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	48 MIPS
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, 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)	2.7V ~ 5.25V
Data Converters	A/D 21x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f38b-b-gqr

1. System Overview

C8051F388/9/A/B devices are fully integrated mixed-signal System-on-a-Chip MCUs. Highlighted features are listed below. Refer to Table 1.1 for specific product feature selection.

- High-speed pipelined 8051-compatible microcontroller core (up to 48 MIPS)
- In-system, full-speed, non-intrusive debug interface (on-chip)
- Supply Voltage Regulator
- True 10-bit 500 ksps differential / single-ended ADC with analog multiplexer
- On-chip Voltage Reference and Temperature Sensor
- On-chip Voltage Comparators (2)
- Precision internal calibrated 48 MHz internal oscillator
- Internal low-frequency oscillator for additional power savings
- Up to 64 kB of on-chip Flash memory
- Up to 4352 Bytes of on-chip RAM (256 + 4 kB)
- External Memory Interface (EMIF) available on 48-pin versions.
- 2 I^2C/SMBus, 2 UARTs, and Enhanced SPI serial interfaces implemented in hardware
- Four general-purpose 16-bit timers
- Programmable Counter/Timer Array (PCA) with five capture/compare modules and Watchdog Timer function
- On-chip Power-On Reset, V_{DD} Monitor, and Missing Clock Detector
- Up to 40 Port I/O (5 V tolerant)

With on-chip Power-On Reset, V_{DD} monitor, Voltage Regulator, Watchdog Timer, and clock oscillator, C8051F388/9/A/B devices are truly stand-alone System-on-a-Chip solutions. The Flash memory can be reprogrammed in-circuit, providing non-volatile data storage, and also allowing field upgrades of the 8051 firmware. User software has complete control of all peripherals, and may individually shut down any or all peripherals for power savings.

The on-chip Silicon Labs 2-Wire (C2) Development Interface allows non-intrusive (uses no on-chip resources), full speed, in-circuit debugging using the production MCU installed in the final application. This debug logic supports inspection and modification of memory and registers, setting breakpoints, single stepping, run and halt commands. All analog and digital peripherals are fully functional while debugging using C2. The two C2 interface pins can be shared with user functions, allowing in-system debugging without occupying package pins.

Each device is specified for 2.7–5.25 V operation over the industrial temperature range (–40 to +85 °C). For voltages above 3.6 V, the on-chip Voltage Regulator must be used. The Port I/O and $\overline{RST}$ pins are tolerant of input signals up to 5 V. C8051F388/9/A/B devices are available in 48-pin TQFP, 32-pin LQFP, or 32-pin QFN packages. See Table 1.1, “Product Selection Guide,” on page 16 for feature and package choices.

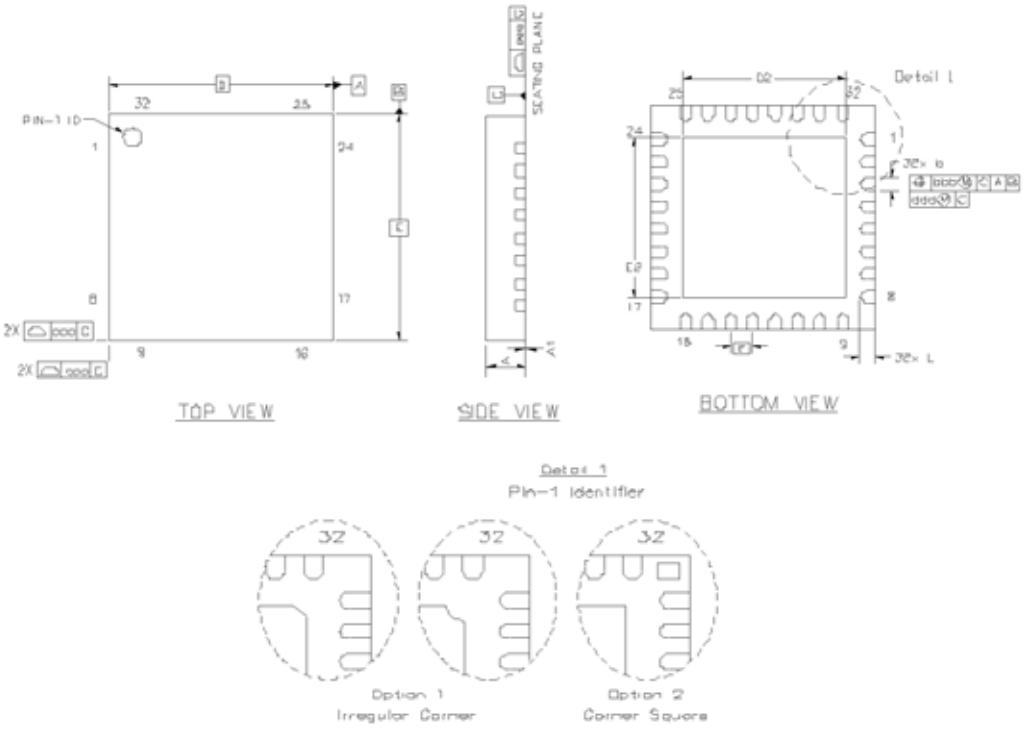

Figure 3.8. QFN-32 Package Drawing

Table 3.6. QFN-32 Package Dimensions

Dimension	Min	Typ	Max
A	0.80	0.85	0.90
A1	0.00	0.02	0.05
b	0.18	0.25	0.30
D		5.00 BSC	
D2	3.20	3.30	3.40
e		0.50 BSC	
E		5.00 BSC	

Dimension	Min	Typ	Max
E2	3.20	3.30	3.40
L	0.35	0.40	0.45
aaa	—	—	0.10
bbb	—	—	0.10
ddd	—	—	0.05
eee	—	—	0.08

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to the JEDEC Solid State Outline MO-220, variation VHHD except for custom features D2, E2, and L which are toleranced per supplier designation.
4. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

5. Electrical Characteristics

5.1. Absolute Maximum Specifications

Table 5.1. Absolute Maximum Ratings

Parameter	Conditions	Min	Typ	Max	Units
Junction Temperature Under Bias		–55	—	125	°C
Storage Temperature		–65	—	150	°C
Voltage on $\overline{RST}$, INT2, or any Port I/O Pin with Respect to GND	$V_{DD} \geq 2.2$ V $V_{DD} < 2.2$ V	–0.3 –0.3	— —	5.8 V_{DD} + 3.6	V V
Voltage on V_{DD} with Respect to GND	Regulator1 in Normal Mode Regulator1 in Bypass Mode	–0.3 –0.3	— —	4.2 1.98	V V
Maximum Total Current through V_{DD} or GND		—	—	500	mA
Maximum Output Current sunk by $\overline{RST}$ or any Port Pin		—	—	100	mA

Note: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

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 6.5. ADC0GTH: ADC0 Greater-Than Data High Byte

Bit	7	6	5	4	3	2	1	0
Name	ADC0GTH[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xC4; SFR Page = All Pages

Bit	Name	Function
7:0	ADC0GTH[7:0]	**ADC0 Greater-Than Data Word High-Order Bits.**

SFR Definition 6.6. ADC0GTL: ADC0 Greater-Than Data Low Byte

Bit	7	6	5	4	3	2	1	0
Name	ADC0GTL[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xC3; SFR Page = All Pages

Bit	Name	Function
7:0	ADC0GTL[7:0]	**ADC0 Greater-Than Data Word Low-Order Bits.**

Table 11.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through Carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through Carry	1	1
SWAP A	Swap nibbles of A	1	1
Data Transfer			
MOV A, Rn	Move Register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move indirect RAM to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to Register	1	1
MOV Rn, direct	Move direct byte to Register	2	2
MOV Rn, #data	Move immediate to Register	2	2
MOV direct, A	Move A to direct byte	2	2
MOV direct, Rn	Move Register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	3
MOV direct, @Ri	Move indirect RAM to direct byte	2	2
MOV direct, #data	Move immediate to direct byte	3	3
MOV @Ri, A	Move A to indirect RAM	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	2	2
MOV @Ri, #data	Move immediate to indirect RAM	2	2
MOV DPTR, #data16	Load DPTR with 16-bit constant	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3
MOVC A, @A+PC	Move code byte relative PC to A	1	3
MOVX A, @Ri	Move external data (8-bit address) to A	1	3
MOVX @Ri, A	Move A to external data (8-bit address)	1	3
MOVX A, @DPTR	Move external data (16-bit address) to A	1	3
MOVX @DPTR, A	Move A to external data (16-bit address)	1	3
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange Register with A	1	1
XCH A, direct	Exchange direct byte with A	2	2
XCH A, @Ri	Exchange indirect RAM with A	1	2
XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	2
Boolean Manipulation			
CLR C	Clear Carry	1	1
CLR bit	Clear direct bit	2	2
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

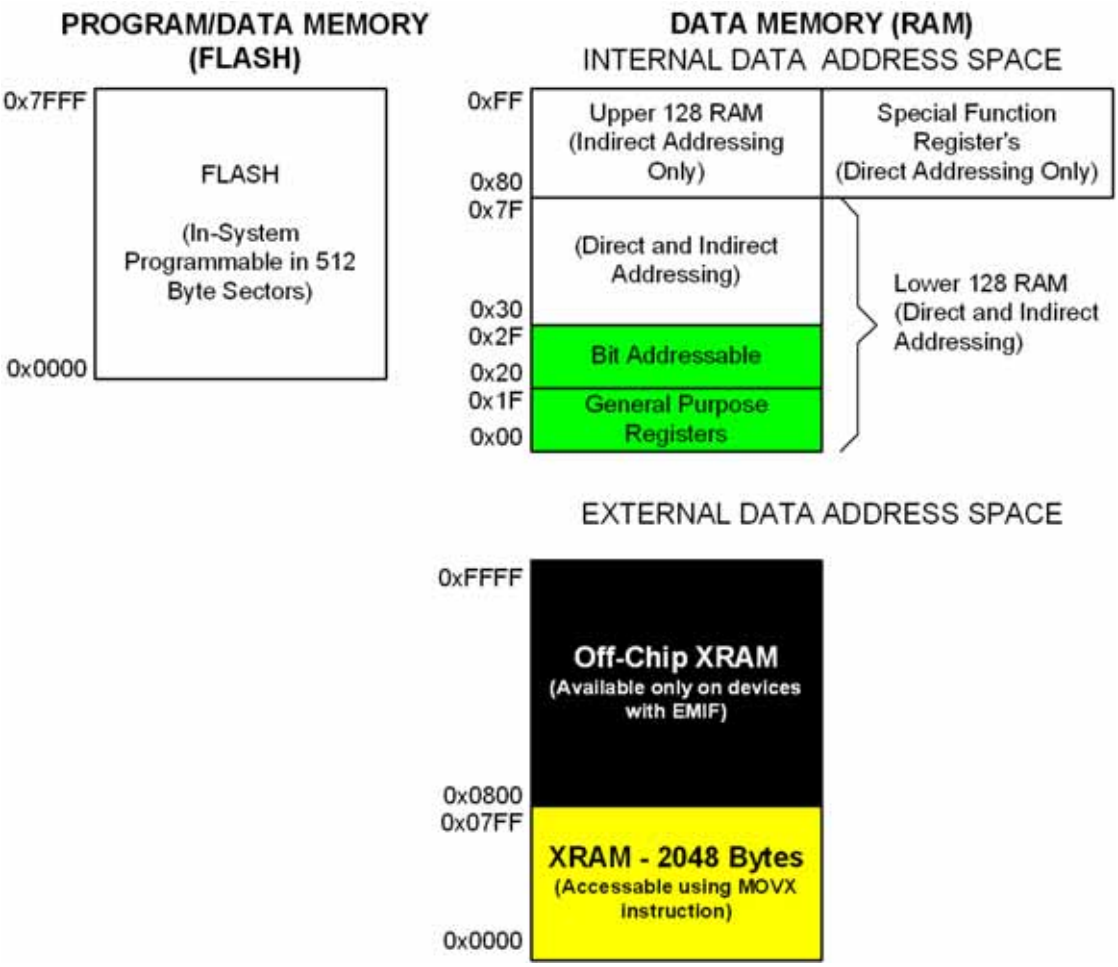

Figure 13.2. On-Chip Memory Map for 32 kB Devices (C8051F38A/B)

13.1. Program Memory

The CIP-51 core has a 64k-byte program memory space. The C8051F388/9/A/B implements 64 or 32 kB of this program memory space as in-system, re-programmable Flash memory. Note that on the C8051F388/9 (64 kB version), addresses above 0xFBFF are reserved.

Program memory is normally assumed to be read-only. However, the CIP-51 can write to program memory by setting the Program Store Write Enable bit (PSCTL.0) and using the MOVX instruction. This feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to Section “18. Flash Memory” on page 129 for further details.

13.2. Data Memory

The CIP-51 includes 256 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.

14.6.2. Multiplexed Mode

14.6.2.1. 16-bit MOVX: EMI0CF[4:2] = 001, 010, or 011

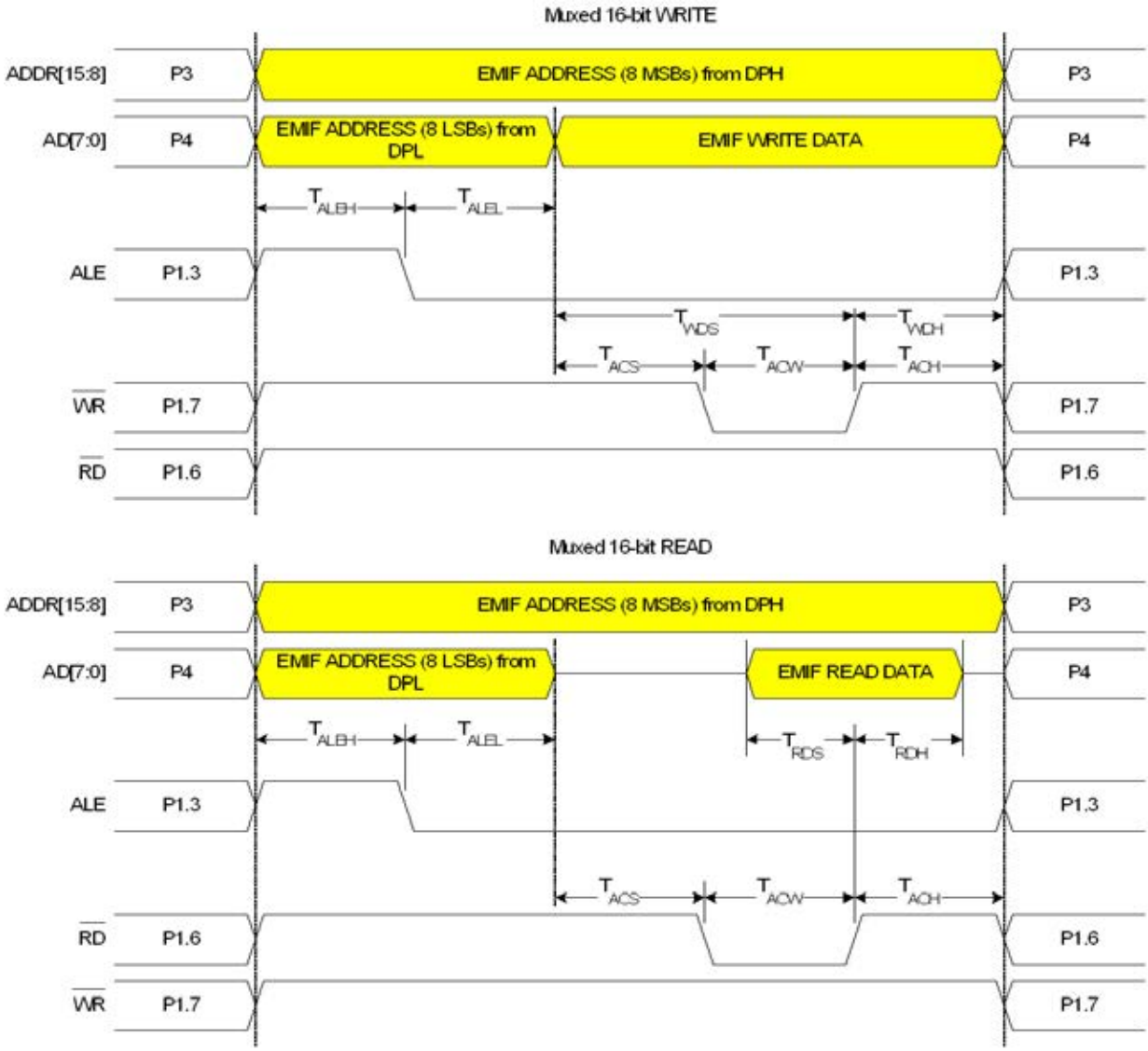

Figure 14.7. Multiplexed 16-bit MOVX Timing

14.6.2.3. 8-bit MOVX with Bank Select: EMI0CF[4:2] = 010

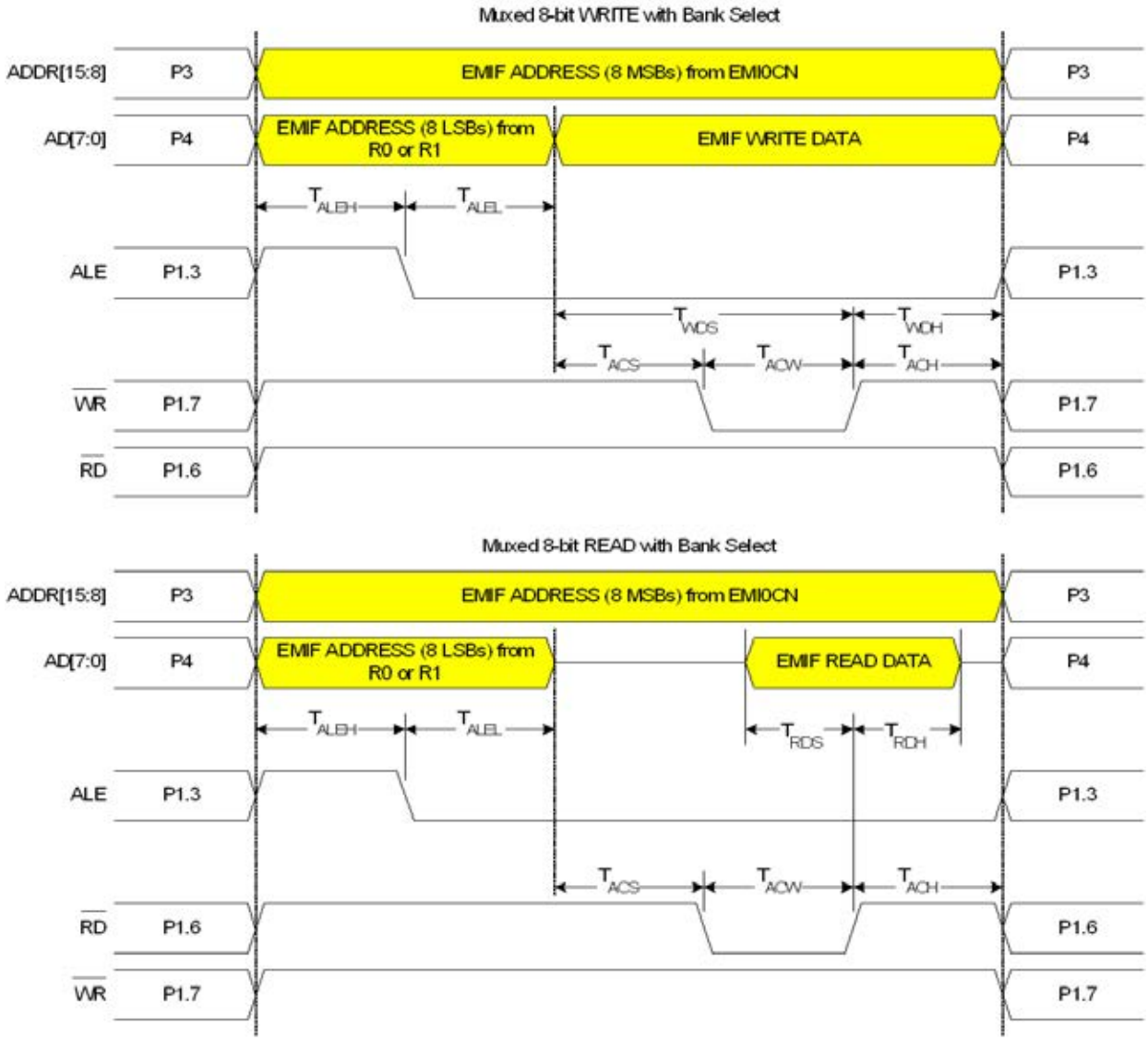

Figure 14.9. Multiplexed 8-bit MOVX with Bank Select Timing

Table 15.1. Special Function Register (SFR) Memory Map

Address	Page	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
F8		SPI0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0	PCA0CPL4	PCA0CPH4	VDM0CN
F0		B	P0MDIN	P1MDIN	P2MDIN	P3MDIN	P4MDIN	EIP1	EIP2
E8		ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2	PCA0CPL3	PCA0CPH3	RSTSRC
E0	0	ACC	XBR0	XBR1	XBR2	IT01CF	SMOD1	EIE1	EIE2
	F					CKCON1			
D8		PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	PCA0CPM3	PCA0CPM4	P3SKIP
D0		PSW	REF0CN	SCON1	SBUF1	P0SKIP	P1SKIP	P2SKIP	
C8	0	TMR2CN	REG01CN	TMR2RLL	TMR2RLH	TMR2L	TMR2H	SMB0ADM	SMB0ADR
	F	TMR5CN		TMR5RLL	TMR5RLH	TMR5L	TMR5H	SMB1ADM	SMB1ADR
C0	0	SMB0CN	SMB0CF	SMB0DAT	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	P4
	F	SMB1CN	SMB1CF	SMB1DAT					
B8	0	IP	CLKMUL	AMX0N	AMX0P	ADC0CF	ADC0L	ADC0H	SFRPAGE
	F		SMBTC						
B0		P3	OSCXCN	OSCICN	OSCICL	SBRLL1	SBRLH1	FLSCL	FLKEY
A8		IE	CLKSEL	EMI0CN		SBCON1		P4MDOUT	PFE0CN
A0		P2	SPI0CFG	SPI0CKR	SPI0DAT	P0MDOUT	P1MDOUT	P2MDOUT	P3MDOUT
98		SCON0	SBUF0	CPT1CN	CPT0CN	CPT1MD	CPT0MD	CPT1MX	CPT0MX
90	0	P1	TMR3CN	TMR3RLL	TMR3RLH	TMR3L	TMR3H		
	F		TMR4CN	TMR4RLL	TMR4RLH	TMR4L	TMR4H		
88		TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
80		P0	SP	DPL	DPH	EMI0TC	EMI0CF	OSCLCN	PCON
		0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

Notes:

1. SFR Addresses ending in 0x0 or 0x8 are bit-addressable locations and can be used with bitwise instructions.
2. Unless indicated otherwise, SFRs are available on both page 0 and page F.

Table 15.2. Special Function Registers (Continued)

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

Register	Address	Page	Description	Page
PCA0CPL2	0xEB	All Pages	PCA Capture 2 Low	276
PCA0CPL3	0xED	All Pages	PCA Capture 3 Low	276
PCA0CPL4	0xFD	All Pages	PCA Capture 4 Low	276
PCA0CPM0	0xDA	All Pages	PCA Module 0 Mode Register	274
PCA0CPM1	0xDB	All Pages	PCA Module 1 Mode Register	274
PCA0CPM2	0xDC	All Pages	PCA Module 2 Mode Register	274
PCA0CPM3	0xDD	All Pages	PCA Module 3 Mode Register	274
PCA0CPM4	0xDE	All Pages	PCA Module 4 Mode Register	274
PCA0H	0xFA	All Pages	PCA Counter High	275
PCA0L	0xF9	All Pages	PCA Counter Low	275
PCA0MD	0xD9	All Pages	PCA Mode	273
PCON	0x87	All Pages	Power Control	74
PFE0CN	0xAF	All Pages	Prefetch Engine Control	84
PSCTL	0x8F	All Pages	Program Store R/W Control	133
PSW	0xD0	All Pages	Program Status Word	83
REF0CN	0xD1	All Pages	Voltage Reference Control	59
REG01CN	0xC9	All Pages	Voltage Regulator 0 and 1 Control	71
RSTSRC	0xEF	All Pages	Reset Source Configuration/Status	128
SBCON1	0xAC	All Pages	UART1 Baud Rate Generator Control	209
SBRLH1	0xB5	All Pages	UART1 Baud Rate Generator High	209
SBRLL1	0xB4	All Pages	UART1 Baud Rate Generator Low	210
SBUF0	0x99	All Pages	UART0 Data Buffer	199
SBUF1	0xD3	All Pages	UART1 Data Buffer	208
SCON0	0x98	All Pages	UART0 Control	198
SCON1	0xD2	All Pages	UART1 Control	206
SFRPAGE	0xBF	All Pages	SFR Page Select	105
SMB0ADM	0xCE	0	SMBus0 Address Mask	180
SMB0ADR	0xCF	0	SMBus0 Address	179
SMB0CF	0xC1	0	SMBus0 Configuration	172
SMB0CN	0xC0	0	SMBus0 Control	176
SMB0DAT	0xC2	0	SMBus0 Data	182
SMB1ADM	0xCE	F	SMBus1 Address Mask	181
SMB1ADR	0xCF	F	SMBus1 Address	180
SMB1CF	0xC1	F	SMBus1 Configuration	172

16.1. MCU Interrupt Sources and Vectors

The C8051F388/9/A/B MCUs support several interrupt sources. Software can simulate an interrupt by setting any interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. MCU interrupt sources, associated vector addresses, priority order and control bits are summarized in Table 16.1. Refer to the datasheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

16.1.1. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (IP, EIP1, or EIP2) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 16.1.

16.1.2. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 6 system clock cycles: 1 clock cycle to detect the interrupt and 5 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 20 system clock cycles: 1 clock cycle to detect the interrupt, 6 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 5 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

Note that the CPU is stalled during Flash write operations. Interrupt service latency will be increased for interrupts occurring while the CPU is stalled. The latency for these situations will be determined by the standard interrupt service procedure (as described above) and the amount of time the CPU is stalled.

16.2. Interrupt Register Descriptions

The SFRs used to enable the interrupt sources and set their priority level are described in this section. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

SFR Definition 16.1. IE: Interrupt Enable

Bit	7	6	5	4	3	2	1	0
Name	EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0
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 = 0xA8; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7	EA	**Enable All Interrupts.** Globally enables/disables all interrupts. It overrides individual interrupt mask settings. 0: Disable all interrupt sources. 1: Enable each interrupt according to its individual mask setting.
6	ESPI0	**Enable Serial Peripheral Interface (SPI0) Interrupt.** This bit sets the masking of the SPI0 interrupts. 0: Disable all SPI0 interrupts. 1: Enable interrupt requests generated by SPI0.
5	ET2	**Enable Timer 2 Interrupt.** This bit sets the masking of the Timer 2 interrupt. 0: Disable Timer 2 interrupt. 1: Enable interrupt requests generated by the TF2L or TF2H flags.
4	ES0	**Enable UART0 Interrupt.** This bit sets the masking of the UART0 interrupt. 0: Disable UART0 interrupt. 1: Enable UART0 interrupt.
3	ET1	**Enable Timer 1 Interrupt.** This bit sets the masking of the Timer 1 interrupt. 0: Disable all Timer 1 interrupt. 1: Enable interrupt requests generated by the TF1 flag.
2	EX1	**Enable External Interrupt 1.** This bit sets the masking of External Interrupt 1. 0: Disable external interrupt 1. 1: Enable interrupt requests generated by the $\overline{\text{INT1}}$ input.
1	ET0	**Enable Timer 0 Interrupt.** This bit sets the masking of the Timer 0 interrupt. 0: Disable all Timer 0 interrupt. 1: Enable interrupt requests generated by the TF0 flag.
0	EX0	**Enable External Interrupt 0.** This bit sets the masking of External Interrupt 0. 0: Disable external interrupt 0. 1: Enable interrupt requests generated by the $\overline{\text{INT0}}$ input.

SFR Definition 16.3. EIE1: Extended Interrupt Enable 1

Bit	7	6	5	4	3	2	1	0
Name	ET3	ECP1	ECP0	EPCA0	EADC0	EWADC0	Reserved	ESMB0
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 = 0xE6; SFR Page = All Pages

Bit	Name	Function
7	ET3	**Enable Timer 3 Interrupt.** This bit sets the masking of the Timer 3 interrupt. 0: Disable Timer 3 interrupts. 1: Enable interrupt requests generated by the TF3L or TF3H flags.
6	ECP1	**Enable Comparator1 (CP1) Interrupt.** This bit sets the masking of the CP1 interrupt. 0: Disable CP1 interrupts. 1: Enable interrupt requests generated by the CP1RIF or CP1FIF flags.
5	ECP0	**Enable Comparator0 (CP0) Interrupt.** This bit sets the masking of the CP0 interrupt. 0: Disable CP0 interrupts. 1: Enable interrupt requests generated by the CP0RIF or CP0FIF flags.
4	EPCA0	**Enable Programmable Counter Array (PCA0) Interrupt.** This bit sets the masking of the PCA0 interrupts. 0: Disable all PCA0 interrupts. 1: Enable interrupt requests generated by PCA0.
3	EADC0	**Enable ADC0 Conversion Complete Interrupt.** This bit sets the masking of the ADC0 Conversion Complete interrupt. 0: Disable ADC0 Conversion Complete interrupt. 1: Enable interrupt requests generated by the AD0INT flag.
2	EWADC0	**Enable Window Comparison ADC0 Interrupt.** This bit sets the masking of ADC0 Window Comparison interrupt. 0: Disable ADC0 Window Comparison interrupt. 1: Enable interrupt requests generated by ADC0 Window Compare flag (AD0WINT).
1	Reserved	Read = 0b, Must Write 0b.
0	ESMB0	**Enable SMBus0 Interrupt.** This bit sets the masking of the SMB0 interrupt. 0: Disable all SMB0 interrupts. 1: Enable interrupt requests generated by SMB0.

SFR Definition 20.2. XBR1: Port I/O Crossbar Register 1

Bit	7	6	5	4	3	2	1	0
Name	WEAKPUD	XBARE	T1E	T0E	ECIE	PCA0ME[2:0]		
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 = 0xE2; SFR Page = All Pages

Bit	Name	Function
7	WEAKPUD	**Port I/O Weak Pullup Disable.** 0: Weak Pullups enabled (except for Ports whose I/O are configured for analog mode). 1: Weak Pullups disabled.
6	XBARE	**Crossbar Enable.** 0: Crossbar disabled. 1: Crossbar enabled.
5	T1E	**T1 Enable.** 0: T1 unavailable at Port pin. 1: T1 routed to Port pin.
4	T0E	**T0 Enable.** 0: T0 unavailable at Port pin. 1: T0 routed to Port pin.
3	ECIE	**PCA0 External Counter Input Enable.** 0: ECI unavailable at Port pin. 1: ECI routed to Port pin.
2:0	PCA0ME[2:0]	**PCA Module I/O Enable Bits.** 000: All PCA I/O unavailable at Port pins. 001: CEX0 routed to Port pin. 010: CEX0, CEX1 routed to Port pins. 011: CEX0, CEX1, CEX2 routed to Port pins. 100: CEX0, CEX1, CEX2, CEX3 routed to Port pins. 101: CEX0, CEX1, CEX2, CEX3 routed to Port pins. 11x: Reserved.

23.2. Data Format

UART1 has a number of available options for data formatting. Data transfers begin with a start bit (logic low), followed by the data bits (sent LSB-first), a parity or extra bit (if selected), and end with one or two stop bits (logic high). The data length is variable between 5 and 8 bits. A parity bit can be appended to the data, and automatically generated and detected by hardware for even, odd, mark, or space parity. The stop bit length is selectable between short (1 bit time) and long (1.5 or 2 bit times), and a multi-processor communication mode is available for implementing networked UART buses. All of the data formatting options can be configured using the SMOD1 register, shown in SFR Definition . Figure 23.2 shows the timing for a UART1 transaction without parity or an extra bit enabled. Figure 23.3 shows the timing for a UART1 transaction with parity enabled (PE1 = 1). Figure 23.4 is an example of a UART1 transaction when the extra bit is enabled (XBE1 = 1). Note that the extra bit feature is not available when parity is enabled, and the second stop bit is only an option for data lengths of 6, 7, or 8 bits.

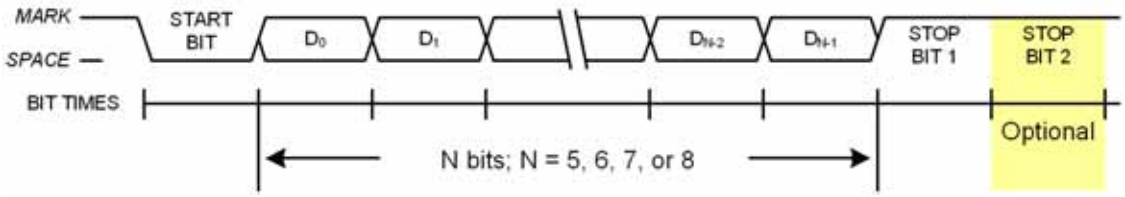

Figure 23.2. UART1 Timing Without Parity or Extra Bit

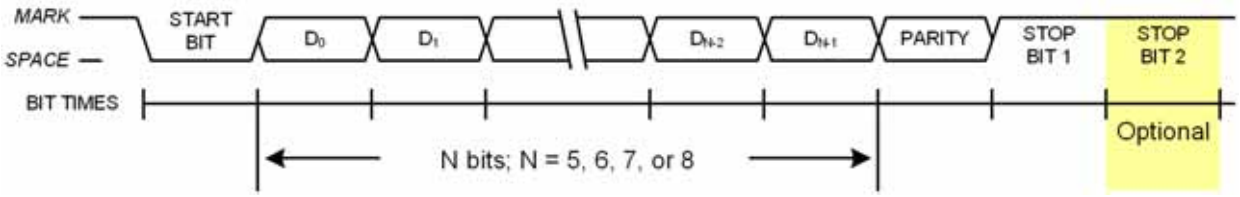

Figure 23.3. UART1 Timing With Parity

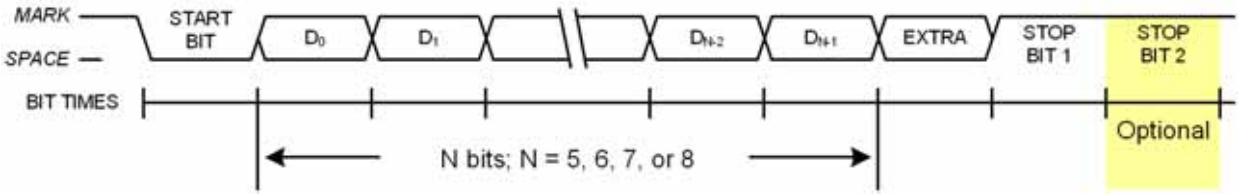

Figure 23.4. UART1 Timing With Extra Bit

24.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

All of the following bits must be cleared by software.

- The SPI Interrupt Flag, SPIF (SPI0CN.7) is set to logic 1 at the end of each byte transfer. This flag can occur in all SPI0 modes.
- The Write Collision Flag, WCOL (SPI0CN.6) is set to logic 1 if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written.This flag can occur in all SPI0 modes.
- The Mode Fault Flag MODF (SPI0CN.5) is set to logic 1 when SPI0 is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN and SPIEN bits in SPI0CN are set to logic 0 to disable SPI0 and allow another master device to access the bus.
- The Receive Overrun Flag RXOVRN (SPI0CN.4) is set to logic 1 when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

24.5. Serial Clock Phase and Polarity

Four combinations of serial clock phase and polarity can be selected using the clock control bits in the SPI0 Configuration Register (SPI0CFG). The CKPHA bit (SPI0CFG.5) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI0CFG.4) selects between an active-high or active-low clock. Both master and slave devices must be configured to use the same clock phase and polarity. SPI0 should be disabled (by clearing the SPIEN bit, SPI0CN.0) when changing the clock phase or polarity. The clock and data line relationships for master mode are shown in Figure 24.5. For slave mode, the clock and data relationships are shown in Figure 24.6 and Figure 24.7. Note that CKPHA should be set to 0 on both the master and slave SPI when communicating between two Silicon Labs C8051 devices.

The SPI0 Clock Rate Register (SPI0CKR) as shown in SFR Definition 24.3 controls the master mode serial clock frequency. This register is ignored when operating in slave mode. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency or 12.5 MHz, whichever is slower. When the SPI is configured as a slave, the maximum data transfer rate (bits/sec) for full-duplex operation is 1/10 the system clock frequency, provided that the master issues SCK, NSS (in 4-wire slave mode), and the serial input data synchronously with the slave's system clock. If the master issues SCK, NSS, and the serial input data asynchronously, the maximum data transfer rate (bits/sec) must be less than 1/10 the system clock frequency. In the special case where the master only wants to transmit data to the slave and does not need to receive data from the slave (i.e. half-duplex operation), the SPI slave can receive data at a maximum data transfer rate (bits/sec) of 1/4 the system clock frequency. This is provided that the master issues SCK, NSS, and the serial input data synchronously with the slave's system clock.

SFR Definition 25.24. TMR5CN: Timer 5 Control

Bit	7	6	5	4	3	2	1	0
Name	TF5H	TF5L	TF5LEN		T5SPLIT	TR5		T5XCLK
Type	R/W	R/W	R/W	R	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC8; SFR Page = F; Bit-Addressable

Bit	Name	Function
7	TF5H	**Timer 5 High Byte Overflow Flag.** Set by hardware when the Timer 5 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 5 overflows from 0xFFFF to 0x0000. When the Timer 5 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 5 interrupt service routine. This bit is not automatically cleared by hardware.
6	TF5L	**Timer 5 Low Byte Overflow Flag.** Set by hardware when the Timer 5 low byte overflows from 0xFF to 0x00. TF5L will be set when the low byte overflows regardless of the Timer 5 mode. This bit is not automatically cleared by hardware.
5	TF5LEN	**Timer 5 Low Byte Interrupt Enable.** When set to 1, this bit enables Timer 5 Low Byte interrupts. If Timer 5 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 5 overflows.
4	Unused	Read = 0b; Write = don't care.
3	T5SPLIT	**Timer 5 Split Mode Enable.** When this bit is set, Timer 5 operates as two 8-bit timers with auto-reload. 0: Timer 5 operates in 16-bit auto-reload mode. 1: Timer 5 operates as two 8-bit auto-reload timers.
2	TR5	**Timer 5 Run Control.** Timer 5 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR5H only; TMR5L is always enabled in split mode.
1	Unused	Read = 0b; Write = don't care.
0	T5XCLK	**Timer 5 External Clock Select.** This bit selects the external clock source for Timer 5. However, the Timer 5 Clock Select bits (T5MH and T5ML in register CKCON1) may still be used to select between the external clock and the system clock for either timer. 0: Timer 5 clock is the system clock divided by 12. 1: Timer 5 clock is the external clock divided by 8 (synchronized with SYSCLK).

SFR Definition 25.28. TMR5H Timer 5 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR5H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCD; SFR Page = F

Bit	Name	Function
7:0	TMR5H[7:0]	**Timer 5 High Byte.** In 16-bit mode, the TMR5H register contains the high byte of the 16-bit Timer 5. In 8-bit mode, TMR5H contains the 8-bit high byte timer value.

26.3.3. High-Speed Output Mode

In High-Speed Output mode, a module's associated CEXn pin is toggled each time a match occurs between the PCA Counter and the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the TOGn, MATn, and ECOMn bits in the PCA0CPMn register enables the High-Speed Output mode. If ECOMn is cleared, the associated pin will retain its state, and not toggle on the next match event.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

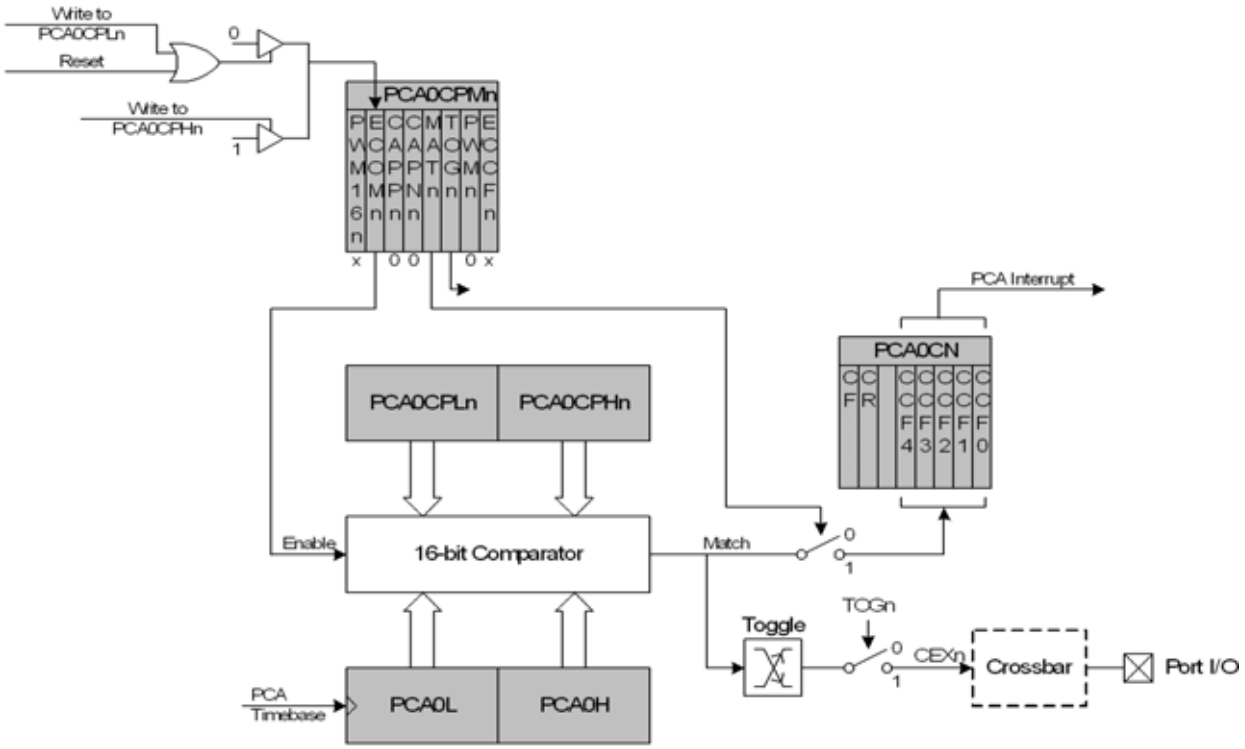

Figure 26.6. PCA High-Speed Output Mode Diagram