode = High

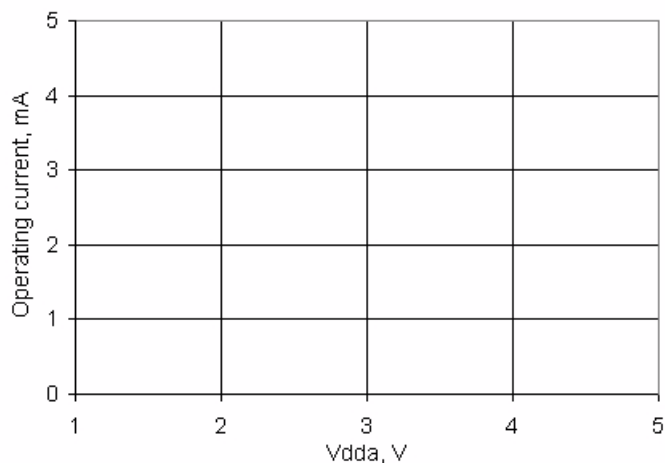


Table 11-19. Opamp AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
GBW	Gain-bandwidth product	Power mode = minimum, 200 pF load	1	—	—	MHz
		Power mode = low, 200 pF load	2	—	—	MHz
		Power mode = medium, 200 pF load	1	—	—	MHz
		Power mode = high, 200 pF load	3	—	—	MHz
SR	Slew rate, 20% - 80%	Power mode = low, 200 pF load	1.1	—	—	V/μs
		Power mode = medium, 200 pF load	0.9	—	—	V/μs
		Power mode = high, 200 pF load	3	—	—	V/μs
e_n	Input noise density	Power mode = high, Vdda = 5 V, at 100 kHz	—	45	—	nV/sqrtHz

Figure 11-10. Open Loop Gain and Phase vs Frequency and Temperature, Power Mode = High, C_I = 15 pF, Vdda = 5V

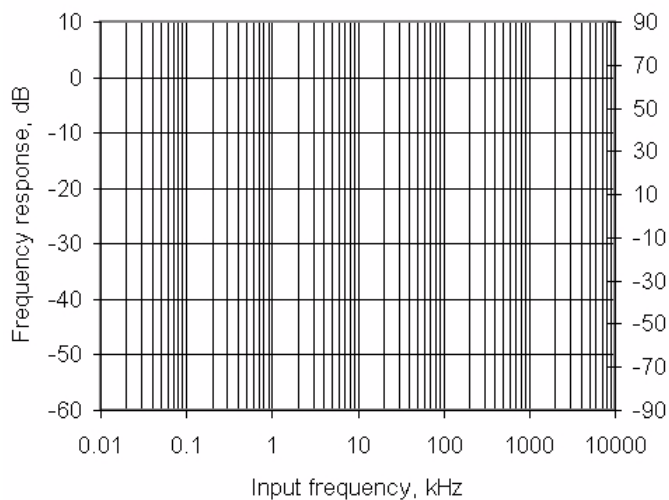


Figure 11-11. Opamp Closed Loop Frequency Response, Gain = 1, Vdda = 5V

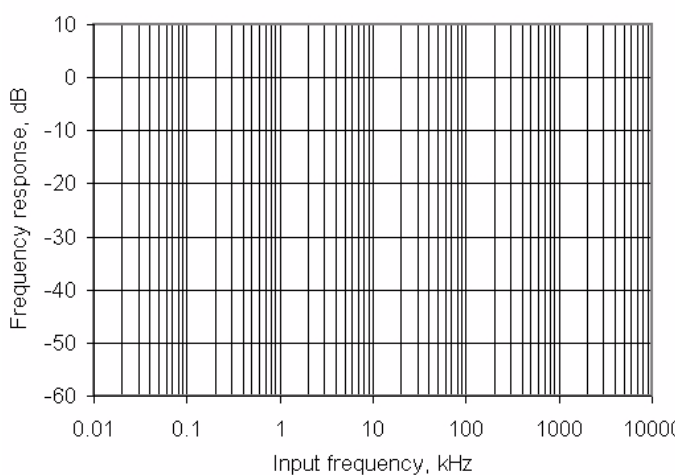


Figure 11-12. Opamp Closed Loop Frequency Response, Gain = 10, Vdda = 5V

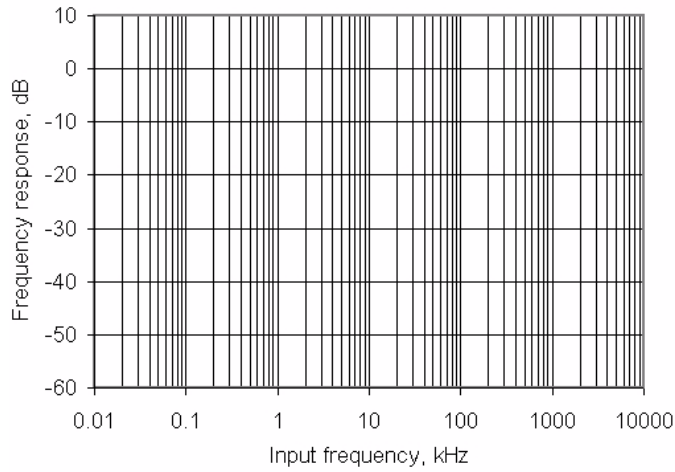


Figure 11-13. Opamp test Circuit for Gain = 10

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

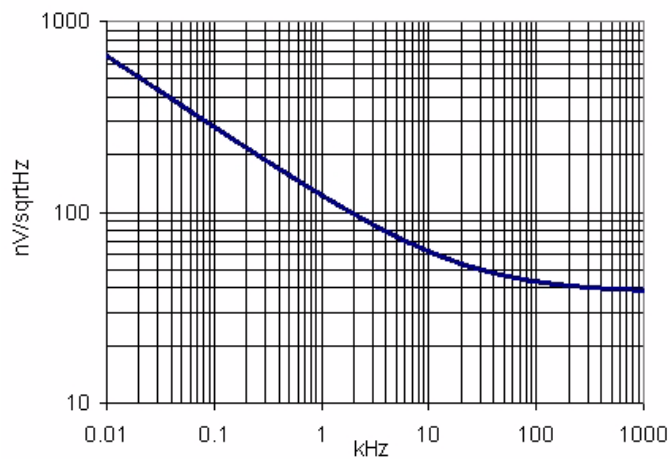
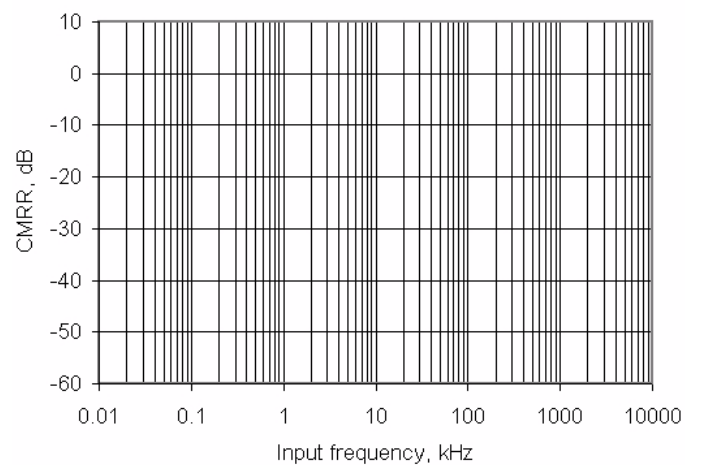


Figure 11-15. Opamp CMRR vs Frequency



11.5.12 LCD Direct Drive

Table 11-40. LCD Direct Drive DC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
I_{CC}	LCD system operating current	Device sleep mode with wakeup at 400-Hz rate to refresh LCDs, bus clock = 3 Mhz, Vddio = Vdda = 3 V, 4 commons, 16 segments, 1/4 duty cycle, 50 Hz frame rate, no glass connected	–	38	–	μA
I_{CC_SEG}	Current per segment driver	Strong drive mode	–	260	–	μA
V_{BIAS}	LCD bias range (V_{BIAS} refers to the main output voltage(V0) of LCD DAC)	$V_{DDA} \geq 3 V$ and $V_{DDA} \geq V_{BIAS}$	2	–	5	V
	LCD bias step size	$V_{DDA} \geq 3 V$ and $V_{DDA} \geq V_{BIAS}$	–	$9.1 \times V_{DDA}$	–	mV
	LCD capacitance per segment/common driver	Drivers may be combined	–	500	5000	pF
	Long term segment offset		–	–	20	mV
I_{OUT}	Output drive current per segment driver)	Vddio = 5.5V, strong drive mode	355	–	710	μA

Table 11-41. LCD Direct Drive AC Specifications

Parameter	Description	Conditions	Min	Typ	Max	Units
f_{LCD}	LCD frame rate		10	50	150	Hz

Figure 11-51. Synchronous Write Cycle Timing

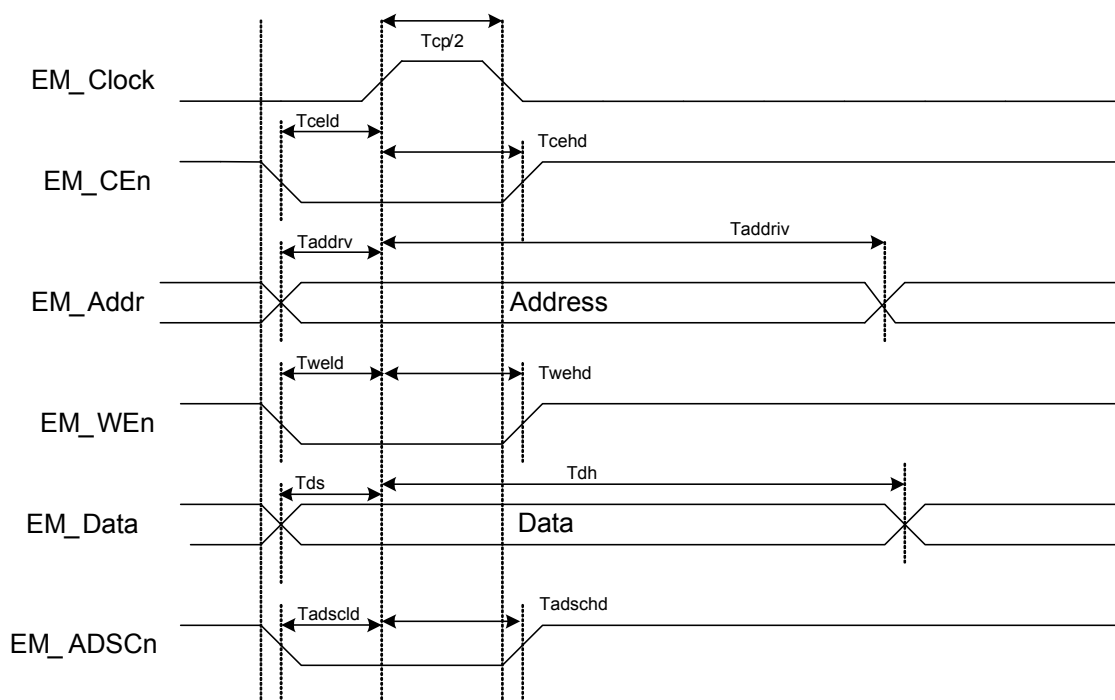


Table 11-66. Synchronous Write Cycle Specifications

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.