



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

Product Status	Active
Core Processor	CIP-51 8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	16
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-UFQFN Exposed Pad
Supplier Device Package	20-QFN (3x3)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f982-c-gmr

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

# List of Tables

Table 2.1. Product Selection Guide	31
Table 3.1. Pin Definitions for the C8051F99x-C8051F98x	32
Table 3.2. QFN-20 Package Dimensions	40
Table 3.3. PCB Land Pattern	42
Table 3.4. QFN-24 Package Dimensions	43
Table 3.5. PCB Land Pattern	45
Table 3.6. QSOP-24 Package Dimensions	46
Table 3.7. PCB Land Pattern	47
Table 4.1. Absolute Maximum Ratings	48
Table 4.2. Global Electrical Characteristics	49
Table 4.3. Port I/O DC Electrical Characteristics	54
Table 4.4. Reset Electrical Characteristics	57
Table 4.5. Power Management Electrical Specifications	58
Table 4.6. Flash Electrical Characteristics	58
Table 4.7. Internal Precision Oscillator Electrical Characteristics	58
Table 4.8. Internal Low-Power Oscillator Electrical Characteristics	58
Table 4.9. SmaRTClock Characteristics	59
Table 4.10. ADC0 Electrical Characteristics	59
Table 4.11. Temperature Sensor Electrical Characteristics	60
Table 4.12. Voltage Reference Electrical Characteristics	61
Table 4.13. IREF0 Electrical Characteristics	62
Table 4.14. Comparator Electrical Characteristics	63
Table 4.15. VREG0 Electrical Characteristics	64
Table 4.16. Capacitive Sense Electrical Characteristics	65
Table 5.1. Representative Conversion Times and Energy Consumption for the SAF	R
ADC with 1.65 V High-Speed VREF	73
Table 8.1. Operation with Auto-scan and Accumulate    1	105
Table 9.1. CIP-51 Instruction Set Summary    1	121
Table 12.1. Special Function Register (SFR) Memory Map (Page 0x0)         1	132
Table 12.2. Special Function Register (SFR) Memory Map (Page 0xF)         1	133
Table 12.3. Special Function Registers    1	134
Table 13.1. Interrupt Summary    1	140
Table 14.1. Flash Security Summary    1	153
Table 15.1. Power Modes         1	162
Iable 16.1. Example 16-bit CRC Outputs    1	173
Table 19.1. Recommended XFCN Settings for Crystal Mode    1	190
Table 19.2. Recommended XFCN Settings for RC and C modes	191
Table 20.1. SmaRTClock Internal Registers         1	198
Table 20.2. SmaRTClock Load Capacitance Settings       2	205
Table 20.3. SmaR I Clock Bias Settings   2	207
Table 21.1. Port I/O Assignment for Analog Functions	217
I able 21.2. Port I/O Assignment for Digital Functions    2	218
Lobia 21.2 Dort I/O Acaignment for External Digital Exect Confure Eurofiana 🦷 🤈	140



# List of Registers

SFR	Definition #	5.1. ADC0CN: ADC0 Control	. 74
SFR	Definition #	5.2. ADC0CF: ADC0 Configuration	. 75
SFR	Definition #	5.3. ADC0AC: ADC0 Accumulator Configuration	. 76
SFR	Definition #	5.4. ADC0PWR: ADC0 Burst Mode Power-Up Time	. 77
SFR	Definition #	5.5. ADC0TK: ADC0 Burst Mode Track Time	. 78
SFR	Definition #	5.6. ADC0H: ADC0 Data Word High Byte	. 79
SFR	Definition #	5.7. ADC0L: ADC0 Data Word Low Byte	. 79
SFR	Definition \$	5.8. ADC0GTH: ADC0 Greater-Than High Byte	. 80
SFR	Definition \$	5.9. ADC0GTL: ADC0 Greater-Than Low Byte	. 80
SFR	Definition \$	5.10. ADC0LTH: ADC0 Less-Than High Byte	. 81
SFR	Definition \$	5.11. ADC0LTL: ADC0 Less-Than Low Byte	. 81
SFR	Definition \$	5.12. ADC0MX: ADC0 Input Channel Select	. 84
SFR	Definition \$	5.13. TOFFH: ADC0 Data Word High Byte	. 87
SFR	Definition #	5.14. TOFFL: ADC0 Data Word Low Byte	. 87
SFR	Definition &	5.15. REF0CN: Voltage Reference Control	. 90
SFR	Definition (	6.1. IREF0CN: Current Reference Control	. 91
SFR	Definition (	6.2. IREF0CF: Current Reference Configuration	. 92
SFR	Definition 7	7.1. CPT0CN: Comparator 0 Control	. 96
SFR	Definition	7.2. CPT0MD: Comparator 0 Mode Selection	. 97
SFR	Definition	7.3. CPT0MX: Comparator0 Input Channel Select	. 99
SFR	Definition 8	8.1. CS0CN: Capacitive Sense Control	107
SFR	Definition 8	8.2. CS0CF: Capacitive Sense Configuration	108
SFR	Definition 8	8.3. CS0DH: Capacitive Sense Data High Byte	109
SFR	Definition 8	8.4. CS0DL: Capacitive Sense Data Low Byte	109
SFR	Definition 8	8.5. CS0SCAN0: Capacitive Sense Channel Scan Mask 0	110
SFR	Definition 8	8.6. CS0SCAN1: Capacitive Sense Channel Scan Mask 1	110
SFR	Definition 8	8.7. CS0SS: Capacitive Sense Auto-Scan Start Channel	111
SFR	Definition 8	8.8. CS0SE: Capacitive Sense Auto-Scan End Channel	111
SFR	Definition 8	8.9. CS0THH: Capacitive Sense Comparator Threshold High Byte	112
SFR	Definition 8	8.10. CS0THL: Capacitive Sense Comparator Threshold Low Byte .	112
SFR	Definition 8	8.11. CS0MD1: Capacitive Sense Mode 1	113
SFR	Definition 8	8.12. CS0MD2: Capacitive Sense Mode 2	114
SFR	Definition 8	8.13. CS0MD3: Capacitive Sense Mode 3	115
SFR	Definition 8	8.14. CS0PM: Capacitive Sense Pin Monitor	116
SFR	Definition 8	8.15. CS0MX: Capacitive Sense Mux Channel Select	118
SFR	Definition 9	9.1. DPL: Data Pointer Low Byte	125
SFR	Definition 9	9.2. DPH: Data Pointer High Byte	125
SFR	Definition 9	9.3. SP: Stack Pointer	126
SFR	Definition 9	9.4. ACC: Accumulator	126
SFR	Definition 9	9.5. B: B Register	126
SFR	Definition 9	9.6. PSW: Program Status Word	127
SFR	Definition <sup>2</sup>	12.1. SFR Page: SFR Page	134
SFR	Definition <sup>7</sup>	13.1. IE: Interrupt Enable	142









\*Note: Signal only available on 'F986, 'F988, and 'F996 devices.

### Figure 3.3. QSOP-24 Pinout Diagram (Top View)



### 5.1. Output Code Formatting

The registers ADC0H and ADC0L contain the high and low bytes of the output conversion code from the ADC at the completion of each conversion. Data can be right-justified or left-justified, depending on the setting of the AD0SJST[2:0]. When the repeat count is set to 1, conversion codes are represented as 10bit unsigned integers. Inputs are measured from 0 to VREF x 1023/1024. Example codes are shown below for both right-justified and left-justified data. Unused bits in the ADC0H and ADC0L registers are set to 0.

Input Voltage	Right-Justified ADC0H:ADC0L (AD0SJST = 000)	Left-Justified ADC0H:ADC0L (AD0SJST = 100)
VREF x 1023/1024	0x03FF	0xFFC0
VREF x 512/1024	0x0200	0x8000
VREF x 256/1024	0x0100	0x4000
0	0x0000	0x0000

When the repeat count is greater than 1, the output conversion code represents the accumulated result of the conversions performed and is updated after the last conversion in the series is finished. Sets of 4, 8, 16, 32, or 64 consecutive samples can be accumulated and represented in unsigned integer format. The repeat count can be selected using the AD0RPT bits in the ADC0AC register. When a repeat count is higher than 1, the ADC output must be right-justified (AD0SJST = 0xx); unused bits in the ADC0H and ADC0L registers are set to 0. The example below shows the right-justified result for various input voltages and repeat counts. Notice that accumulating  $2^n$  samples is equivalent to left-shifting by *n* bit positions when all samples returned from the ADC have the same value.

Input Voltage	Repeat Count = 4	Repeat Count = 16	Repeat Count = 64
V <sub>REF</sub> x 1023/1024	0x0FFC	0x3FF0	0xFFC0
V <sub>REF</sub> x 512/1024	0x0800	0x2000	0x8000
V <sub>REF</sub> x 511/1024	0x07FC	0x1FF0	0x7FC0
0	0x0000	0x0000	0x0000

The AD0SJST bits can be used to format the contents of the 16-bit accumulator. The accumulated result can be shifted right by 1, 2, or 3 bit positions. Based on the principles of oversampling and averaging, the effective ADC resolution increases by 1 bit each time the oversampling rate is increased by a factor of 4. The example below shows how to increase the effective ADC resolution by 1, 2, and 3 bits to obtain an effective ADC resolution of 11-bit, 12-bit, or 13-bit respectively without CPU intervention.

Input Voltage	Repeat Count = 4 Shift Right = 1 11-Bit Result	Repeat Count = 16 Shift Right = 2 12-Bit Result	Repeat Count = 64 Shift Right = 3 13-Bit Result
V <sub>REF</sub> x 1023/1024	0x07F7	0x0FFC	0x1FF8
V <sub>REF</sub> x 512/1024	0x0400	0x0800	0x1000
V <sub>REF</sub> x 511/1024	0x03FE	0x04FC	0x0FF8
0	0x0000	0x0000	0x0000



### 5.2.2. Tracking Modes

Each ADC0 conversion must be preceded by a minimum tracking time in order for the converted result to be accurate. The minimum tracking time is given in Table 4.10. The AD0TM bit in register ADC0CN controls the ADC0 track-and-hold mode. In its default state when Burst Mode is disabled, the ADC0 input is continuously tracked, except when a conversion is in progress. When the AD0TM bit is logic 1, ADC0 operates in low-power track-and-hold mode. In this mode, each conversion is preceded by a tracking period of 3 SAR clocks (after the start-of-conversion signal). When the CNVSTR signal is used to initiate conversions in low-power tracking mode, ADC0 tracks only when CNVSTR is low; conversion begins on the rising edge of CNVSTR (see Figure 5.2). Tracking can also be disabled (shutdown) when the device is in low power standby or sleep modes. Low-power track-and-hold mode is also useful when AMUX settings are frequently changed, due to the settling time requirements described in "5.2.4. Settling Time Requirements" on page 71.



Figure 5.2. 10-Bit ADC Track and Conversion Example Timing (BURSTEN = 0)



### 8.1. Configuring Port Pins as Capacitive Sense Inputs

In order for a port pin to be measured by CS0, that port pin must be configured as an analog input (see "21. Port Input/Output"). Configuring the input multiplexer to a port pin not configured as an analog input will cause the capacitive sense comparator to output incorrect measurements.

### 8.2. Initializing the Capacitive Sensing Peripheral

The following procedure is recommended for properly initializing the CS0 peripheral:

- 1. Enable the CS0 block (CS0EN = 1) before performing any other initializations.
- 2. Initialize the Start of Conversion Mode Select bits (CS0CM[2:0]) to the desired mode.
- 3. Continue initializing all remaining CS0 registers.

### 8.3. Capacitive Sense Start-Of-Conversion Sources

A capacitive sense conversion can be initiated in one of eight ways, depending on the programmed state of the CS0 start of conversion bits (CS0CF6:4). Conversions may be initiated by one of the following:

- 1. Writing a 1 to the CS0BUSY bit of register CS0CN
- 2. Timer 0 overflow
- 3. Timer 2 overflow
- 4. Timer 1 overflow
- 5. Timer 3 overflow
- 6. Convert continuously
- 7. Convert continuously with auto-scan enabled
- 8. Perform a single scan of all enabled channels

Conversions can be configured to be initiated continuously through one of two methods. CS0 can be configured to convert at a single channel continuously or it can be configured to convert continuously with auto-scan enabled. When configured to convert continuously, conversions will begin after the CS0BUSY bit in CS0CF has been set. An interrupt will be generated if CS0 conversion complete interrupts are enabled by setting the ECSCPT bit (EIE2.0).

Single Scan Mode allows all channels enabled in the CS0SCAN0 and CS0SCAN1 registers to be scanned in a single pass. An end of scan interrupt can be enabled to trigger once all selected channels have been converted. See Section "8.9. Automatic Scanning (Method 2—CS0SMEN = 1)" on page 104 for more details about this mode.

The CS0 module uses a method of successive approximation to determine the value of an external capacitance. The number of bits the CS0 module converts is adjustable using the CS0CR bits in register CS0MD2. Conversions are 13 bits long by default, but they can be adjusted to 12, 13, 14, or 16 bits depending on the needs of the application. Unconverted bits will be set to 0. Shorter conversion lengths produce faster conversion rates, and vice-versa. Applications can take advantage of faster conversion rates when the unconverted bits fall below the noise floor.

**Note:** CS0 conversion complete interrupt behavior depends on the settings of the CS0 accumulator. If CS0 is configured to accumulate multiple conversions on an input channel, a CS0 conversion complete interrupt will be generated only after the last conversion completes.



# SFR Definition 9.3. SP: Stack Pointer

Bit	7	6	5	4	3	2	1	0
Name				SP[	7:0]			
Туре	R/W							
Reset	0	0	0	0	0	1	1	1
SFR Pag	SFR Page = All; SFR Address = 0x81							

Bit	Name	Function
7:0	SP[7:0]	Stack Pointer.
		The Stack Pointer holds the location of the top of the stack. The stack pointer is incre- mented before every PUSH operation. The SP register defaults to 0x07 after reset.

## SFR Definition 9.4. ACC: Accumulator

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

SFR Page = All; SFR Address = 0xE0; Bit-Addressable

Bit	Name	Function
7:0	ACC[7:0]	Accumulator.
		This register is the accumulator for arithmetic operations.

### SFR Definition 9.5. B: B Register

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

SFR Page = All; SFR Address = 0xF0; Bit-Addressable

Bit	Name	Function
7:0	B[7:0]	B Register.
		This register serves as a second accumulator for certain arithmetic operations.



## SFR Definition 13.6. EIP2: Extended Interrupt Priority 2

Bit	7	6	5	4	3	2	1	0
Name		PCSEOS	PCSDC	PCSCPT		PRTC0F	PMAT	PWARN
Туре	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = All; SFR Address = 0xF7

Bit	Name	Function
7	Unused	Read = 0b. Write = Don't care.
6	PCSEOS	Capacitive Sense End of Scan Interrupt Priority Control. 0: Capacitive Sense End of Scan interrupt set to low priority level. 1: Capacitive Sense End of Scan interrupt set to high priority level.
5	PCSDC	Capacitive Sense Digital Comparator Interrupt Priority Control. 0: Capacitive Sense Digital Comparator interrupt set to low priority level. 1: Capacitive Sense Digital Comparator interrupt set to high priority level.
4	PCSCPT	Capacitive Sense Conversion Complete Interrupt Priority Control. 0: Capacitive Sense Conversion Complete interrupt set to low priority level. 1: Capacitive Sense Conversion Complete interrupt set to high priority level.
3	Unused	Read = 0b. Write = Don't care.
2	PRTC0F	<ul> <li>SmaRTClock Oscillator Fail Interrupt Priority Control.</li> <li>This bit sets the priority of the SmaRTClock Alarm interrupt.</li> <li>O: SmaRTClock Alarm interrupt set to low priority level.</li> <li>1: SmaRTClock Alarm interrupt set to high priority level.</li> </ul>
1	PMAT	Port Match Interrupt Priority Control. This bit sets the priority of the Port Match Event interrupt. 0: Port Match interrupt set to low priority level. 1: Port Match interrupt set to high priority level.
0	PWARN	<ul> <li>Supply Monitor Early Warning Interrupt Priority Control.</li> <li>This bit sets the priority of the Supply Monitor Early Warning interrupt.</li> <li>0: Supply Monitor Early Warning interrupt set to low priority level.</li> <li>1: Supply Monitor Early Warning interrupt set to high priority level.</li> </ul>



### 14.5.2. PSWE Maintenance

- 1. 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 both PSWE and PSEE both to a 1 to erase Flash pages.
- 2. Minimize the number of variable accesses while PSWE is set to a 1. Handle pointer address updates and loop 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 website.
- 3. 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.
- 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.
- 5. 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.

#### 14.5.3. System Clock

- 1. If operating from an external crystal, be advised that crystal performance is susceptible to electrical interference and is sensitive to layout and to changes in temperature. If the system is operating in an electrically noisy environment, use the internal oscillator or use an external CMOS clock.
- 2. If operating from the external oscillator, switch to the internal oscillator during Flash write or erase operations. The external oscillator can continue to run, and the CPU can switch back to the external oscillator after the Flash operation has completed.

Additional Flash recommendations and example code can be found in "AN201: Writing to Flash from Firm-ware", available from the Silicon Laboratories web site.



## SFR Definition 19.2. OSCICN: Internal Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	IOSCEN	IFRDY			Reserv	ved[5:0]		
Туре	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	Varies	Varies	Varies	Varies	Varies	Varies

SFR Page = 0x0; SFR Address = 0xB2

Bit	Name	Function						
7	IOSCEN	Internal Oscillator Enable.						
		0: Internal oscillator disabled.						
		1: Internal oscillator enabled.						
6	IFRDY	ternal Oscillator Frequency Ready Flag.						
		: Internal oscillator is not running at its programmed frequency.						
		1: Internal oscillator is running at its programmed frequency.						
5:0	Reserved	Must perform read-modify-write.						
Notes:								
1.	Read-modify-v register.	vrite operations such as ORL and ANL must be used to set or clear the enable bit of this						
2.	OSCBIAS (RE	G0CN.4) must be set to 1 before enabling the precision internal oscillator.						



## SFR Definition 19.4. OSCXCN: External Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	XCLKVLD	×	(OSCMD[2:0	)]	Reserved	XFCN[2:0]		
Туре	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0xB1

Bit	Name	Function						
7	XCLKVLD	External Oscillator Valid Flag.						
		<ul> <li>Provides External Oscillator status and is valid at all times for all modes of operation except External CMOS Clock Mode and External CMOS Clock Mode with divide by</li> <li>In these modes, XCLKVLD always returns 0.</li> <li>External Oscillator is unused or not yet stable.</li> <li>External Oscillator is running and stable.</li> </ul>						
6:4	XOSCMD	External Oscillator Mode Bits.						
		Configures the external oscillator circuit to the selected mode.						
		0x: External Oscillator circuit disabled.						
		10: External CMOS Clock Mode.						
		11: External CMOS Clock Mode with divide by 2 stage.						
		00: RC Oscillator Mode.						
		101: Capacitor Oscillator Mode.						
		110: Crystal Oscillator Mode.						
		111: Crystal Oscillator Mode with divide by 2 stage.						
3	Reserved	Read = 0b. Must Write 0b.						
2:0	XFCN	External Oscillator Frequency Control Bits.						
		Controls the external oscillator bias current.						
		000-111: See Table 19.1 on page 190 (Crystal Mode) or Table 19.2 on page 191 (RC or C Mode) for recommended settings.						



# SFR Definition 22.1. SMB0CF: SMBus Clock/Configuration

Bit	7	6	5	4	3	2	1	0
Name	ENSMB	INH	BUSY	EXTHOLD	SMBTOE	SMBFTE	SMBC	:S[1:0]
Туре	R/W	R/W	R	R/W	R/W	R/W	R/	W
Reset	0	0	0	0	0	0	0	0

### SFR Page = 0x0; SFR Address = 0xC1

Bit	Name	Function
7	ENSMB	SMBus Enable.
		This bit enables the SMBus interface when set to 1. When enabled, the interface constantly monitors the SDA and SCL pins.
6	INH	SMBus Slave Inhibit.
		When this bit is set to logic 1, the SMBus does not generate an interrupt when slave events occur. This effectively removes the SMBus slave from the bus. Master Mode interrupts are not affected.
5	BUSY	SMBus 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	EXTHOLD	SMBus Setup and Hold Time Extension Enable.
		This bit controls the SDA setup and hold times according to Table 22.2.
		0: SDA Extended Setup and Hold Times disabled.
		1: SDA Extended Setup and Hold Times enabled.
3	SMBTOE	SMBus SCL Timeout Detection Enable.
		This bit enables SCL low timeout detection. If set to logic 1, the SMBus forces Timer 3 to reload while SCL is high and allows Timer 3 to count when SCL goes low. If Timer 3 is configured to Split Mode, only the High Byte of the timer is held in reload while SCL is high. Timer 3 should be programmed to generate interrupts at 25 ms, and the Timer 3 interrupt service routine should reset SMBus communication.
2	SMBFTE	SMBus Free Timeout Detection Enable.
		When this bit is set to logic 1, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods.
1:0	SMBCS[1:0]	SMBus Clock Source Selection.
		These two bits select the SMBus clock source, which is used to generate the SMBus bit rate. The selected device should be configured according to Equation 22.1. 00: Timer 0 Overflow 01: Timer 1 Overflow 10:Timer 2 High Byte Overflow 11: Timer 2 Low Byte Overflow



### 22.4.2. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information (see SFR Definition 22.2). The higher four bits of SMB0CN (MASTER, TXMODE, STA, and STO) form a status vector that can be used to jump to service routines. MASTER indicates whether a device is the master or slave during the current transfer. TXMODE indicates whether the device is transmitting or receiving data for the current byte.

STA and STO indicate that a START and/or STOP has been detected or generated since the last SMBus interrupt. STA and STO are also used to generate START and STOP conditions when operating as a master. Writing a 1 to STA will cause the SMBus interface to enter Master Mode and generate a START when the bus becomes free (STA is not cleared by hardware after the START is generated). Writing a 1 to STO while in Master Mode will cause the interface to generate a STOP and end the current transfer after the next ACK cycle. If STO and STA are both set (while in Master Mode), a STOP followed by a START will be generated.

The ARBLOST bit indicates that the interface has lost an arbitration. This may occur anytime the interface is transmitting (master or slave). A lost arbitration while operating as a slave indicates a bus error condition. ARBLOST is cleared by hardware each time SI is cleared.

The SI bit (SMBus Interrupt Flag) is set at the beginning and end of each transfer, after each byte frame, or when an arbitration is lost; see Table 22.3 for more details.

**Important Note About the SI Bit:** The SMBus interface is stalled while SI is set; thus SCL is held low, and the bus is stalled until software clears SI.

#### 22.4.2.1.Software ACK Generation

When the EHACK bit in register SMB0ADM is cleared to 0, the firmware on the device must detect incoming slave addresses and ACK or NACK the slave address and incoming data bytes. As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received during the last ACK cycle. ACKRQ is set each time a byte is received, indicating that an outgoing ACK value is needed. When ACKRQ is set, software should write the desired outgoing value to the ACK bit before clearing SI. A NACK will be generated if software does not write the ACK bit before clearing SI. SDA will reflect the defined ACK value immediately following a write to the ACK bit; however SCL will remain low until SI is cleared. If a received slave address is not acknowledged, further slave events will be ignored until the next START is detected.

#### 22.4.2.2.Hardware ACK Generation

When the EHACK bit in register SMB0ADM is set to 1, automatic slave address recognition and ACK generation is enabled. More detail about automatic slave address recognition can be found in Section 22.4.3. As a receiver, the value currently specified by the ACK bit will be automatically sent on the bus during the ACK cycle of an incoming data byte. As a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. The ACKRQ bit is not used when hardware ACK generation is enabled. If a received slave address is NACKed by hardware, further slave events will be ignored until the next START is detected, and no interrupt will be generated.

Table 22.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 22.5 for SMBus status decoding using the SMB0CN register.



### 22.5. SMBus Transfer Modes

The SMBus interface may be configured to operate as master and/or slave. At any particular time, it will be operating in one of the following four modes: Master Transmitter, Master Receiver, Slave Transmitter, or Slave Receiver. The SMBus interface enters Master Mode any time a START is generated, and remains in Master Mode until it loses an arbitration or generates a STOP. An SMBus interrupt is generated at the end of all SMBus byte frames. Note that the position of the ACK interrupt when operating as a receiver depends on whether hardware ACK generation is enabled. As a receiver, the interrupt for an ACK occurs **before** the ACK with hardware ACK generation disabled, and **after** the ACK when hardware ACK generation is enabled. As a transmitter, interrupts occur **after** the ACK, regardless of whether hardware ACK generation is enabled or not.

### 22.5.1. Write Sequence (Master)

During a write sequence, an SMBus master writes data to a slave device. The master in this transfer will be a transmitter during the address byte, and a transmitter during all data bytes. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 0 (WRITE). The master then transmits one or more bytes of serial data. After each byte is transmitted, an acknowledge bit is generated by the slave. The transfer is ended when the STO bit is set and a STOP is generated. Note that the interface will switch to Master Receiver Mode if SMB0DAT is not written following a Master Transmitter interrupt. Figure 22.5 shows a typical master write sequence. Two transmit data bytes are shown, though any number of bytes may be transmitted. Notice that all of the 'data byte transferred' interrupts occur **after** the ACK cycle in this mode, regardless of whether hardware ACK generation is enabled.



Figure 22.5. Typical Master Write Sequence



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 24.2 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 24.3 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 24.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.



Figure 24.2. Multiple-Master Mode Connection Diagram



Figure 24.3. 3-Wire Single Master and 3-Wire Single Slave Mode Connection Diagram



### 25.3.3. SmaRTClock/External Oscillator Capture Mode

The Capture Mode in Timer 3 allows either SmaRTClock or the external oscillator period to be measured against the system clock or the system clock divided by 12. SmaRTClock and the external oscillator period can also be compared against each other.

Setting TF3CEN to 1 enables the SmaRTClock/External Oscillator Capture Mode for Timer 3. In this mode, T3SPLIT should be set to 0, as the full 16-bit timer is used.

When Capture Mode is enabled, a capture event will be generated either every SmaRTClock rising edge or every 8 external clock cycles, depending on the T3XCLK1 setting. When the capture event occurs, the contents of Timer 3 (TMR3H:TMR3L) are loaded into the Timer 3 reload registers (TMR3RLH:TMR3RLL) and the TF3H flag is set (triggering an interrupt if Timer 3 interrupts are enabled). By recording the difference between two successive timer capture values, the SmaRTClock or external clock period 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.

For example, if T3ML = 1b, T3XCLK1 = 0b, and TF3CEN = 1b, Timer 3 will clock every SYSCLK and capture every SmaRTClock rising edge. If SYSCLK is 24.5 MHz and the difference between two successive captures is 350 counts, then the SmaRTClock period is as follows:

350 x (1 / 24.5 MHz) = 14.2 μs.

This mode allows software to determine the exact frequency of the external oscillator in C and RC mode or the time between consecutive SmaRTClock rising edges, which is useful for determining the SmaRTClock frequency.



Figure 25.9. Timer 3 Capture Mode Block Diagram



# SFR Definition 25.14. TMR3RLL: Timer 3 Reload Register Low Byte

Bit	7	6	5	4	3	2	1	0			
Name	TMR3RLL[7:0]										
Туре				R/	W						
Reset	0	0 0 0 0 0 0 0 0									
SFR Pa	SFR Page = 0x0; SFR Address = 0x92										
<b>D</b> ''											

Bit	Name	Function
7:0	TMR3RLL[7:0]	Timer 3 Reload Register Low Byte.
		TMR3RLL holds the low byte of the reload value for Timer 3.

## SFR Definition 25.15. TMR3RLH: Timer 3 Reload Register High Byte

Bit	7 6 5 4 3 2 1 0								
Nam	TMR3RLH[7:0]								
Тур	R/W								
Rese	et O	0	0	0	0	0	0	0	
SFR F	Page = 0x0; SF	R Address :	= 0x93						
Bit	Name		Function						
7:0	TMR3RLH[7:0	] Timer 3 F	ſimer 3 Reload Register High Byte.						
		TMR3RL	MR3RLH holds the high byte of the reload value for Timer 3.						



298

### 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 26.5. PCA Software Timer Mode Diagram



## SFR Definition 26.3. PCA0PWM: PCA PWM Configuration

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

### SFR Page = 0x0; SFR Address = 0xDF

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	<b>Cycle Overflow Interrupt Enable.</b> 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	<ul> <li>Cycle Overflow Flag.</li> <li>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.</li> <li>0: No overflow has occurred since the last time this bit was cleared.</li> <li>1: An overflow has occurred since the last time this bit was cleared.</li> </ul>
4:2	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 config- ured to16-bit PWM mode. 00: 8 bits. 01: 9 bits. 10: 10 bits. 11: 11 bits.

