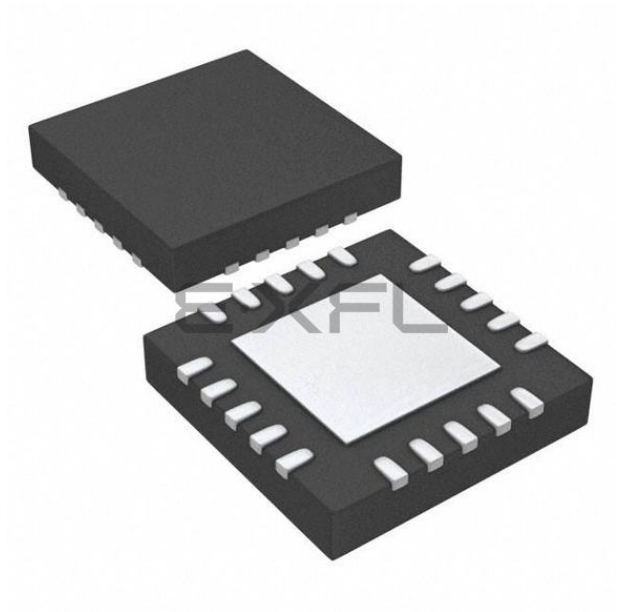

Welcome to E-XFL.COM

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, WDT
Number of I/O	17
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	20-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f399-a-gmr

List of Figures

3. C8051F33x Compatibility

The C8051F39x/37x family is designed to be a pin and code compatible replacement for the C8051F33x device family, with an enhanced feature set. The C8051F39x/37x device should function as a drop-in replacement for the C8051F33x devices in most applications. Table 3.1 lists recommended replacement part numbers for C8051F33x devices. See “3.1. Hardware Incompatibilities” to determine if any changes are necessary when upgrading an existing C8051F33x design to the C8051F39x/37x.

Table 3.1. C8051F33x Replacement Part Numbers

C8051F33x Part Number	C8051F39x/37x Part Number
C8051F330-GM	C8051F396-A-GM
C8051F331-GM	C8051F397-A-GM
C8051F332-GM	C8051F398-A-GM
C8051F333-GM	C8051F399-A-GM
C8051F334-GM	C8051F398-A-GM
C8051F335-GM	C8051F399-A-GM
C8051F336-GM	C8051F392-A-GM
C8051F337-GM	C8051F393-A-GM
C8051F338-GM	C8051F390-A-GM
C8051F339-GM	C8051F391-A-GM

3.1. Hardware Incompatibilities

While the C8051F39x/37x family includes a number of new features not found on the C8051F33x family, there are some differences that should be considered for any design port.

- **Internal High-Frequency Oscillator**: The undivided high-frequency oscillator on the C8051F39x/37x is 49 MHz, whereas the undivided high-frequency oscillator on the C8051F33x is 24.5 MHz. Correspondingly, the internal high frequency divide ratios (IFCN) have doubled. Thus, firmware written for the C8051F33x where the CLKSL[1:0] = 00b will result in the same SYSCLK frequency on the C8051F39x/37x.
- **Fabrication Technology**: The C8051F39x/37x is manufactured using a different technology process than the C8051F33x. As a result, many of the electrical performance parameters will have subtle differences. These differences should not affect most systems but it is nonetheless important to review the electrical parameters for any blocks that are used in the design, and ensure they are compatible with the existing hardware.
- **5 V Tolerance**: The port I/O pins on the C8501F39x/37x are not 5 V tolerant, whereas the port I/O pins on the C8051F33x are 5 V tolerant.
- **Lock Byte Address**: The lock byte for C8051F39x/7x devices with 16 kB of Flash resides at address 0x3FFF, whereas the lock byte for C8051F33x devices with 16 kB of Flash resides at address 0x3DFF. The lock byte for C8051F39x/7x devices with 8 kB of Flash resides at address 0x1FFF, whereas the lock byte for C8051F33x devices with 8 kB of Flash resides at address 0x1DFF.

Table 7.13. Voltage Reference Electrical Characteristics

V_{DD} = 3.0 V, –40 to +105 °C (C8051F39x), –40 to +85 °C (C8051F37x), unless otherwise specified.

Parameter	Test Condition	Min	Typ	Max	Unit
Internal Reference (REFBE = 1)					
Output Voltage	2.4 V Setting	2.37	2.4	2.43	V
	1.2 V Setting	1.18	1.2	1.22	V
VREF Short-Circuit Current		—	—	8	mA
VREF Temperature Coefficient		—	33	—	ppm/°C
Load Regulation	Load = 0 to 200 µA to AGND	—	6	—	µV/µA
VREF Turn-on Time 1	4.7 µF tantalum, 0.1 µF ceramic bypass	—	1.5	—	ms
VREF Turn-on Time 2	0.1 µF ceramic bypass	—	110	—	µs
Power Supply Rejection	2.4 V Setting	—	3.5	—	mV/V
	1.2 V Setting	—	1.1	—	mV/V
External Reference (REFBE = 0)					
Input Voltage Range		1.0	—	V_{DD}	V
Input Current	Sample Rate = 200 ksps; VREF = 3.0 V	—	3	—	µA
Power Specifications					
Supply Current	REFBE = “1” or TEMPE = “1”	—	70	100	µA

Table 7.14. Voltage Regulator Electrical Characteristics

V_{DD} = 3.0 V, –40 to +105 °C (C8051F39x), –40 to +85 °C (C8051F37x), unless otherwise specified.

Parameter	Test Condition	Min	Typ	Max	Unit
Output Voltage		1.73	1.78	1.83	V
Power Supply Sensitivity	Constant Temperature	—	0.5	—	%/V
Temperature Sensitivity	Constant Supply	—	55	—	ppm/°C

SFR Definition 9.1. ADC0CF: ADC0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	AD0SC[4:0]					AD0LJST		
Type	R/W					R/W	R/W	
Reset	1	1	1	1	1	0	0	0

SFR Address = 0xBC; SFR Page = All Pages

Bit	Name	Function
7:3	AD0SC[4:0]	**ADC0 SAR Conversion Clock Period Bits.** SAR Conversion clock is derived from system clock by the following equation, where *AD0SC* refers to the 5-bit value held in bits AD0SC4–0. SAR Conversion clock requirements are given in the ADC specification Table 7.10. $AD0SC = \frac{SYSCLK}{CLK_{SAR}} - 1$
2	AD0LJST	**ADC0 Left Justify Select.** 0: Data in ADC0H:ADC0L registers are right-justified. 1: Data in ADC0H:ADC0L registers are left-justified.
1:0	Reserved	Must Write 00b.

10.1. Calibration

The uncalibrated temperature sensor output is extremely linear and suitable for relative temperature measurements (see Table 7.11 on page 40 for specifications). For absolute temperature measurements, offset and/or gain calibration is recommended.

Figure 10.2 shows the typical temperature sensor error assuming a 1-point calibration at 0 °C. **Parameters that affect ADC measurement, in particular the voltage reference value, will also affect temperature measurement.**

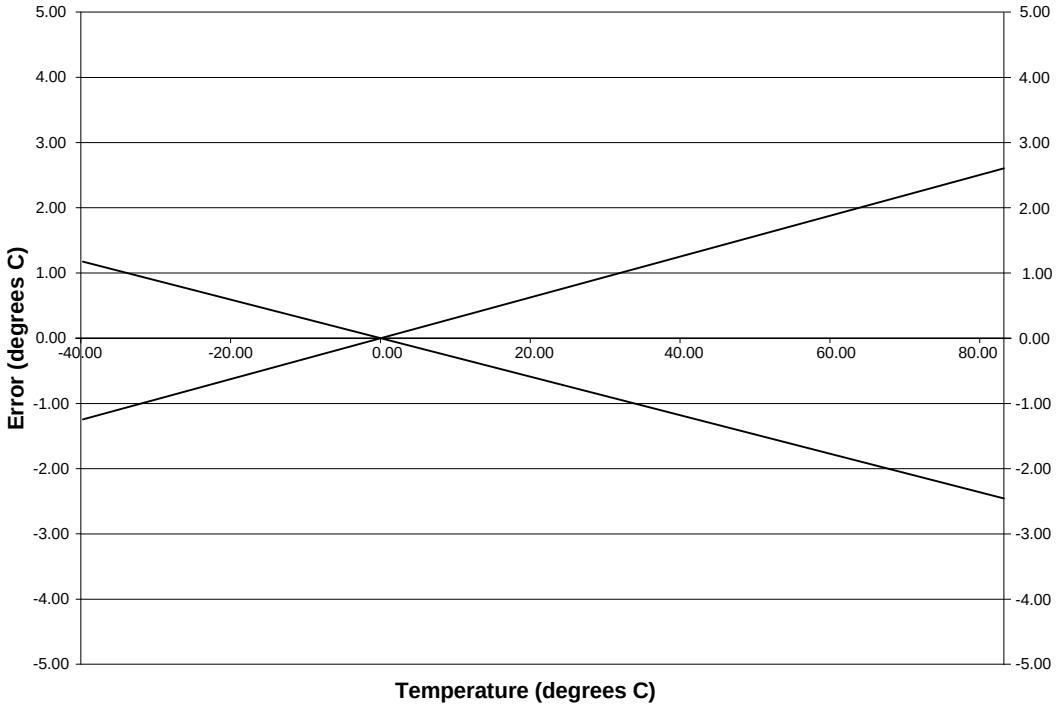

Figure 10.2. Temperature Sensor Error with 1-Point Calibration at 0 °C

13. Voltage Regulator

C8051F39x/37x devices include an internal regulator that regulates the internal core supply from a V_{DD} supply of 1.8 to 3.6 V. The regulator has two power-saving modes built in to help reduce current consumption in low-power applications. These modes are accessed through the REG0CN register.

13.1. Power Modes

Under default conditions, the internal regulator will remain on when the device enters STOP mode. This allows any enabled reset source to generate a reset for the device and bring the device out of STOP mode. For additional power savings, the STOPCF bit can be used to shut down the regulator and the internal power network of the device when the part enters STOP mode. When STOPCF is set to 1, the $\overline{RST}$ pin and a full power cycle of the device are the only methods of generating a reset.

SFR Definition 13.1. REG0CN: Voltage Regulator Control

Bit	7	6	5	4	3	2	1	0
Name					STOPCF			
Type	R/W				R/W	R/W		
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC9; SFR Page = All Pages

Bit	Name	Function
7:4	Reserved	Must Write 0000b.
3	STOPCF	**Stop Mode Configuration.** This bit configures the regulator's behavior when the device enters STOP mode. 0: Regulator is still active in STOP mode. Any enabled reset source will reset the device. 1: Regulator is shut down in STOP mode. Only the $\overline{RST}$ pin or power cycle can reset the device.
2:0	Reserved	Must Write 000b.

14. Comparator0

C8051F39x/37x devices include an on-chip programmable voltage comparator, Comparator0, shown in Figure 14.1.

The Comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a synchronous “latched” output (CP0), or an asynchronous “raw” output (CP0A). The asynchronous CP0A signal is available even when the system clock is not active. This allows the Comparator to operate and generate an output with the device in STOP mode. When assigned to a Port pin, the Comparator output may be configured as open drain or push-pull (see Section “27.4. Port I/O Initialization” on page 180). Comparator0 may also be used as a reset source (see Section “24.5. Comparator0 Reset” on page 159), or as a trigger to kill a PCA output channel.

The Comparator0 inputs are selected by the comparator input multiplexer, as detailed in Section “14.1. Comparator Multiplexer” on page 80.

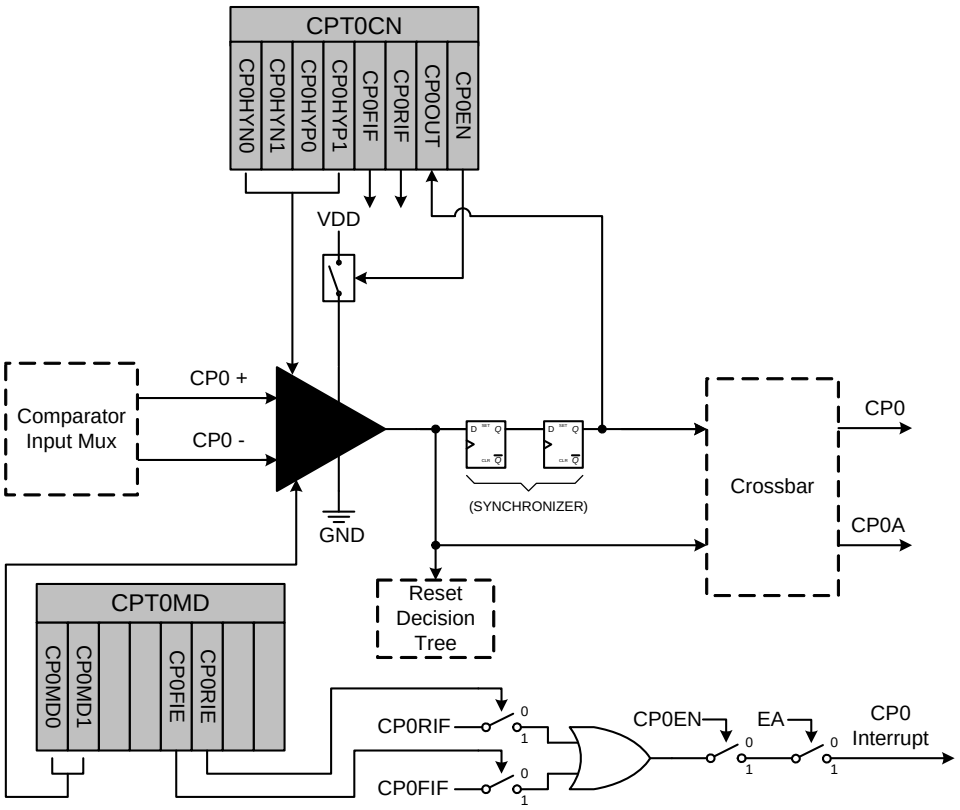

Figure 14.1. Comparator0 Functional Block Diagram

The Comparator output can be polled in software, used as an interrupt source, and/or routed to a Port pin. When routed to a Port pin, the Comparator output is available asynchronous or synchronous to the system clock; the asynchronous output is available even in STOP mode (with no system clock active). When disabled, the Comparator output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, and the power supply to the comparator is turned off. See Section “27.3. Priority Crossbar Decoder” on page 178 for details on configuring Comparator outputs via the digital Crossbar. Comparator inputs can be externally driven from –0.25 V to (V_{DD}) + 0.25 V without damage or upset. The complete Comparator electrical specifications are given in Section “7. Electrical Characteristics” on page 32.

Table 15.1. 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
XRL A, #data	Exclusive-OR immediate to A	2	2
XRL direct, A	Exclusive-OR A to direct byte	2	2

Table 15.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
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	2
JC rel	Jump if Carry is set	2	2/4
JNC rel	Jump if Carry is not set	2	2/4
JB bit, rel	Jump if direct bit is set	3	3/5
JNB bit, rel	Jump if direct bit is not set	3	3/5
JBC bit, rel	Jump if direct bit is set and clear bit	3	3/5
Program Branching			
ACALL addr11	Absolute subroutine call	2	4*
LCALL addr16	Long subroutine call	3	5*
RET	Return from subroutine	1	6*
RETI	Return from interrupt	1	6*
AJMP addr11	Absolute jump	2	4*
LJMP addr16	Long jump	3	5*
SJMP rel	Short jump (relative address)	2	4*
JMP @A+DPTR	Jump indirect relative to DPTR	1	4*
JZ rel	Jump if A equals zero	2	2/4*
JNZ rel	Jump if A does not equal zero	2	2/4*
CJNE A, direct, rel	Compare direct byte to A and jump if not equal	3	4/6*
CJNE A, #data, rel	Compare immediate to A and jump if not equal	3	3/5*
CJNE Rn, #data, rel	Compare immediate to Register and jump if not equal	3	3/5*
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	4/6*
DJNZ Rn, rel	Decrement Register and jump if not zero	2	2/4*
DJNZ direct, rel	Decrement direct byte and jump if not zero	3	3/5*
NOP	No operation	1	1
* Clock cycles for branch instructions with prefetch enabled, Align = 0, FLRT = 0			

SFR Definition 20.3. IPH: Interrupt Priority High

Bit	7	6	5	4	3	2	1	0
Name		PHSPI0	PHT2	PHS0	PHT1	PHX1	PHT0	PHX0
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

SFR Address = 0x84; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7	Unused	Read = 1, Write = Don't Care.
6	PHSPI0	**Serial Peripheral Interface (SPI0) Interrupt Priority Control MSB.** This bit sets the MSB of the priority field for the SPI0 interrupt.
5	PHT2	**Timer 2 Interrupt Priority Control MSB.** This bit sets the MSB of the priority field for the Timer 2 interrupt.
4	PHS0	**UART0 Interrupt Priority Control MSB.** This bit sets the MSB of the priority field for the UART0 interrupt.
3	PHT1	**Timer 1 Interrupt Priority Control MSB.** This bit sets the MSB of the priority field for the Timer 1 interrupt.
2	PHX1	**External Interrupt 1 Priority Control MSB.** This bit sets the MSB of the priority field for the External Interrupt 1 interrupt.
1	PHT0	**Timer 0 Interrupt Priority Control MSB.** This bit sets the MSB of the priority field for the Timer 0 interrupt.
0	PHX0	**External Interrupt 0 Priority Control MSB.** This bit sets the MSB of the priority field for the External Interrupt 0 interrupt.

21.4. Flash Write and Erase Guidelines

Any system which contains routines which write or erase Flash memory from software involves some risk that the write or erase routines will execute unintentionally if the CPU is operating outside its specified operating range of V_{DD}, system clock frequency, or temperature. This accidental execution of Flash modifying code can result in alteration of Flash memory contents causing a system failure that is only recoverable by re-Flashing the code in the device. The following guidelines are recommended for any system which contains routines which write or erase Flash from code.

21.4.1. V_{DD} Maintenance and the V_{DD} Monitor

1. If the system power supply is subject to voltage or current "spikes," add sufficient transient protection devices to the power supply to ensure that the supply voltages listed in the Absolute Maximum Ratings table are not exceeded.
2. Make certain that the minimum V_{DD} rise time specification of 1 ms is met. If the system cannot meet this rise time specification, then add an external V_{DD} brownout circuit to the $\overline{RST}$ pin of the device that holds the device in reset until V_{DD} reaches 2.7 V and re-asserts $\overline{RST}$ if V_{DD} drops below 2.7 V.
3. Enable the on-chip V_{DD} monitor and enable the V_{DD} monitor as a reset source as early in code as possible. This should be the first set of instructions executed after the Reset Vector. For 'C'-based systems, this will involve modifying the startup code added by the C compiler. See your compiler documentation for more details. Make certain that there are no delays in software between enabling the V_{DD} monitor and enabling the V_{DD} monitor as a reset source. Code examples showing this can be found in “AN201: Writing to Flash from Firmware", available from the Silicon Laboratories web site.
4. As an added precaution, explicitly enable the V_{DD} monitor and enable the V_{DD} monitor as a reset source inside the functions that write and erase Flash memory. The V_{DD} monitor enable instructions should be placed just after the instruction to set PSWE to a 1, but before the Flash write or erase operation instruction.
5. Make certain that all writes to the RSTSRC (Reset Sources) register use direct assignment operators and explicitly DO NOT use the bit-wise operators (such as AND or OR). For example, "RSTSRC = 0x02" is correct. "RSTSRC |= 0x02" is incorrect.
6. Make certain that all writes to the RSTSRC register explicitly set the PORSF bit to a '1'. Areas to check are initialization code which enables other reset sources, such as the Missing Clock Detector or Comparator, for example, and instructions which force a Software Reset. A global search on "RSTSRC" can quickly verify this.

21.4.2. PSWE Maintenance

7. Reduce the number of places in code where the PSWE bit (b0 in PSCTL) is set to a '1'. There should be exactly one routine in code that sets PSWE to a '1' to write Flash bytes and one routine in code that sets PSWE and PSEE both to a '1' to erase Flash pages.
8. Minimize the number of variable accesses while PSWE is set to a '1'. Handle pointer address updates and loop variable maintenance outside the "PSWE = 1;... PSWE = 0;" area. Code examples showing this can be found in “AN201: Writing to Flash from Firmware", available from the Silicon Laboratories web site.
9. Disable interrupts prior to setting PSWE to a '1' and leave them disabled until after PSWE has been reset to 0. Any interrupts posted during the Flash write or erase operation will be serviced in priority order after the Flash operation has been completed and interrupts have been re-enabled by software.
10. Make certain that the Flash write and erase pointer variables are not located in XRAM. See your compiler documentation for instructions regarding how to explicitly locate variables in different memory areas.
11. Add address bounds checking to the routines that write or erase Flash memory to ensure that a routine called with an illegal address does not result in modification of the Flash.

27.3. Priority Crossbar Decoder

The Priority Crossbar Decoder (Figure 27.3) assigns a priority to each I/O function, starting at the top with UART0. When a digital resource is selected, the least-significant unassigned Port pin is assigned to that resource (excluding UART0, which is always at pins 4 and 5). If a Port pin is assigned, the Crossbar skips that pin when assigning the next selected resource. Additionally, the Crossbar will skip Port pins whose associated bits in the PnSKIP registers are set. The PnSKIP registers allow software to skip Port pins that are to be used for analog input, dedicated functions, or GPIO.

Important Note on Crossbar Configuration: If a Port pin is claimed by a peripheral without use of the Crossbar, its corresponding PnSKIP bit should be set. This applies to P0.0 if VREF is used, P0.3 and/or P0.2 if the external oscillator circuit is enabled, P0.6 if the ADC or IDAC is configured to use the external conversion start signal (CNVSTR), and any selected ADC or Comparator inputs. The Crossbar skips selected pins as if they were already assigned, and moves to the next unassigned pin.

Figure 27.3 shows all of the potential peripheral-to-pin assignments available to the crossbar. Note that this does not mean any peripheral can always be assigned to the highlighted pins. The actual pin assignments are determined by the priority of the enabled peripherals.

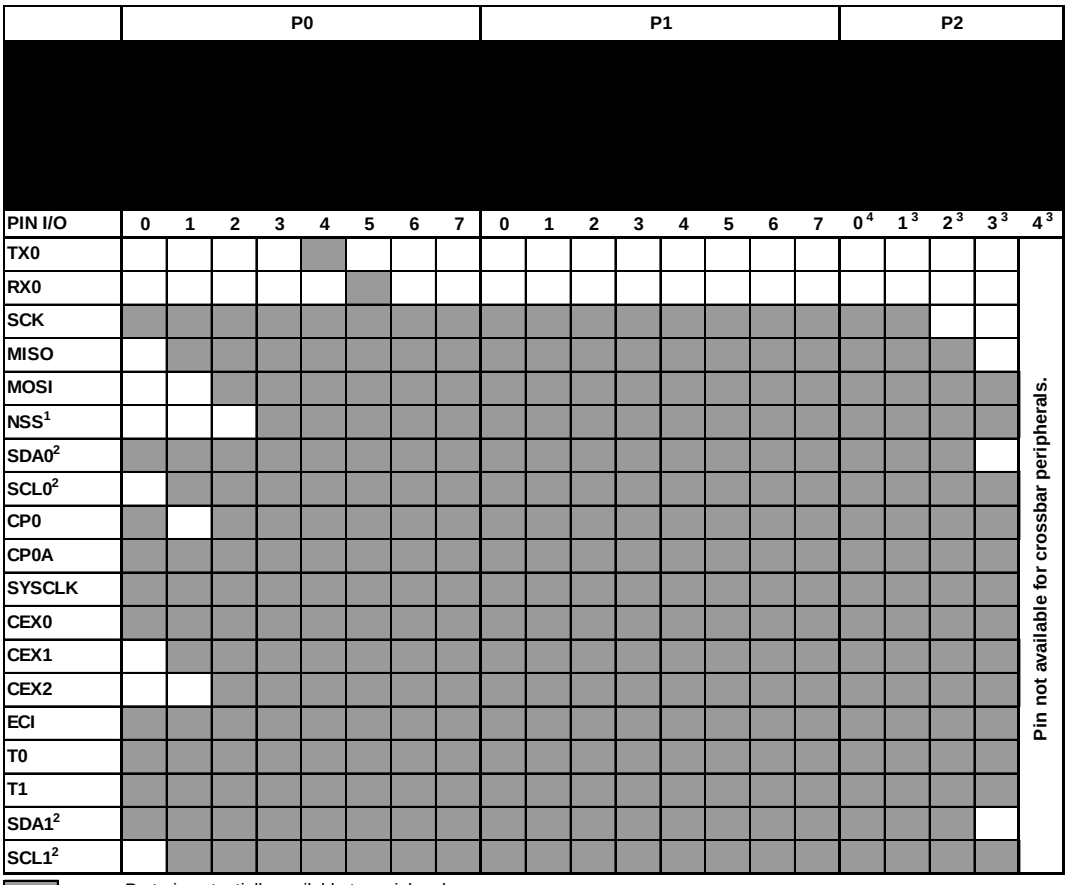

Figure 27.3. Crossbar Priority Decoder - Possible Pin Assignments

SFR Definition 28.2. SMB1CF: SMBus Clock/Configuration

Bit	7	6	5	4	3	2	1	0
Name	ENSMB1	INH1	BUSY1	EXTHOLD1	SMB1TOE	SMB1FTE	SMB1CS[1:0]	
Type	R/W	R/W	R	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC1; SFR Page = F

Bit	Name	Function
7	ENSMB1	**SMBus1 Enable.** This bit enables the SMBus1 interface when set to 1. When enabled, the interface constantly monitors the SDA1 and SCL1 pins.
6	INH1	**SMBus1 Slave Inhibit.** When this bit is set to logic 1, the SMBus1 does not generate an interrupt when slave events occur. This effectively removes the SMBus1 slave from the bus. Master Mode interrupts are not affected.
5	BUSY1	**SMBus1 Busy Indicator.** This bit is set to logic 1 by hardware when a transfer is in progress. It is cleared to logic 0 when a STOP or free-timeout is sensed.
4	EXTHOLD1	**SMBus1 Setup and Hold Time Extension Enable.** This bit controls the SDA1 setup and hold times according to Table 28.2. 0: SDA1 Extended Setup and Hold Times disabled. 1: SDA1 Extended Setup and Hold Times enabled.
3	SMB1TOE	**SMBus1 SCL Timeout Detection Enable.** This bit enables SCL low timeout detection. If set to logic 1, the SMBus1 forces Timer 4 to reload while SCL1 is high and allows Timer 4 to count when SCL1 goes low. If Timer 4 is configured to Split Mode, only the High Byte of the timer is held in reload while SCL1 is high. Timer 4 should be programmed to generate interrupts at 25 ms, and the Timer 4 interrupt service routine should reset SMBus1 communication.
2	SMB1FTE	**SMBus1 Free Timeout Detection Enable.** When this bit is set to logic 1, the bus will be considered free if SCL1 and SDA1 remain high for more than 10 SMBus clock source periods.
1:0	SMB1CS[1:0]	**SMBus1 Clock Source Selection.** These two bits select the SMBus1 clock source, which is used to generate the SMBus1 bit rate. The selected device should be configured according to Equation 28.1. 00: Timer 0 Overflow 01: Timer 5 Overflow 10: Timer 2 High Byte Overflow 11: Timer 2 Low Byte Overflow

28.4.6. Data Register

The SMBus Data register SMBnDAT holds a byte of serial data to be transmitted or one that has just been received. Software may safely read or write to the data register when the SIn flag is set. Software should not attempt to access the SMBnDAT register when the SMBus is enabled and the SIn flag is cleared to logic 0, as the interface may be in the process of shifting a byte of data into or out of the register.

Data in SMBnDAT is always shifted out MSB first. After a byte has been received, the first bit of received data is located at the MSB of SMBnDAT. While data is being shifted out, data on the bus is simultaneously being shifted in. SMBnDAT always contains the last data byte present on the bus. In the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data or address in SMBnDAT.

SFR Definition 28.10. SMB0DAT: SMBus Data

Bit	7	6	5	4	3	2	1	0
Name	SMB0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC2; SFR Page = 0

Bit	Name	Function
7:0	SMB0DAT[7:0]	**SMBus0 Data.** The SMB0DAT register contains a byte of data to be transmitted on the SMBus0 serial interface or a byte that has just been received on the SMBus0 serial interface. The CPU can read from or write to this register whenever the SI0 serial interrupt flag (SMB0CN.0) is set to logic 1. The serial data in the register remains stable as long as the SI0 flag is set. When the SI0 flag is not set, the system may be in the process of shifting data in/out and the CPU should not attempt to access this register.

Table 28.5. SMBus Status Decoding: Hardware ACK Disabled (EHACK = 0)

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

29.1. Enhanced Baud Rate Generation

The UART0 baud rate is generated by Timer 1 in 8-bit auto-reload mode. The TX clock is generated by TL1; the RX clock is generated by a copy of TL1 (shown as RX Timer in Figure 29.2), which is not user-accessible. Both TX and RX Timer overflows are divided by two to generate the TX and RX baud rates. The RX Timer runs when Timer 1 is enabled, and uses the same reload value (TH1). However, an RX Timer reload is forced when a START condition is detected on the RX pin. This allows a receive to begin any time a START is detected, independent of the TX Timer state.

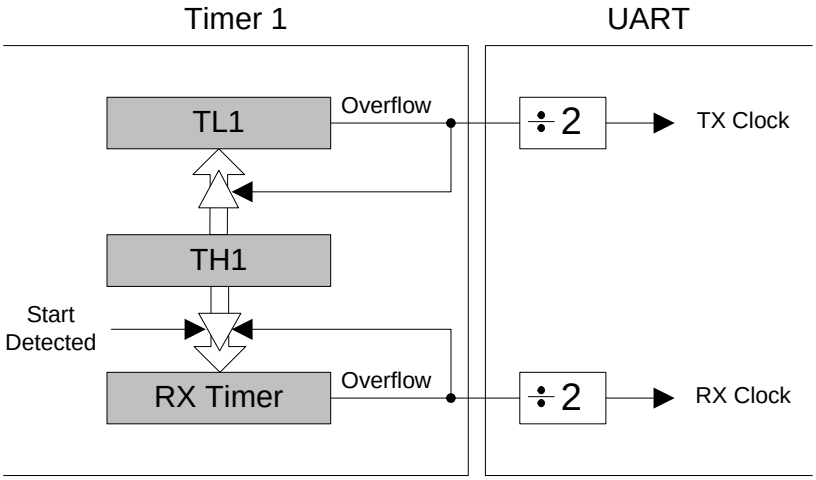

Figure 29.2. UART0 Baud Rate Logic

Timer 1 should be configured for Mode 2, 8-bit auto-reload (see Section “31.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload” on page 247). The Timer 1 reload value should be set so that overflows will occur at two times the desired UART baud rate frequency. Note that Timer 1 may be clocked by one of six sources: SYSCLK, SYSCLK/4, SYSCLK/12, SYSCLK/48, the external oscillator clock/8, or an external input T1. For any given Timer 1 clock source, the UART0 baud rate is determined by Equation 29.1-A and Equation 29.1-B.

A) $$\text{UartBaudRate} = \frac{1}{2} \times \text{T1\_Overflow\_Rate}$$

B) $$\text{T1\_Overflow\_Rate} = \frac{\text{T1}_{\text{CLK}}}{256 - \text{TH1}}$$

Equation 29.1. UART0 Baud Rate

Where $T1_{CLK}$ is the frequency of the clock supplied to Timer 1, and *T1H* is the high byte of Timer 1 (reload value). Timer 1 clock frequency is selected as described in Section “31. Timers” on page 242. A quick reference for typical baud rates and system clock frequencies is given in Table 29.1 through Table 29.2. The internal oscillator may still generate the system clock when the external oscillator is driving Timer 1.

29.2.2. 9-Bit UART

9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB80 (SCON0.3), which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB80 (SCON0.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: (1) RI0 must be logic 0, and (2) if MCE0 is logic 1, the 9th bit must be logic 1 (when MCE0 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80, and the RI0 flag is set to 1. If the above conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set to 1. A UART0 interrupt will occur if enabled when either TI0 or RI0 is set to 1.

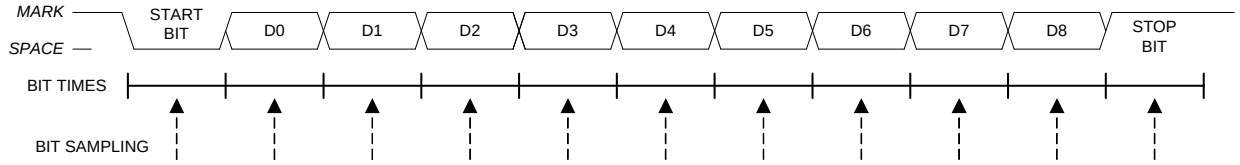

Figure 29.5. 9-Bit UART Timing Diagram

30.2. SPI0 Master Mode Operation

A SPI master device initiates all data transfers on a SPI bus. SPI0 is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.6). Writing a byte of data to the SPI0 data register (SPI0DAT) when in master mode writes to the transmit buffer. If the SPI shift register is empty, the byte in the transmit buffer is moved to the shift register, and a data transfer begins. The SPI0 master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set. While the SPI0 master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPI0DAT.

When configured as a master, SPI0 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI0 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI0CN.6) and SPIEN (SPI0CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI0CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI0 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 30.2 shows a connection diagram between two master devices in multiple-master mode.

3-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. In this mode, NSS is not used, and is not mapped to an external port pin through the crossbar. Any slave devices that must be addressed in this mode should be selected using general-purpose I/O pins. Figure 30.3 shows a connection diagram between a master device in 3-wire master mode and a slave device.

4-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 1. In this mode, NSS is configured as an output pin, and can be used as a slave-select signal for a single SPI device. In this mode, the output value of NSS is controlled (in software) with the bit NSSMD0 (SPI0CN.2). Additional slave devices can be addressed using general-purpose I/O pins. Figure 30.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

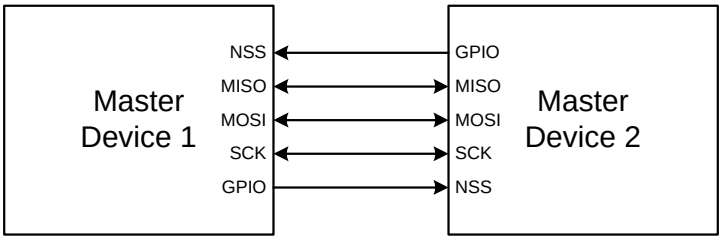

Figure 30.2. Multiple-Master Mode Connection Diagram

SFR Definition 31.10. TMR2RLL: Timer 2 Reload Register Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2RLL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCA; SFR Page = 0

Bit	Name	Function
7:0	TMR2RLL[7:0]	**Timer 2 Reload Register Low Byte.** TMR2RLL holds the low byte of the reload value for Timer 2.

SFR Definition 31.11. TMR2RLH: Timer 2 Reload Register High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2RLH[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCB; SFR Page = 0

Bit	Name	Function
7:0	TMR2RLH[7:0]	**Timer 2 Reload Register High Byte.** TMR2RLH holds the high byte of the reload value for Timer 2.

SFR Definition 31.12. 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 = 0

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 32.3. PCA0PWM: PCA PWM Configuration

Bit	7	6	5	4	3	2	1	0
Name	ARSEL	ECOV	COVF				CLSEL[1:0]	
Type	R/W	R/W	R/W	R	R	R	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xF7; SFR Page = All Pages

Bit	Name	Function
7	ARSEL	**Auto-Reload Register Select.** This bit selects whether to read and write the normal PCA capture/compare registers (PCA0CPn), or the Auto-Reload registers at the same SFR addresses. This function is used to define the reload value for 9, 10, and 11-bit PWM modes. In all other modes, the Auto-Reload registers have no function. 0: Read/Write Capture/Compare Registers at PCA0CPHn and PCA0CPLn. 1: Read/Write Auto-Reload Registers at PCA0CPHn and PCA0CPLn.
6	ECOV	**Cycle Overflow Interrupt Enable.** This bit sets the masking of the Cycle Overflow Flag (COVF) interrupt. 0: COVF will not generate PCA interrupts. 1: A PCA interrupt will be generated when COVF is set.
5	COVF	**Cycle Overflow Flag.** This bit indicates an overflow of the 8th, 9th, 10th, or 11th bit of the main PCA counter (PCA0). The specific bit used for this flag depends on the setting of the Cycle Length Select bits. The bit can be set by hardware or software, but must be cleared by software. 0: No overflow has occurred since the last time this bit was cleared. 1: An overflow has occurred since the last time this bit was cleared.
4:2	Unused	Unused. Read = 000b; Write = Don't care.
1:0	CLSEL[1:0]	**Cycle Length Select.** When 16-bit PWM mode is not selected, these bits select the length of the PWM cycle, between 8, 9, 10, or 11 bits. This affects all channels configured for PWM which are not using 16-bit PWM mode. These bits are ignored for individual channels configured to16-bit PWM mode. 00: 8 bits. 01: 9 bits. 10: 10 bits. 11: 11 bits.