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	EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	40
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 32x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f501-iqr

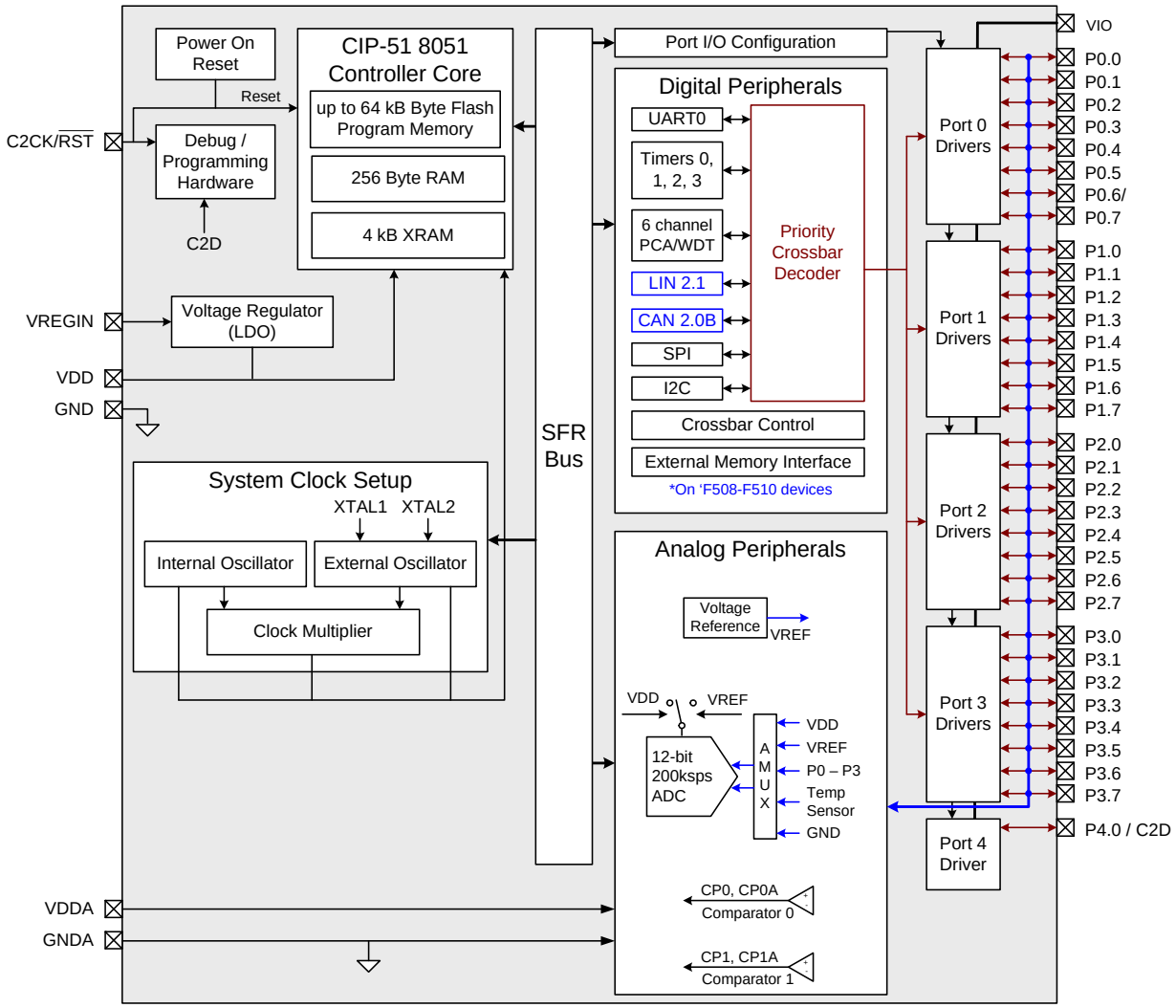

Figure 1.2. C8051F508/9-F510/1 Block Diagram

3. Pin Definitions

Table 3.1. Pin Definitions for the C8051F50x/F51x

Name	Pin 'F500/1/4/5 (48-pin)	Pin F508/9-F510/1 (40-pin)	Pin 'F502/3/6/7 (32-pin)	Type	Description
VDD	4	4	4		Digital Supply Voltage. Must be connected.
GND	6	6	6		Digital Ground. Must be connected.
VDDA	5	5	5		Analog Supply Voltage. Must be connected.
GNDA	7	7	7		Analog Ground. Must be connected.
VREGIN	3	3	3		Voltage Regulator Input
VIO	2	2	2		Port I/O Supply Voltage. Must be connected.
$\overline{\text{RST}}$/ C2CK	12	10	10	D I/O D I/O	Device Reset. Open-drain output of internal POR or V_{DD} Monitor. An external source can initiate a system reset by driving this pin low. Clock signal for the C2 Debug Interface.
C2D	11	—	—	D I/O	Bi-directional data 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 20.29 for a description. 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 20.24 for a description. Bi-directional data signal for the C2 Debug Interface.
P0.0	8	8	8	D I/O or A In	Port 0.0. See SFR Definition 20.12 for a description.
P0.1	1	1	1	D I/O or A In	Port 0.1
P0.2	48	40	32	D I/O or A In	Port 0.2
P0.3	47	39	31	D I/O or A In	Port 0.3
P0.4	46	38	30	D I/O or A In	Port 0.4
P0.5	45	37	29	D I/O or A In	Port 0.5

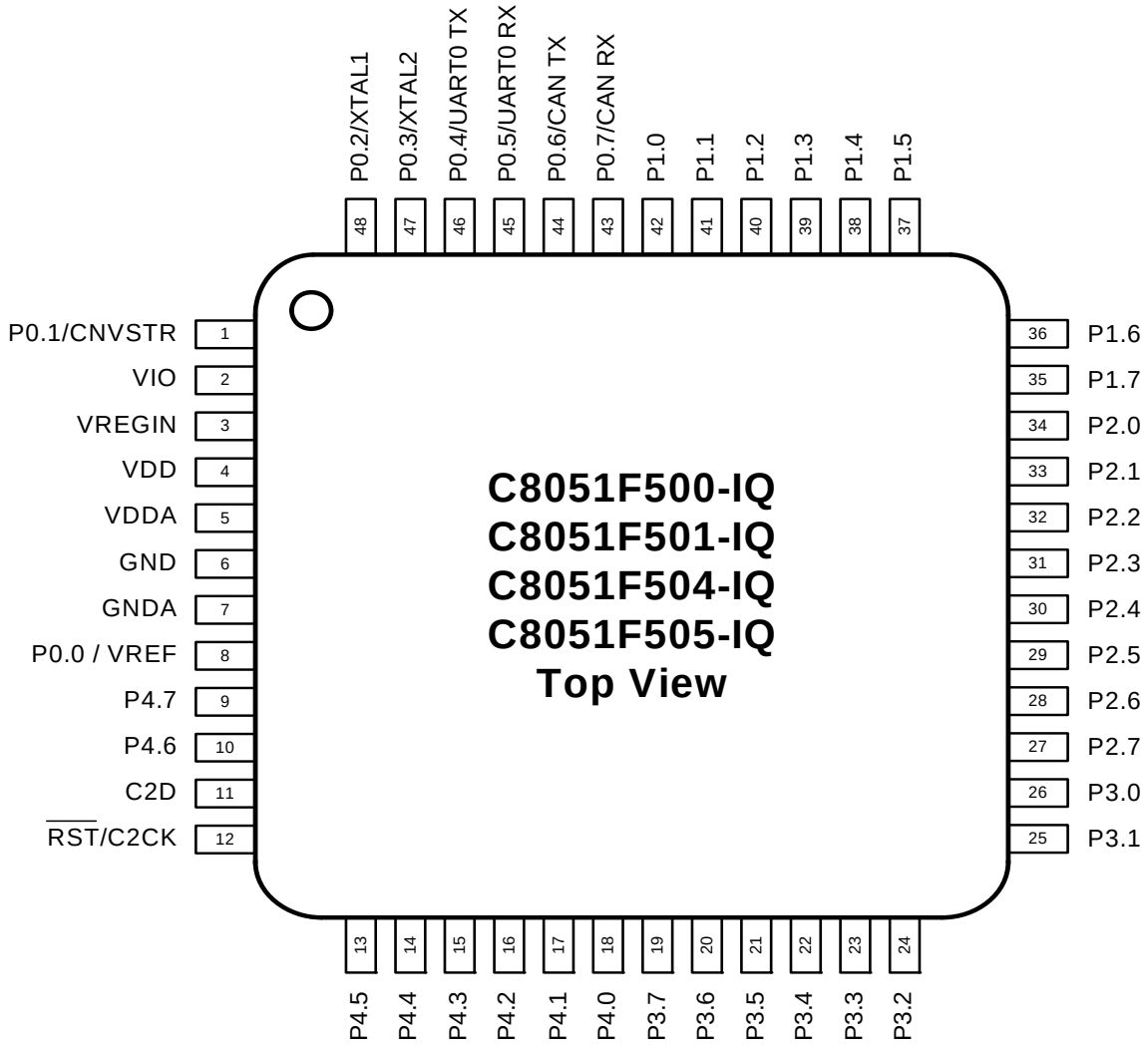

Figure 3.1. QFP-48 Pinout Diagram (Top View)

been accumulated. Similarly, the Window Comparator will not compare the result to the greater-than and less-than registers until “repeat count” conversions have been accumulated.

Note: When using Burst Mode, care must be taken to issue a convert start signal no faster than once every four SYSCLK periods. This includes external convert start signals.

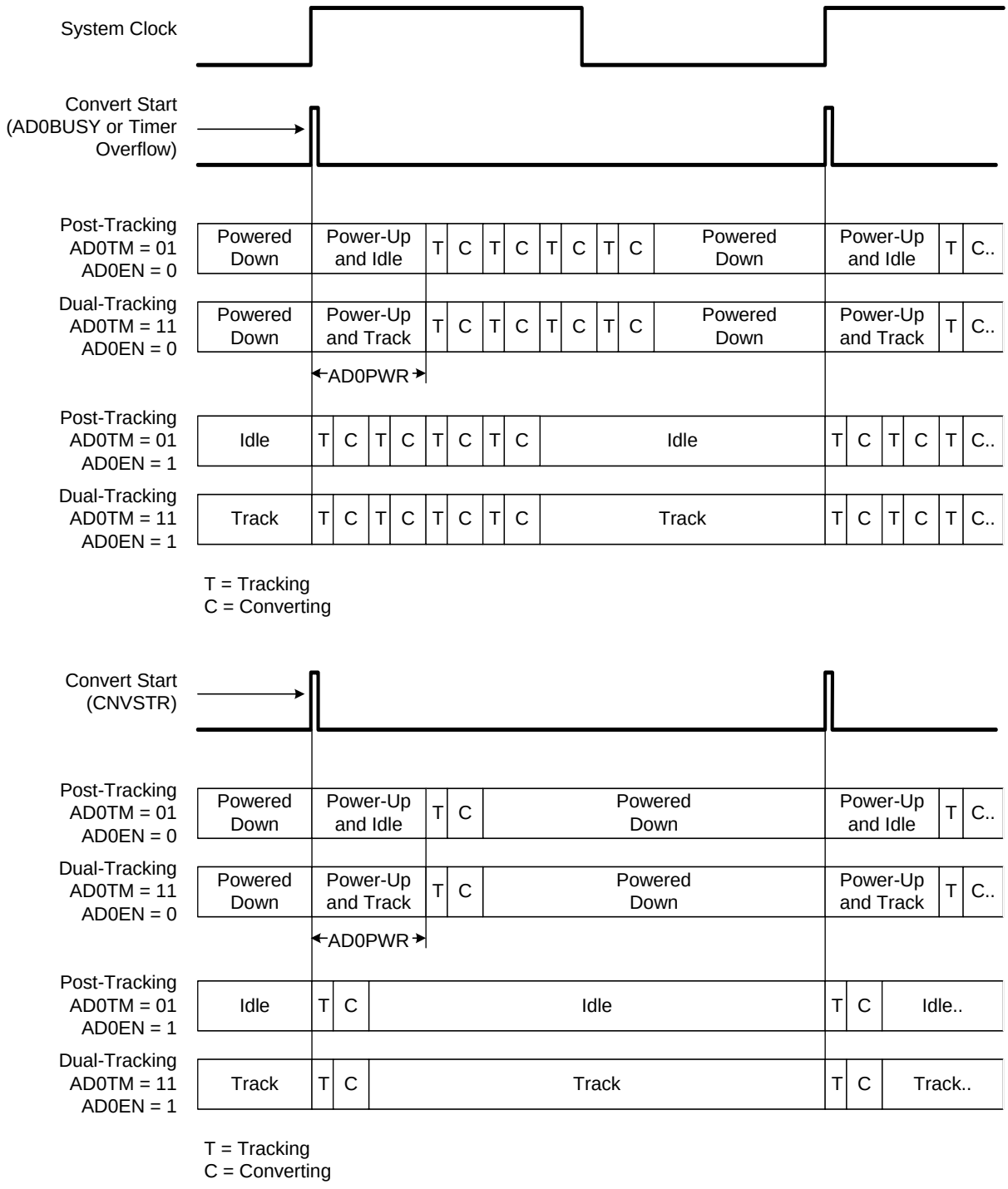

Figure 6.4. 12-Bit ADC Burst Mode Example With Repeat Count Set to 4

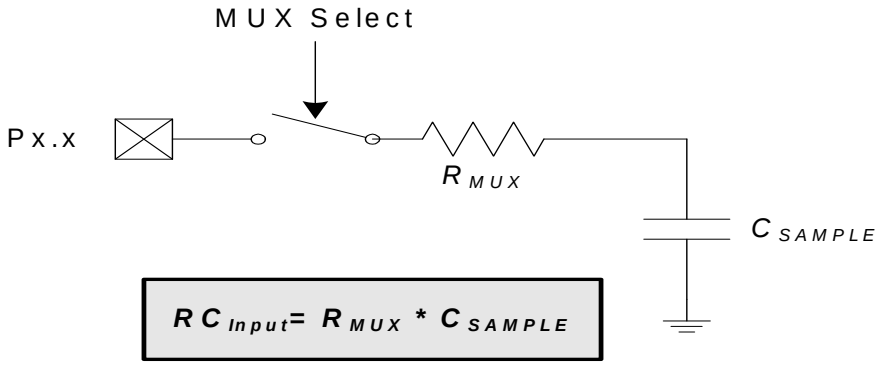

Figure 6.5. ADC0 Equivalent Input Circuit

6.3. Selectable Gain

ADC0 on the C8051F50x/F51x family of devices implements a selectable gain adjustment option. By writing a value to the gain adjust address range, the user can select gain values between 0 and 1.016.

For example, three analog sources to be measured have full-scale outputs of 5.0 V, 4.0 V, and 3.0 V, respectively. Each ADC measurement would ideally use the full dynamic range of the ADC with an internal voltage reference of 1.5 V or 2.2 V (set to 2.2 V for this example). When selecting the first source (5.0 V full-scale), a gain value of 0.44 (5 V full scale x 0.44 = 2.2 V full scale) provides a full-scale signal of 2.2 V when the input signal is 5.0 V. Likewise, a gain value of 0.55 (4 V full scale x 0.55 = 2.2 V full scale) for the second source and 0.73 (3 V full scale x 0.73 = 2.2 V full scale) for the third source provide full-scale ADC0 measurements when the input signal is full-scale.

Additionally, some sensors or other input sources have small part-to-part variations that must be accounted for to achieve accurate results. In this case, the programmable gain value could be used as a calibration value to eliminate these part-to-part variations.

6.3.1. Calculating the Gain Value

The ADC0 selectable gain feature is controlled by 13 bits in three registers. ADC0GNH contains the 8 upper bits of the gain value and ADC0GNL contains the 4 lower bits of the gain value. The final GAINADD bit (ADC0GNA.0) controls an optional extra 1/64 (0.016) of gain that can be added in addition to the ADC0GNH and ADC0GNL gain. The ADC0GNA.0 bit is set to 1 after a power-on reset.

The equivalent gain for the ADC0GNH, ADC0GNL and ADC0GNA registers is as follows:

$$gain = \left(\frac{GAIN}{4096}\right) + GAINADD \times \left(\frac{1}{64}\right)$$

Equation 6.2. Equivalent Gain from the ADC0GNH and ADC0GNL Registers

Where:
GAIN is the 12-bit word of ADC0GNH[7:0] and ADC0GNL[7:4]
GAINADD is the value of the GAINADD bit (ADC0GNA.0)
gain is the equivalent gain value from 0 to 1.016

Gain Register Definition 6.1. ADC0GNH: ADC0 Selectable Gain High Byte

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

Indirect Address = 0x04;

Bit	Name	Function
7:0	GAINH[7:0]	**ADC0 Gain High Byte.** See Section 6.3.1 for details on calculating the value for this register.
Note: This register is accessed indirectly; See Section 6.3.2 for details for writing this register.		

Gain Register Definition 6.2. ADC0GNL: ADC0 Selectable Gain Low Byte

Bit	7	6	5	4	3	2	1	0
Name	GAINL[3:0]				Reserved	Reserved	Reserved	Reserved
Type	W				W	W	W	W
Reset	0	0	0	0	0	0	0	0

Indirect Address = 0x07;

Bit	Name	Function
7:4	GAINL[3:0]	**ADC0 Gain Lower 4 Bits.** See Figure 6.3.1 for details for setting this register. This register is only accessed indirectly through the ADC0H and ADC0L register.
3:0	Reserved	Must Write 0000b
Note: This register is accessed indirectly; See Section 6.3.2 for details for writing this register.		

SFR Definition 6.7. ADC0CN: ADC0 Control

Bit	7	6	5	4	3	2	1	0
Name	AD0EN	BURSTEN	AD0INT	AD0BUSY	AD0WINT	AD0LJST	AD0CM[1:0]	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE8; SFR Page = 0x00; Bit-Addressable

Bit	Name	Function		
7	AD0EN	**ADC0 Enable Bit.** 0: ADC0 Disabled. ADC0 is in low-power shutdown. 1: ADC0 Enabled. ADC0 is active and ready for data conversions.		
6	BURSTEN	**ADC0 Burst Mode Enable Bit.** 0: Burst Mode Disabled. 1: Burst Mode Enabled.		
5	AD0INT	**ADC0 Conversion Complete Interrupt Flag.** 0: ADC0 has not completed a data conversion since AD0INT was last cleared. 1: ADC0 has completed a data conversion.		
4	AD0BUSY	**ADC0 Busy Bit.**	**Read:** 0: ADC0 conversion is not in progress. 1: ADC0 conversion is in progress.	**Write:** 0: No Effect. 1: Initiates ADC0 Conversion if AD0CM[1:0] = 00b
3	AD0WINT	**ADC0 Window Compare Interrupt Flag.** This bit must be cleared by software 0: ADC0 Window Comparison Data match has not occurred since this flag was last cleared. 1: ADC0 Window Comparison Data match has occurred.		
2	AD0LJST	**ADC0 Left Justify Select Bit.** 0: Data in ADC0H:ADC0L registers is right-justified 1: Data in ADC0H:ADC0L registers is left-justified. This option should not be used with a repeat count greater than 1 (when AD0RPT[1:0] is 01b, 10b, or 11b).		
1:0	AD0CM[1:0]	**ADC0 Start of Conversion Mode Select.** 00: ADC0 start-of-conversion source is write of 1 to AD0BUSY. 01: ADC0 start-of-conversion source is overflow of Timer 1. 10: ADC0 start-of-conversion source is rising edge of external CNVSTR. 11: ADC0 start-of-conversion source is overflow of Timer 2.		

Table 13.3. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	Description	Page
IT01CF	0xE4	INT0/INT1 Configuration	128
LIN0ADR	0xD3	LIN0 Address	208
LIN0CF	0xC9	LIN0 Configuration	208
LIN0DAT	0xD2	LIN0 Data	209
OSCICN	0xA1	Internal Oscillator Control	168
OSCICRS	0xA2	Internal Oscillator Coarse Control	169
OSCIFIN	0x9E	Internal Oscillator Fine Calibration	169
OSCXCN	0x9F	External Oscillator Control	173
P0	0x80	Port 0 Latch	191
P0MASK	0xF2	Port 0 Mask Configuration	187
P0MAT	0xF1	Port 0 Match Configuration	187
P0MDIN	0xF1	Port 0 Input Mode Configuration	192
P0MDOUT	0xA4	Port 0 Output Mode Configuration	192
P0SKIP	0xD4	Port 0 Skip	193
P1	0x90	Port 1 Latch	193
P1MASK	0xF4	Port 1 Mask Configuration	188
P1MAT	0xF3	Port 1 Match Configuration	188
P1MDIN	0xF2	Port 1 Input Mode Configuration	194
P1MDOUT	0xA5	Port 1 Output Mode Configuration	194
P1SKIP	0xD5	Port 1 Skip	195
P2	0xA0	Port 2 Latch	195
P2MASK	0xB2	Port 2 Mask Configuration	189
P2MAT	0xB1	Port 2 Match Configuration	189
P2MDIN	0xF3	Port 2 Input Mode Configuration	196
P2MDOUT	0xA6	Port 2 Output Mode Configuration	196
P2SKIP	0xD6	Port 2 Skip	197
P3	0xB0	Port 3 Latch	197
P3MASK	0xAF	Port 3 Mask Configuration	190
P3MAT	0xAE	Port 3 Match Configuration	190
P3MDIN	0xF4	Port 3 Input Mode Configuration	198
P3MDOUT	0xAE	Port 3 Output Mode Configuration	198
P3SKIP	0xD7	Port 3 Skip	199
P4	0xB5	Port 4 Latch	199
P4MDOUT	0xAF	Port 4 Output Mode Configuration	200
PCA0CN	0xD8	PCA Control	300
PCA0CPH0	0xFC	PCA Capture 0 High	305

15.4. 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.

15.4.1. 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. The on-chip V_{DD} monitor is turned on and enabled as a reset source by default by the hardware. If it is disabled by the firmware, use the following recommendations when re-enabling the V_{DD} monitor. Turn on the V_{DD} monitor and enable it 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.
3. 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.
4. 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.
5. 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.

15.4.2. 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 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 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.
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.
4. 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.

18. External Data Memory Interface and On-Chip XRAM

For C8051F50x/F51x devices, 4 kB of RAM are included on-chip and mapped into the external data memory space (XRAM). Additionally, an External Memory Interface (EMIF) is available on the C8051F500/1/4/5 and C8051F508/9-F510/1 devices, which can be used to access off-chip data memories and memory-mapped devices connected to the GPIO ports. The external memory space may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using the MOVX indirect addressing mode using R0 or R1. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the External Memory Interface Control Register (EMI0CN, shown in SFR Definition 18.1).

Note: The MOVX instruction can also be used for writing to the Flash memory. See Section "15. Flash Memory" on page 129 for details. The MOVX instruction accesses XRAM by default.

18.1. Accessing XRAM

The XRAM memory space is accessed using the MOVX instruction. The MOVX instruction has two forms, both of which use an indirect addressing method. The first method uses the Data Pointer, DPTR, a 16-bit register which contains the effective address of the XRAM location to be read from or written to. The second method uses R0 or R1 in combination with the EMI0CN register to generate the effective XRAM address. Examples of both of these methods are given below.

18.1.1. 16-Bit MOVX Example

The 16-bit form of the MOVX instruction accesses the memory location pointed to by the contents of the DPTR register. The following series of instructions reads the value of the byte at address 0x1234 into the accumulator A:

```
MOV    DPTR, #1234h        ; load DPTR with 16-bit address to read (0x1234)
MOVX   A, @DPTR            ; load contents of 0x1234 into accumulator A
```

The above example uses the 16-bit immediate MOV instruction to set the contents of DPTR. Alternately, the DPTR can be accessed through the SFR registers DPH, which contains the upper 8-bits of DPTR, and DPL, which contains the lower 8-bits of DPTR.

18.1.2. 8-Bit MOVX Example

The 8-bit form of the MOVX instruction uses the contents of the EMI0CN SFR to determine the upper 8-bits of the effective address to be accessed and the contents of R0 or R1 to determine the lower 8-bits of the effective address to be accessed. The following series of instructions read the contents of the byte at address 0x1234 into the accumulator A.

```
MOV    EMI0CN, #12h        ; load high byte of address into EMI0CN
MOV    R0, #34h            ; load low byte of address into R0 (or R1)
MOVX   a, @R0              ; load contents of 0x1234 into accumulator A
```

Table 18.2. EMIF Pinout (C8051F508/9-F510/1)

Multiplexed Mode	
Signal Name	**Port Pin**
$\overline{RD}$	P1.6
$\overline{WR}$	P1.7
ALE	P1.5
D0/A0	P3.0
D1/A1	P3.1
D2/A2	P3.2
D3/A3	P3.3
D4/A4	P3.4
D5/A5	P3.5
D6/A6	P3.6
D7/A7	P3.7
A8	P2.0
A9	P2.1
A10	P2.2
A11	P2.3
A12	P2.4
A13	P2.5
A14	P2.6
A15	P2.7

SFR Definition 18.2. EMI0CF: External Memory Configuration

Bit	7	6	5	4	3	2	1	0
Name				EMD2	EMD[1:0]		EALE[1:0]	
Type	R/W							
Reset	0	0	0	0	0	0	1	1

SFR Address = 0xB2; SFR Page = 0x0F

Bit	Name	Function
7:5	Unused	Read = 000b; Write = Don't Care.
4	EMD2	**EMIF Multiplex Mode Select Bit.** 0: EMIF operates in multiplexed address/data mode 1: EMIF operates in non-multiplexed mode (separate address and data pins)
3:2	EMD[1:0]	**EMIF Operating Mode Select Bits.** 00: Internal Only: MOVX accesses on-chip XRAM only. All effective addresses alias to on-chip memory space 01: Split Mode without Bank Select: Accesses below the 4 kB boundary are directed on-chip. Accesses above the 4 kB boundary are directed off-chip. 8-bit off-chip MOVX operations use current contents of the Address high port latches to resolve the upper address byte. To access off chip space, EMI0CN must be set to a page that is not contained in the on-chip address space. 10: Split Mode with Bank Select: Accesses below the 4 kB boundary are directed on-chip. Accesses above the 4 kB boundary are directed off-chip. 8-bit off-chip MOVX operations uses the contents of EMI0CN to determine the high-byte of the address. 11: External Only: MOVX accesses off-chip XRAM only. On-chip XRAM is not visible to the CPU.
1:0	EALE[1:0]	**ALE Pulse-Width Select Bits.** These bits only have an effect when EMD2 = 0. 00: ALE high and ALE low pulse width = 1 SYSCLK cycle. 01: ALE high and ALE low pulse width = 2 SYSCLK cycles. 10: ALE high and ALE low pulse width = 3 SYSCLK cycles. 11: ALE high and ALE low pulse width = 4 SYSCLK cycles.

18.6.2.3. 8-bit MOVX with Bank Select: EMI0CF[4:2] = 010

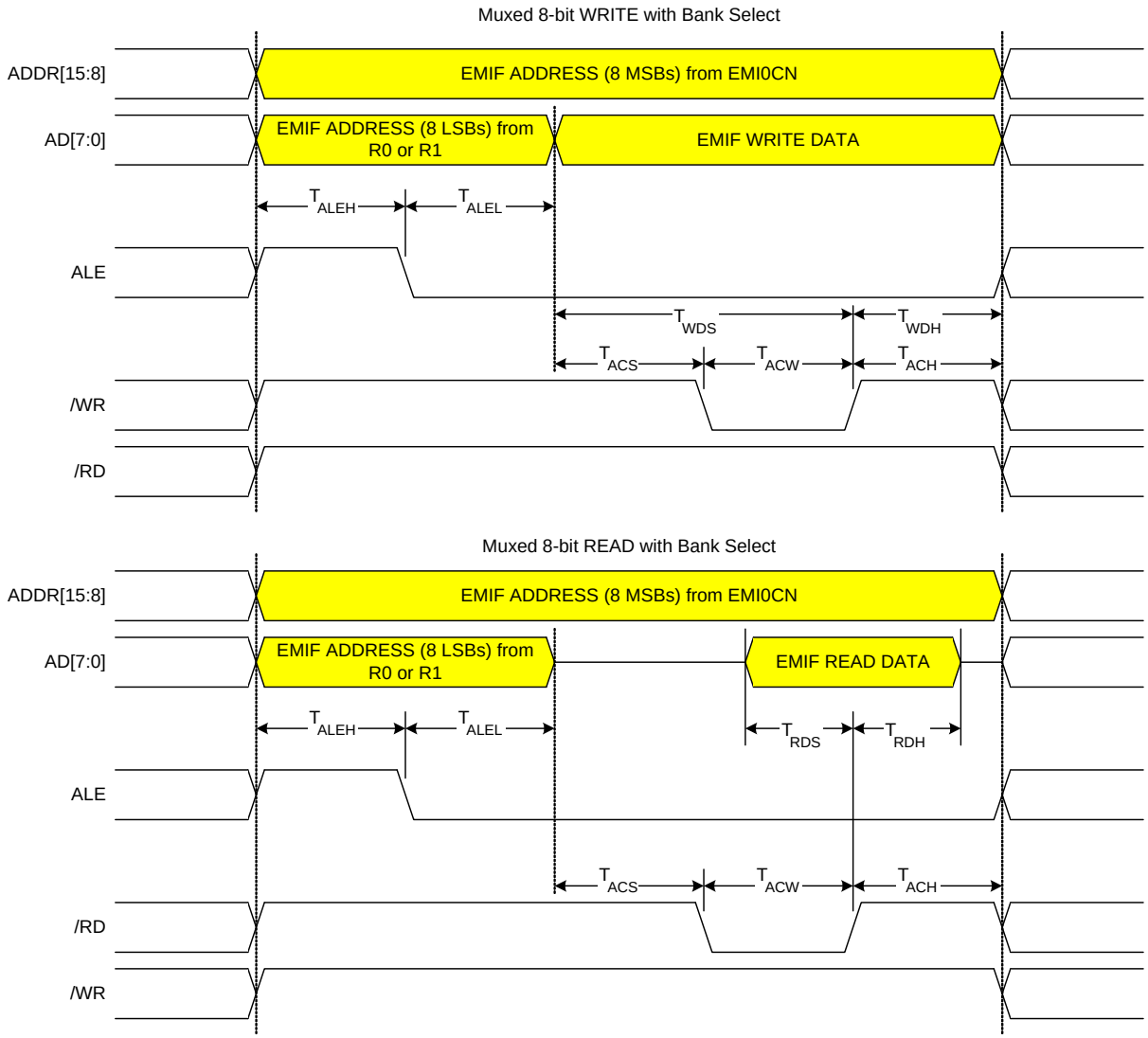

Figure 18.9. Multiplexed 8-bit MOVX with Bank Select Timing

Table 18.3. AC Parameters for External Memory Interface

Parameter	Description	Min*	Max*	Units
T_{ACS}	Address/Control Setup Time	0	3 x T_{SYSCLK}	ns
T_{ACW}	Address/Control Pulse Width	1 x T_{SYSCLK}	16 x T_{SYSCLK}	ns
T_{ACH}	Address/Control Hold Time	0	3 x T_{SYSCLK}	ns
T_{ALEH}	Address Latch Enable High Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{ALEL}	Address Latch Enable Low Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{WDS}	Write Data Setup Time	1 x T_{SYSCLK}	19 x T_{SYSCLK}	ns
T_{WDH}	Write Data Hold Time	0	3 x T_{SYSCLK}	ns
T_{RDS}	Read Data Setup Time	20		ns
T_{RDH}	Read Data Hold Time	0		ns
***Note:** T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).				

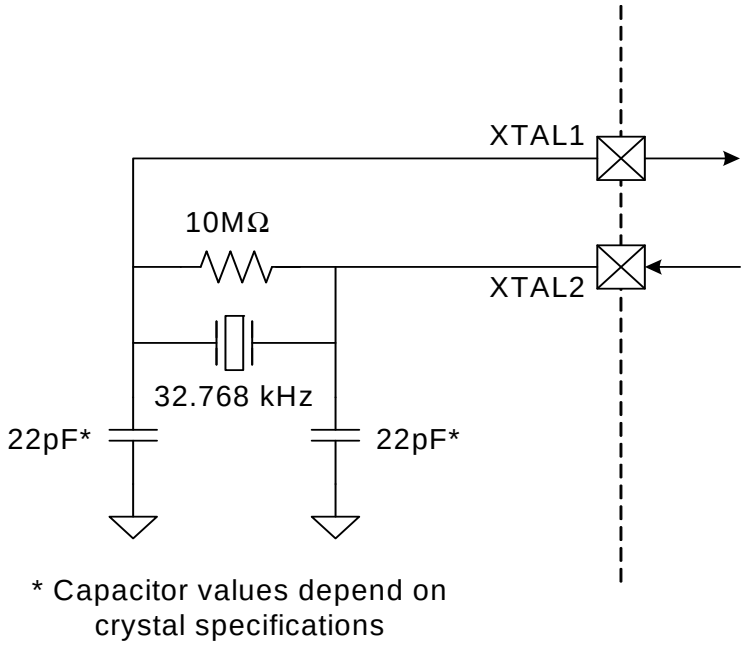

Figure 19.3. External 32.768 kHz Quartz Crystal Oscillator Connection Diagram

19.4.2. External RC Example

If an RC network is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 19.1, Option 2. The capacitor should be no greater than 100 pF; however for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, first select the RC network value to produce the desired frequency of oscillation, according to Equation , where f = the frequency of oscillation in MHz, C = the capacitor value in pF, and R = the pull-up resistor value in kΩ.

$$f = 1.23 \times 10^3 / (R \times C)$$

Equation 19.1. RC Mode Oscillator Frequency

For example: If the frequency desired is 100 kHz, let R = 246 kΩ and C = 50 pF:

f = 1.23(10^3)/RC = 1.23(10^3)/[246 x 50] = 0.1 MHz = 100 kHz

Referring to the table in SFR Definition 19.6, the required XFCN setting is 010b.

19.4.3. External Capacitor Example

If a capacitor is used as an external oscillator for the MCU, the circuit should be configured as shown in Figure 19.1, Option 3. The capacitor should be no greater than 100 pF; however for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, select the capacitor to be used and find the frequency of oscillation according to Equation 19.2, where f = the frequency of oscillation in MHz, C = the capacitor value in pF, and V_{DD} = the MCU power supply in Volts.

20.1.3. Interfacing Port I/O in a Multi-Voltage System

All Port I/O are capable of interfacing to digital logic operating at a supply voltage higher than VDD and less than 5.25 V. Connect the VIO pin to the voltage source of the interface logic.

20.2. Assigning Port I/O Pins to Analog and Digital Functions

Port I/O pins P0.0–P3.7 can be assigned to various analog, digital, and external interrupt functions. P4.0-P4.7 can be assigned to only digital functions. The Port pins assigned to analog functions should be configured for analog I/O, and Port pins assigned to digital or external interrupt functions should be configured for digital I/O.

20.2.1. Assigning Port I/O Pins to Analog Functions

Table 20.1 shows all available analog functions that require Port I/O assignments. **Port pins selected for these analog functions should have their corresponding bit in PnSKIP set to 1.** This reserves the pin for use by the analog function and does not allow it to be claimed by the Crossbar. Table 20.1 shows the potential mapping of Port I/O to each analog function.

Table 20.1. Port I/O Assignment for Analog Functions

Analog Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
ADC Input	P0.0–P3.7*	ADC0MX, PnSKIP
Comparator0 or Compartor1 Input	P0.0–P2.7	CPT0MX, CPT1MX, PnSKIP
Voltage Reference (VREF0)	P0.0	REF0CN, PnSKIP
External Oscillator in Crystal Mode (XTAL1)	P0.2	OSCXCN, PnSKIP
External Oscillator in RC, C, or Crystal Mode (XTAL2)	P0.3	OSCXCN, PnSKIP
***Note:** P3.1–P3.7 are only available on the 48-pin and 40-pin packages		

20.2.2. Assigning Port I/O Pins to Digital Functions

Any Port pins not assigned to analog functions may be assigned to digital functions or used as GPIO. Most digital functions rely on the Crossbar for pin assignment; however, some digital functions bypass the Crossbar in a manner similar to the analog functions listed above. **Port pins used by these digital functions and any Port pins selected for use as GPIO should have their corresponding bit in PnSKIP set to 1.** Table 20.2 shows all available digital functions and the potential mapping of Port I/O to each digital function.

Table 20.2. Port I/O Assignment for Digital Functions

Digital Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
UART0, SPI0, SMBus, CAN0, LIN0, CP0, CP0A, CP1, CP1A, SYSCLK, PCA0 (CEX0-5 and ECI), T0 or T1.	Any Port pin available for assignment by the Crossbar. This includes P0.0–P4.7* pins which have their PnSKIP bit set to 0. **Note:** The Crossbar will always assign UART0 pins to P0.4 and P0.5 and always assign CAN0 to P0.6 and P0.7.	XBR0, XBR1, XBR2
Any pin used for GPIO	P0.0–P4.7*	P0SKIP, P1SKIP, P2SKIP, P3SKIP
***Note:** P3.1–P3.7 and P4.0 are only available on the 48-pin and 40pin packages. P4.1–P4.7 are only available on the 48-pin package. A skip register is not available for P4.		

21.1. Software Interface with the LIN Controller

The selection of the mode (Master or Slave) and the automatic baud rate feature are done though the LIN0 Control Mode (LIN0CF) register. The other LIN registers are accessed indirectly through the two SFRs LIN0 Address (LIN0ADR) and LIN0 Data (LIN0DAT). The LIN0ADR register selects which LIN register is targeted by reads/writes of the LIN0DAT register. The full list of indirectly-accessible LIN registers is given in Table 21.4 on page 210.

21.2. LIN Interface Setup and Operation

The hardware based LIN controller allows for the implementation of both Master and Slave nodes with minimal firmware overhead and complete control of the interface status while allowing for interrupt and polled mode operation.

The first step to use the controller is to define the basic characteristics of the node:

Mode—Master or Slave

Baud Rate—Either defined manually or using the autobaud feature (slave mode only)

Checksum Type—Select between classic or enhanced checksum, both of which are implemented in hardware.

21.2.1. Mode Definition

Following the LIN specification, the controller implements in hardware both the Slave and Master operating modes. The mode is configured using the MODE bit (LIN0CF.6).

21.2.2. Baud Rate Options: Manual or Autobaud

The LIN controller can be selected to have its baud rate calculated manually or automatically. A master node must always have its baud rate set manually, but slave nodes can choose between a manual or automatic setup. The configuration is selected using the ABAUD bit (LIN0CF.5).

Both the manual and automatic baud rate configurations require additional setup. The following sections explain the different options available and their relation with the baud rate, along with the steps necessary to achieve the required baud rate.

21.2.3. Baud Rate Calculations: Manual Mode

The baud rate used by the LIN controller is a function of the System Clock (SYSCLK) and the LIN timing registers according to the following equation:

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

The prescaler, divider and multiplier factors are part of the LIN0DIV and LIN0MUL registers and can assume values in the following range:

Table 21.1. Baud Rate Calculation Variable Ranges

Factor	Range
prescaler	0…3
multiplier	0…31
divider	200…511

Important Note: The minimum system clock (SYSCLK) to operate the LIN controller is 8 MHz.

26.2. Timer 2

Timer 2 is a 16-bit timer formed by two 8-bit SFRs: TMR2L (low byte) and TMR2H (high byte). Timer 2 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T2SPLIT bit (TMR2CN.3) defines the Timer 2 operation mode.

Timer 2 may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. The external clock mode is ideal for real-time clock (RTC) functionality, where the internal oscillator drives the system clock while Timer 2 (and/or the PCA) is clocked by an external precision oscillator. Note that the external oscillator source divided by 8 is synchronized with the system clock.

26.2.1. 16-bit Timer with Auto-Reload

When T2SPLIT (TMR2CN.3) is zero, Timer 2 operates as a 16-bit timer with auto-reload. Timer 2 can be clocked by SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 2 reload registers (TMR2RLH and TMR2RLL) is loaded into the Timer 2 register as shown in Figure 26.4, and the Timer 2 High Byte Overflow Flag (TMR2CN.7) is set. If Timer 2 interrupts are enabled (if IE.5 is set), an interrupt will be generated on each Timer 2 overflow. Additionally, if Timer 2 interrupts are enabled and the TF2LEN bit is set (TMR2CN.5), an interrupt will be generated each time the lower 8 bits (TMR2L) overflow from 0xFF to 0x00.

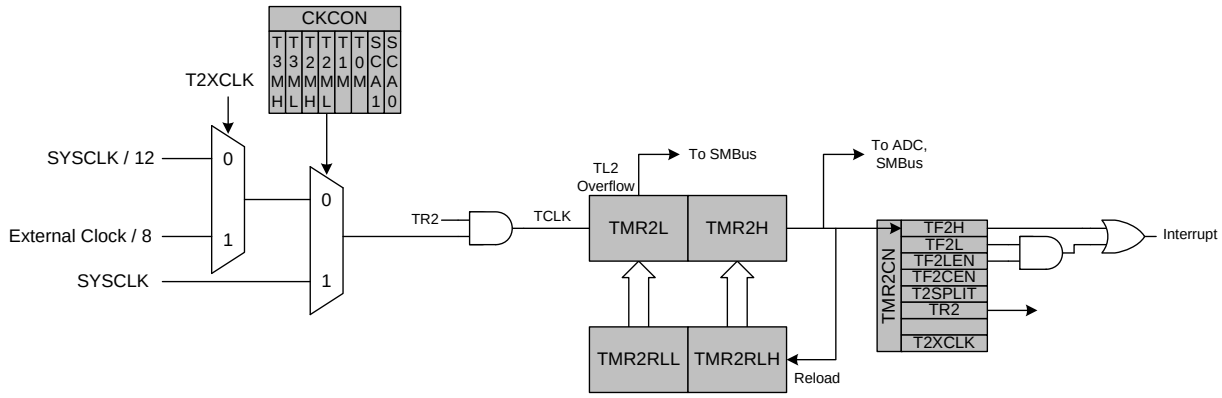

Figure 26.4. Timer 2 16-Bit Mode Block Diagram

26.2.2. 8-bit Timers with Auto-Reload

When T2SPLIT is set, Timer 2 operates as two 8-bit timers (TMR2H and TMR2L). Both 8-bit timers operate in auto-reload mode as shown in Figure 26.5. TMR2RLL holds the reload value for TMR2L; TMR2RLH holds the reload value for TMR2H. The TR2 bit in TMR2CN handles the run control for TMR2H. TMR2L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Timer 2 Clock Select bits (T2MH and T2ML in CKCON) select either SYSCLK or the clock defined by the Timer 2 External Clock Select bit (T2XCLK in TMR2CN), as follows:

26.3. Timer 3

Timer 3 is a 16-bit timer formed by two 8-bit SFRs: TMR3L (low byte) and TMR3H (high byte). Timer 3 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T3SPLIT bit (TMR3CN.3) defines the Timer 3 operation mode.

Timer 3 may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. The external clock mode is ideal for real-time clock (RTC) functionality, where the internal oscillator drives the system clock while Timer 3 (and/or the PCA) is clocked by an external precision oscillator. Note that the external oscillator source divided by 8 is synchronized with the system clock.

26.3.1. 16-bit Timer with Auto-Reload

When T3SPLIT (TMR3CN.3) is zero, Timer 3 operates as a 16-bit timer with auto-reload. Timer 3 can be clocked by SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 3 reload registers (TMR3RLH and TMR3RLL) is loaded into the Timer 3 register as shown in Figure 26.7, and the Timer 3 High Byte Overflow Flag (TMR3CN.7) is set. If Timer 3 interrupts are enabled, an interrupt will be generated on each Timer 3 overflow. Additionally, if Timer 3 interrupts are enabled and the TF3LEN bit is set (TMR3CN.5), an interrupt will be generated each time the lower 8 bits (TMR3L) overflow from 0xFF to 0x00.

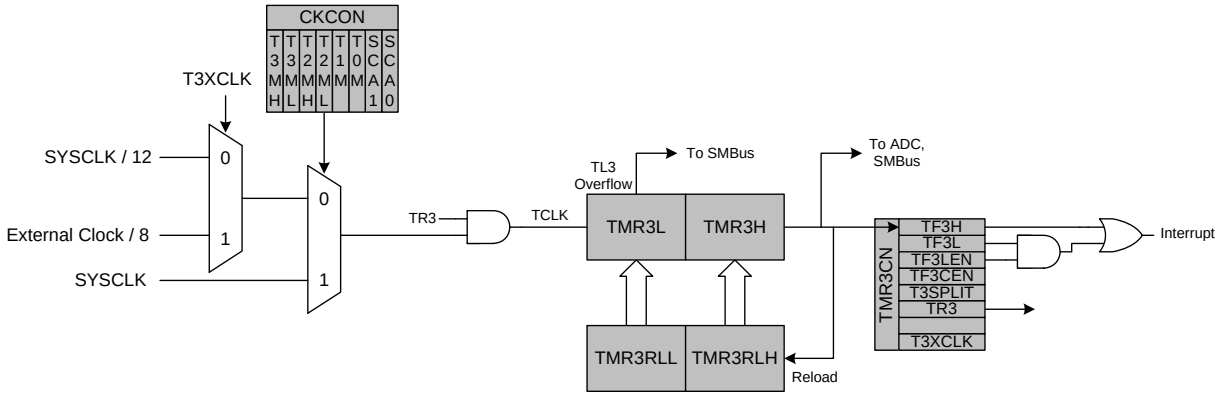

Figure 26.7. Timer 3 16-Bit Mode Block Diagram

26.3.2. 8-bit Timers with Auto-Reload

When T3SPLIT is set, Timer 3 operates as two 8-bit timers (TMR3H and TMR3L). Both 8-bit timers operate in auto-reload mode as shown in Figure 26.8. TMR3RLL holds the reload value for TMR3L; TMR3RLH holds the reload value for TMR3H. The TR3 bit in TMR3CN handles the run control for TMR3H. TMR3L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Timer 3 Clock Select bits (T3MH and T3ML in CKCON) select either SYSCLK or the clock defined by the Timer 3 External Clock Select bit (T3XCLK in TMR3CN), as follows: