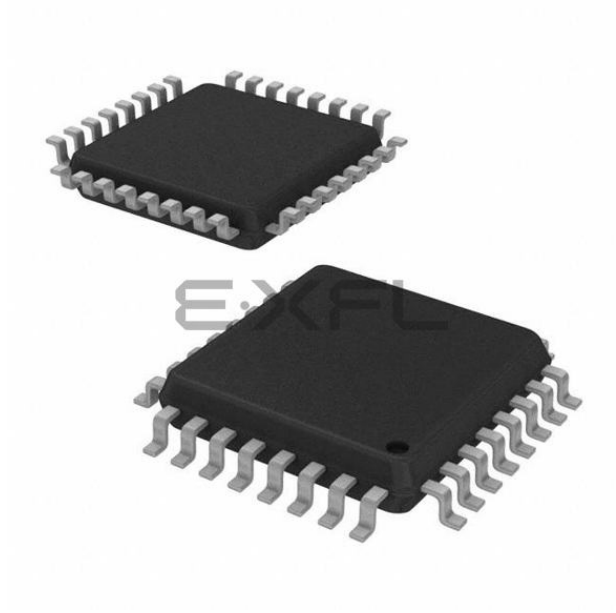


Welcome to E-XFL.COM

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	Active
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	25
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 25x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f567-iq

4.4. QFN-24 Package Specifications

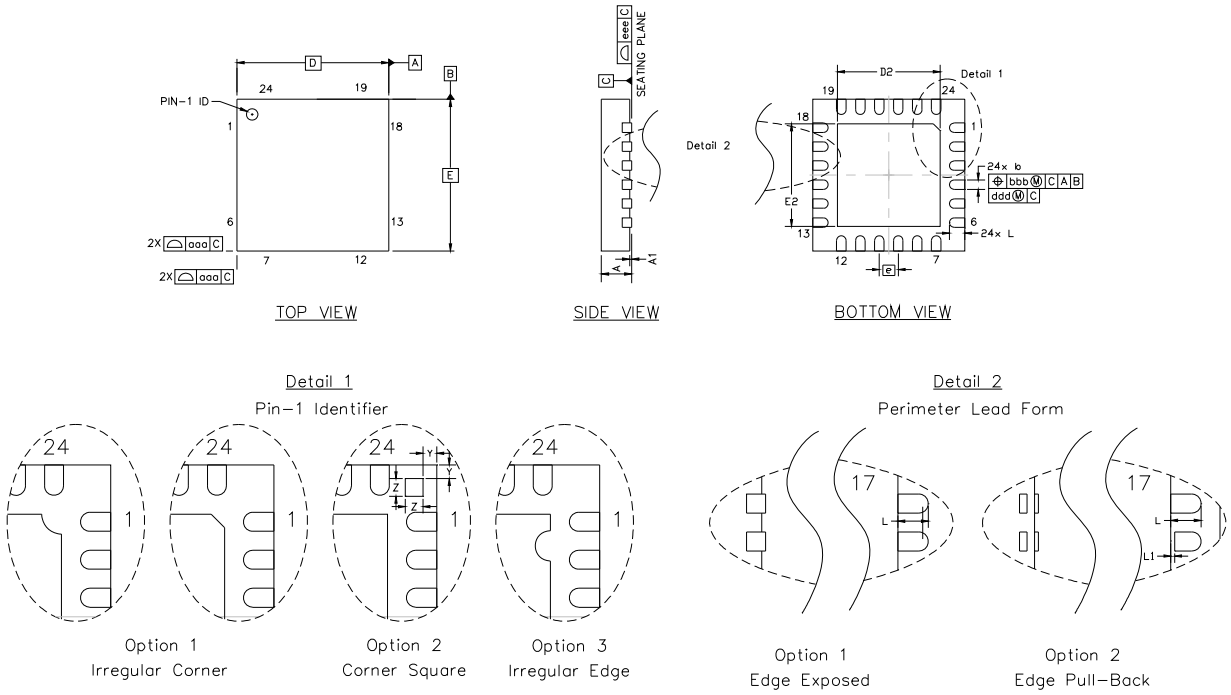


Figure 4.7. QFN-24 Package Drawing

Table 4.7. QFN-24 Package Dimensions

Dimension	Min	Typ	Max	Dimension	Min	Typ	Max
A	0.70	0.75	0.80	L	0.30	0.40	0.50
A1	0.00	0.02	0.05	L1	0.00		0.15
b	0.18	0.25	0.30	aaa			0.15
D	4.00 BSC			bbb			0.10
D2	2.55	2.70	2.80	ddd			0.05
e	0.50 BSC			eee			0.08
E	4.00 BSC			Z		0.24	
E2	2.55	2.70	2.80	Y		0.18	

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to JEDEC Solid State Outline MO-220, variation WGGD, except for custom features D2, E2, Z, Y, and L which are toleranced per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

5. Electrical Characteristics

5.1. Absolute Maximum Specifications

Table 5.1. Absolute Maximum Ratings

Parameter	Conditions	Min	Typ	Max	Units
Ambient Temperature under Bias		–55	—	135	°C
Storage Temperature		–65	—	150	°C
Voltage on V_{REGIN} with Respect to GND		–0.3	—	5.5	V
Voltage on V_{DD} with Respect to GND		–0.3	—	2.8	V
Voltage on VDDA with Respect to GND		–0.3	—	2.8	V
Voltage on V_{IO} with Respect to GND		–0.3	—	5.5	V
Voltage on any Port I/O Pin or $\overline{RST}$ with Respect to GND		–0.3	—	V_{IO} + 0.3	V
Maximum Total Current through V_{REGIN} or GND		—	—	500	mA
Maximum Output Current Sunk by $\overline{RST}$ or any Port Pin		—	—	100	mA
Maximum Output Current Sourced by any Port Pin		—	—	100	mA

Note: Stresses outside of the range of the "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions outside of those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

SFR Definition 6.13. ADC0MX: ADC0 Channel Select

Bit	7	6	5	4	3	2	1	0
Name			ADC0MX[5:0]					
Type	R	R	R/W					
Reset	0	0	1	1	1	1	1	1

SFR Address = 0xBB; SFR Page = 0x00;

Bit	Name	Function
7:6	Unused	Read = 00b; Write = Don't Care.
5:0	AMX0P[5:0]	**AMUX0 Positive Input Selection.** 000000: P0.0 000001: P0.1 000010: P0.2 000011: P0.3 000100: P0.4 000101: P0.5 000110: P0.6 000111: P0.7 001000: P1.0 001001: P1.1 001010: P1.2 001011: P1.3 001100: P1.4 001101: P1.5 001110: P1.6 001111: P1.7 010000: P2.0 010001: P2.1 010010: P2.2 (Only available on 40-pin and 32-pin package devices) 010011: P2.3 (Only available on 40-pin and 32-pin package devices) 010100: P2.4 (Only available on 40-pin and 32-pin package devices) 010101: P2.5 (Only available on 40-pin and 32-pin package devices) 010110: P2.6 (Only available on 40-pin and 32-pin package devices) 010111: P2.7 (Only available on 40-pin and 32-pin package devices) 011000: P3.0 (Only available on 40-pin and 32-pin package devices) 011001: P3.1 (Only available on 40-pin package devices) 011010: P3.2 (Only available on 40-pin package devices) 011011: P3.3 (Only available on 40-pin package devices) 011100: P3.4 (Only available on 40-pin package devices) 011101: P3.5 (Only available on 40-pin package devices) 011110: P3.6 (Only available on 40-pin package devices) 011111: P3.7 (Only available on 40-pin package devices) 100000–101111: Reserved 110000: Temp Sensor 110001: V_{DD} 110010–111111: GND

Notes on Registers, Operands and Addressing Modes:
Rn—Register R0–R7 of the currently selected register bank. **@Ri**—Data RAM location addressed indirectly through R0 or R1. **rel**—8-bit, signed (two's complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps. **direct**—8-bit internal data location's address. This could be a direct-access Data RAM location (0x00–0x7F) or an SFR (0x80–0xFF). **#data**—8-bit constant **#data16**—16-bit constant **bit**—Direct-accessed bit in Data RAM or SFR **addr11**—11-bit destination address used by ACALL and AJMP. The destination must be within the same 2 kB page of program memory as the first byte of the following instruction. **addr16**—16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 64 kB program memory space. There is one unused opcode (0xA5) that performs the same function as NOP. All mnemonics copyrighted © Intel Corporation 1980.

10.3. CIP-51 Register Descriptions

Following are descriptions of SFRs related to the operation of the CIP-51 System Controller. Reserved bits should not be set to logic l. Future product versions may use these bits to implement new features in which case the reset value of the bit will be logic 0, selecting the feature's default state. Detailed descriptions of the remaining SFRs are included in the sections of the datasheet associated with their corresponding system function.

18.3. Clock Multiplier

The Clock Multiplier generates an output clock which is 4 times the input clock frequency scaled by a programmable factor of 1, 2/3, 2/4 (or 1/2), 2/5, 2/6 (or 1/3), or 2/7. The Clock Multiplier's input can be selected from the external oscillator, or the internal or external oscillators divided by 2. This produces three possible base outputs which can be scaled by a programmable factor: Internal Oscillator x 2, External Oscillator x 2, or External Oscillator x 4. See Section 18.1 on page 157 for details on system clock selection.

The Clock Multiplier is configured via the CLKMUL register (SFR Definition 18.5). The procedure for configuring and enabling the Clock Multiplier is as follows:

1. Reset the Multiplier by writing 0x00 to register CLKMUL.
2. Select the Multiplier input source via the MULSEL bits.
3. Select the Multiplier output scaling factor via the MULDIV bits
4. Enable the Multiplier with the MULEN bit (CLKMUL | = 0x80).
5. Delay for >5 µs.
6. Initialize the Multiplier with the MULINIT bit (CLKMUL | = 0xC0).
7. Poll for MULRDY $\geq$ 1.

Important Note: When using an external oscillator as the input to the Clock Multiplier, the external source must be enabled and stable before the Multiplier is initialized. See "18.4. External Oscillator Drive Circuit" on page 164 for details on selecting an external oscillator source.

The Clock Multiplier allows faster operation of the CIP-51 core and is intended to generate an output frequency between 25 and 50 MHz. The clock multiplier can also be used with slow input clocks. However, if the clock is below the minimum Clock Multiplier input frequency (FCM_{min}), the generated clock will consist of four fast pulses followed by a long delay until the next input clock rising edge. The average frequency of the output is equal to 4x the input, but the instantaneous frequency may be faster. See Figure 18.2 below for more information.

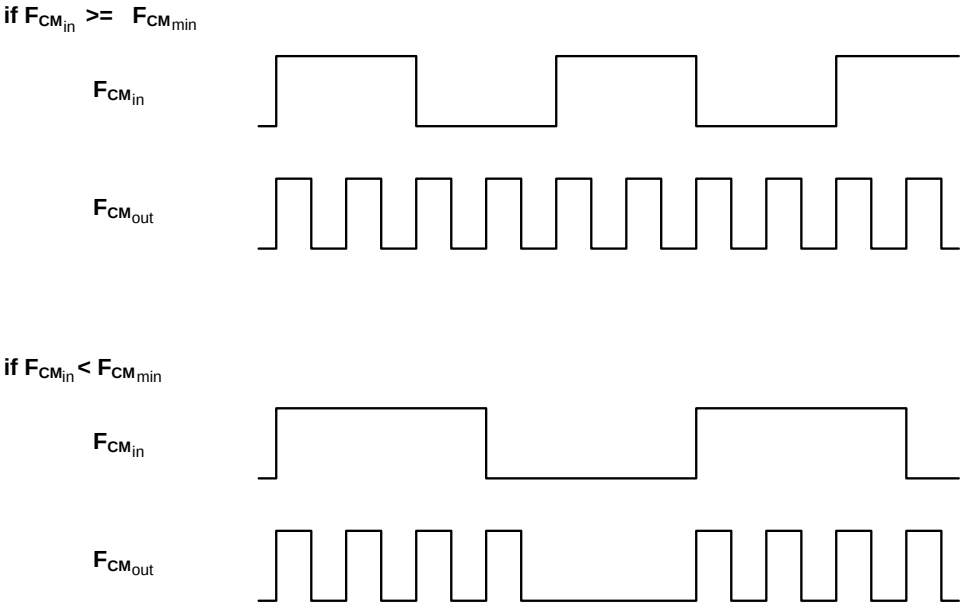


Figure 18.2. Example Clock Multiplier Output

Note: When VIO rises faster than VDD, which can happen when VREGIN and VIO are tied together, a delay created between GPIO power (VIO) and the logic controlling GPIO (VDD) results in a temporary unknown state at the GPIO pins. When VIO rises faster than VDD, the GPIO may enter the following states: floating, glitch low, or glitch high. Cross coupling VIO and VDD with a 4.7 µF capacitor mitigates the root cause of the problem by allowing VIO and VDD to rise at the same rate.

19.1. Port I/O Modes of Operation

Port pins P0.0–P4.0 use the Port I/O cell shown in Figure 19.2. Each Port I/O cell can be configured by software for analog I/O or digital I/O using the PnMDIN registers. On reset, all Port I/O cells default to a high impedance state with weak pull-ups enabled until the Crossbar is enabled (XBARE = 1).

19.1.1. Port Pins Configured for Analog I/O

Any pins to be used as Comparator or ADC inputs, external oscillator inputs, or VREF should be configured for analog I/O (PnMDIN.n = 0). When a pin is configured for analog I/O, its weak pullup, digital driver, and digital receiver are disabled. Port pins configured for analog I/O will always read back a value of 0.

Configuring pins as analog I/O saves power and isolates the Port pin from digital interference. Port pins configured as digital inputs may still be used by analog peripherals; however, this practice is not recommended and may result in measurement errors.

19.1.2. Port Pins Configured For Digital I/O

Any pins to be used by digital peripherals (UART, SPI, SMBus, etc.), external digital event capture functions, or as GPIO should be configured as digital I/O (PnMDIN.n = 1). For digital I/O pins, one of two output modes (push-pull or open-drain) must be selected using the PnMDOUT registers.

Push-pull outputs (PnMDOUT.n = 1) drive the Port pad to the VIO or GND supply rails based on the output logic value of the Port pin. Open-drain outputs have the high side driver disabled; therefore, they only drive the Port pad to GND when the output logic value is 0 and become high impedance inputs (both high low drivers turned off) when the output logic value is 1.

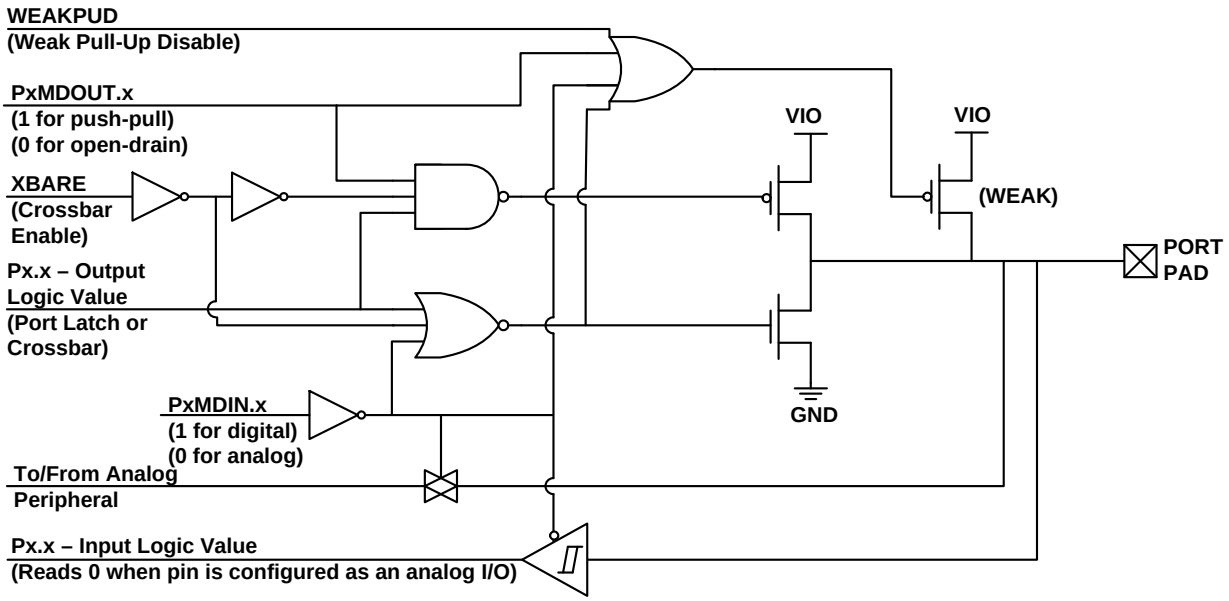


Figure 19.2. Port I/O Cell Block Diagram

SFR Definition 19.6. P1MASK: Port 1 Mask Register

Bit	7	6	5	4	3	2	1	0
Name	P1MASK[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF4; SFR Page = 0x00

Bit	Name	Function
7:0	P1MASK[7:0]	**Port 1 Mask Value.** Selects P1 pins to be compared to the corresponding bits in P1MAT. 0: P1.n pin logic value is ignored and cannot cause a Port Mismatch event. 1: P1.n pin logic value is compared to P1MAT.n.

SFR Definition 19.7. P1MAT: Port 1 Match Register

Bit	7	6	5	4	3	2	1	0
Name	P1MAT[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xF3; SFR Page = 0x00

Bit	Name	Function
7:0	P1MAT[7:0]	**Port 1 Match Value.** Match comparison value used on Port 1 for bits in P1MAT which are set to 1. 0: P1.n pin logic value is compared with logic LOW. 1: P1.n pin logic value is compared with logic HIGH.

SFR Definition 19.10. P3MASK: Port 3 Mask Register

Bit	7	6	5	4	3	2	1	0
Name	P3MASK[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xAF; SFR Page = 0x00

Bit	Name	Function
7:0	P3MASK[7:0]	**Port 1 Mask Value.** Selects P3 pins to be compared to the corresponding bits in P3MAT. 0: P3.n pin logic value is ignored and cannot cause a Port Mismatch event. 1: P3.n pin logic value is compared to P3MAT.n.
Note: P3.0 is available on 40-pin and 32-pin packages. P3.1-P3.7 are available on 40-pin packages		

SFR Definition 19.11. P3MAT: Port 3 Match Register

Bit	7	6	5	4	3	2	1	0
Name	P3MAT[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xAE; SFR Page = 0x00

Bit	Name	Function
7:0	P3MAT[7:0]	**Port 3 Match Value.** Match comparison value used on Port 3 for bits in P3MAT which are set to 1. 0: P3.n pin logic value is compared with logic LOW. 1: P3.n pin logic value is compared with logic HIGH.
Note: P3.0 is available on 40-pin and 32-pin packages. P3.1-P3.7 are available on 40-pin packages		

LIN Register Definition 20.11. LIN0ID: LIN0 Identifier Register

Bit	7	6	5	4	3	2	1	0
Name			ID[5:0]					
Type	R	R	R/W					
Reset	0	0	0	0	0	0	0	0

Indirect Address = 0x0E

Bit	Name	Function
7:6	Unused	Read = 00b; Write = Don't Care.
5:0	ID[5:0]	**LIN Identifier Bits.** These bits form the data identifier. If the LINSIZE bits (LIN0SIZE[3:0]) are 1111b, bits ID[5:4] are used to determine the data size and are interpreted as follows: 00: 2 bytes 01: 2 bytes 10: 4 bytes 11: 8 bytes

Table 22.4. SMBus Status Decoding

Mode	Values Read				Current SMbus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Master Transmitter	1110	0	0	X	A master START was generated.	Load slave address + R/W into SMB0DAT.	0	0	X	1100
	1100	0	0	0	A master data or address byte was transmitted; NACK received.	Set STA to restart transfer.	1	0	X	1110
						Abort transfer.	0	1	X	—
		0	0	1	A master data or address byte was transmitted; ACK received.	Load next data byte into SMB0DAT.	0	0	X	1100
						End transfer with STOP.	0	1	X	—
						End transfer with STOP and start another transfer.	1	1	X	—
						Send repeated START.	1	0	X	1110
						Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT).	0	0	X	1000
Master Receiver	1000	1	0	X	A master data byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1	1000
						Send NACK to indicate last byte, and send STOP.	0	1	0	—
						Send NACK to indicate last byte, and send STOP followed by START.	1	1	0	1110
						Send ACK followed by repeated START.	1	0	1	1110
						Send NACK to indicate last byte, and send repeated START.	1	0	0	1110
						Send ACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	1	1100
						Send NACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	0	1100

the RI0 flag will be set. Note: when MCE0 = 1, RI0 will only be set if the extra bit was equal to 1. Data can be read from the receive FIFO by reading the SBUF0 register. The SBUF0 register represents the oldest byte in the FIFO. After SBUF0 is read, the next byte in the FIFO is immediately loaded into SBUF0, and space is made available in the FIFO for another incoming byte. If enabled, an interrupt will occur when RI0 is set. RI0 can only be cleared to '0' by software when there is no more information in the FIFO. The recommended procedure to empty the FIFO contents is as follows:

1. Clear RI0 to 0.
2. Read SBUF0.
3. Check RI0, and repeat at step 1 if RI0 is set to 1.

If the extra bit function is enabled (XBE0 = 1) and the parity function is disabled (PE0 = 0), the extra bit for the oldest byte in the FIFO can be read from the RBX0 bit (SCON0.2). If the extra bit function is not enabled, the value of the stop bit for the oldest FIFO byte will be presented in RBX0. When the parity function is enabled (PE0 = 1), hardware will check the received parity bit against the selected parity type (selected with S0PT[1:0]) when receiving data. If a byte with parity error is received, the PERR0 flag will be set to 1. This flag must be cleared by software. Note: when parity is enabled, the extra bit function is not available.

Note: The UART Receive FIFO pointer can be corrupted if the UART receives a byte and firmware reads a byte from the FIFO at the same time. When this occurs, firmware will lose the received byte and the FIFO receive overrun flag (OVR0) will also be set to 1. Systems using the UART Receive FIFO should ensure that the FIFO isn't accessed by hardware and firmware at the same time. In other words, firmware should ensure to read the FIFO before the next byte is received.

24. Enhanced Serial Peripheral Interface (SPI0)

The Enhanced Serial Peripheral Interface (SPI0) provides access to a flexible, full-duplex synchronous serial bus. SPI0 can operate as a master or slave device in both 3-wire or 4-wire modes, and supports multiple masters and slaves on a single SPI bus. The slave-select (NSS) signal can be configured as an input to select SPI0 in slave mode, or to disable Master Mode operation in a multi-master environment, avoiding contention on the SPI bus when more than one master attempts simultaneous data transfers. NSS can also be configured as a chip-select output in master mode, or disabled for 3-wire operation. Additional general purpose port I/O pins can be used to select multiple slave devices in master mode.

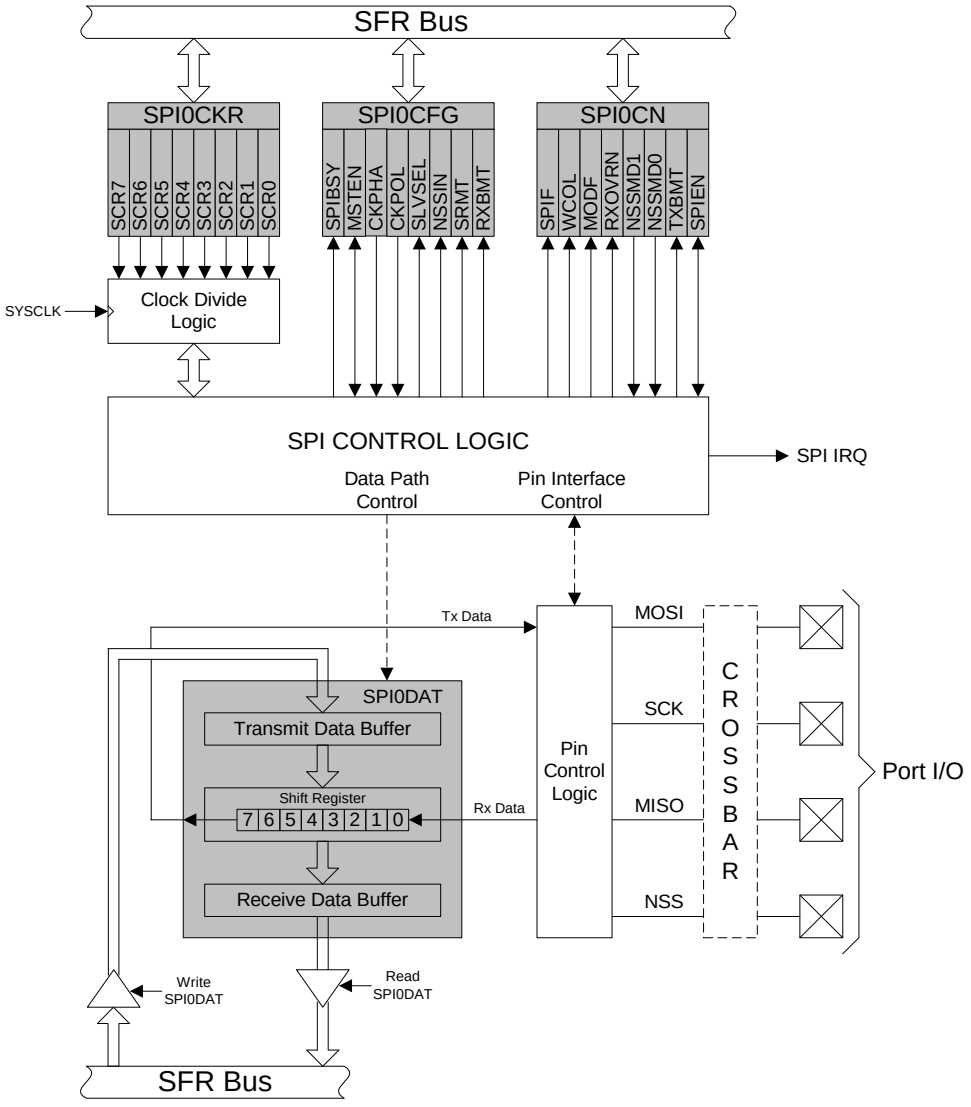
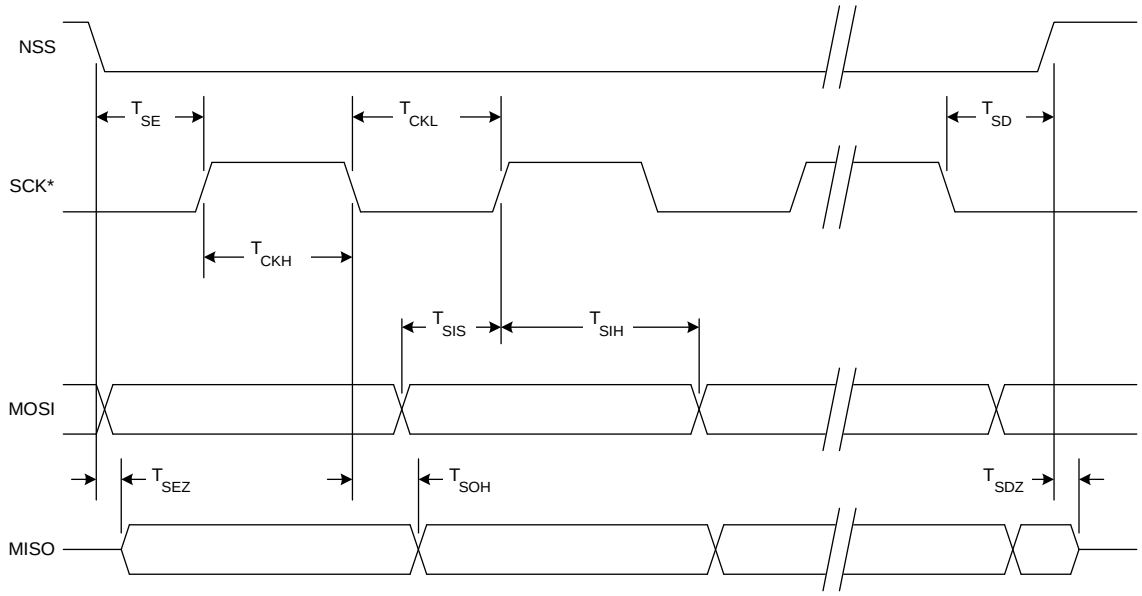
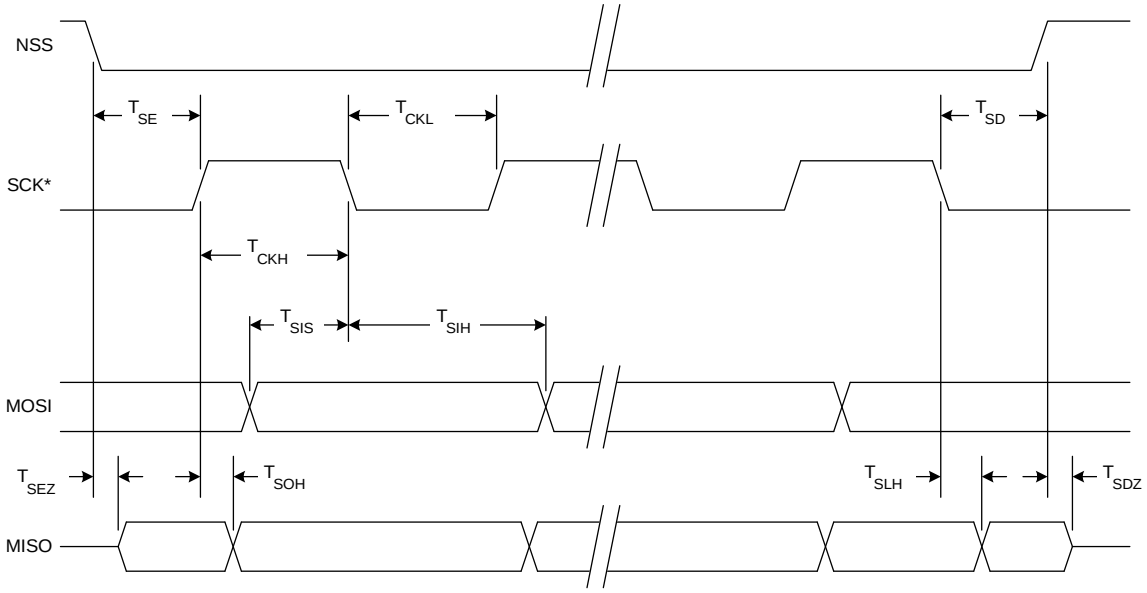


Figure 24.1. SPI Block Diagram



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 24.10. SPI Slave Timing (CKPHA = 0)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 24.11. SPI Slave Timing (CKPHA = 1)

25. Timers

Each MCU includes four counter/timers: two are 16-bit counter/timers compatible with those found in the standard 8051, and two are 16-bit auto-reload timer for use with the ADC, SMBus, or for general purpose use. These timers can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2 and Timer 3 offer 16-bit and split 8-bit timer functionality with auto-reload.

Timer 0 and Timer 1 Modes	Timer 2 Modes	Timer 3 Modes
13-bit counter/timer	16-bit timer with auto-reload	16-bit timer with auto-reload
16-bit counter/timer		
8-bit counter/timer with auto-reload	Two 8-bit timers with auto-reload	Two 8-bit timers with auto-reload
Two 8-bit counter/timers (Timer 0 only)		

Timers 0 and 1 may be clocked by one of five sources, determined by the Timer Mode Select bits (T1M–T0M) and the Clock Scale bits (SCA1–SCA0). The Clock Scale bits define a pre-scaled clock from which Timer 0 and/or Timer 1 may be clocked (See SFR Definition 25.1 for pre-scaled clock selection).Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock.

Timer 2 and Timer 3 may be clocked by the system clock, the system clock divided by 12, or the external oscillator clock source divided by 8.

Timer 0 and Timer 1 may also be operated as counters. When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin (T0 or T1). Events with a frequency of up to one-fourth the system clock frequency can be counted. The input signal need not be periodic, but it should be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.

SFR Definition 25.8. TMR2CN: Timer 2 Control

Bit	7	6	5	4	3	2	1	0
Name	TF2H	TF2L	TF2LEN	TF2CEN	T2SPLIT	TR2		T2XCLK
Type	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC8; Bit-Addressable; SFR Page = 0x00

Bit	Name	Function
7	TF2H	**Timer 2 High Byte Overflow Flag.** Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit is not automatically cleared by hardware.
6	TF2L	**Timer 2 Low Byte Overflow Flag.** Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. TF2L will be set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware.
5	TF2LEN	**Timer 2 Low Byte Interrupt Enable.** When set to 1, this bit enables Timer 2 Low Byte interrupts. If Timer 2 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 2 overflows.
4	TF2CEN	**Timer 2 Capture Mode Enable.** 0: Timer 2 Capture Mode is disabled. 1: Timer 2 Capture Mode is enabled.
3	T2SPLIT	**Timer 2 Split Mode Enable.** When this bit is set, Timer 2 operates as two 8-bit timers with auto-reload. 0: Timer 2 operates in 16-bit auto-reload mode. 1: Timer 2 operates as two 8-bit auto-reload timers.
2	TR2	**Timer 2 Run Control.** Timer 2 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in split mode.
1	Unused	Read = 0b; Write = Don’t Care
0	T2XCLK	**Timer 2 External Clock Select.** This bit selects the external clock source for Timer 2. If Timer 2 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the external clock and the system clock for either timer. 0: Timer 2 clock is the system clock divided by 12. 1: Timer 2 clock is the external clock divided by 8 (synchronized with SYSCLK).

SFR Definition 25.11. TMR2L: Timer 2 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCC; SFR Page = 0x00

Bit	Name	Function
7:0	TMR2L[7:0]	**Timer 2 Low Byte.** In 16-bit mode, the TMR2L register contains the low byte of the 16-bit Timer 2. In 8-bit mode, TMR2L contains the 8-bit low byte timer value.

SFR Definition 25.12. TMR2H Timer 2 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCD; SFR Page = 0x00

Bit	Name	Function
7:0	TMR2H[7:0]	**Timer 2 High Byte.** In 16-bit mode, the TMR2H register contains the high byte of the 16-bit Timer 2. In 8-bit mode, TMR2H contains the 8-bit high byte timer value.

T3MH	T3XCLK	TMR3H Clock Source
0	0	SYSCLK/12
0	1	External Clock/8
1	X	SYSCLK

T3ML	T3XCLK	TMR3L Clock Source
0	0	SYSCLK/12
0	1	External Clock/8
1	X	SYSCLK

The TF3H bit is set when TMR3H overflows from 0xFF to 0x00; the TF3L bit is set when TMR3L overflows from 0xFF to 0x00. When Timer 3 interrupts are enabled, an interrupt is generated each time TMR3H overflows. If Timer 3 interrupts are enabled and TF3LEN (TMR3CN.5) is set, an interrupt is generated each time either TMR3L or TMR3H overflows. When TF3LEN is enabled, software must check the TF3H and TF3L flags to determine the source of the Timer 3 interrupt. The TF3H and TF3L interrupt flags are not cleared by hardware and must be manually cleared by software.

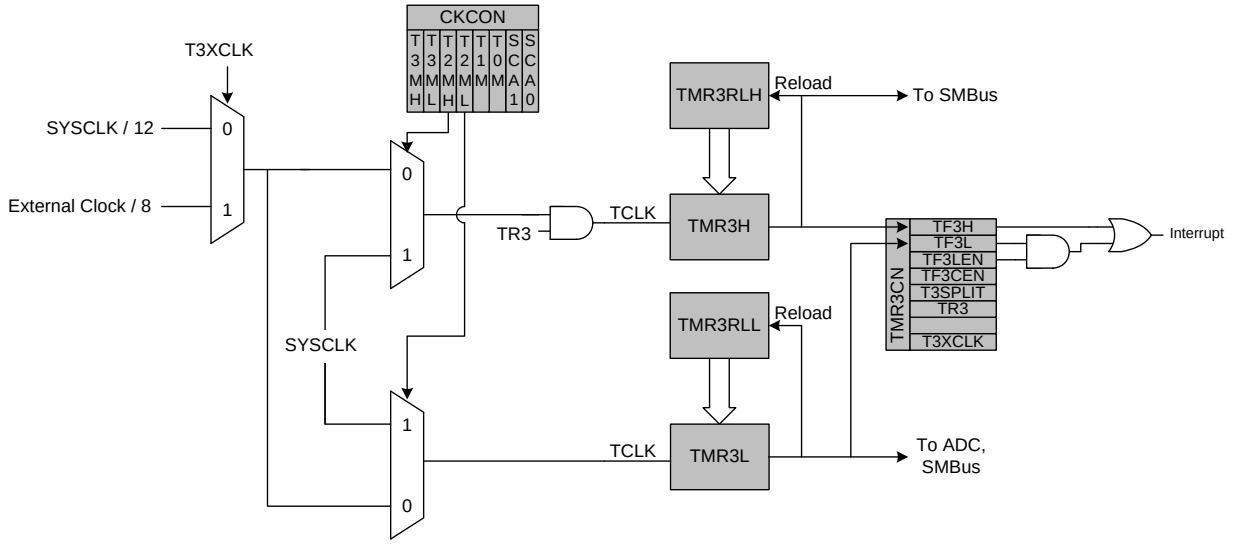


Figure 25.8. Timer 3 8-Bit Mode Block Diagram

25.3.3. External Oscillator Capture Mode

Capture Mode allows the external oscillator to be measured against the system clock. Timer 3 can be clocked from the system clock, or the system clock divided by 12, depending on the T3ML (CKCON.6), and T3XCLK bits. When a capture event is generated, the contents of Timer 3 (TMR3H:TMR3L) are loaded into the Timer 3 reload registers (TMR3RLH:TMR3RLL) and the TF3H flag is set. A capture event is generated by the falling edge of the clock source being measured, which is the external oscillator/8. By recording the difference between two successive timer capture values, the external oscillator frequency can be determined with respect to the Timer 3 clock. The Timer 3 clock should be much faster than the capture clock to achieve an accurate reading. Timer 3 should be in 16-bit auto-reload mode when using Capture Mode.

If the SYSCLK is 24 MHz and the difference between two successive captures is 5861, then the external clock frequency is as follows:

24 MHz/(5861/8) = 0.032754 MHz or 32.754 kHz

This mode allows software to determine the external oscillator frequency when an RC network or capacitor is used to generate the clock source.

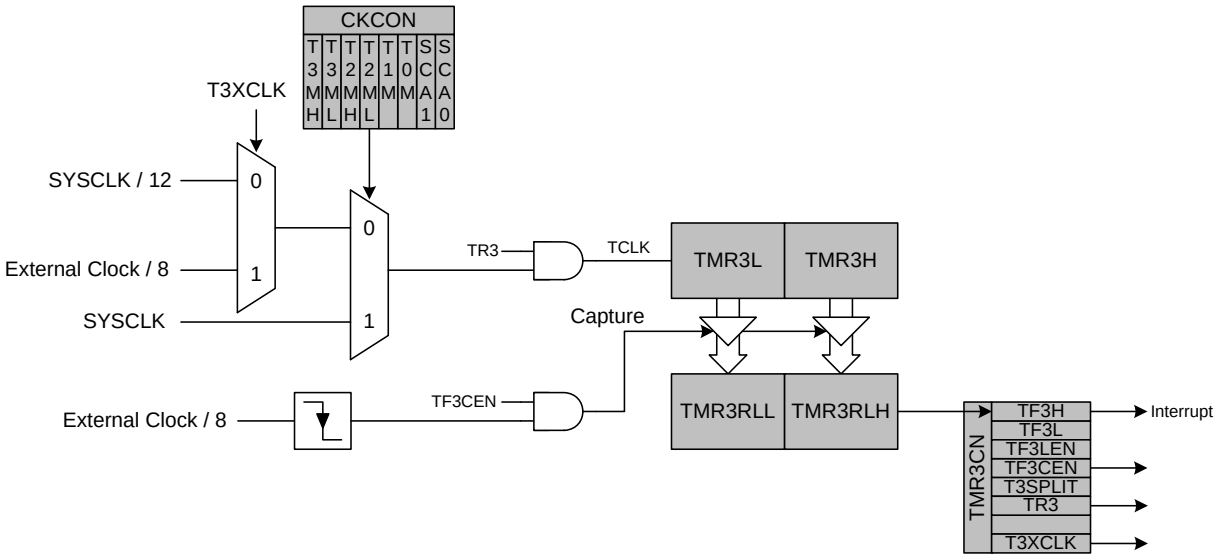


Figure 25.9. Timer 3 External Oscillator Capture Mode Block Diagram

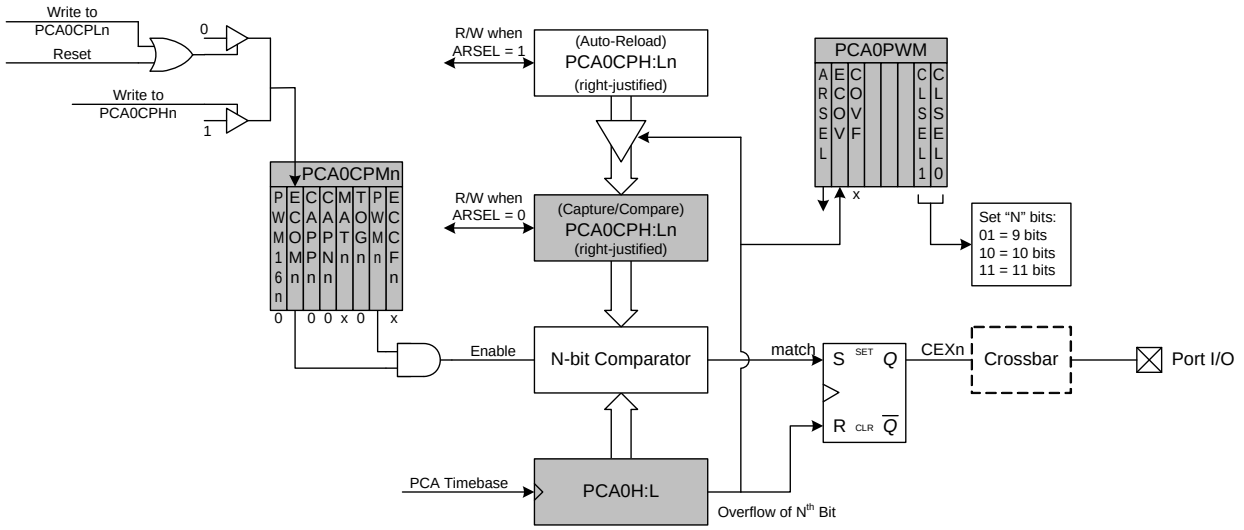


Figure 26.9. PCA 9, 10 and 11-Bit PWM Mode Diagram

26.3.6. 16-Bit Pulse Width Modulator Mode

A PCA module may also be operated in 16-Bit PWM mode. 16-bit PWM mode is independent of the other (8/9/10/11-bit) PWM modes. In this mode, the 16-bit capture/compare module defines the number of PCA clocks for the low time of the PWM signal. When the PCA counter matches the module contents, the output on CEXn is asserted high; when the 16-bit counter overflows, CEXn is asserted low. To output a varying duty cycle, new value writes should be synchronized with PCA CCFn match interrupts. 16-Bit PWM Mode is enabled by setting the ECOMn, PWMn, and PWM16n bits in the PCA0CPMn register. For a varying duty cycle, match interrupts should be enabled (ECCFn = 1 AND MATn = 1) to help synchronize the capture/compare register writes. If the MATn bit is set to 1, the CCFn flag for the module will be set each time a 16-bit comparator match (rising edge) occurs. The CF flag in PCA0CN can be used to detect the overflow (falling edge). The duty cycle for 16-Bit PWM Mode is given by Equation 26.4.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

$$\text{Duty Cycle} = \frac{(65536 - \text{PCA0CPn})}{65536}$$

Equation 26.4. 16-Bit PWM Duty Cycle

Using Equation 26.4, the largest duty cycle is 100% (PCA0CPn = 0), and the smallest duty cycle is 0.0015% (PCA0CPn = 0xFFFF). A 0% duty cycle may be generated by clearing the ECOMn bit to 0.

27. C2 Interface

C8051F55x/56x/57x devices include an on-chip Silicon Labs 2-Wire (C2) debug interface to allow Flash programming and in-system debugging with the production part installed in the end application. The C2 interface uses a clock signal (C2CK) and a bi-directional C2 data signal (C2D) to transfer information between the device and a host system. See the C2 Interface Specification for details on the C2 protocol.

27.1. C2 Interface Registers

The following describes the C2 registers necessary to perform Flash programming through the C2 interface. All C2 registers are accessed through the C2 interface as described in the C2 Interface Specification.

C2 Register Definition 27.1. C2ADD: C2 Address

Bit	7	6	5	4	3	2	1	0
Name	C2ADD[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function	
7:0	C2ADD[7:0]	**C2 Address.** The C2ADD register is accessed via the C2 interface to select the target Data register for C2 Data Read and Data Write commands.	
		Address	**Description**
		0x00	Selects the Device ID register for Data Read instructions
		0x01	Selects the Revision ID register for Data Read instructions
		0x02	Selects the C2 Flash Programming Control register for Data Read/Write instructions
		0xB4	Selects the C2 Flash Programming Data register for Data Read/Write instructions