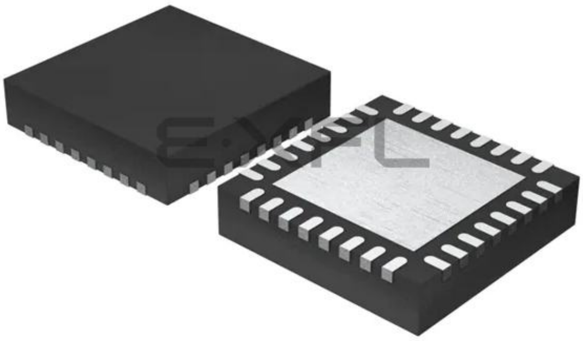
Silicon Labs - C8051F560-IMR Datasheet



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), CANbus, LINbus, SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	25
Program Memory Size	32KB (32K 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-VFQFN Exposed Pad
Supplier Device Package	32-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f560-imr

3. Pin Definitions

Table 3.1. Pin Definitions for the C8051F55x/56x/57x

Name	Pin	Pin	Pin	Type	Description
	40-pin packages	32-pin packages	24-pin packages		
VDD	4	4	3		Digital Supply Voltage. Must be connected.
GND	6	6	4		Digital Ground. Must be connected.
VDDA	5	5	—		Analog Supply Voltage. Must be connected.
GNDA	7	7	5		Analog Ground. Must be connected.
VREGIN	3	3	2		Voltage Regulator Input
VIO	2	2	1		Port I/O Supply Voltage. Must be connected.
$\overline{\text{RST}}$/ C2CK	10	10	8	D I/O D I/O	Device Reset. Open-drain output of internal POR or V_{DD} Monitor. Clock signal for the C2 Debug Interface.
P4.0/ C2D	9	—	—	D I/O or A In D I/O	Port 4.0. See SFR Definition 19.28. Bi-directional data signal for the C2 Debug Interface.
P3.0/ C2D		9	—	D I/O or A In D I/O	Port 3.0. See SFR Definition 19.24. Bi-directional data signal for the C2 Debug Interface.
P2.1/ C2D		—	7	D I/O or A In D I/O	Port 2.1. See SFR Definition 19.20. Bi-directional data signal for the C2 Debug Interface.
P0.0	8	8	6	D I/O or A In	Port 0.0. See SFR Definition 19.12.
P0.1	1	1	24	D I/O or A In	Port 0.1
P0.2	40	32	23	D I/O or A In	Port 0.2
P0.3	39	31	22	D I/O or A In	Port 0.3
P0.4	38	30	21	D I/O or A In	Port 0.4
P0.5	37	29	20	D I/O or A In	Port 0.5
P0.6	36	28	19	D I/O or A In	Port 0.6
P0.7	35	27	18	D I/O or A In	Port 0.7

Table 10.1. CIP-51 Instruction Set Summary

Mnemonic	Description	Bytes	Clock Cycles
Arithmetic Operations			
ADD A, Rn	Add register to A	1	1
ADD A, direct	Add direct byte to A	2	2
ADD A, @Ri	Add indirect RAM to A	1	2
ADD A, #data	Add immediate to A	2	2
ADDC A, Rn	Add register to A with carry	1	1
ADDC A, direct	Add direct byte to A with carry	2	2
ADDC A, @Ri	Add indirect RAM to A with carry	1	2
ADDC A, #data	Add immediate to A with carry	2	2
SUBB A, Rn	Subtract register from A with borrow	1	1
SUBB A, direct	Subtract direct byte from A with borrow	2	2
SUBB A, @Ri	Subtract indirect RAM from A with borrow	1	2
SUBB A, #data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	2
INC @Ri	Increment indirect RAM	1	2
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC direct	Decrement direct byte	2	2
DEC @Ri	Decrement indirect RAM	1	2
INC DPTR	Increment Data Pointer	1	1
MUL AB	Multiply A and B	1	4
DIV AB	Divide A by B	1	8
DA A	Decimal adjust A	1	1
Logical Operations			
ANL A, Rn	AND Register to A	1	1
ANL A, direct	AND direct byte to A	2	2
ANL A, @Ri	AND indirect RAM to A	1	2
ANL A, #data	AND immediate to A	2	2
ANL direct, A	AND A to direct byte	2	2
ANL direct, #data	AND immediate to direct byte	3	3
ORL A, Rn	OR Register to A	1	1
ORL A, direct	OR direct byte to A	2	2
ORL A, @Ri	OR indirect RAM to A	1	2
ORL A, #data	OR immediate to A	2	2
ORL direct, A	OR A to direct byte	2	2
ORL direct, #data	OR immediate to direct byte	3	3
XRL A, Rn	Exclusive-OR Register to A	1	1
XRL A, direct	Exclusive-OR direct byte to A	2	2
XRL A, @Ri	Exclusive-OR indirect RAM to A	1	2

Note: Certain instructions take a variable number of clock cycles to execute depending on instruction alignment and the FLRT setting (SFR Definition 14.3).

SFR Definition 12.3. SFRNEXT: SFR Next

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

SFR Address = 0x85; SFR Page = All Pages

Bit	Name	Function
7:0	SFRNEXT[7:0]	**SFR Page Bits.** This is the value that will go to the SFR Page register upon a return from interrupt. Write: Sets the SFR Page contained in the second byte of the SFR Stack. This will cause the SFRPAGE SFR to have this SFR page value upon a return from interrupt. Read: Returns the value of the SFR page contained in the second byte of the SFR stack. SFR page context is retained upon interrupts/return from interrupts in a 3 byte SFR Page Stack: SFRPAGE is the first entry, SFRNEXT is the second, and SFRLAST is the third entry. The SFR stack bytes may be used alter the context in the SFR Page Stack, and will not cause the stack to “push” or “pop”. Only interrupts and return from interrupts cause pushes and pops of the SFR Page Stack.

Table 12.2. Special Function Register (SFR) Memory Map for Page 0x0C

	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
F8			CAN0IF2DA2L	CAN0IF2DA2H	CAN0IF2DB1L	CAN0IF2DB1H	CAN0IF2DB2L	CAN0IF2DB2H
F0	B (All Pages)		CAN0IF2A2L	CAN0IF2A2H			CAN0IF2DA1L	CAN0IF2DA1H
E8			CAN0IF2M1L	CAN0IF2M1H	CAN0IF2M2L	CAN0IF2M2H	CAN0IF2A1L	CAN0IF2A1H
E0	ACC (All Pages)		CAN0IF2CML	CAN0IF2CMH			EIE1 (All Pages)	EIE2 (All Pages)
D8			CAN0IF1DB1L	CAN0IF1DB1H	CAN0IF1DB2L	CAN0IF1DB2H	CAN0IF2CRL	CAN0IF2CRH
D0	PSW (All Pages)		CAN0IF1MCL	CAN0IF1MCH	CAN0IF1DA1L	CAN0IF1DA1H	CAN0IF1DA2L	CAN0IF1DA2H
C8			CAN0IF1A1L	CAN0IF1A1H	CAN0IF1A2L	CAN0IF1A2H	CAN0IF2MCL	CAN0IF2MCH
C0	CAN0CN		CAN0IF1CML	CAN0IF1CMH	CAN0IF1M1L	CAN0IF1M1H	CAN0IF1M2L	CAN0IF1M2H
B8	IP (All Pages)		CAN0MV1L	CAN0MV1H	CAN0MV2L	CAN0MV2H	CAN0IF1CRL	CAN0IF1CRH
B0	P3 (All Pages)		CAN0IP2L	CAN0IP2H		P4 (All Pages)	FLSCL (All Pages)	FLKEY (All Pages)
A8	IE (All Pages)		CAN0ND1L	CAN0ND1H	CAN0ND2L	CAN0ND2H	CAN0IP1L	CAN0IP1H
A0	P2 (All Pages)	CAN0BRPE	CAN0TR1L	CAN0TR1H	CAN0TR2L	CAN0TR2H		SFRPAGE (All Pages)
98	SCON0 (All Pages)		CAN0BTL	CAN0BTH	CAN0IIDL	CAN0IIDH	CAN0TST	
90	P1 (All Pages)		CAN0CFG		CAN0STAT		CAN0ERRL	CAN0ERRH
88	TCON (All Pages)	TMOD (All Pages)	TL0 (All Pages)	TL1 (All Pages)	TH0 (All Pages)	TH1 (All Pages)	CKCON (All Pages)	
80	P0 (All Pages)	SP (All Pages)	DPL (All Pages)	DPH (All Pages)		SFRNEXT (All Pages)	SFRLAST (All Pages)	PCON (All Pages)
	0(8) (bit addressable)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

13.2. Interrupt Register Descriptions

The SFRs used to enable the interrupt sources and set their priority level are described in this section. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

SFR Definition 18.3. OSCICRS: Internal Oscillator Coarse Calibration

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

SFR Address = 0xA2; SFR Page = 0x0F

Bit	Name	Function
7	Unused	Read = 0; Write = Don’t Care
6:0	OSCICRS[6:0]	**Internal Oscillator Coarse Calibration Bits.** These bits determine the internal oscillator period. When set to 0000000b, the internal oscillator operates at its slowest setting. When set to 1111111b, the internal oscillator operates at its fastest setting. The reset value is factory calibrated to generate an internal oscillator frequency of 24 MHz.

SFR Definition 18.4. OSCIFIN: Internal Oscillator Fine Calibration

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

SFR Address = 0x9E; SFR Page = 0x0F

Bit	Name	Function
7:6	Unused	Read = 00b; Write = Don’t Care
5:0	OSCIFIN[5:0]	**Internal Oscillator Fine Calibration Bits.** These bits are fine adjustment for the internal oscillator period. The reset value is factory calibrated to generate an internal oscillator frequency of 24 MHz.

SFR Definition 19.2. XBR1: Port I/O Crossbar Register 1

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1E	T0E	ECIE	PCA0ME[2:0]			SYSCKE	Reserved
Type	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE2; SFR Page = 0x0F

Bit	Name	Function
7	T1E	**T1 Enable.** 0: T1 unavailable at Port pin. 1: T1 routed to Port pin.
6	T0E	**T0 Enable.** 0: T0 unavailable at Port pin. 1: T0 routed to Port pin.
5	ECIE	**PCA0 External Counter Input Enable.** 0: ECI unavailable at Port pin. 1: ECI routed to Port pin.
4:2	PCA0ME[2:0]	**PCA Module I/O Enable Bits.** 000: All PCA I/O unavailable at Port pins. 001: CEX0 routed to Port pin. 010: CEX0, CEX1 routed to Port pins. 011: CEX0, CEX1, CEX2 routed to Port pins. 100: CEX0, CEX1, CEX2, CEX3 routed to Port pins. 101: CEX0, CEX1, CEX2, CEX3, CEX4 routed to Port pins. 110: CEX0, CEX1, CEX2, CEX3, CEX4, CEX5 routed to Port pins. 111: RESERVED
1	SYSCKE	**/SYSCLK Output Enable.** 0: /SYSCLK unavailable at Port pin. 1: /SYSCLK output routed to Port pin.
0	Reserved	Always Write to 0.

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		

SFR Definition 19.23. P2SKIP: Port 2 Skip

Bit	7	6	5	4	3	2	1	0
Name	P2SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD6; SFR Page = 0x0F

Bit	Name	Function
7:0	P2SKIP[7:0]	**Port 2 Crossbar Skip Enable Bits.** These bits select Port 2 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar. 0: Corresponding P2.n pin is not skipped by the Crossbar. 1: Corresponding P2.n pin is skipped by the Crossbar.

Note: P2.2-P2.7 are available on 40-pin and 32-pin packages.

SFR Definition 19.24. P3: Port 3

Bit	7	6	5	4	3	2	1	0
Name	P3[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xB0; SFR Page = All Pages; Bit-Addressable

Bit	Name	Description	Write	Read
7:0	P3[7:0]	**Port 3 Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P3.n Port pin is logic LOW. 1: P3.n Port pin is logic HIGH.

Note: P3.0 is available on 40-pin and 32-pin packages. P3.1-P3.7 are available on 40-pin packages

$$\text{multiplier} = \frac{20000}{\text{baud\_rate}} - 1$$

$$\text{prescaler} = \ln\left[\frac{\text{SYSCLK}}{(\text{multiplier} + 1) \times \text{baud\_rate} \times 200}\right] \times \frac{1}{\ln 2} - 1$$

$$\text{divider} = \frac{\text{SYSCLK}}{(2^{(\text{prescaler} + 1)} \times (\text{multiplier} + 1) \times \text{baud\_rate})}$$

In all of these equations, the results must be rounded down to the nearest integer.

The following example shows the steps for calculating the baud rate values for a Master node running at 24 MHz and communicating at 19200 bits/sec. First, calculate the multiplier:

$$\text{multiplier} = \frac{20000}{19200} - 1 = 0.0417 \cong 0$$

Next, calculate the prescaler:

$$\text{prescaler} = \ln\frac{24000000}{(0 + 1) \times 19200 \times 200} \times \frac{1}{\ln 2} - 1 = 1.644 \cong 1$$

Finally, calculate the divider:

$$\text{divider} = \frac{24000000}{2^{(1 + 1)} \times (0 + 1) \times 19200} = 312.5 \cong 312$$

These values lead to the following baud rate:

$$\text{baud\_rate} = \frac{24000000}{2^{(1 + 1)} \times (0 + 1) \times 312} \cong 19230.77$$

The following code programs the interface in Master mode, using the Enhanced Checksum and enables the interface to operate at 19230 bits/sec using a 24 MHz system clock.

```
   LIN0CF   = 0x80;                        // Activate the interface
   LIN0CF  |= 0x40;                        // Set the node as a Master

   LIN0ADR  = 0x0D;                        // Point to the LIN0MUL register
   // Initialize the register (prescaler, multiplier and bit 8 of divider)
   LIN0DAT  = ( 0x01 << 6 ) + ( 0x00 << 1 ) + ( ( 0x138 & 0x0100 ) >> 8 );
   LIN0ADR   = 0x0C;                       // Point to the LIN0DIV register
   LIN0DAT   = (unsigned char)_0x138;      // Initialize LIN0DIV

   LIN0ADR   = 0x0B;                       // Point to the LIN0SIZE register
   LIN0DAT  |= 0x80;                       // Initialize the checksum as Enhanced

   LIN0ADR   = 0x08;                       // Point to LIN0CTRL register
   LIN0DAT   = 0x0C;                       // Reset any error and the interrupt
```

Table 20.2 includes the configuration values required for the typical system clocks and baud rates:

Table 21.2. Standard CAN Registers and Reset Values

CAN Addr.	Name	SFR Name (High)	SFR Addr.	SFR Name (Low)	SFR Addr.	16-bit SFR	Reset Value
0x50	IF2 Data A 2	CAN0IF2DA2H	0xFB	CAN0IF2DA2L	0xFA	CAN0IF2DA2	0x0000
0x52	IF2 Data B 1	CAN0IF2DB1H	0xFD	CAN0IF2DB1L	0xFC	CAN0IF2DB1	0x0000
0x54	IF2 Data B 2	CAN0IF2DB2H	0xFF	CAN0IF2DB2L	0xFE	CAN0IF2DB2	0x0000
0x80	Transmission Request 1[1]	CAN0TR1H	0xA3	CAN0TR1L	0xA2	CAN0TR1	0x0000
0x82	Transmission Request 2[1]	CAN0TR2H	0xA5	CAN0TR2L	0xA4	CAN0TR2	0x0000
0x90	New Data 1[1]	CAN0ND1H	0xAB	CAN0ND1L	0xAA	CAN0ND1	0x0000
0x92	New Data 2[1]	CAN0ND2H	0xAD	CAN0ND2L	0xAC	CAN0ND2	0x0000
0xA0	Interrupt Pending 1[1]	CAN0IP1H	0xAF	CAN0IP1L	0xAE	CAN0IP1	0x0000
0xA2	Interrupt Pending 2 [1]	CAN0IP2H	0xB3	CAN0IP2L	0xB2	CAN0IP2	0x0000
0xB0	Message Valid 1[1]	CAN0MV1H	0xBB	CAN0MV1L	0xBA	CAN0MV1	0x0000
0xB2	Message Valid 2[1]	CAN0MV2H	0xBD	CAN0MV2L	0xBC	CAN0MV2	0x0000

Notes:

1. Read-only register.
2. Write-enabled by CCE.
3. The reset value of CAN0TST could also be r0000000b, where r signifies the value of the CAN RX pin.
4. Write-enabled by Test.

SFR Definition 21.1. CAN0CFG: CAN Clock Configuration

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

SFR Address = 0x92; SFR Page = 0x0C

Bit	Name	Function
7:2	Unused	Read = 000000b; Write = Don’t Care.
1:0	SYSDIV[1:0]	**CAN System Clock Divider Bits.** The CAN controller clock is derived from the CIP-51 system clock. The CAN controller clock must be less than or equal to 25 MHz. 00: CAN controller clock = System Clock/1. 01: CAN controller clock = System Clock/2. 10: CAN controller clock = System Clock/4. 11: CAN controller clock = System Clock/8.

24.3. SPI0 Slave Mode Operation

When SPI0 is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPI0 logic counts SCK edges. When 8 bits have been shifted through the shift register, the SPIF flag is set to logic 1, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPI0DAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPI0DAT. Writes to SPI0DAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data, the SPI will load the shift register with the transmit buffer's contents after the last SCK edge of the next (or current) SPI transfer.

When configured as a slave, SPI0 can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPI0 is enabled when NSS is logic 0, and disabled when NSS is logic 1. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 24.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

3-wire slave mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. NSS is not used in this mode, and is not mapped to an external port pin through the crossbar. Since there is no way of uniquely addressing the device in 3-wire slave mode, SPI0 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI0 with the SPIEN bit. Figure 24.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

24.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

All of the following bits must be cleared by software.

1. The SPI Interrupt Flag, SPIF (SPI0CN.7) is set to logic 1 at the end of each byte transfer. This flag can occur in all SPI0 modes.
2. The Write Collision Flag, WCOL (SPI0CN.6) is set to logic 1 if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written.This flag can occur in all SPI0 modes.
3. The Mode Fault Flag MODF (SPI0CN.5) is set to logic 1 when SPI0 is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN and SPIEN bits in SPI0CN are set to logic 0 to disable SPI0 and allow another master device to access the bus.
4. The Receive Overrun Flag RXOVRN (SPI0CN.4) is set to logic 1 when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

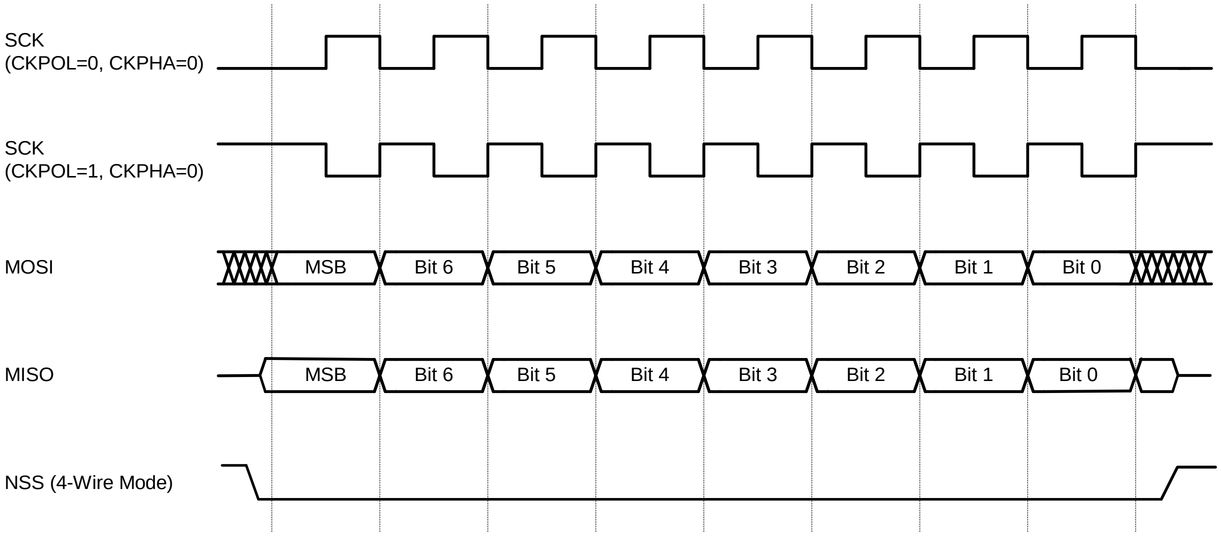


Figure 24.6. Slave Mode Data/Clock Timing (CKPHA = 0)

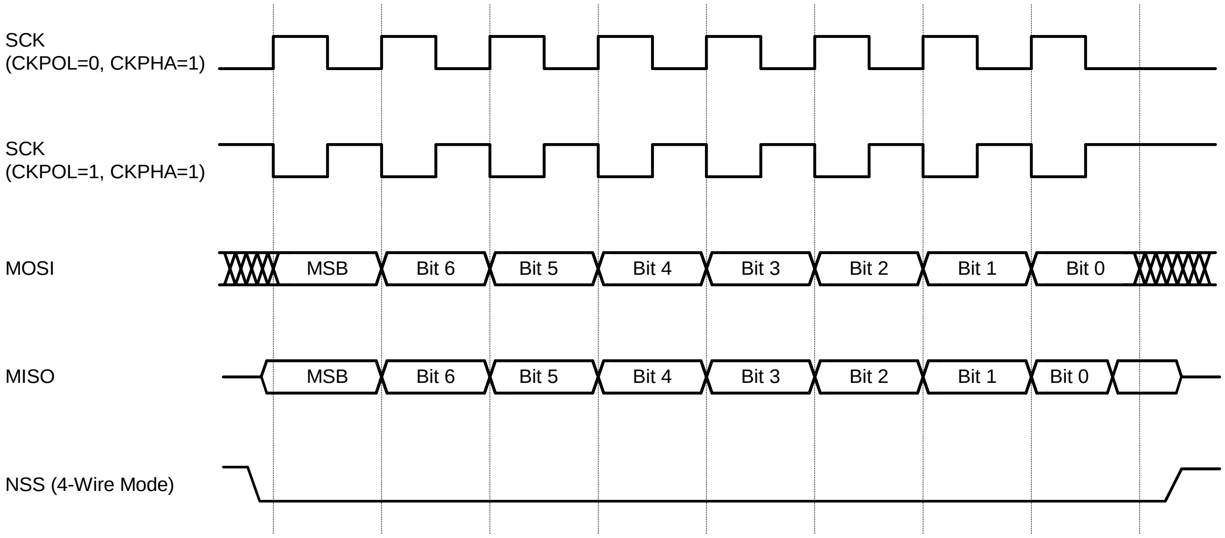


Figure 24.7. Slave Mode Data/Clock Timing (CKPHA = 1)

24.6. SPI Special Function Registers

SPI0 is accessed and controlled through four special function registers in the system controller: SPI0CN Control Register, SPI0DAT Data Register, SPI0CFG Configuration Register, and SPI0CKR Clock Rate Register. The four special function registers related to the operation of the SPI0 Bus are described in the following figures.

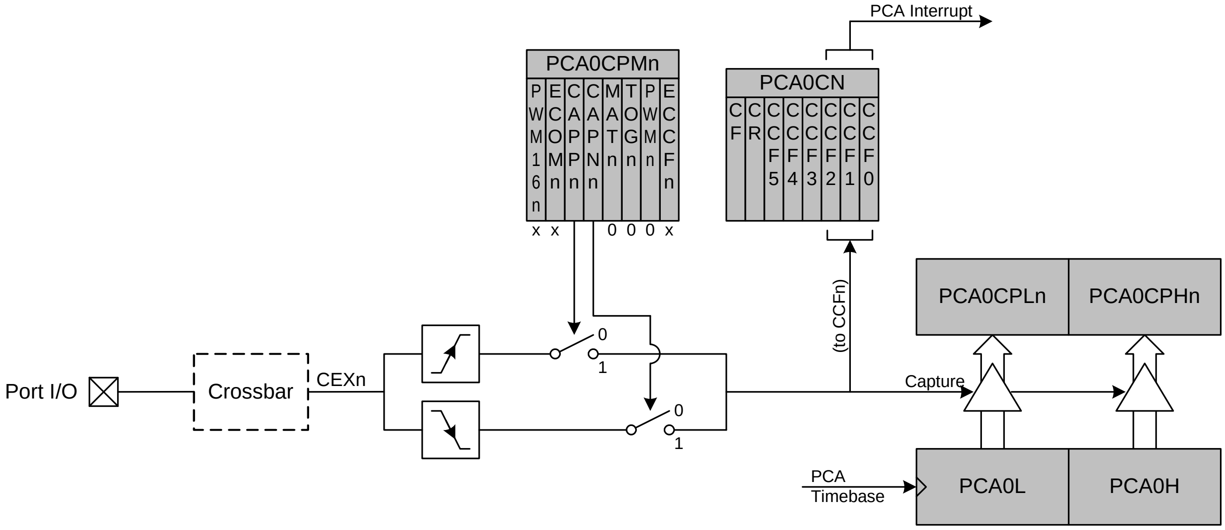


Figure 26.4. PCA Capture Mode Diagram

Note: The CEXn input signal must remain high or low for at least 2 system clock cycles to be recognized by the hardware.

26.3.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA counter/timer value is compared to the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

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.

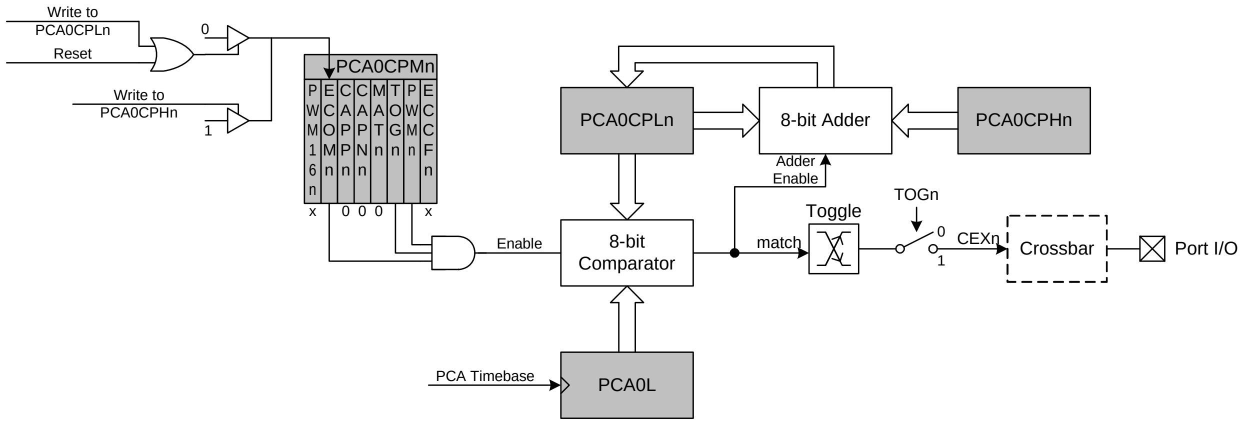


Figure 26.7. PCA Frequency Output Mode

26.3.5. 8-bit, 9-bit, 10-bit and 11-bit Pulse Width Modulator Modes

Each module can be used independently to generate a pulse width modulated (PWM) output on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA counter/timer, and the setting of the PWM cycle length (8, 9, 10 or 11-bits). For backwards-compatibility with the 8-bit PWM mode available on other devices, the 8-bit PWM mode operates slightly different than 9, 10 and 11-bit PWM modes. **It is important to note that all channels configured for 8/9/10/11-bit PWM mode will use the same cycle length.** It is not possible to configure one channel for 8-bit PWM mode and another for 11-bit mode (for example). However, other PCA channels can be configured to Pin Capture, High-Speed Output, Software Timer, Frequency Output, or 16-bit PWM mode independently.

26.3.5.1. 8-bit Pulse Width Modulator Mode

The duty cycle of the PWM output signal in 8-bit PWM mode is varied using the module's PCA0CPLn capture/compare register. When the value in the low byte of the PCA counter/timer (PCA0L) is equal to the value in PCA0CPLn, the output on the CEXn pin will be set. When the count value in PCA0L overflows, the CEXn output will be reset (see Figure 26.8). Also, when the counter/timer low byte (PCA0L) overflows from 0xFF to 0x00, PCA0CPLn is reloaded automatically with the value stored in the module’s capture/compare high byte (PCA0CPHn) without software intervention. Setting the ECOMn and PWMn bits in the PCA0CPMn register, and setting the CLSEL bits in register PCA0PWM to 00b enables 8-Bit Pulse Width Modulator mode. If the MATn bit is set to 1, the CCFn flag for the module will be set each time an 8-bit comparator match (rising edge) occurs. The COVF flag in PCA0PWM can be used to detect the overflow (falling edge), which will occur every 256 PCA clock cycles. The duty cycle for 8-Bit PWM Mode is given in Equation 26.2.

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.

$$\text{Duty Cycle} = \frac{(256 - \text{PCA0CPHn})}{256}$$

Equation 26.2. 8-Bit PWM Duty Cycle

Using Equation 26.2, the largest duty cycle is 100% (PCA0CPHn = 0), and the smallest duty cycle is 0.39% (PCA0CPHn = 0xFF). A 0% duty cycle may be generated by clearing the ECOMn bit to 0.

Table 26.3. Watchdog Timer Timeout Intervals[1]

System Clock (Hz)	PCA0CPL5	Timeout Interval (ms)
24,000,000	255	32.8
24,000,000	128	16.5
24,000,000	32	4.2
3,000,000	255	262.1
3,000,000	128	132.1
3,000,000	32	33.8
187,500[2]	255	4194
187,500[2]	128	2114
187,500[2]	32	541

Notes:

1. Assumes SYSCLK/12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.
2. Internal SYSCLK reset frequency = Internal Oscillator divided by 128.

SFR Definition 26.2. PCA0MD: PCA Mode

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CIDL	WDTE	WDLCK		CPS[2:0]			ECF
Type	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

SFR Address = 0xD9; SFR Page = 0x00

Bit	**Name**	**Function**
7	CIDL	**PCA Counter/Timer Idle Control.** Specifies PCA behavior when CPU is in Idle Mode. 0: PCA continues to function normally while the system controller is in Idle Mode. 1: PCA operation is suspended while the system controller is in Idle Mode.
6	WDTE	**Watchdog Timer Enable** If this bit is set, PCA Module 5 is used as the watchdog timer. 0: Watchdog Timer disabled. 1: PCA Module 5 enabled as Watchdog Timer.
5	WDLCK	**Watchdog Timer Lock** This bit locks/unlocks the Watchdog Timer Enable. When WDLCK is set, the Watchdog Timer may not be disabled until the next system reset. 0: Watchdog Timer Enable unlocked. 1: Watchdog Timer Enable locked.
4	Unused	Read = 0b, Write = Don't care.
3:1	CPS[2:0]	**PCA Counter/Timer Pulse Select.** These bits select the timebase source for the PCA counter 000: System clock divided by 12 001: System clock divided by 4 010: Timer 0 overflow 011: High-to-low transitions on ECI (max rate = system clock divided by 4) 100: System clock 101: External clock divided by 8 (synchronized with the system clock) 11x: Reserved
0	ECF	**PCA Counter/Timer Overflow Interrupt Enable.** This bit sets the masking of the PCA Counter/Timer Overflow (CF) interrupt. 0: Disable the CF interrupt. 1: Enable a PCA Counter/Timer Overflow interrupt request when CF (PCA0CN.7) is set.
Note:	When the WDTE bit is set to 1, the other bits in the PCA0MD register cannot be modified. To change the contents of the PCA0MD register, the Watchdog Timer must first be disabled.	