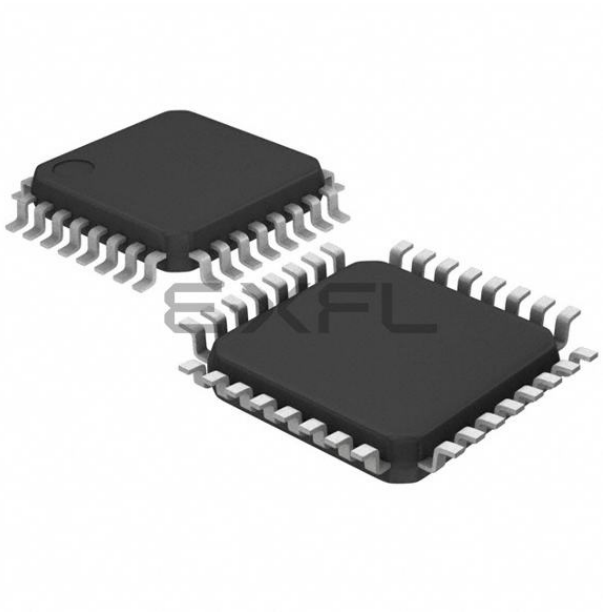

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	SMBus (2-Wire/I²C), CANbus, LINbus, SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	25
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 25x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-QFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f560-iq

Table 5.3. Port I/O DC Electrical Characteristics

V_{DD} = 1.8 to 2.75 V, –40 to +125 °C unless otherwise specified.

Parameters	Conditions	Min	Typ	Max	Units
Output High Voltage	I_{OH} = –3 mA, Port I/O push-pull	V_{IO} – 0.4	—	—	V
	I_{OH} = –10 µA, Port I/O push-pull	V_{IO} – 0.02	—	—	
	I_{OH} = –10 mA, Port I/O push-pull	—	V_{IO} – 0.7	—	
Output Low Voltage	**V_{IO} = 1.8 V:**				
	I_{OL} = 70 µA	—	—	50	
	I_{OL} = 8.5 mA	—	—	750	
	V_{IO} = 2.7 V:				
	I_{OL} = 70 µA	—	—	45	mV
	I_{OL} = 8.5 mA	—	—	550	
	V_{IO} = 5.25 V:				
	I_{OL} = 70 µA	—	—	40	
	I_{OL} = 8.5 mA	—	—	400	
Input High Voltage	V_{REGIN} = 5.25 V	0.7 x VIO	—		V
Input Low Voltage	V_{REGIN} = 2.7 V	—	—	0.3 x VIO	V
Input Leakage Current	Weak Pullup Off	—	—	±2	µA
	Weak Pullup On, V_{IO} = 2.1 V, V_{IN} = 0 V, V_{DD} = 1.8 V	—	7	9	
	Weak Pullup On, V_{IO} = 2.6 V, V_{IN} = 0 V, V_{DD} = 2.6 V	—	17	22	
	Weak Pullup On, V_{IO} = 5.0 V, V_{IN} = 0 V, V_{DD} = 2.6 V	—	49	115	

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.

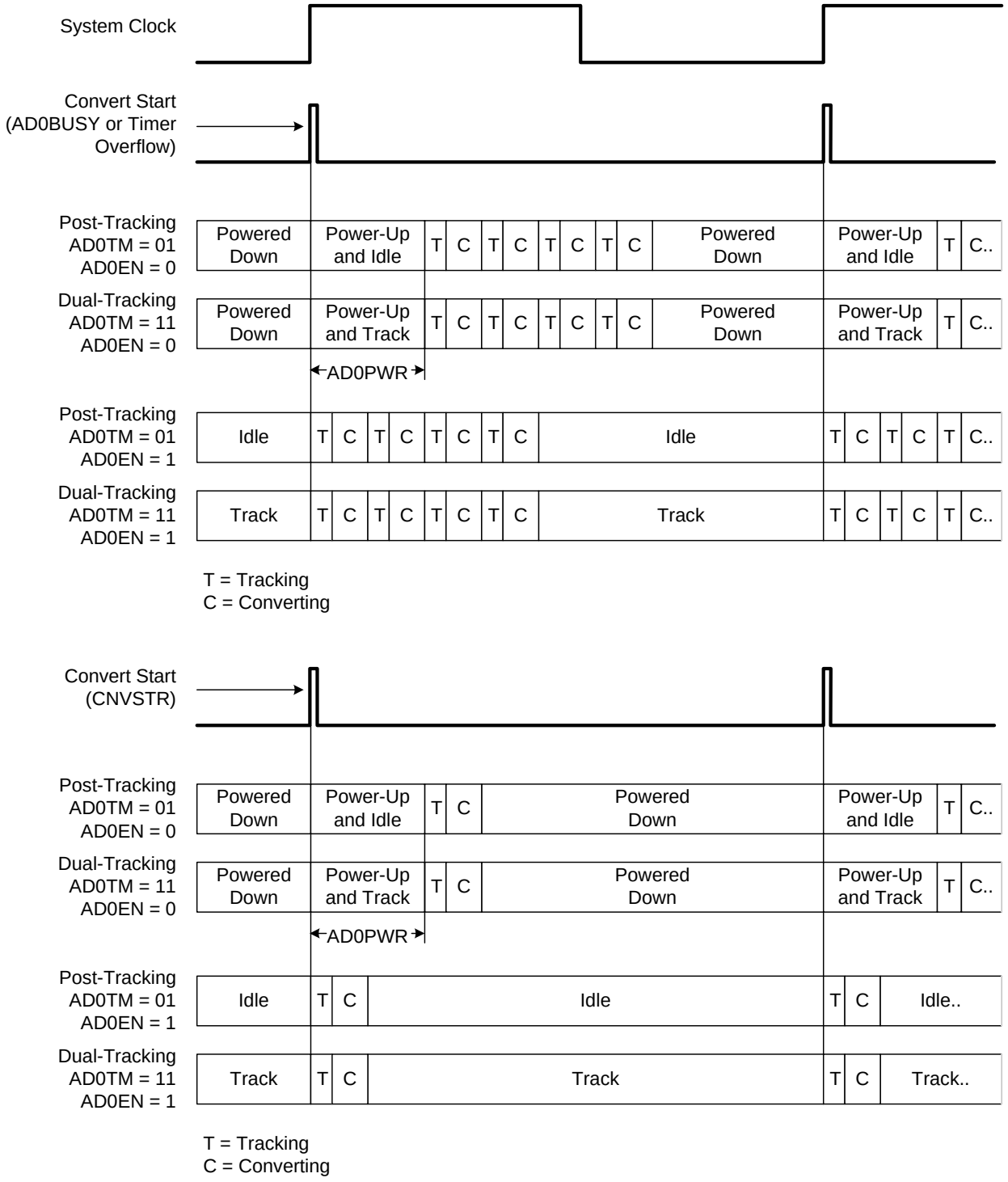

Figure 6.4. 12-Bit ADC Burst Mode Example With Repeat Count Set to 4

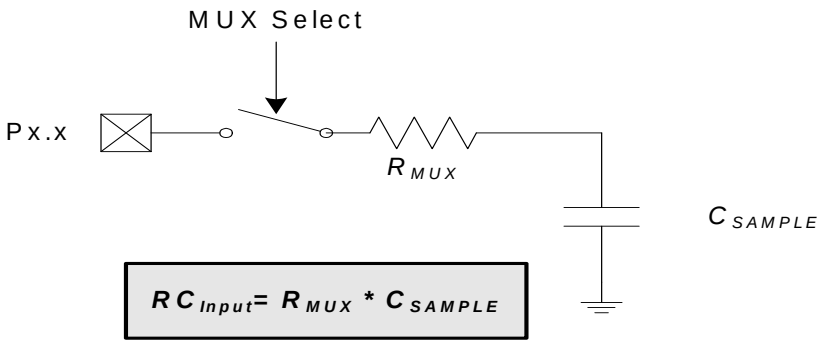

Figure 6.5. ADC0 Equivalent Input Circuit

6.3. Selectable Gain

ADC0 on the C8051F55x/56x/57x 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

SFR Definition 6.8. ADC0TK: ADC0 Tracking Mode Select

Bit	7	6	5	4	3	2	1	0
Name	AD0PWR[3:0]				AD0TM[1:0]		AD0TK[1:0]	
Type	R/W				R/W		R/W	
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xBA; SFR Page = 0x00

Bit	Name	Function
7:4	AD0PWR[3:0]	**ADC0 Burst Power-up Time.** For BURSTEN = 0: ADC0 Power state controlled by AD0EN For BURSTEN = 1, AD0EN = 1: ADC0 remains enabled and does not enter the very low power state For BURSTEN = 1, AD0EN = 0: ADC0 enters the very low power state and is enabled after each convert start signal. The Power-up time is programmed according the following equation: $AD0PWR = \frac{Tstartup}{200ns} - 1$ or $Tstartup = (AD0PWR + 1)200ns$
3:2	AD0TM[1:0]	**ADC0 Tracking Mode Enable Select Bits.** 00: Reserved. 01: ADC0 is configured to Post-Tracking Mode. 10: ADC0 is configured to Pre-Tracking Mode. 11: ADC0 is configured to Dual Tracking Mode.
1:0	AD0TK[1:0]	**ADC0 Post-Track Time.** 00: Post-Tracking time is equal to 2 SAR clock cycles + 2 FCLK cycles. 01: Post-Tracking time is equal to 4 SAR clock cycles + 2 FCLK cycles. 10: Post-Tracking time is equal to 8 SAR clock cycles + 2 FCLK cycles. 11: Post-Tracking time is equal to 16 SAR clock cycles + 2 FCLK cycles.

6.4. Programmable Window Detector

The ADC Programmable Window Detector continuously compares the ADC0 output registers to user-programmed limits, and notifies the system when a desired condition is detected. This is especially effective in an interrupt-driven system, saving code space and CPU bandwidth while delivering faster system response times. The window detector interrupt flag (AD0WINT in register ADC0CN) can also be used in polled mode. The ADC0 Greater-Than (ADC0GTH, ADC0GTL) and Less-Than (ADC0LTH, ADC0LTL) registers hold the comparison values. The window detector flag can be programmed to indicate when measured data is inside or outside of the user-programmed limits, depending on the contents of the ADC0 Less-Than and ADC0 Greater-Than registers.

SFR Definition 8.6. CPT1MX: Comparator1 MUX Selection

Bit	7	6	5	4	3	2	1	0
Name	CMX1N[3:0]				CMX1P[3:0]			
Type	R/W				R/W			
Reset	0	1	1	1	0	1	1	1

SFR Address = 0x9F; SFR Page = 0x00

Bit	Name	Function
7:4	CMX1N[3:0]	**Comparator1 Negative Input MUX Selection.** 0000: P0.1 0001: P0.3 0010: P0.5 0011: P0.7 0100: P1.1 0101: P1.3 0110: P1.5 0111: P1.7 1000: P2.1 1001: P2.3 (only available on 40-pin and 32-pin devices) 1010: P2.5 (only available on 40-pin and 32-pin devices) 1011: P2.7 (only available on 40-pin and 32-pin devices) 1100–1111: None
3:0	CMX1P[3:0]	**Comparator1 Positive Input MUX Selection.** 0000: P0.0 0001: P0.2 0010: P0.4 0011: P0.6 0100: P1.0 0101: P1.2 0110: P1.4 0111: P1.6 1000: P2.0 1001: P2.2 (only available on 40-pin and 32-pin devices) 1010: P2.4 (only available on 40-pin and 32-pin devices) 1011: P2.6 (only available on 40-pin and 32-pin devices) 1100–1111: None

Table 10.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
SETB C	Set Carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement Carry	1	1
CPL bit	Complement direct bit	2	2
ANL C, bit	AND direct bit to Carry	2	2
ANL C, /bit	AND complement of direct bit to Carry	2	2
ORL C, bit	OR direct bit to carry	2	2
ORL C, /bit	OR complement of direct bit to Carry	2	2
MOV C, bit	Move direct bit to Carry	2	2
MOV bit, C	Move Carry to direct bit	2	2
JC rel	Jump if Carry is set	2	2/(4-6)*
JNC rel	Jump if Carry is not set	2	2/(4-6)*
JB bit, rel	Jump if direct bit is set	3	3/(5-7)*
JNB bit, rel	Jump if direct bit is not set	3	3/(5-7)*
JBC bit, rel	Jump if direct bit is set and clear bit	3	3/(5-7)*
Program Branching			
ACALL addr11	Absolute subroutine call	2	4-6*
LCALL addr16	Long subroutine call	3	5-7*
RET	Return from subroutine	1	6-8*
RETI	Return from interrupt	1	6-8*
AJMP addr11	Absolute jump	2	4-6*
LJMP addr16	Long jump	3	5-7*
SJMP rel	Short jump (relative address)	2	4-6*
JMP @A+DPTR	Jump indirect relative to DPTR	1	3-5*
JZ rel	Jump if A equals zero	2	2/(4-6)*
JNZ rel	Jump if A does not equal zero	2	2/(4-6)*
CJNE A, direct, rel	Compare direct byte to A and jump if not equal	3	4/(6-8)*
CJNE A, #data, rel	Compare immediate to A and jump if not equal	3	3/(6-8)*
CJNE Rn, #data, rel	Compare immediate to Register and jump if not equal	3	3/(5-7)*
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	4/(6-8)*
DJNZ Rn, rel	Decrement Register and jump if not zero	2	2/(4-6)*
DJNZ direct, rel	Decrement direct byte and jump if not zero	3	3/(5-7)*
NOP	No operation	1	1

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

SFR Definition 10.1. DPL: Data Pointer Low Byte

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

SFR Address = 0x82; SFR Page = All Pages

Bit	Name	Function
7:0	DPL[7:0]	**Data Pointer Low.** The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed Flash memory or XRAM.

SFR Definition 10.2. DPH: Data Pointer High Byte

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

SFR Address = 0x83; SFR Page = All Pages

Bit	Name	Function
7:0	DPH[7:0]	**Data Pointer High.** The DPH register is the high byte of the 16-bit DPTR. DPTR is used to access indirectly addressed Flash memory or XRAM.

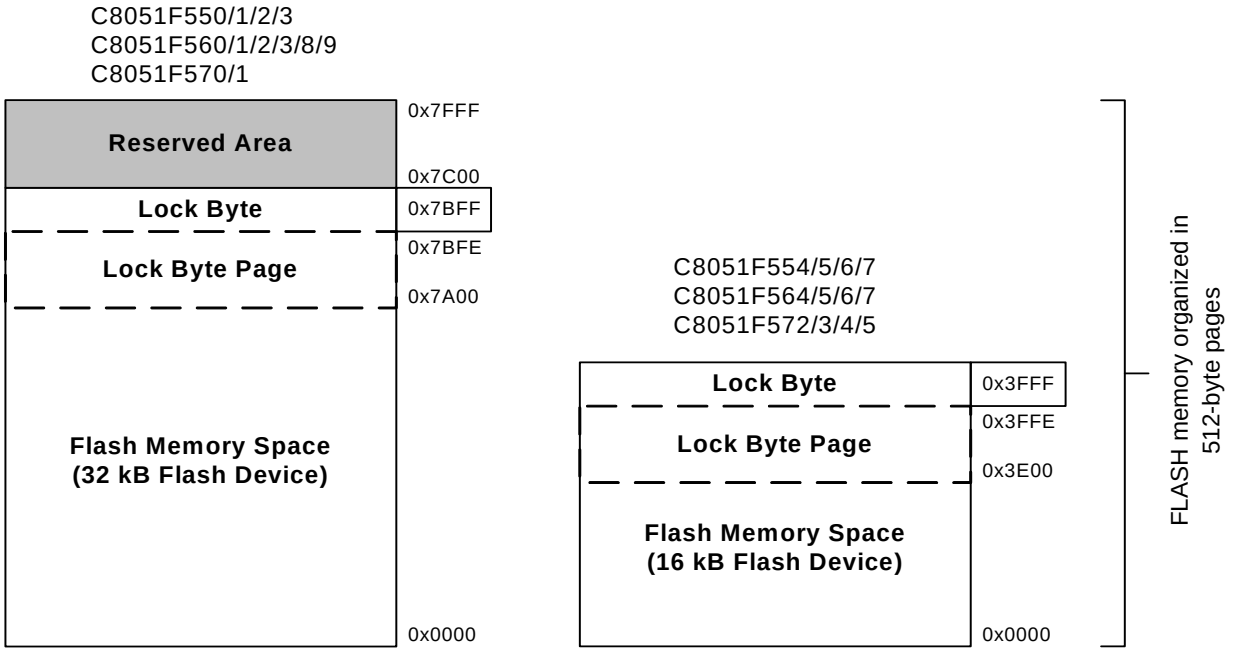

Figure 11.2. Flash Program Memory Map

11.1.1. MOVX Instruction and Program Memory

The MOVX instruction in an 8051 device is typically used to access external data memory. On the C8051F55x/56x/57x devices, the MOVX instruction is normally used to read and write on-chip XRAM, but can be re-configured to write and erase on-chip Flash memory space. MOVC instructions are always used to read Flash memory, while MOVX write instructions are used to erase and write Flash. This Flash access feature provides a mechanism for the C8051F55x/56x/57x to update program code and use the program memory space for non-volatile data storage. Refer to Section “14. Flash Memory” on page 124 for further details.

11.2. Data Memory

The C8051F55x/56x/57x devices include 2304 bytes of RAM data memory. 256 bytes of this memory is mapped into the internal RAM space of the 8051. The other 2048 bytes of this memory is on-chip “external” memory. The data memory map is shown in Figure 11.1 for reference.

11.2.1. Internal RAM

There are 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and scratch pad memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may either be addressed as bytes or as 128 bit locations accessible with the direct addressing mode.

The upper 128 bytes of data memory are accessible only by indirect addressing. This region occupies the same address space as the Special Function Registers (SFR) but is physically separate from the SFR space. The addressing mode used by an instruction when accessing locations above 0x7F determines whether the CPU accesses the upper 128 bytes of data memory space or the SFRs. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 11.1 illustrates the data memory organization of the

C8051F55x/56x/57x.

11.2.1.1. General Purpose Registers

The lower 32 bytes of data memory, locations 0x00 through 0x1F, may be addressed as four banks of general-purpose registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in SFR Definition 10.6). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

11.2.1.2. Bit Addressable Locations

In addition to direct access to data memory organized as bytes, the sixteen data memory locations at 0x20 through 0x2F are also accessible as 128 individually addressable bits. Each bit has a bit address from 0x00 to 0x7F. Bit 0 of the byte at 0x20 has bit address 0x00 while bit7 of the byte at 0x20 has bit address 0x07. Bit 7 of the byte at 0x2F has bit address 0x7F. A bit access is distinguished from a full byte access by the type of instruction used (bit source or destination operands as opposed to a byte source or destination).

The MCS-51™ assembly language allows an alternate notation for bit addressing of the form XX.B where XX is the byte address and B is the bit position within the byte. For example, the instruction:

```
MOV    C, 22.3h
```

moves the Boolean value at 0x13 (bit 3 of the byte at location 0x22) into the Carry flag.

11.2.1.3. Stack

A programmer's stack can be located anywhere in the 256-byte data memory. The stack area is designated using the Stack Pointer (SP) SFR. The SP will point to the last location used. The next value pushed on the stack is placed at SP+1 and then SP is incremented. A reset initializes the stack pointer to location 0x07. Therefore, the first value pushed on the stack is placed at location 0x08, which is also the first register (R0) of register bank 1. Thus, if more than one register bank is to be used, the SP should be initialized to a location in the data memory not being used for data storage. The stack depth can extend up to 256 bytes.

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

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

(bit addressable)

14. Flash Memory

On-chip, re-programmable Flash memory is included for program code and non-volatile data storage. The Flash memory can be programmed in-system, a single byte at a time, through the C2 interface or by software using the MOVX instruction. Once cleared to logic 0, a Flash bit must be erased to set it back to logic 1. Flash bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operation is not required. Code execution is stalled during a Flash write/erase operation. Refer to Table 5.5 for complete Flash memory electrical characteristics.

14.1. Programming The Flash Memory

The simplest means of programming the Flash memory is through the C2 interface using programming tools provided by Silicon Labs or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program Flash memory, see Section “27. C2 Interface” on page 300.

To ensure the integrity of Flash contents, it is strongly recommended that the on-chip V_{DD} Monitor be enabled in any system that includes code that writes and/or erases Flash memory from software. See Section 14.4 for more details. Before performing any Flash write or erase procedure, set the FLEWT bit in Flash Scale register (FLSCL) to 1. Also, note that 8-bit MOVX instructions cannot be used to erase or write to Flash memory at addresses higher than 0x00FF.

For –I (Industrial Grade) parts, parts programmed at a cold temperature below 0 °C may exhibit weakly programmed flash memory bits. If programmed at 0 °C or higher, there is no problem reading Flash across the entire temperature range of -40 °C to 125 °C. This temperature restriction does not apply to –A (Automotive Grade) devices.

14.1.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before Flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, Flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a Flash write or erase is attempted before the key codes have been written properly. The Flash lock resets after each write or erase; the key codes must be written again before a following Flash operation can be performed. The FLKEY register is detailed in SFR Definition 14.2.

SFR Definition 17.2. EMI0CF: External Memory Configuration

Bit	7	6	5	4	3	2	1	0
Name				Reserved	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	Reserved	Read = 0b; Must Write 0b.
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 2 kB boundary are directed on-chip. Accesses above the 2 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 2 kB boundary are directed on-chip. Accesses above the 2 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.

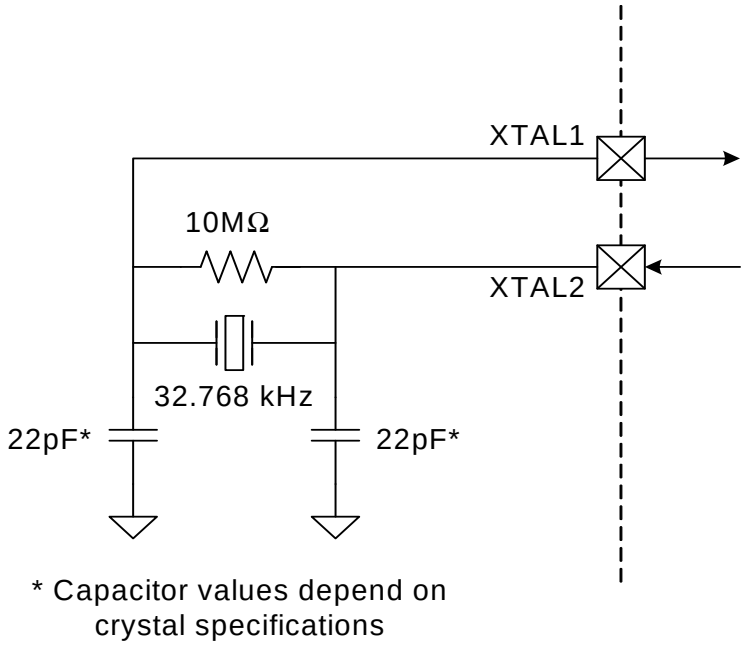

Figure 18.3. External 32.768 kHz Quartz Crystal Oscillator Connection Diagram

18.4.2. 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 18.1, 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 18.1, 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 18.1. 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 18.6, the required XFCN setting is 010b.

18.4.3. External Capacitor Example

If a capacitor is used as an external oscillator for the MCU, the circuit should be configured as shown in Figure 18.1, 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 , where f = the frequency of oscillation in MHz, C = the capacitor value in pF, and V_{DD} = the MCU power supply in Volts.

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.

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.

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.

22.5.2. Read Sequence (Master)

During a read sequence, an SMBus master reads data from a slave device. The master in this transfer will be a transmitter during the address byte, and a receiver 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 1 (READ). Serial data is then received from the slave on SDA while the SMBus outputs the serial clock. The slave transmits one or more bytes of serial data. An interrupt is generated after each received byte.

Software must write the ACK bit at that time to ACK or NACK the received byte. Writing a 1 to the ACK bit generates an ACK; writing a 0 generates a NACK. Software should write a 0 to the ACK bit for the last data transfer, to transmit a NACK. The interface exits Master Receiver Mode after the STO bit is set and a STOP is generated. The interface will switch to Master Transmitter Mode if SMB0DAT is written while an active Master Receiver. Figure 22.6 shows a typical master read sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the ‘data byte transferred’ interrupts occur **before** the ACK cycle in this mode.

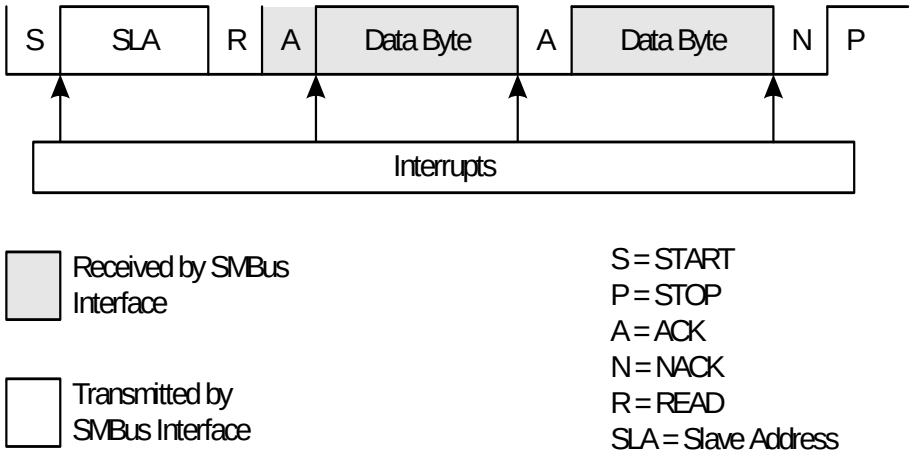

Figure 22.6. Typical Master Read Sequence

23. UART0

UART0 is an asynchronous, full duplex serial port offering a variety of data formatting options. A dedicated baud rate generator with a 16-bit timer and selectable prescaler is included, which can generate a wide range of baud rates (details in Section “23.1. Baud Rate Generator” on page 235). A received data FIFO allows UART0 to receive up to three data bytes before data is lost and an overflow occurs.

UART0 has six associated SFRs. Three are used for the Baud Rate Generator (SBCON0, SBRLH0, and SBRLL0), two are used for data formatting, control, and status functions (SCON0, SMOD0), and one is used to send and receive data (SBUF0). The single SBUF0 location provides access to both the transmit holding register and the receive FIFO. **Writes to SBUF0 always access the Transmit register. Reads of SBUF0 always access the first byte of the Receive FIFO; it is not possible to read data from the Transmit Holding Register.**

With UART0 interrupts enabled, an interrupt is generated each time a transmit is completed (TI0 is set in SCON0), or a data byte has been received (RI0 is set in SCON0). The UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART0 interrupt (transmit complete or receive complete). If additional bytes are available in the Receive FIFO, the RI0 bit cannot be cleared by software.

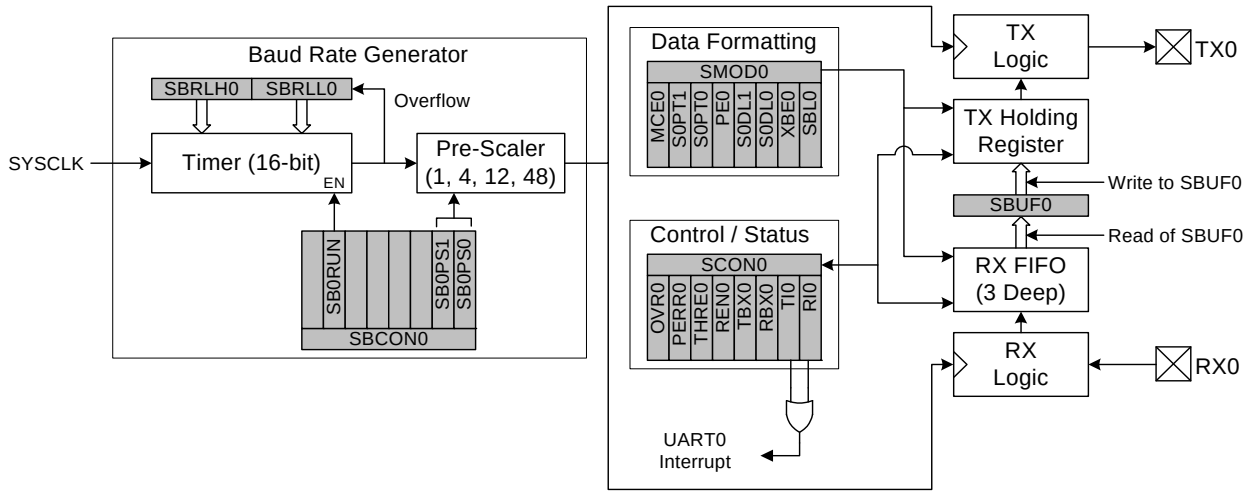

Figure 23.1. UART0 Block Diagram

23.1. Baud Rate Generator

The UART0 baud rate is generated by a dedicated 16-bit timer which runs from the controller’s core clock (SYSCLK) and has prescaler options of 1, 4, 12, or 48. The timer and prescaler options combined allow for a wide selection of baud rates over many clock frequencies.

The baud rate generator is configured using three registers: SBCON0, SBRLH0, and SBRLL0. The UART0 Baud Rate Generator Control Register (SBCON0, SFR Definition 23.4) enables or disables the baud rate generator and selects the prescaler value for the timer. The baud rate generator must be enabled for UART0 to function. Registers SBRLH0 and SBRLL0 contain a 16-bit reload value for the dedicated 16-bit timer. The internal timer counts up from the reload value on every clock tick. On timer overflows (0xFFFF to 0x0000), the timer is reloaded. The baud rate for UART0 is defined in Equation 23.1, where “BRG Clock” is the baud rate generator’s selected clock source. For reliable UART operation, it is recommended that the UART baud rate is not configured for baud rates faster than SYSCLK/16.

SFR Definition 23.5. SBRLH0: UART0 Baud Rate Generator Reload High Byte

Bit	7	6	5	4	3	2	1	0
Name	SBRLH0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xAD; SFR Page = 0x0F

Bit	Name	Function
7:0	SBRLH0[7:0]	**High Byte of Reload Value for UART0 Baud Rate Generator.** This value is loaded into the high byte of the UART0 baud rate generator when the counter overflows from 0xFFFF to 0x0000.

SFR Definition 23.6. SBRLL0: UART0 Baud Rate Generator Reload Low Byte

Bit	7	6	5	4	3	2	1	0
Name	SBRLL0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xAC; SFR Page = 0x0F

Bit	Name	Function
7:0	SBRLL0[7:0]	**Low Byte of Reload Value for UART0 Baud Rate Generator.** This value is loaded into the low byte of the UART0 baud rate generator when the counter overflows from 0xFFFF to 0x0000.

SFR Definition 26.5. PCA0L: PCA Counter/Timer Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0[7:0]							
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 = 0xF9; SFR Page = 0x00

Bit	Name	Function
7:0	PCA0[7:0]	**PCA Counter/Timer Low Byte.** The PCA0L register holds the low byte (LSB) of the 16-bit PCA Counter/Timer.

Note: When the WDTE bit is set to 1, the PCA0L register cannot be modified by software. To change the contents of the PCA0L register, the Watchdog Timer must first be disabled.

SFR Definition 26.6. PCA0H: PCA Counter/Timer High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0[15:8]							
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 = 0xFA; SFR Page = 0x00

Bit	Name	Function
7:0	PCA0[15:8]	**PCA Counter/Timer High Byte.** The PCA0H register holds the high byte (MSB) of the 16-bit PCA Counter/Timer. Reads of this register will read the contents of a “snapshot” register, whose contents are updated only when the contents of PCA0L are read (see Section 26.1).

Note: When the WDTE bit is set to 1, the PCA0H register cannot be modified by software. To change the contents of the PCA0H register, the Watchdog Timer must first be disabled.

SFR Definition 26.7. PCA0CPLn: PCA Capture Module Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPn[7:0]							
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 Addresses: PCA0CPL0 = 0xFB, PCA0CPL1 = 0xE9, PCA0CPL2 = 0xEB, PCA0CPL3 = 0xED, PCA0CPL4 = 0xFD, PCA0CPL5 = 0xCE; SFR Page (all registers) = 0x00

Bit	Name	Function
7:0	PCA0CPn[7:0]	**PCA Capture Module Low Byte.** The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n. This register address also allows access to the low byte of the corresponding PCA channel's auto-reload value for 9, 10, or 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will clear the module's ECOMn bit to a 0.		

SFR Definition 26.8. PCA0CPHn: PCA Capture Module High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPn[15:8]							
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 Addresses: PCA0CPH0 = 0xFC, PCA0CPH1 = 0xEA, PCA0CPH2 = 0xEC, PCA0CPH3 = 0xEE, PCA0CPH4 = 0xFE, PCA0CPH5 = 0xCF; SFR Page (all registers) = 0x00

Bit	Name	Function
7:0	PCA0CPn[15:8]	**PCA Capture Module High Byte.** The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n. This register address also allows access to the high byte of the corresponding PCA channel's auto-reload value for 9, 10, or 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will set the module's ECOMn bit to a 1.		