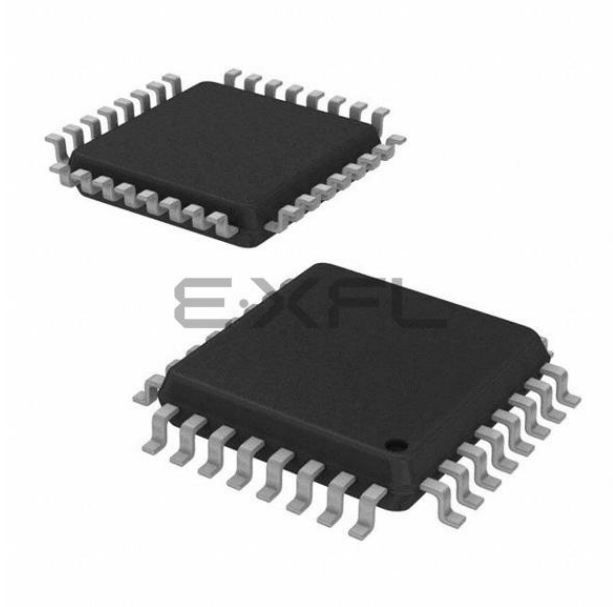

Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	48MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART, USB
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 ~ 3.6V
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/c8051f343-gqr

5.4.1. Window Detector In Single-Ended Mode

Figure 5.6 shows two example window comparisons for right-justified, single-ended data, with ADC0LTH:ADC0LTL = 0x0080 (128d) and ADC0GTH:ADC0GTL = 0x0040 (64d). In single-ended mode, the input voltage can range from '0' to VREF x (1023/1024) with respect to GND, and is represented by a 10-bit unsigned integer value. In the left example, an AD0WINT interrupt will be generated if the ADC0 conversion word (ADC0H:ADC0L) is within the range defined by ADC0GTH:ADC0GTL and ADC0LTH:ADC0LTL (if 0x0040 < ADC0H:ADC0L < 0x0080). In the right example, and AD0WINT interrupt will be generated if the ADC0 conversion word is outside of the range defined by the ADC0GT and ADC0LT registers (if ADC0H:ADC0L < 0x0040 or ADC0H:ADC0L > 0x0080). Figure 5.7 shows an example using left-justified data with equivalent ADC0GT and ADC0LT register settings.

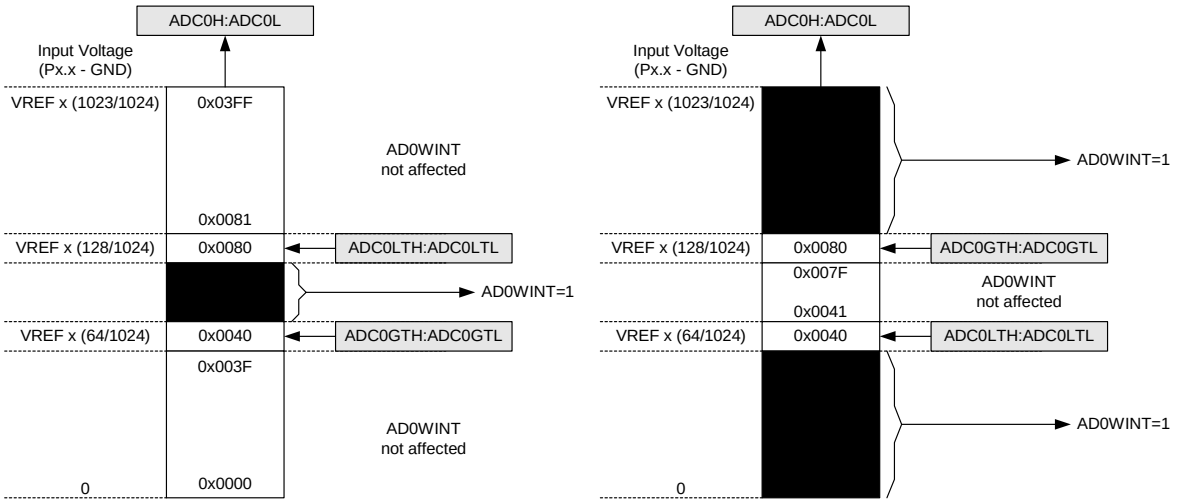

Figure 5.6. ADC Window Compare Example: Right-Justified Single-Ended Data

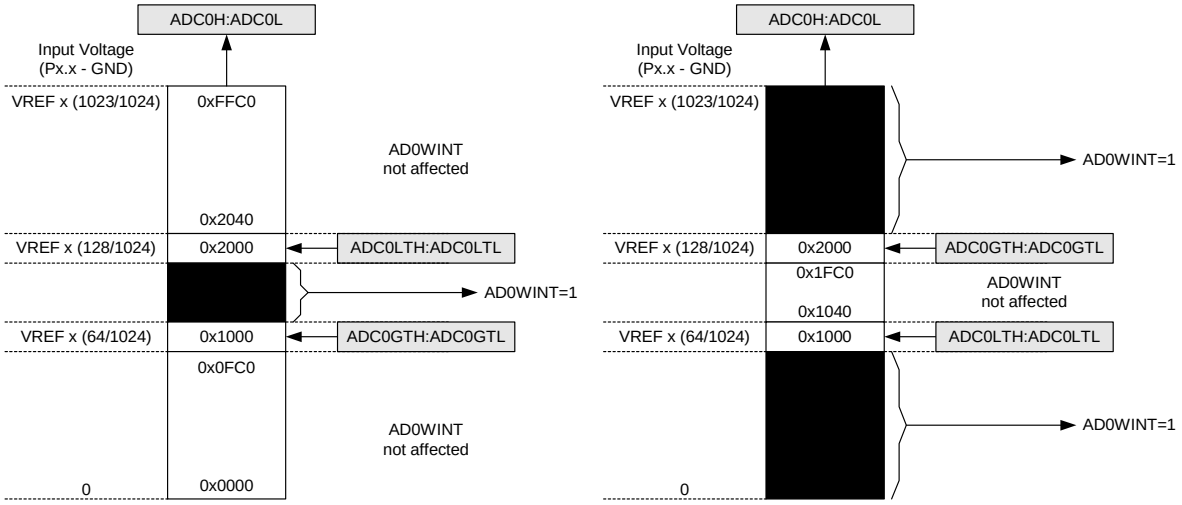

Figure 5.7. ADC Window Compare Example: Left-Justified Single-Ended Data

SFR Definition 7.5. CPT1MX: Comparator1 MUX Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	CMX1N2	CMX1N1	CMX1N0	-	CMX1P2	CMX1P1	CMX1P0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x9E

Bit7: UNUSED. Read = 0b, Write = don't care.
Bits6–4: CMX1N2–CMX1N0: Comparator1 Negative Input MUX Select.
These bits select which Port pin is used as the Comparator1 negative input.

CMX1N2	CMX1N1	CMX1N0	Negative Input (32-pin Package)	Negative Input (48-pin Package)
0	0	0	P1.3	P2.3
0	0	1	P1.7	P3.1
0	1	0	P2.3	P4.0
0	1	1	P2.7	P4.6
1	0	0	P0.5	P1.2

Bit3: UNUSED. Read = 0b, Write = don't care.
Bits2–0: CMX1P1–CMX1P0: Comparator1 Positive Input MUX Select.
These bits select which Port pin is used as the Comparator1 positive input.

CMX1P2	CMX1P1	CMX1P0	Positive Input (32-pin Package)	Positive Input (48-pin Package)
0	0	0	P1.2	P2.2
0	0	1	P1.6	P3.0
0	1	0	P2.2	P3.7
0	1	1	P2.6	P4.5
1	0	0	P0.4	P1.1

Note that the port pins used by the comparator depend on the package type (32-pin or 48-pin).

SFR Definition 7.6. CPT1MD: Comparator1 Mode Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	CP1RIE	CP1FIE	-	-	CP1MD1	CP1MD0	00000010
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x9C

Bits7–6: UNUSED. Read = 00b, Write = don't care.
Bit5: CP1RIE: Comparator1 Rising-Edge Interrupt Enable.
0: Comparator1 rising-edge interrupt disabled.
1: Comparator1 rising-edge interrupt enabled.
Bit4: CP1FIE: Comparator1 Falling-Edge Interrupt Enable.
0: Comparator1 falling-edge interrupt disabled.
1: Comparator1 falling-edge interrupt enabled.
Bits1–0: CP1MD1–CP1MD0: Comparator1 Mode Select.
These bits select the response time for Comparator1.

Mode	CP1MD1	CP1MD0	CP1 Response Time*
0	0	0	Fastest Response
1	0	1	
2	1	0	
3	1	1	Lowest Power

* See Table 7.1 for response time parameters.

SFR Definition 9.10. EIP1: Extended Interrupt Priority 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PT3	PCP1	PCP0	PPCA0	PADC0	PWADC0	PUSB0	PSMB0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xF6

Bit7: PT3: Timer 3 Interrupt Priority Control.
This bit sets the priority of the Timer 3 interrupt.
0: Timer 3 interrupts set to low priority level.
1: Timer 3 interrupts set to high priority level.

Bit6: PCP1: Comparator1 (CP1) Interrupt Priority Control.
This bit sets the priority of the CP1 interrupt.
0: CP1 interrupt set to low priority level.
1: CP1 interrupt set to high priority level.

Bit5: PCP0: Comparator0 (CP0) Interrupt Priority Control.
This bit sets the priority of the CP0 interrupt.
0: CP0 interrupt set to low priority level.
1: CP0 interrupt set to high priority level.

Bit4: PPCA0: Programmable Counter Array (PCA0) Interrupt Priority Control.
This bit sets the priority of the PCA0 interrupt.
0: PCA0 interrupt set to low priority level.
1: PCA0 interrupt set to high priority level.

Bit3: PADC0 ADC0 Conversion Complete Interrupt Priority Control.
This bit sets the priority of the ADC0 Conversion Complete interrupt.
0: ADC0 Conversion Complete interrupt set to low priority level.
1: ADC0 Conversion Complete interrupt set to high priority level.

Bit2: PWADC0: ADC0 Window Comparator Interrupt Priority Control.
This bit sets the priority of the ADC0 Window interrupt.
0: ADC0 Window interrupt set to low priority level.
1: ADC0 Window interrupt set to high priority level.

Bit1: PUSB0: USB0 Interrupt Priority Control.
This bit sets the priority of the USB0 interrupt.
0: USB0 interrupt set to low priority level.
1: USB0 interrupt set to high priority level.

Bit0: PSMB0: SMBus (SMB0) Interrupt Priority Control.
This bit sets the priority of the SMB0 interrupt.
0: SMB0 interrupt set to low priority level.
1: SMB0 interrupt set to high priority level.

SFR Definition 9.13. IT01CF: INT0/INT1 Configuration

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
IN1PL	IN1SL2	IN1SL1	IN1SL0	IN0PL	IN0SL2	IN0SL1	IN0SL0	00000001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xE4

Note: Refer to SFR Definition 21.1 for INT0/1 edge- or level-sensitive interrupt selection.

Bit7: IN1PL: $\overline{\text{INT1}}$ Polarity
0: $\overline{\text{INT1}}$ input is active low.
1: $\overline{\text{INT1}}$ input is active high.

Bits6–4: IN1SL2–0: $\overline{\text{INT1}}$ Port Pin Selection Bits
These bits select which Port pin is assigned to $\overline{\text{INT1}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT1}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to '1' the corresponding bit in register P0SKIP).

IN1SL2–0	$\overline{\text{INT1}}$ Port Pin
000	P0.0
001	P0.1
010	P0.2
011	P0.3
100	P0.4
101	P0.5
110	P0.6
111	P0.7

Bit3: IN0PL: $\overline{\text{INT0}}$ Polarity
0: $\overline{\text{INT0}}$ interrupt is active low.
1: $\overline{\text{INT0}}$ interrupt is active high.

Bits2–0: INT0SL2–0: $\overline{\text{INT0}}$ Port Pin Selection Bits
These bits select which Port pin is assigned to $\overline{\text{INT0}}$. Note that this pin assignment is independent of the Crossbar. $\overline{\text{INT0}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to '1' the corresponding bit in register P0SKIP).

IN0SL2–0	$\overline{\text{INT0}}$ Port Pin
000	P0.0
001	P0.1
010	P0.2
011	P0.3
100	P0.4
101	P0.5
110	P0.6
111	P0.7

Table 11.1. Reset Electrical Characteristics

–40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
$\overline{\text{RST}}$ Output Low Voltage	I_{OL} = 8.5 mA, V_{DD} = 2.7 to 3.6 V			0.6	V
$\overline{\text{RST}}$ Input High Voltage		0.7 x V_{DD}			V
$\overline{\text{RST}}$ Input Low Voltage				0.3 x V_{DD}	
$\overline{\text{RST}}$ Input Pull-Up Current	$\overline{\text{RST}}$ = 0.0 V		25	40	µA
V_{DD} POR Threshold (V_{RST})		2.40	2.55	2.70	V
Missing Clock Detector Timeout	Time from last system clock rising edge to reset initiation	100	220	500	µs
Reset Time Delay	Delay between release of any reset source and code execution at location 0x0000	5.0			µs
Minimum $\overline{\text{RST}}$ Low Time to Generate a System Reset		15			µs
V_{DD} Monitor Turn-on Time		100			µs
V_{DD} Monitor Supply Current			20	50	µA

12. 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 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 12.1 for complete Flash memory electrical characteristics.

12.1. 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 “23. C2 Interface” on page 271**.

To ensure the integrity of Flash contents, it is strongly recommended that the V_{DD} monitor be left enabled in any system which writes or erases Flash memory from code. It is also crucial to ensure that the FLRT bit in register FLSCL be set to '1' if a clock speed higher than 25 MHz is being used for the device.

12.1.1. 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 12.2.

12.1.2. Flash Erase Procedure

The Flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before writing to Flash memory using MOVX, Flash write operations must be enabled by: (1) Writing the Flash key codes in sequence to the Flash Lock register (FLKEY); and (2) Setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1 (this directs the MOVX writes to target Flash memory). The PSWE bit remains set until cleared by software.

A write to Flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in Flash. **A byte location to be programmed must be erased before a new value is written.** The Flash memory is organized in 512-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 512-byte page, perform the following steps:

Step 1. Disable interrupts (recommended).
Step 2. Write the first key code to FLKEY: 0xA5.
Step 3. Write the second key code to FLKEY: 0xF1.
Step 4. Set the PSEE bit (register PSCTL).
Step 5. Set the PSWE bit (register PSCTL).
Step 6. Using the MOVX instruction, write a data byte to any location within the 512-byte page to be erased.
Step 7. Clear the PSWE bit (register PSCTL).
Step 8. Clear the PSEE bit (register PSCTI).

12.1.3. Flash Write Procedure

Bytes in Flash memory can be written one byte at a time, or in groups of two. The FLBWE bit in register PFE0CN (SFR Definition 10.1) controls whether a single byte or a block of two bytes is written to Flash during a write operation. When FLBWE is cleared to ‘0’, the Flash will be written one byte at a time. When FLBWE is set to ‘1’, the Flash will be written in two-byte blocks. Block writes are performed in the same amount of time as single-byte writes, which can save time when storing large amounts of data to Flash memory.During a single-byte write to Flash, bytes are written individually, and a Flash write will be performed after each MOVX write instruction. The recommended procedure for writing Flash in single bytes is:

Step 1. Disable interrupts.
Step 2. Clear the FLBWE bit (register PFE0CN) to select single-byte write mode.
Step 3. Set the PSWE bit (register PSCTL).
Step 4. Clear the PSEE bit (register PSCTL).
Step 5. Write the first key code to FLKEY: 0xA5.
Step 6. Write the second key code to FLKEY: 0xF1.
Step 7. Using the MOVX instruction, write a single data byte to the desired location within the 512-byte sector.
Step 8. Clear the PSWE bit.
Step 9. Re-enable interrupts.

Steps 5-7 must be repeated for each byte to be written.

For block Flash writes, the Flash write procedure is only performed after the last byte of each block is written with the MOVX write instruction. A Flash write block is two bytes long, from even addresses to odd addresses. Writes must be performed sequentially (i.e. addresses ending in 0b and 1b must be written in order). The Flash write will be performed following the MOVX write that targets the address ending in 1b. If a byte in the block does not need to be updated in Flash, it should be written to 0xFF. The recommended procedure for writing Flash in blocks is:

Step 1. Disable interrupts.
Step 2. Set the FLBWE bit (register PFE0CN) to select block write mode.
Step 3. Set the PSWE bit (register PSCTL).
Step 4. Clear the PSEE bit (register PSCTL).
Step 5. Write the first key code to FLKEY: 0xA5.
Step 6. Write the second key code to FLKEY: 0xF1.
Step 7. Using the MOVX instruction, write the first data byte to the even block location (ending in 0b).
Step 8. Write the first key code to FLKEY: 0xA5.
Step 9. Write the second key code to FLKEY: 0xF1.
Step 10. Using the MOVX instruction, write the second data byte to the odd block location (ending in 1b).
Step 11. Clear the PSWE bit.
Step 12. Re-enable interrupts.

Steps 5–10 must be repeated for each block to be written.

13.7.1.2.8-bit MOVX without Bank Select: EMI0CF[4:2] = ‘101’ or ‘111’.

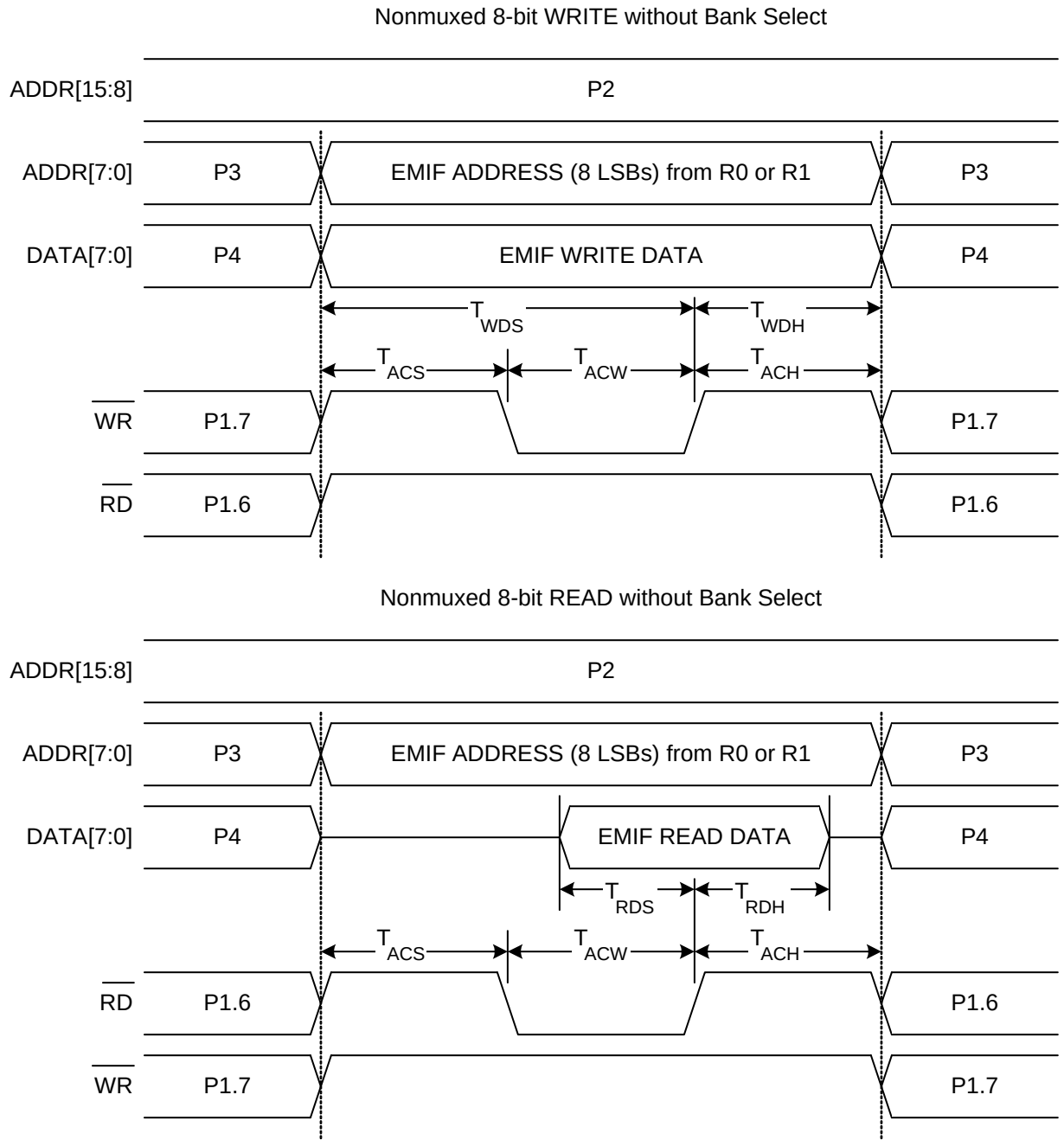

Figure 13.6. Non-multiplexed 8-bit MOVX without Bank Select Timing

13.7.2. Multiplexed Mode

13.7.2.1.16-bit MOVX: EMI0CF[4:2] = ‘001’, ‘010’, or ‘011’.

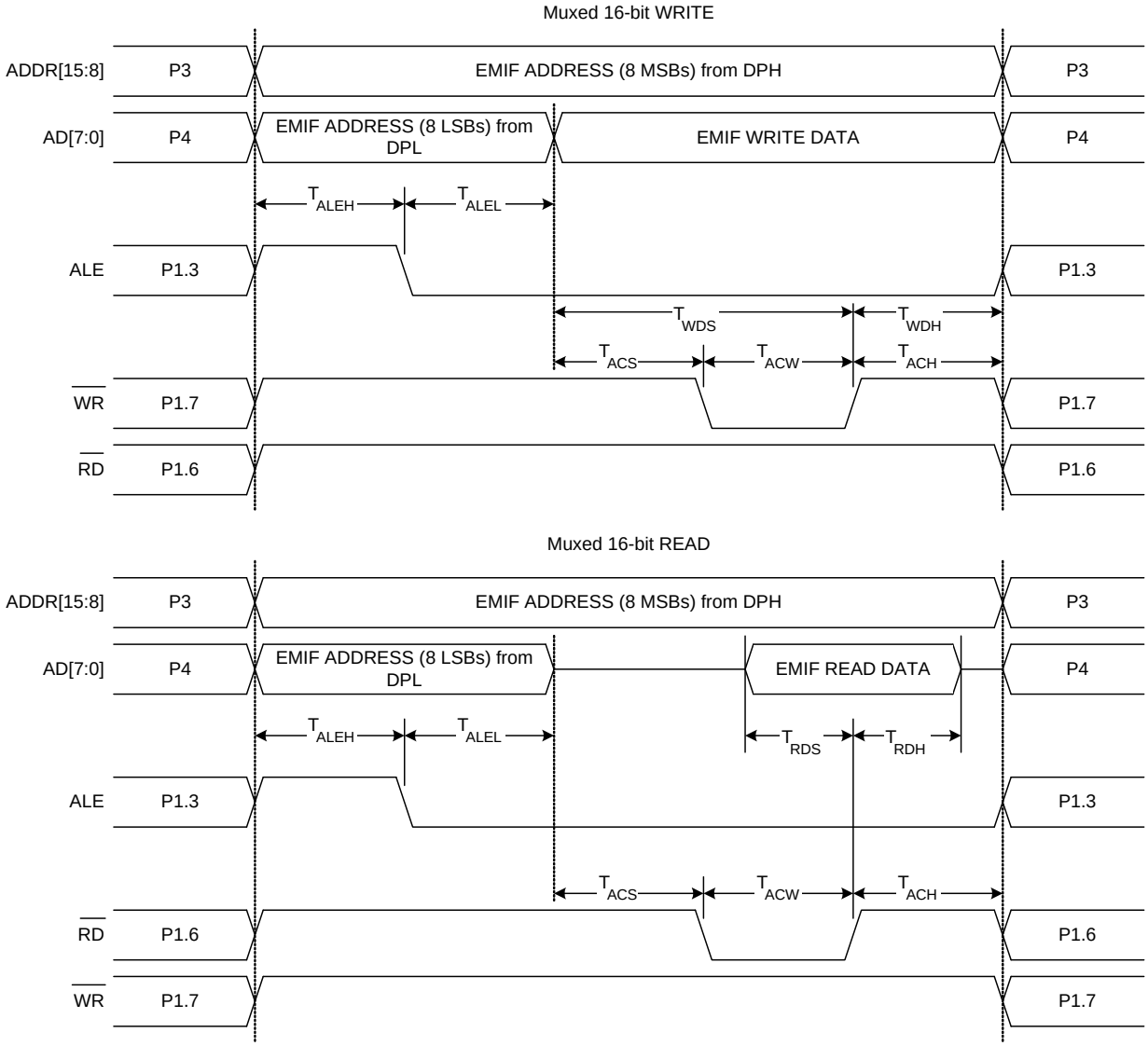

Figure 13.8. Multiplexed 16-bit MOVX Timing

SFR Definition 15.8. P1: Port1 Latch

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0x90

Bits7–0: P1.[7:0]
Write - Output appears on I/O pins per Crossbar Registers (when XBARE = ‘1’).
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P1MDOUT.n bit = 0).
Read - Always reads ‘0’ if selected as analog input in register P1MDIN. Directly reads Port pin when configured as digital input.
0: P1.n pin is logic low.
1: P1.n pin is logic high.

SFR Definition 15.9. P1MDIN: Port1 Input Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xF2

Bits7–0: Analog Input Configuration Bits for P1.7–P1.0 (respectively).
Port pins configured as analog inputs have their weak pull-up, digital driver, and digital receiver disabled.
0: Corresponding P1.n pin is configured as an analog input.
1: Corresponding P1.n pin is not configured as an analog input.

SFR Definition 15.10. P1MDOUT: Port1 Output Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xA5

Bits7–0: Output Configuration Bits for P1.7–P1.0 (respectively): ignored if corresponding bit in register P1MDIN is logic 0.
0: Corresponding P1.n Output is open-drain.
1: Corresponding P1.n Output is push-pull.

USB Register Definition 16.17. E0CSR: USB0 Endpoint0 Control

R/W	R/W	R/W	R	R/W	R/W	R/W	R	Reset Value
SSUEND	SOPRDY	SDSTL	SUEND	DATAEND	STSTL	INPRDY	OPRDY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x11

Bit7: SSUEND: Serviced Setup End
Write: Software should set this bit to ‘1’ after servicing a Setup End (bit SUEND) event. Hardware clears the SUEND bit when software writes ‘1’ to SSUEND.
Read: This bit always reads ‘0’.

Bit6: SOPRDY: Serviced OPRDY
Write: Software should write ‘1’ to this bit after servicing a received Endpoint0 packet. The OPRDY bit will be cleared by a write of ‘1’ to SOPRDY.
Read: This bit always reads ‘0’.

Bit5: SDSTL: Send Stall
Software can write ‘1’ to this bit to terminate the current transfer (due to an error condition, unexpected transfer request, etc.). Hardware will clear this bit to ‘0’ when the STALL handshake is transmitted.

Bit4: SUEND: Setup End
Hardware sets this read-only bit to ‘1’ when a control transaction ends before software has written ‘1’ to the DATAEND bit. Hardware clears this bit when software writes ‘1’ to SSUEND.

Bit3: DATAEND: Data End
Software should write ‘1’ to this bit:
1. When writing ‘1’ to INPRDY for the last outgoing data packet.
2. When writing ‘1’ to INPRDY for a zero-length data packet.
3. When writing ‘1’ to SOPRDY after servicing the last incoming data packet.

This bit is automatically cleared by hardware.

Bit2: STSTL: Sent Stall
Hardware sets this bit to ‘1’ after transmitting a STALL handshake signal. This flag must be cleared by software.

Bit1: INPRDY: IN Packet Ready
Software should write ‘1’ to this bit after loading a data packet into the Endpoint0 FIFO for transmit. Hardware clears this bit and generates an interrupt under either of the following conditions:
1. The packet is transmitted.
2. The packet is overwritten by an incoming SETUP packet.
3. The packet is overwritten by an incoming OUT packet.

Bit0: OPRDY: OUT Packet Ready
Hardware sets this read-only bit and generates an interrupt when a data packet has been received. This bit is cleared only when software writes ‘1’ to the SOPRDY bit.

A Bulk or Interrupt pipe can be shut down (or Halted) by writing ‘1’ to the SDSTL bit (EOUTCSRL.5). While SDSTL = ‘1’, hardware will respond to all OUT requests with a STALL condition. Each time hardware generates a STALL condition, an interrupt will be generated and the STSTL bit (EOUTCSRL.6) set to ‘1’. The STSTL bit must be reset to ‘0’ by firmware.

Hardware will automatically set OPRDY when a packet is ready in the OUT FIFO. Note that if double buffering is enabled for the target endpoint, it is possible for two packets to be ready in the OUT FIFO at a time. In this case, hardware will set OPRDY to ‘1’ immediately after firmware unloads the first packet and resets OPRDY to ‘0’. A second interrupt will be generated in this case.

16.13.2.Endpoints1-3 OUT Isochronous Mode

When the ISO bit (EOUTCSRH.6) is set to ‘1’, the target endpoint operates in Isochronous (ISO) mode. Once an endpoint has been configured for ISO OUT mode, the host will send exactly one data per USB frame; the location of the data packet within each frame may vary, however. Because of this, it is recommended that double buffering be enabled for ISO OUT endpoints.

Each time a data packet is received, hardware will load the received data packet into the endpoint FIFO, set the OPRDY bit (EOUTCSRL.0) to ‘1’, and generate an interrupt (if enabled). Firmware would typically use this interrupt to unload the data packet from the endpoint FIFO and reset the OPRDY bit to ‘0’.

If a data packet is received when there is no room in the endpoint FIFO, an interrupt will be generated and the OVRUN bit (EOUTCSRL.2) set to ‘1’. If USB0 receives an ISO data packet with a CRC error, the data packet will be loaded into the endpoint FIFO, OPRDY will be set to ‘1’, an interrupt (if enabled) will be generated, and the DATAERR bit (EOUTCSRL.3) will be set to ‘1’. Software should check the DATAERR bit each time a data packet is unloaded from an ISO OUT endpoint FIFO.

USB Register Definition 16.22. EOUTCSRH: USB0 OUT Endpoint Control High Byte

R/W	R/W	R/W	R/W	R	R	R	R	Reset Value
DBOEN	ISO	-	-	-	-	-	-	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address:
								0x15

Bit7: DBOEN: Double-buffer Enable
0: Double-buffering disabled for the selected OUT endpoint.
1: Double-buffering enabled for the selected OUT endpoint.
Bit6: ISO: Isochronous Transfer Enable
This bit enables/disables isochronous transfers on the current endpoint.
0: Endpoint configured for bulk/interrupt transfers.
1: Endpoint configured for isochronous transfers.
Bits5–0: Unused. Read = 000000b; Write = don't care.

USB Register Definition 16.23. EOUTCNTL: USB0 OUT Endpoint Count Low

R	R	R	R	R	R	R	R	Reset Value
EOCL								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address:
								0x16

Bits7–0: EOCL: OUT Endpoint Count Low Byte
EOCL holds the lower 8-bits of the 10-bit number of data bytes in the last received packet in the current OUT endpoint FIFO. This number is only valid while OPRDY = '1'.

USB Register Definition 16.24. EOUTCNTH: USB0 OUT Endpoint Count High

R	R	R	R	R	R	R	R	Reset Value
-	-	-	-	-	-	E0CH		00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address:
								0x17

Bits7–2: Unused. Read = 00000. Write = don't care.
Bits1–0: EOCH: OUT Endpoint Count High Byte
EOCH holds the upper 2-bits of the 10-bit number of data bytes in the last received packet in the current OUT endpoint FIFO. This number is only valid while OPRDY = '1'.

SFR Definition 17.1. SMB0CF: SMBus Clock/Configuration

R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	Reset Value
ENSMB	INH	BUSY	EXTHOLD	SMBTOE	SMBFTE	SMBCS1	SMBCS0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xC1

Bit7: ENSMB: SMBus Enable.
This bit enables/disables the SMBus interface. When enabled, the interface constantly monitors the SDA and SCL pins.
0: SMBus interface disabled.
1: SMBus interface enabled.

Bit6: INH: SMBus Slave Inhibit.
When this bit is set to logic 1, the SMBus does not generate an interrupt when slave events occur. This effectively removes the SMBus slave from the bus. Master Mode interrupts are not affected.
0: SMBus Slave Mode enabled.
1: SMBus Slave Mode inhibited.

Bit5: BUSY: SMBus Busy Indicator.
This bit is set to logic 1 by hardware when a transfer is in progress. It is cleared to logic 0 when a STOP or free-timeout is sensed.

Bit4: EXTHOLD: SMBus Setup and Hold Time Extension Enable.
This bit controls the SDA setup and hold times according to.
0: SDA Extended Setup and Hold Times disabled.
1: SDA Extended Setup and Hold Times enabled.

Bit3: SMBTOE: SMBus SCL Timeout Detection Enable.
This bit enables SCL low timeout detection. If set to logic 1, the SMBus forces Timer 3 to reload while SCL is high and allows Timer 3 to count when SCL goes low. Timer 3 should be programmed to generate interrupts at 25 ms, and the Timer 3 interrupt service routine should reset SMBus communication.

Bit2: SMBFTE: SMBus Free Timeout Detection Enable.
When this bit is set to logic 1, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods.

Bits1–0: SMBCS1-SMBCS0: SMBus Clock Source Selection.
These two bits select the SMBus clock source, which is used to generate the SMBus bit rate. The selected device should be configured according to Equation 17.1.

SMBCS1	SMBCS0	SMBus Clock Source
0	0	Timer 0 Overflow
0	1	Timer 1 Overflow
1	0	Timer 2 High Byte Overflow
1	1	Timer 2 Low Byte Overflow

21. Timers

Each MCU includes four counter/timers: two are 16-bit counter/timers compatible with those found in the standard 8051, and two are 16-bit auto-reload timer for use with the ADC, SMBus, USB (frame measurements), Low-Frequency Oscillator (period measurements), or for general purpose use. These timers can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2 and Timer 3 offer 16-bit and split 8-bit timer functionality with auto-reload.

Timer 0 and Timer 1 Modes:	Timer 2 Modes:	Timer 3 Modes:
13-bit counter/timer	16-bit timer with auto-reload	16-bit timer with auto-reload
16-bit counter/timer		
8-bit counter/timer with auto-reload	Two 8-bit timers with auto-reload	Two 8-bit timers with auto-reload
Two 8-bit counter/timers (Timer 0 only)		

Timers 0 and 1 may be clocked by one of five sources, determined by the Timer Mode Select bits (T1M-T0M) and the Clock Scale bits (SCA1-SCA0). The Clock Scale bits define a pre-scaled clock from which Timer 0 and/or Timer 1 may be clocked (See SFR Definition 21.3 for pre-scaled clock selection).

Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock. Timer 2 and Timer 3 may be clocked by the system clock, the system clock divided by 12, or the external oscillator clock source divided by 8.

Timer 0 and Timer 1 may also be operated as counters. When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin (T0 or T1). Events with a frequency of up to one-fourth the system clock's frequency can be counted. The input signal need not be periodic, but it should be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.

21.1. Timer 0 and Timer 1

Each timer is implemented as a 16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Control register (TCON) is used to enable Timer 0 and Timer 1 as well as indicate status. Timer 0 interrupts can be enabled by setting the ET0 bit in the IE register (**Section “9.3.5. Interrupt Register Descriptions” on page 90**); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register (**Section 9.3.5**). Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits T1M1-T0M0 in the Counter/Timer Mode register (TMOD). Each timer can be configured independently. Each operating mode is described below.

21.1.1. Mode 0: 13-bit Counter/Timer

Timer 0 and Timer 1 operate as 13-bit counter/timers in Mode 0. The following describes the configuration and operation of Timer 0. However, both timers operate identically, and Timer 1 is configured in the same manner as described for Timer 0.

The TH0 register holds the eight MSBs of the 13-bit counter/timer. TL0 holds the five LSBs in bit positions TL0.4-TL0.0. The three upper bits of TL0 (TL0.7-TL0.5) are indeterminate and should be masked out or ignored when reading. As the 13-bit timer register increments and overflows from 0x1FFF (all ones) to 0x0000, the timer overflow flag TF0 (TCON.5) is set and an interrupt will occur if Timer 0 interrupts are enabled.

The C/T0 bit (TMOD.2) selects the counter/timer's clock source. When C/T0 is set to logic 1, high-to-low transitions at the selected Timer 0 input pin (T0) increment the timer register (Refer to **Section "15.1. Priority Crossbar Decoder" on page 144** for information on selecting and configuring external I/O pins). Clearing C/T selects the clock defined by the T0M bit (CKCON.3). When T0M is set, Timer 0 is clocked by the system clock. When T0M is cleared, Timer 0 is clocked by the source selected by the Clock Scale bits in CKCON (see SFR Definition 21.3).

Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or the input signal $\overline{INT0}$ is active as defined by bit IN0PL in register INT01CF (see SFR Definition 9.13). Setting GATE0 to '1' allows the timer to be controlled by the external input signal $\overline{INT0}$ (see **Section "9.3.5. Interrupt Register Descriptions" on page 90**), facilitating pulse width measurements.

TR0	GATE0	$\overline{INT0}$	Counter/Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0	Disabled
1	1	1	Enabled

X = Don't Care

Setting TR0 does not force the timer to reset. The timer registers should be loaded with the desired initial value before the timer is enabled.

TL1 and TH1 form the 13-bit register for Timer 1 in the same manner as described above for TL0 and TH0. Timer 1 is configured and controlled using the relevant TCON and TMOD bits just as with Timer 0. The input signal $\overline{INT1}$ is used with Timer 1; the $\overline{INT1}$ polarity is defined by bit IN1PL in register INT01CF (see SFR Definition 9.13).

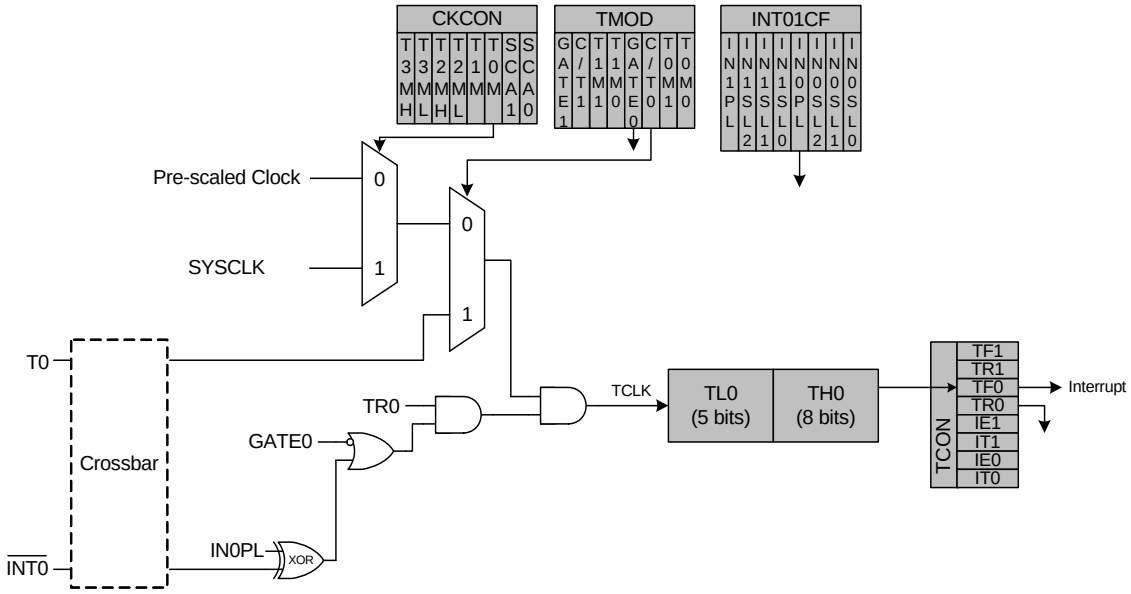

Figure 21.1. T0 Mode 0 Block Diagram

21.1.2. Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

22. Programmable Counter Array (PCA0)

The Programmable Counter Array (PCA0) provides enhanced timer functionality while requiring less CPU intervention than the standard 8051 counter/timers. The PCA consists of a dedicated 16-bit counter/timer and five 16-bit capture/compare modules. Each capture/compare module has its own associated I/O line (CEXn) which is routed through the Crossbar to Port I/O when enabled (See **Section "15.1. Priority Crossbar Decoder" on page 144** for details on configuring the Crossbar). The counter/timer is driven by a programmable timebase that can select between six sources: system clock, system clock divided by four, system clock divided by twelve, the external oscillator clock source divided by 8, Timer 0 overflow, or an external clock signal on the ECI input pin. Each capture/compare module may be configured to operate independently in one of six modes: Edge-Triggered Capture, Software Timer, High-Speed Output, Frequency Output, 8-Bit PWM, or 16-Bit PWM (each mode is described in **Section "22.2. Capture/Compare Modules" on page 257**). The external oscillator clock option is ideal for real-time clock (RTC) functionality, allowing the PCA to be clocked by a precision external oscillator while the internal oscillator drives the system clock. The PCA is configured and controlled through the system controller's Special Function Registers. The PCA block diagram is shown in Figure 22.1

Important Note: The PCA Module 4 may be used as a watchdog timer (WDT), and is enabled in this mode following a system reset. Access to certain PCA registers is restricted while WDT mode is enabled. See **Section 22.3** for details.

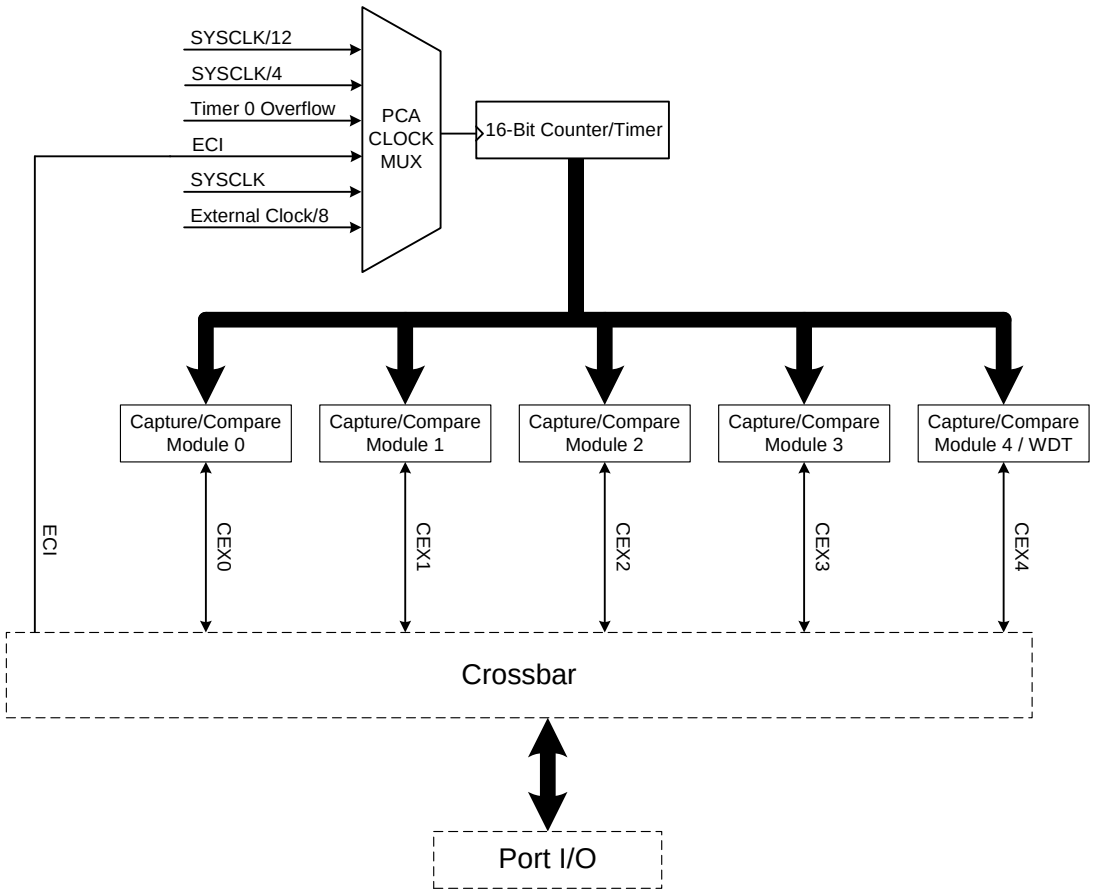

Figure 22.1. PCA Block Diagram

23. C2 Interface

C8051F34x 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.

23.1. C2 Interface Registers

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

C2 Register Definition 23.1. C2ADD: C2 Address

Reset Value: 00000000

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bits7–0: 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
0xAD	Selects the C2 Flash Programming Data register for Data Read/Write instructions

C2 Register Definition 23.2. DEVICEID: C2 Device ID

Reset Value: 00001111

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

This read-only register returns the 8-bit device ID: 0x0F (C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D).