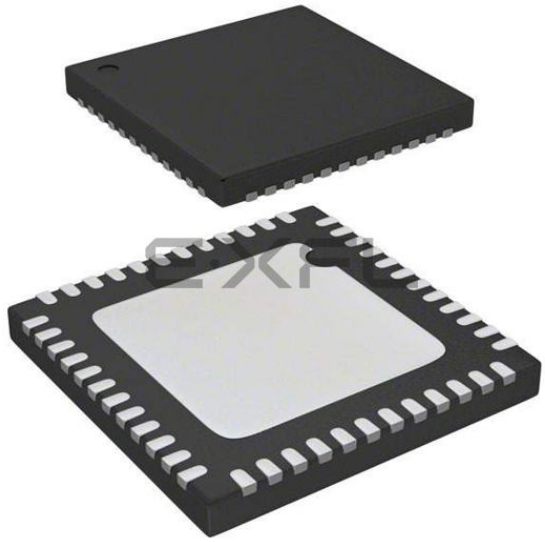

Welcome to E-XFL.COM

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	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8.25K 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-VFQFN Exposed Pad
Supplier Device Package	48-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f581-imr

Table 3.1. Pin Definitions for the C8051F58x/F59x (Continued)

Name	Pin F580/1/4/5 (48-pin)	Pin F588/9-F590/1 (40-pin)	Pin F582/3/6/7 (32-pin)	Type	Description
P0.6	44	36	28	D I/O or A In	Port 0.6
P0.7	43	35	27	D I/O or A In	Port 0.7
P1.0	42	34	26	D I/O or A In	Port 1.0. See SFR Definition 20.17 for a description.
P1.1	41	33	25	D I/O or A In	Port 1.1.
P1.2	40	32	24	D I/O or A In	Port 1.2.
P1.3	39	31	23	D I/O or A In	Port 1.3.
P1.4	38	30	22	D I/O or A In	Port 1.4.
P1.5	37	29	21	D I/O or A In	Port 1.5.
P1.6	36	28	20	D I/O or A In	Port 1.6.
P1.7	35	27	19	D I/O or A In	Port 1.7.
P2.0	34	26	18	D I/O or A In	Port 2.0. See SFR Definition 20.21 for a description.
P2.1	33	25	17	D I/O or A In	Port 2.1.
P2.2	32	24	16	D I/O or A In	Port 2.2.
P2.3	31	23	15	D I/O or A In	Port 2.3.
P2.4	30	22	14	D I/O or A In	Port 2.4.
P2.5	29	21	13	D I/O or A In	Port 2.5.
P2.6	28	20	12	D I/O or A In	Port 2.6.
P2.7	27	19	11	D I/O or A In	Port 2.7.
P3.0	26	18	—	D I/O or A In	Port 3.0. See SFR Definition 20.25 for a description.
P3.1	25	17	—	D I/O or A In	Port 3.1.
P3.2	24	16	—	D I/O or A In	Port 3.2.
P3.3	23	15	—	D I/O or A In	Port 3.3.
P3.4	22	14	—	D I/O or A In	Port 3.4.
P3.5	21	13	—	D I/O or A In	Port 3.5.
P3.6	20	12	—	D I/O or A In	Port 3.6.

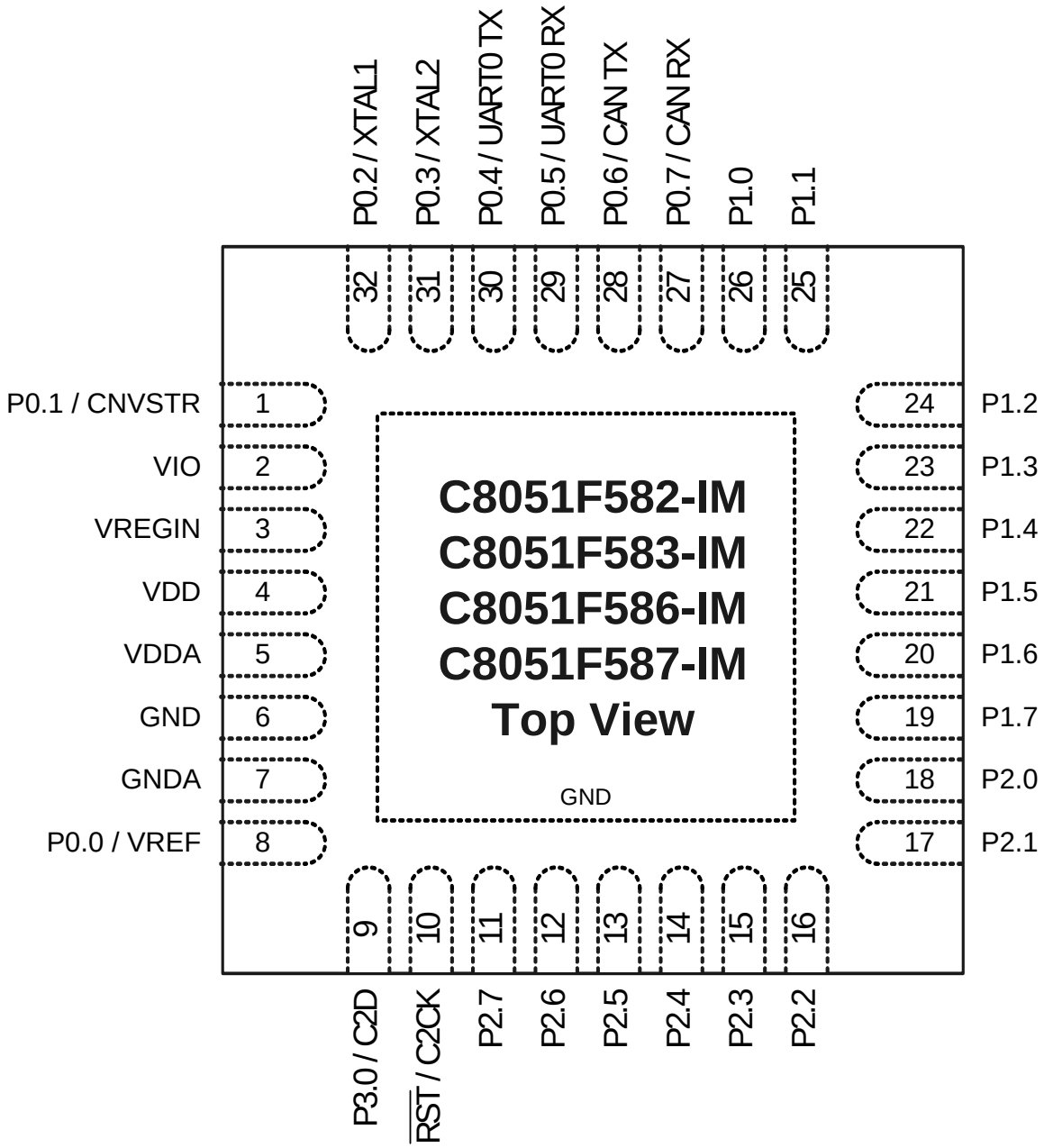

Figure 3.5. QFN-32 Pinout Diagram (Top View)

For example, if ADC0GNH = 0xFC, ADC0GNL = 0x00, and GAINADD = 1, GAIN = 0xFC0 = 4032, and the resulting equation is as follows:

$$GAIN = \left(\frac{4032}{4096}\right) + 1 \times \left(\frac{1}{64}\right) = 0.984 + 0.016 = 1.0$$

The table below equates values in the ADC0GNH, ADC0GNL, and ADC0GNA registers to the equivalent gain using this equation.

ADC0GNH Value	ADC0GNL Value	GAINADD Value	GAIN Value	Equivalent Gain
0xFC (default)	0x00 (default)	1 (default)	4032 + 64	1.0 (default)
0x7C	0x00	1	1984 + 64	0.5
0xBC	0x00	1	3008 + 64	0.75
0x3C	0x00	1	960 + 64	0.25
0xFF	0xF0	0	4095 + 0	~1.0
0xFF	0xF0	1	4096 + 64	1.016

For any desired gain value, the GAIN registers can be calculated by the following:

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

Equation 6.3. Calculating the ADC0GNH and ADC0GNL Values from the Desired Gain

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

When calculating the value of GAIN to load into the ADC0GNH and ADC0GNL registers, the GAINADD bit can be turned on or off to reach a value closer to the desired gain value.

For example, the initial example in this section requires a gain of 0.44 to convert 5 V full scale to 2.2 V full scale. Using Equation 6.3:

$$GAIN = \left(0.44 - GAINADD \times \left(\frac{1}{64}\right)\right) \times 4096$$

If GAINADD is set to 1, this makes the equation:

$$GAIN = \left(0.44 - 1 \times \left(\frac{1}{64}\right)\right) \times 4096 = 0.424 \times 4096 = 1738 = 0x06CA$$

The actual gain from setting GAINADD to 1 and ADC0GNH and ADC0GNL to 0x6CA is 0.4399. A similar gain can be achieved if GAINADD is set to 0 with a different value for ADC0GNH and ADC0GNL.

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.		

9. Comparators

The C8051F58x/F59x devices include three on-chip programmable voltage Comparators. A block diagram of the comparators is shown in Figure 9.1, where “n” is the comparator number (0, 1, or 2). The three Comparators operate identically except that Comparator0 can also be used a reset source.

Each 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, CP1, CP2), or an asynchronous “raw” output (CP0A, CP1A, CP2A). The asynchronous signal is available even when the system clock is not active. This allows the Comparators to operate and generate an output with the device in STOP mode. When assigned to a Port pin, the Comparator outputs may be configured as open drain or push-pull (see Section “20.4. Port I/O Initialization” on page 195). Comparator0 may also be used as a reset source (see Section “17.5. Comparator0 Reset” on page 155).

The Comparator0 inputs are selected in the CPT0MX register (SFR Definition 9.7). The CMX0P1-CMX0P0 bits select the Comparator0 positive input; the CMX0N1-CMX0N0 bits select the Comparator0 negative input. The Comparator1 inputs are selected in the CPT1MX register (SFR Definition 9.8). The CMX1P1-CMX1P0 bits select the Comparator1 positive input; the CMX1N1-CMX1N0 bits select the Comparator1 negative input. The Comparator2 inputs are selected in the CPT2MX register (SFR Definition 9.9). The CMX2P1-CMX2P0 bits select the Comparator1 positive input; the CMX2N1-CMX2N0 bits select the Comparator2 negative input.

Important Note About Comparator Inputs: The Port pins selected as Comparator inputs should be configured as analog inputs in their associated Port configuration register, and configured to be skipped by the Crossbar (for details on Port configuration, see Section “20.1. Port I/O Modes of Operation” on page 190).

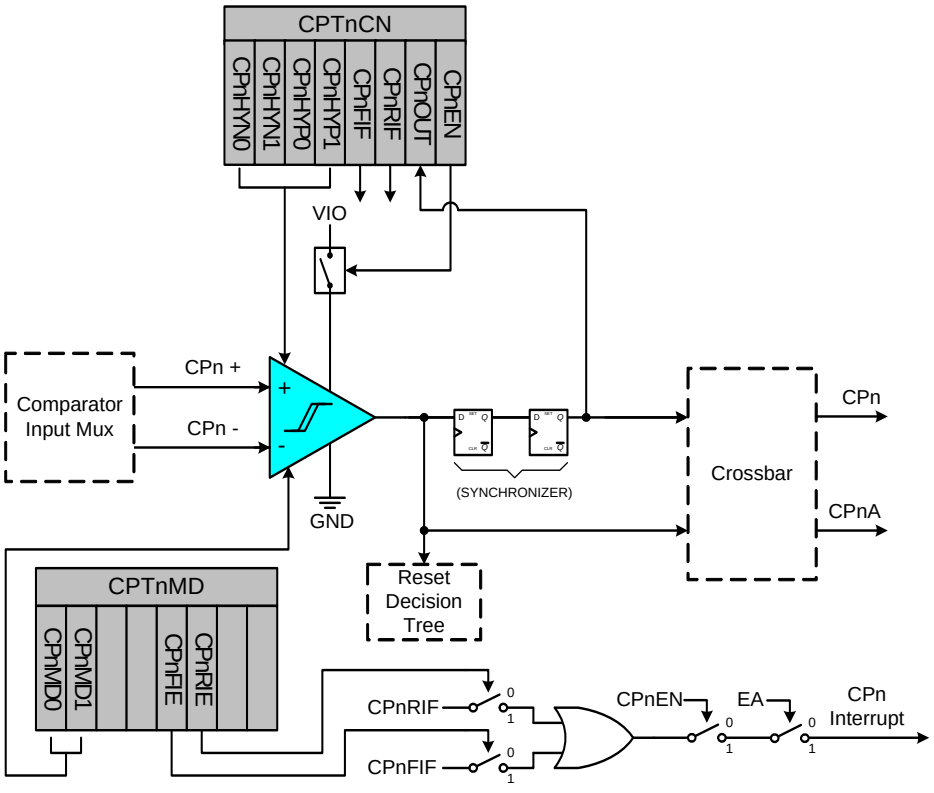

Figure 9.1. Comparator Functional Block Diagram

SFR Definition 9.4. CPT1MD: Comparator1 Mode Selection

Bit	7	6	5	4	3	2	1	0
Name			CP1RIE	CP1FIE			CP1MD[1:0]	
Type	R	R	R/W	R/W	R	R	R/W	
Reset	0	0	0	0	0	0	1	0

SFR Address = 0x9E; SFR Page = 0x00

Bit	Name	Function
7:6	Unused	Read = 00b, Write = Don't Care.
5	CP1RIE	**Comparator1 Rising-Edge Interrupt Enable.** 0: Comparator1 Rising-edge interrupt disabled. 1: Comparator1 Rising-edge interrupt enabled.
4	CP1FIE	**Comparator1 Falling-Edge Interrupt Enable.** 0: Comparator1 Falling-edge interrupt disabled. 1: Comparator1 Falling-edge interrupt enabled.
3:2	Unused	Read = 00b, Write = don't care.
1:0	CP1MD[1:0]	**Comparator1 Mode Select.** These bits affect the response time and power consumption for Comparator1. 00: Mode 0 (Fastest Response Time, Highest Power Consumption) 01: Mode 1 10: Mode 2 11: Mode 3 (Slowest Response Time, Lowest Power Consumption)

Table 13.1. Special Function Register (SFR) Memory Map for Pages 0x00, 0x10, and 0x0F

90	00 10 0F	P1 (All Pages)	TMR3CN TMR5CN	TMR3RLL TMR5CAPL	TMR3RLH TMR5CAPH	TMR3L TMR5L	TMR3H TMR5H	TMR5CF	CLKMUL
88	00 10 0F	TCON (All Pages)	TMOD (All Pages)	TL0 (All Pages)	TL1 (All Pages)	TH0 (All Pages)	TH1 (All Pages)	CKCON (All Pages)	PSCTL CLKSEL
80	00 10 0F	P0 (All Pages)	SP (All Pages)	DPL (All Pages)	DPH (All Pages)	SFR0CN	SFRNEXT (All Pages)	SFRLAST (All Pages)	PCON (All Pages)
		0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

(bit addressable)

Table 13.3. Special Function Registers (Continued)

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

Register	Address	Description	Page
P4MDOUT	0xAF	Port 4 Output Mode Configuration	213
PCA0CN	0xD8	PCA0 Control	327
PCA0CPH0	0xFC	PCA0 Capture 0 High	332
PCA0CPH1	0xEA	PCA0 Capture 1 High	332
PCA0CPH2	0xEC	PCA0 Capture 2 High	332
PCA0CPH3	0xEE	PCA0 Capture 3 High	332
PCA0CPH4	0xFE	PCA0 Capture 4 High	332
PCA0CPH5	0xCF	PCA0 Capture 5 High	332
PCA0CPL0	0xFB	PCA0 Capture 0 Low	332
PCA0CPL1	0xE9	PCA0 Capture 1 Low	332
PCA0CPL2	0xEB	PCA0 Capture 2 Low	332
PCA0CPL3	0xED	PCA0 Capture 3 Low	332
PCA0CPL4	0xFD	PCA0 Capture 4 Low	332
PCA0CPL5	0xCE	PCA0 Capture 5 Low	332
PCA0CPM0	0xDA	PCA0 Module 0 Mode Register	330
PCA0CPM1	0xDB	PCA0 Module 1 Mode Register	330
PCA0CPM2	0xDC	PCA0 Module 2 Mode Register	330
PCA0CPM3	0xDD	PCA0 Module 3 Mode Register	330
PCA0CPM4	0xDE	PCA0 Module 4 Mode Register	330
PCA0CPM5	0xDF	PCA0 Module 5 Mode Register	330
PCA0H	0xFA	PCA0 Counter High	331
PCA0L	0xF9	PCA0 Counter Low	331
PCA0MD	0xD9	PCA0 Mode	328
PCA0PWM	0xD9	PCA0 PWM Configuration	329
PCA1CN	0xD8	PCA1 Control	345
PCA1CPH6	0xFC	PCA1 Capture 6 High	350
PCA1CPH7	0xEA	PCA1 Capture 7 High	350
PCA1CPH8	0xEC	PCA1 Capture 8 High	350
PCA1CPH9	0xEE	PCA1 Capture 9 High	350
PCA1CPH10	0xFE	PCA1 Capture 10 High	350
PCA1CPH11	0xCF	PCA1 Capture 11 High	350
PCA1CPL6	0xFB	PCA1 Capture 6 Low	350
PCA1CPL7	0xE9	PCA1 Capture 7 Low	350
PCA1CPL8	0xEB	PCA1 Capture 8 Low	350
PCA1CPL9	0xED	PCA1 Capture 9 Low	350
PCA1CPL10	0xFD	PCA1 Capture 10 Low	350

18. External Data Memory Interface and On-Chip XRAM

For C8051F58x/F59x devices, 8 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 C8051F580/1/4/5 and C8051F588/9-F590/1 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 138 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
```

18.5. Memory Mode Selection

The external data memory space can be configured in one of four modes, shown in Figure 18.3, based on the EMIF Mode bits in the EMI0CF register (SFR Definition 18.2). These modes are summarized below. More information about the different modes can be found in Section “18.6. Timing” on page 167.

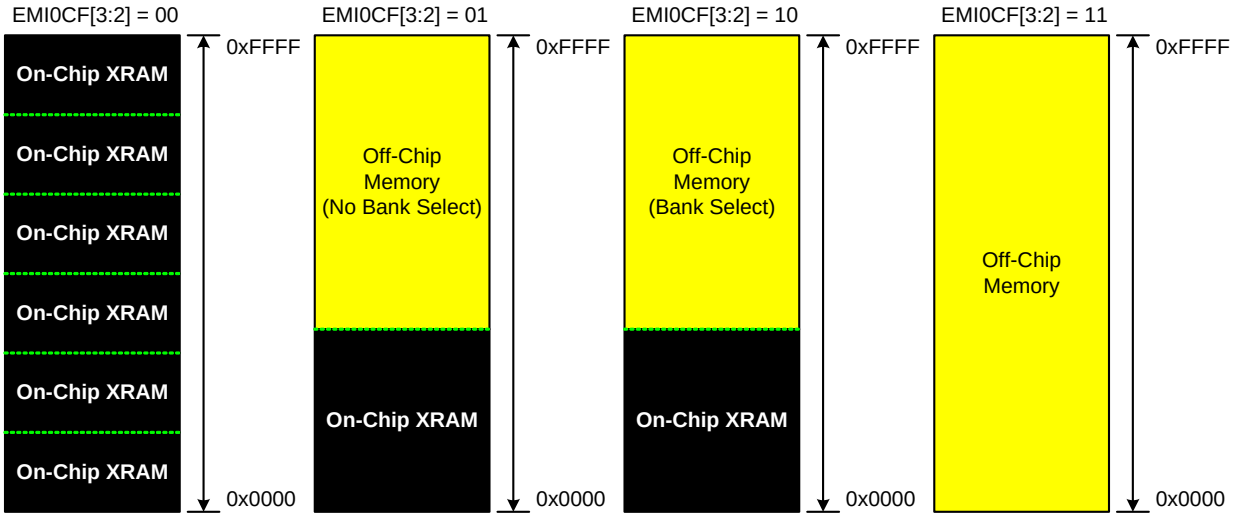

Figure 18.3. EMIF Operating Modes

18.5.1. Internal XRAM Only

When bits EMI0CF[3:2] are set to 00, all MOVX instructions will target the internal XRAM space on the device. Memory accesses to addresses beyond the populated space will wrap on 8 kB boundaries. As an example, the addresses 0x2000 and 0x4000 both evaluate to address 0x0000 in on-chip XRAM space.

- 8-bit MOVX operations use the contents of EMI0CN to determine the high-byte of the effective address and R0 or R1 to determine the low-byte of the effective address.
- 16-bit MOVX operations use the contents of the 16-bit DPTR to determine the effective address.

18.5.2. Split Mode without Bank Select

When bit EMI0CF.[3:2] are set to 01, the XRAM memory map is split into two areas, on-chip space and off-chip space.

- Effective addresses below the internal XRAM size boundary will access on-chip XRAM space.
- Effective addresses above the internal XRAM size boundary will access off-chip space.
- 8-bit MOVX operations use the contents of EMI0CN to determine whether the memory access is on-chip or off-chip. However, in the “No Bank Select” mode, an 8-bit MOVX operation will not drive the upper 8-bits A[15:8] of the Address Bus during an off-chip access. This allows the user to manipulate the upper address bits at will by setting the Port state directly via the port latches. This behavior is in contrast with “Split Mode with Bank Select” described below. The lower 8-bits of the Address Bus A[7:0] are driven, determined by R0 or R1.
- 16-bit MOVX operations use the contents of DPTR to determine whether the memory access is on-chip or off-chip, and unlike 8-bit MOVX operations, the full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

19.3. Clock Multiplier

The Clock Multiplier generates an output clock which is 4 times the input clock frequency scaled by a programmable factor of 1, 2/3, 2/4 (or 1/2), 2/5, 2/6 (or 1/3), or 2/7. The Clock Multiplier's input can be selected from the external oscillator, or the internal or external oscillators divided by 2. This produces three possible base outputs which can be scaled by a programmable factor: Internal Oscillator x 2, External Oscillator x 2, or External Oscillator x 4. See Section 19.1 on page 176 for details on system clock selection.

The Clock Multiplier is configured via the CLKMUL register (SFR Definition 19.5). The procedure for configuring and enabling the Clock Multiplier is as follows:

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

Important Note: When using an external oscillator as the input to the Clock Multiplier, the external source must be enabled and stable before the Multiplier is initialized. See "19.4. External Oscillator Drive Circuit" on page 183 for details on selecting an external oscillator source.

The Clock Multiplier allows faster operation of the CIP-51 core and is intended to generate an output frequency between 25 and 50 MHz. The clock multiplier can also be used with slow input clocks. However, if the clock is below the minimum Clock Multiplier input frequency (FCM_{min}), the generated clock will consist of four fast pulses followed by a long delay until the next input clock rising edge. The average frequency of the output is equal to 4x the input, but the instantaneous frequency may be faster. See Figure 19.2 below for more information.

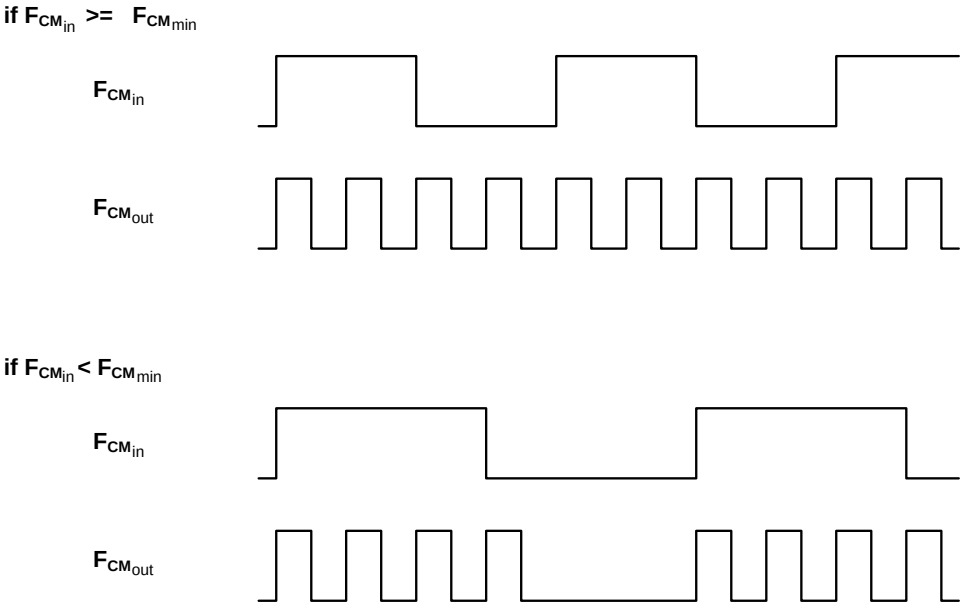

Figure 19.2. Example Clock Multiplier Output

SFR Definition 19.5. CLKMUL: Clock Multiplier

Bit	7	6	5	4	3	2	1	0
Name	MULEN	MULINIT	MULRDY	MULDIV[2:0]			MULSEL[1:0]	
Type	R/W	R/W	R	R/W			R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x97; SFR Page = 0x0F;

Bit	Name	Function
7	MULEN	**Clock Multiplier Enable.** 0: Clock Multiplier disabled. 1: Clock Multiplier enabled.
6	MULINIT	**Clock Multiplier Initialize.** This bit is 0 when the Clock Multiplier is enabled. Once enabled, writing a 1 to this bit will initialize the Clock Multiplier. The MULRDY bit reads 1 when the Clock Multiplier is stabilized.
5	MULRDY	**Clock Multiplier Ready.** 0: Clock Multiplier is not ready. 1: Clock Multiplier is ready (PLL is locked).
4:2	MULDIV[2:0]	**Clock Multiplier Output Scaling Factor.** 000: Clock Multiplier Output scaled by a factor of 1. 001: Clock Multiplier Output scaled by a factor of 1. 010: Clock Multiplier Output scaled by a factor of 1. 011: Clock Multiplier Output scaled by a factor of 2/3*. 100: Clock Multiplier Output scaled by a factor of 2/4 (1/2). 101: Clock Multiplier Output scaled by a factor of 2/5*. 110: Clock Multiplier Output scaled by a factor of 2/6 (1/3). 111: Clock Multiplier Output scaled by a factor of 2/7*. ***Note:** The Clock Multiplier output duty cycle is not 50% for these settings.
1:0	MULSEL[1:0]	**Clock Multiplier Input Select.** These bits select the clock supplied to the Clock Multiplier

MULSEL[1:0]	Selected Input Clock	Clock Multiplier Output for MULDIV[2:0] = 000b
00	Internal Oscillator	Internal Oscillator x 2
01	External Oscillator	External Oscillator x 2
10	Internal Oscillator	Internal Oscillator x 4
11	External Oscillator	External Oscillator x 4

Notes: The maximum system clock is 50 MHz, and so the Clock Multiplier output should be scaled accordingly. If Internal Oscillator x 2 or External Oscillator x 2 is selected using the MULSEL bits, MULDIV[2:0] is ignored.

SFR Definition 20.16. P0SKIP: Port 0 Skip

Bit	7	6	5	4	3	2	1	0
Name	P0SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD4; SFR Page = 0x0F

Bit	Name	Function
7:0	P0SKIP[7:0]	**Port 0 Crossbar Skip Enable Bits.** These bits select Port 0 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar. 0: Corresponding P0.n pin is not skipped by the Crossbar. 1: Corresponding P0.n pin is skipped by the Crossbar.

SFR Definition 20.17. P1: Port 1

Bit	7	6	5	4	3	2	1	0
Name	P1[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0x90; SFR Page = All Pages; Bit-Addressable

Bit	Name	Description	Write	Read
7:0	P1[7:0]	**Port 1 Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P1.n Port pin is logic LOW. 1: P1.n Port pin is logic HIGH.

22.2. CAN Registers

CAN registers are classified as follows:

1. **CAN Controller Protocol Registers**: CAN control, interrupt, error control, bus status, test modes.
2. **Message Object Interface Registers**: Used to configure 32 Message Objects, send and receive data to and from Message Objects. The CIP-51 MCU accesses the CAN message RAM via the Message Object Interface Registers. Upon writing a message object number to an IF1 or IF2 Command Request Register, the contents of the associated Interface Registers (IF1 or IF2) will be transferred to or from the message object in CAN RAM.
3. **Message Handler Registers**: These read only registers are used to provide information to the CIP-51 MCU about the message objects (MSGVLD flags, Transmission Request Pending, New Data Flags) and Interrupts Pending (which Message Objects have caused an interrupt or status interrupt condition).

For the registers other than CAN0CFG, refer to the Bosch CAN User's Guide for information on the function and use of the CAN Control Protocol Registers.

22.2.1. CAN Controller Protocol Registers

The CAN Control Protocol Registers are used to configure the CAN controller, process interrupts, monitor bus status, and place the controller in test modes.

The registers are: CAN Control Register (CAN0CN), CAN Clock Configuration (CAN0CFG), CAN Status Register (CAN0STA), CAN Test Register (CAN0TST), Error Counter Register, Bit Timing Register, and the Baud Rate Prescaler (BRP) Extension Register.

22.2.2. Message Object Interface Registers

There are two sets of Message Object Interface Registers used to configure the 32 Message Objects that transmit and receive data to and from the CAN bus. Message objects can be configured for transmit or receive, and are assigned arbitration message identifiers for acceptance filtering by all CAN nodes.

Message Objects are stored in Message RAM, and are accessed and configured using the Message Object Interface Registers.

22.2.3. Message Handler Registers

The Message Handler Registers are read only registers. The message handler registers provide interrupt, error, transmit/receive requests, and new data information.

23.5.1. 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 23.5 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.

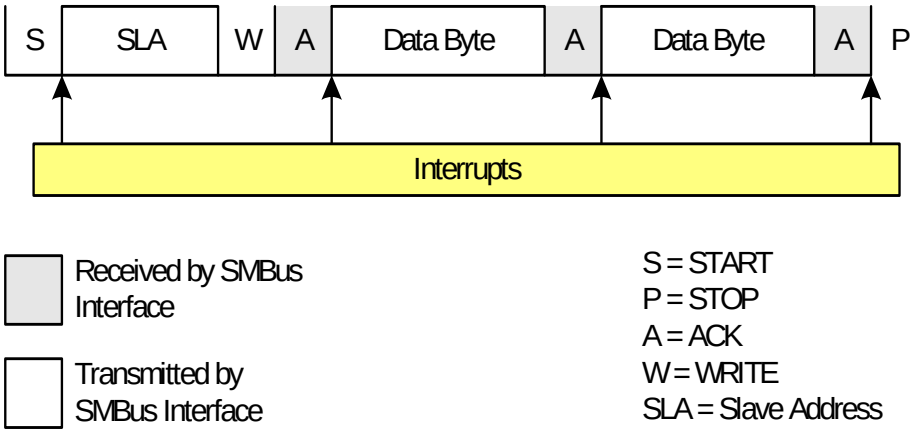

Figure 23.5. Typical Master Write Sequence

25.2. Operational Modes

UART1 provides standard asynchronous, full duplex communication. The UART mode (8-bit or 9-bit) is selected by the S1MODE bit (SCON1.7). Typical UART connection options are shown in Figure 25.3.

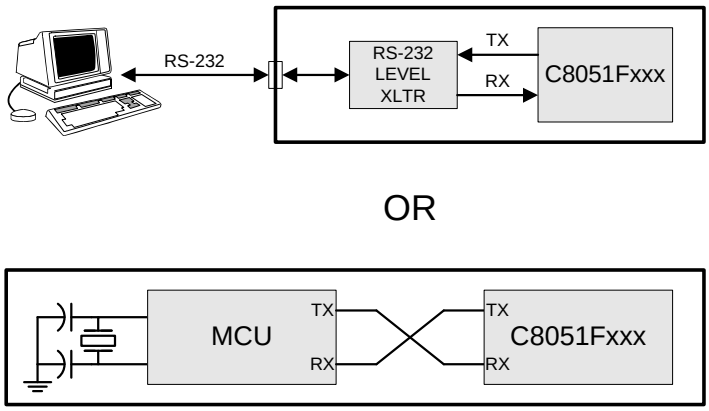

Figure 25.3. UART Interconnect Diagram

25.2.1. 8-Bit UART

8-Bit UART mode uses a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted LSB first from the TX1 pin and received at the RX1 pin. On receive, the eight data bits are stored in SBUF1 and the stop bit goes into RB81 (SCON1.2).

Data transmission begins when software writes a data byte to the SBUF1 register. The TI1 Transmit Interrupt Flag (SCON1.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN1 Receive Enable bit (SCON1.4) is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF1 receive register if the following conditions are met: RI0 must be logic 0, and if MCE0 is logic 1, the stop bit must be logic 1. In the event of a receive data overrun, the first received 8 bits are latched into the SBUF1 receive register and the following overrun data bits are lost.

If these conditions are met, the eight bits of data is stored in SBUF1, the stop bit is stored in RB81 and the RI1 flag is set. If these conditions are not met, SBUF1 and RB81 will not be loaded and the RI1 flag will not be set. An interrupt will occur if enabled when either TI1 or RI1 is set.

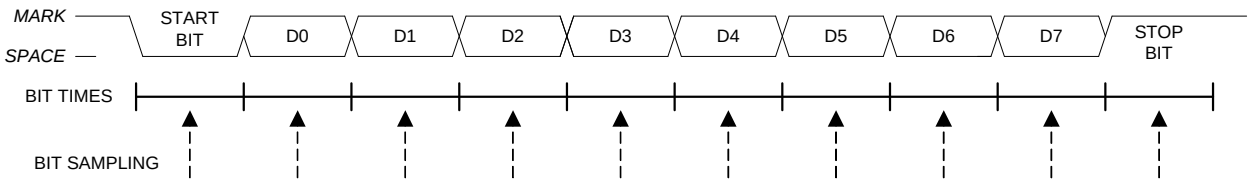

Figure 25.4. 8-Bit UART Timing Diagram

25.2.2. 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 TB81 (SCON1.3), 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 RB81 (SCON1.2) and the stop bit is ignored.

SFR Definition 27.18. TMRnCN: Timer 4 and 5 Control

Bit	7	6	5	4	3	2	1	0
Name	TFn	EXFn			EXEn	TRn	CTn	CPRLn
Type	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

TMR4CN SFR Address = 0xC8; Bit-Addressable; SFR Page = 0x10

TMR5CN SFR Address = 0x91; SFR Page = 0x10;

Bit	Name	Function
7	TFn	**Timer 4 and 5 16-bit Overflow/Underflow Flag.** Set by hardware when either the Timer overflows from 0xFFFF to 0x0000, underflows from the value placed in TMRnCAPH:TMRnCAPL to 0xFFFF (in Auto-reload Mode), or underflows from 0x0000 to 0xFFFF (in Capture Mode). When the Timer interrupt is enabled, setting this bit causes the CPU to vector to the Timer interrupt service routine. This bit is not automatically cleared by hardware.
6	EXFn	**Timer 4 and 5 External Flag.** Set by hardware when either a capture or reload is caused by a high-to-low transition on the TnEX input pin and EXENn is logic 1. This bit is not automatically cleared by hardware.
5:4	Reserved	Must Write 00b.
3	EXEn	**Timer 4 and 5 External Enable.** Enables high-to-low transitions on TnEX to trigger captures, reloads, and control the direction of the timer/counter (up or down count). If DCENn = 1, TnEX will determine if the timer counts up or down when in Auto-reload Mode. If EXENn = 1, TnEX should be configured as a digital input. 0: Transitions on the TnEX pin are ignored. 1: Transitions on the TnEX pin cause capture, reload, or control the direction of timer count (up or down) as follows: **Capture Mode**: '1'-to-'0' Transition on TnEX pin causes TMRnCAPH:TMRnCAPL to capture timer value. **Auto-Reload Mode**: DCENn = 0: '1'-to-'0' transition causes reload of timer and sets the EXFn Flag. DCENn = 1: TnEX logic level controls direction of timer (up or down).
2	TRn	**Timer 4 and 5 Run Control.** 0: Timer is disabled. 1: Timer enabled and running / counting.
1	CTn	**Timer 4 and 5 Counter / Timer Select.** 0: Timer Function: Timer incremented by clocked defined in TnM1:TnM0 (TMRnCF). 1: Counter Function: Timer incremented by high-to-low transitions on TnEX pin.
0	CPRLn	**Timer 4 and 5 Capture / Reload Select.** 0: Timer is in Auto-Reload mode. 1: Timer is in Capture mode.

28.5. Register Descriptions for PCA0

Following are detailed descriptions of the special function registers related to the operation of the PCA.

SFR Definition 28.1. PCA0CN: PCA0 Control

Bit	7	6	5	4	3	2	1	0
Name	CF	CR	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
Type	R/W	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 = 0xD8; Bit-Addressable; SFR Page = 0x00

Bit	Name	Function
7	CF	**PCA0 Counter/Timer Overflow Flag.** Set by hardware when the PCA0 Counter/Timer overflows from 0xFFFF to 0x0000. When the Counter/Timer Overflow (CF) interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
6	CR	**PCA0 Counter/Timer Run Control.** This bit enables/disables the PCA0 Counter/Timer. 0: PCA0 Counter/Timer disabled. 1: PCA0 Counter/Timer enabled.
5	CCF5	**PCA0 Module 5 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF5 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
4	CCF4	**PCA0 Module 4 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF4 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
3	CCF3	**PCA0 Module 3 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF3 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
2	CCF2	**PCA0 Module 2 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF2 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
1	CCF1	**PCA0 Module 1 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF1 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
0	CCF0	**PCA0 Module 0 Capture/Compare Flag.** This bit is set by hardware when a match or capture occurs. When the CCF0 interrupt is enabled, setting this bit causes the CPU to vector to the PCA0 interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

SFR Definition 29.3. PCA1PWM: PCA1 PWM Configuration

Bit	7	6	5	4	3	2	1	0
Name	ARSEL1	ECOV1	COVF1				CLSEL1[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 = 0xDA; SFR Page = 0x0F

Bit	Name	Function
7	ARSEL1	**Auto-Reload Register Select.** This bit selects whether to read and write the normal PCA1 capture/compare registers (PCA1CPn), 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 PCA1CPHn and PCA1CPLn. 1: Read/Write Auto-Reload Registers at PCA1CPHn and PCA1CPLn.
6	ECOV1	**Cycle Overflow Interrupt Enable.** This bit sets the masking of the Cycle Overflow Flag (COVF1) interrupt. 0: COVF1 will not generate PCA1 interrupts. 1: A PCA1 interrupt will be generated when COVF1 is set.
5	COVF1	**Cycle Overflow Flag.** This bit indicates an overflow of the 8th, 9th, 10th, or 11th bit of the main PCA1 counter (PCA1). 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	Read = 000b; Write = Don't care.
1:0	CLSEL1[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.

30. C2 Interface

C8051F58x/F59x devices include an on-chip Silicon Labs 2-Wire (C2) debug interface to allow Flash programming and in-system debugging with the production part installed in the end application. The C2 interface uses a clock signal (C2CK) and a bi-directional C2 data signal (C2D) to transfer information between the device and a host system. See the C2 Interface Specification for details on the C2 protocol.

30.1. C2 Interface Registers

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

C2 Register Definition 30.1. C2ADD: C2 Address

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

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