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	Discontinued at Digi-Key
Core Processor	8051
Core Size	8-Bit
Speed	100MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	39
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f363-gq

1.2. On-Chip Memory

The CIP-51 has a standard 8051 program and data address configuration. It includes 256 bytes of data RAM, with the upper 128 bytes dual-mapped. Indirect addressing accesses the upper 128 bytes of general purpose RAM, and direct addressing accesses the 128 byte SFR address space. The lower 128 bytes of RAM are accessible via direct and indirect addressing. The first 32 bytes are addressable as four banks of general purpose registers, and the next 16 bytes can be byte addressable or bit addressable.

Program memory consists of 32/16 kB of Flash. This memory may be reprogrammed in-system in 1024 byte sectors, and requires no special off-chip programming voltage. See Figure 1.6 for the MCU system memory map.

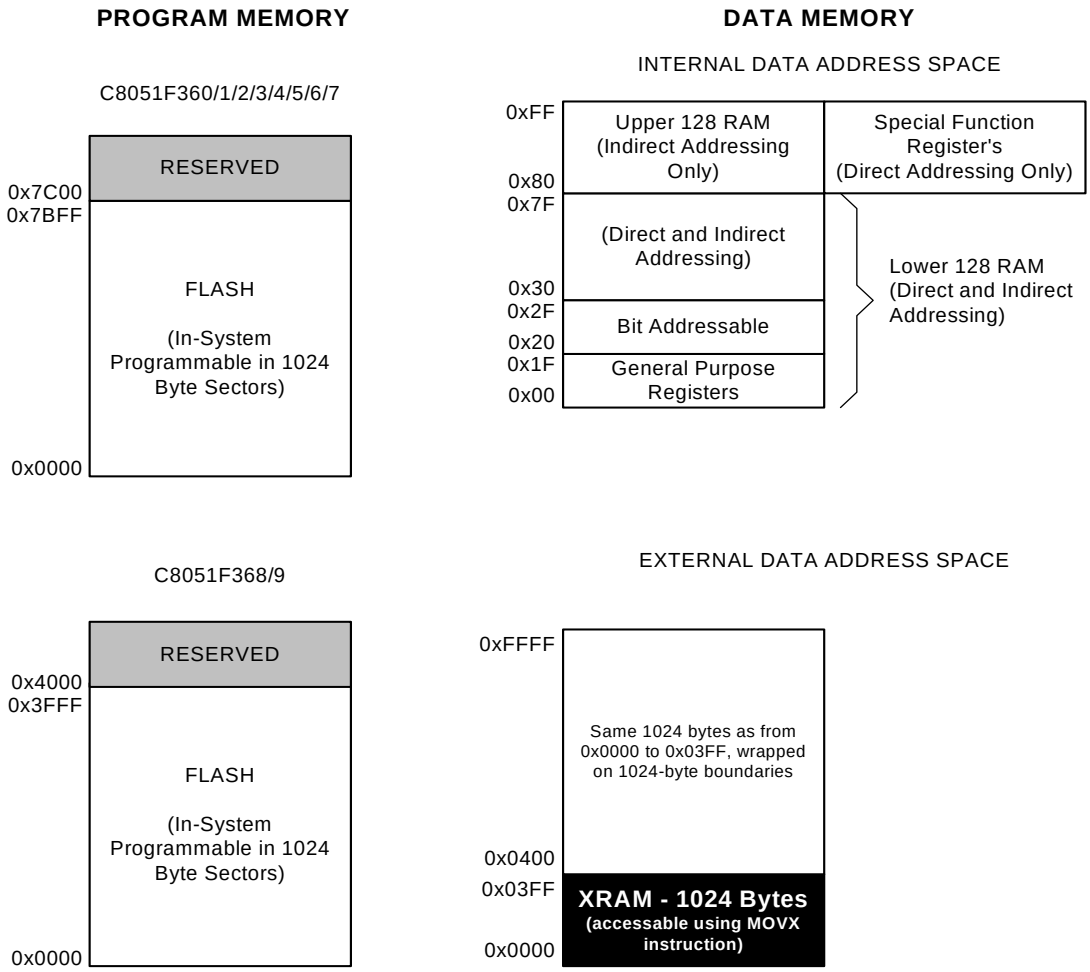

Figure 1.6. On-Board Memory Map

1.3. On-Chip Debug Circuitry

The C8051F36x devices include on-chip Silicon Labs 2-Wire (C2) debug circuitry that provides non-intrusive, full speed, in-circuit debugging of the production part *installed in the end application.*

Silicon Labs' debugging system supports inspection and modification of memory and registers, breakpoints, and single stepping. No additional target RAM, program memory, timers, or communications chan-

SFR Definition 5.3. ADC0CF: ADC0 Configuration

SFR Page: all pages
SFR Address: 0xBC

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD0SC4	AD0SC3	AD0SC2	AD0SC1	AD0SC0	AD0LJST	–	–	11111000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–3: AD0SC4–0: ADC0 SAR Conversion Clock Period Bits.
SAR Conversion clock is derived from system clock by the following equation, where *AD0SC* refers to the 5-bit value held in bits AD0SC4–0. SAR Conversion clock requirements are given in Table 5.1.

$$AD0SC = \frac{SYSCLK}{CLK_{SAR}} - 1$$

Bit 2: AD0LJST: ADC0 Left Justify Select.
0: Data in ADC0H:ADC0L registers are right-justified.
1: Data in ADC0H:ADC0L registers are left-justified.
Bits 1–0: UNUSED. Read = 00b; Write = don't care.

SFR Definition 5.4. ADC0H: ADC0 Data Word MSB

SFR Page: all pages
SFR Address: 0xBE

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	

Bits 7–0: ADC0 Data Word High-Order Bits.
For AD0LJST = 0: Bits 7–2 are the sign extension of Bit1. Bits 1–0 are the upper 2 bits of the 10-bit ADC0 Data Word.
For AD0LJST = 1: Bits 7–0 are the most-significant bits of the 10-bit ADC0 Data Word.

SFR Definition 5.5. ADC0L: ADC0 Data Word LSB

SFR Page: all pages
SFR Address: 0xBD

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	

Bits 7–0: ADC0 Data Word Low-Order Bits.
For AD0LJST = 0: Bits 7–0 are the lower 8 bits of the 10-bit Data Word.
For AD0LJST = 1: Bits 7–6 are the lower 2 bits of the 10-bit Data Word. Bits 5–0 will always read '0'.

SFR Definition 5.6. ADC0CN: ADC0 Control

SFR Page: **all pages**
SFR Address: **0xE8** (bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD0EN	AD0TM	AD0INT	AD0BUSY	AD0WINT	AD0CM2	AD0CM1	AD0CM0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: AD0EN: ADC0 Enable Bit.
0: ADC0 Disabled. ADC0 is in low-power shutdown.
1: ADC0 Enabled. ADC0 is active and ready for data conversions.

Bit 6: AD0TM: ADC0 Track Mode Bit.
0: Normal Track Mode: When ADC0 is enabled, tracking is continuous unless a conversion is in progress.
1: Low-power Track Mode: Tracking Defined by AD0CM2-0 bits (see below).

Bit 5: AD0INT: ADC0 Conversion Complete Interrupt Flag.
0: ADC0 has not completed a data conversion since the last time AD0INT was cleared.
1: ADC0 has completed a data conversion.

Bit 4: AD0BUSY: ADC0 Busy Bit.
Read:
0: ADC0 conversion is complete or a conversion is not currently in progress. AD0INT is set to logic ‘1’ on the falling edge of AD0BUSY.
1: ADC0 conversion is in progress.
Write:
0: No Effect.
1: Initiates ADC0 Conversion if AD0CM2-0 = 000b

Bit 3: AD0WINT: ADC0 Window Compare Interrupt Flag.
0: ADC0 Window Comparison Data match has not occurred since this flag was last cleared.
1: ADC0 Window Comparison Data match has occurred.

Bits 2–0: AD0CM2–0: ADC0 Start of Conversion Mode Select.
When AD0TM = 0:
000: ADC0 conversion initiated on every write of ‘1’ to AD0BUSY.
001: ADC0 conversion initiated on overflow of Timer 0.
010: ADC0 conversion initiated on overflow of Timer 2.
011: ADC0 conversion initiated on overflow of Timer 1.
100: ADC0 conversion initiated on rising edge of external CNVSTR.
101: ADC0 conversion initiated on overflow of Timer 3.
11x: Reserved.
When AD0TM = 1:
000: Tracking initiated on write of ‘1’ to AD0BUSY and lasts 3 SAR clocks, followed by conversion.
001: Tracking initiated on overflow of Timer 0 and lasts 3 SAR clocks, followed by conversion.
010: Tracking initiated on overflow of Timer 2 and lasts 3 SAR clocks, followed by conversion.
011: Tracking initiated on overflow of Timer 1 and lasts 3 SAR clocks, followed by conversion.
100: ADC0 tracks only when CNVSTR input is logic low; conversion starts on rising CNVSTR edge.
101: Tracking initiated on overflow of Timer 3 and lasts 3 SAR clocks, followed by conversion.
11x: Reserved.

SFR Definition 5.9. ADC0LTH: ADC0 Less-Than Data High Byte

SFR Page: all pages
SFR Address: 0xC6

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	

Bits 7–0: High byte of ADC0 Less-Than Data Word.

SFR Definition 5.10. ADC0LTL: ADC0 Less-Than Data Low Byte

SFR Page: all pages
SFR Address: 0xC5

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	

Bits 7–0: Low byte of ADC0 Less-Than Data Word.

Table 9.3. Special Function Registers (Continued)

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

Register	Address	SFR Page	Description	Page No.
SP	0x81	All Pages	Stack Pointer	page 102
SPI0CFG	0xA1	All Pages	SPI Configuration	page 239
SPI0CKR	0xA2	All Pages	SPI Clock Rate Control	page 241
SPI0CN	0xF8	All Pages	SPI Control	page 240
SPI0DAT	0xA3	All Pages	SPI Data	page 241
TCON	0x88	All Pages	Timer/Counter Control	page 250
TH0	0x8C	All Pages	Timer/Counter 0 High Byte	page 253
TH1	0x8D	All Pages	Timer/Counter 1 High Byte	page 253
TL0	0x8A	All Pages	Timer/Counter 0 Low Byte	page 253
TL1	0x8B	All Pages	Timer/Counter 1 Low Byte	page 253
TMOD	0x89	All Pages	Timer/Counter Mode	page 251
TMR2CN	0xC8	All Pages	Timer/Counter 2 Control	page 256
TMR2H	0xCD	All Pages	Timer/Counter 2 High Byte	page 257
TMR2L	0xCC	All Pages	Timer/Counter 2 Low Byte	page 257
TMR2RLH	0xCB	All Pages	Timer 2 Reload Register High Byte	page 257
TMR2RLL	0xCA	All Pages	Timer 2 Reload Register Low Byte	page 257
TMR3CN	0x91	All Pages	Timer 3 Control	page 260
TMR3H	0x95	All Pages	Timer 3 High Byte	page 261
TMR3L	0x94	All Pages	Timer 3 Low Byte	page 261
TMR3RLH	0x93	All Pages	Timer 3 Reload Register High Byte	page 261
TMR3RLL	0x92	All Pages	Timer 3 Reload Register Low Byte	page 261
VDM0CN	0xFF	All Pages	V_{DD} Monitor Control	page 131
XBR0	0xE1	F	Port I/O Crossbar Control 0	page 187
XBR1	0xE2	F	Port I/O Crossbar Control 1	page 188

Notes:
1. Refers to a register in the C8051F360/1/2/6/7/8/9 only.
2. Refers to a register in the C8051F360/3 only.

11.3. Operating in Multiply and Accumulate Mode

MAC0 operates in Multiply and Accumulate (MAC) mode when the MAC0MS bit (MAC0CF.0) is cleared to ‘0’. When operating in MAC mode, MAC0 performs a 16-by-16 bit multiply on the contents of the MAC0A and MAC0B registers, and adds the result to the contents of the 40-bit MAC0 accumulator. Figure 11.4 shows the MAC0 pipeline. There are three stages in the pipeline, each of which takes exactly one SYSCLK cycle to complete. The MAC operation is initiated with a write to the MAC0BL register. After the MAC0BL register is written, MAC0A and MAC0B are multiplied on the first SYSCLK cycle. During the second stage of the MAC0 pipeline, the results of the multiplication are added to the current accumulator contents, and the result of the addition is stored in the MAC0 accumulator. The status flags in the MAC0STA register are set after the end of the second pipeline stage. During the second stage of the pipeline, the next multiplication can be initiated by writing to the MAC0BL register, if it is desired. The rounded (and optionally, saturated) result is available in the MAC0RNDH and MAC0RNDL registers at the end of the third pipeline stage. If the MAC0CA bit (MAC0CF.3) is set to ‘1’ when the MAC operation is initiated, the accumulator and all MAC0STA flags will be cleared during the next cycle of the controller’s clock (SYSCLK). The MAC0CA bit will clear itself to ‘0’ when the clear operation is complete.

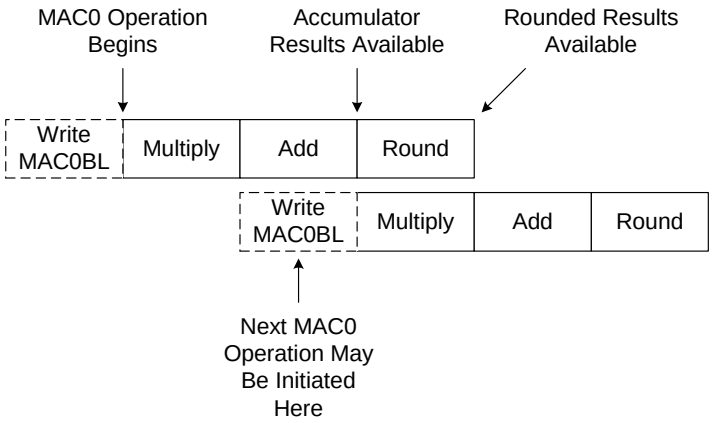

Figure 11.4. MAC0 Pipeline

11.4. Operating in Multiply Only Mode

MAC0 operates in Multiply Only mode when the MAC0MS bit (MAC0CF.0) is set to ‘1’. Multiply Only mode is identical to Multiply and Accumulate mode, except that the multiplication result is added with a value of zero before being stored in the MAC0 accumulator (i.e. it overwrites the current accumulator contents). The result of the multiplication is available in the MAC0 accumulator registers at the end of the second MAC0 pipeline stage (two SYSCLKs after writing to MAC0BL). As in MAC mode, the rounded result is available in the MAC0 Rounding Registers after the third pipeline stage. Note that in Multiply Only mode, the MAC0HO flag is not affected.

11.5. Accumulator Shift Operations

MAC0 contains a 1-bit arithmetic shift function which can be used to shift the contents of the 40-bit accumulator left or right by one bit. The accumulator shift is initiated by writing a ‘1’ to the MAC0SC bit (MAC0CF.5), and takes one SYSCLK cycle (the rounded result is available in the MAC0 Rounding Registers after a second SYSCLK cycle, and MAC0SC is cleared to ‘0’). The direction of the arithmetic shift is controlled by the MAC0SD bit (MAC0CF.4). When this bit is cleared to ‘0’, the MAC0 accumulator will shift left. When the MAC0SD bit is set to ‘1’, the MAC0 accumulator will shift right. Right-shift operations are sign-extended with the current value of bit 39. Note that the status flags in the MAC0STA register are not affected by shift operations.

SFR Definition 11.2. MAC0STA: MAC0 Status

SFR Page: 0
SFR Address: 0xCF

R	R	R	R	R/W	R/W	R/W	R/W	Reset Value
–	–	–	–	MAC0HO	MAC0Z	MAC0SO	MAC0N	00000100
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

Bits 7–4: UNUSED: Read = 0000b, Write = don't care.

Bit 3: MAC0HO: Hard Overflow Flag.
This bit is set to '1' whenever an overflow out of the MAC0OVR register occurs during a MAC operation (i.e. when MAC0OVR changes from 0x7F to 0x80 or from 0x80 to 0x7F). The hard overflow flag must be cleared in software by directly writing it to '0', or by resetting the MAC logic using the MAC0CA bit in register MAC0CF.

Bit 2: MAC0Z: Zero Flag.
This bit is set to '1' if a MAC0 operation results in an Accumulator value of zero. If the result is non-zero, this bit will be cleared to '0'.

Bit 1: MAC0SO: Soft Overflow Flag.
This bit is set to '1' when a MAC operation causes an overflow into the sign bit (bit 31) of the MAC0 Accumulator. If the overflow condition is corrected after a subsequent MAC operation, this bit is cleared to '0'.

Bit 0: MAC0N: Negative Flag.
If the MAC Accumulator result is negative, this bit will be set to '1'. If the result is positive or zero, this flag will be cleared to '0'.

Note: The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

SFR Definition 11.3. MAC0AH: MAC0 A High Byte

SFR Page: 0
SFR Address: 0xA5

R	R	R	R	R	R	R	R	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: High Byte (bits 15–8) of MAC0 A Register.

12.1. Power-On Reset

During power-up, the device is held in a reset state and the $\overline{\text{RST}}$ pin is driven low until V_{DD} settles above V_{RST}. A delay occurs before the device is released from reset; the delay decreases as the V_{DD} ramp time increases (V_{DD} ramp time is defined as how fast V_{DD} ramps from 0 V to V_{RST}). Figure 12.2. plots the power-on and V_{DD} Monitor reset timing. For ramp times less than 1 ms, the power-on reset delay ($T_{PORDelay}$) is typically less than 0.3 ms.

Note: The maximum V_{DD} ramp time is 1 ms; slower ramp times may cause the device to be released from reset before V_{DD} reaches the V_{RST} level.

On exit from a power-on reset, the PORSF flag (RSTSRC.1) is set by hardware to logic '1'. When PORSF is set, all of the other reset flags in the RSTSRC Register are indeterminate (PORSF is cleared by all other resets). Since all resets cause program execution to begin at the same location (0x0000) software can read the PORSF flag to determine if a power-up was the cause of reset. The content of internal data memory should be assumed to be undefined after a power-on reset. The V_{DD} Monitor is enabled following a power-on reset.

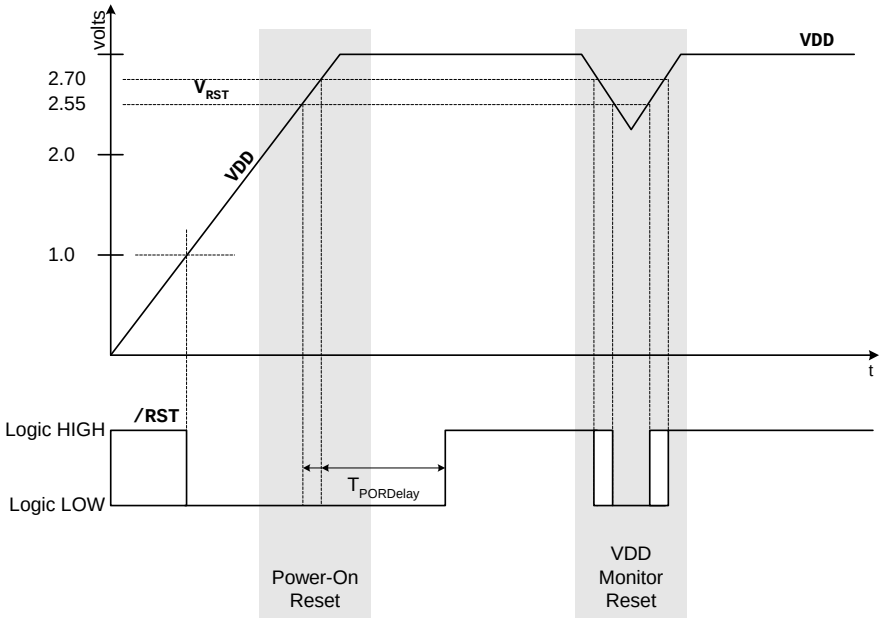

Figure 12.2. Power-On and V_{DD} Monitor Reset Timing

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 13.2.

13.1.2. Erasing Flash Pages From Software

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) the PSWE and PSEE bits must be set to '1' (this directs the MOVX writes to target Flash memory); and (2) Writing the Flash key codes in sequence to the Flash Lock register (FLKEY). 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 should be erased before a new value is written.** The Flash memory is organized in 1024-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 1024-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 PSEE (PSCTL.1) to enable Flash erases.
Step 5. Set PSWE (PSCTL.0) to redirect MOVX commands to write to Flash.
Step 6. Use the MOVX instruction to write a data byte to any location within the page to be erased.
Step 7. Clear PSEE to disable Flash erases.
Step 8. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 9. Re-enable interrupts.

13.1.3. Writing Flash Memory From Software

Bytes in Flash memory can be written one byte at a time, or in small blocks. The CHBLKW bit in register CCH0CN (SFR Definition 14.1) controls whether a single byte or a block of bytes is written to Flash during a write operation. When CHBLKW is cleared to '0', the Flash will be written one byte at a time. When CHBLKW is set to '1', the Flash will be written in blocks of four bytes for addresses in code space. 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.

For single-byte writes to Flash, bytes are written individually, and the Flash write is performed after each MOVX write instruction. The recommended procedure for writing Flash in single bytes is as follows:

Step 1. Disable interrupts.
Step 2. Clear CHBLKW (register CCH0CN) to select single-byte write mode.
Step 3. Write the first key code to FLKEY: 0xA5.
Step 4. Write the second key code to FLKEY: 0xF1.
Step 5. Set PSWE (register PSCTL) to redirect MOVX commands to write to Flash.
Step 6. Clear the PSEE bit (register PSCTL).
Step 7. Use the MOVX instruction to write a data byte to the desired location (repeat as necessary).
Step 8. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 9. Re-enable interrupts.

Certain types of instruction data or certain blocks of code can also be excluded from caching. The destinations of RETI instructions are, by default, excluded from caching. To enable caching of RETI destinations, the CHRETI bit (CCH0CN.3) can be set to ‘1’. It is generally not beneficial to cache RETI destinations unless the same instruction is likely to be interrupted repeatedly (such as a code loop that is waiting for an interrupt to happen). Instructions that are part of an interrupt service routine (ISR) can also be excluded from caching. By default, ISR instructions are cached, but this can be disabled by clearing the CHISR bit (CCH0CN.2) to ‘0’. The other information that can be explicitly excluded from caching are the data returned by MOVC instructions. Clearing the CHMOV bit (CCH0CN.1) to ‘0’ will disable caching of MOVC data. If MOVC caching is allowed, it can be restricted to only use slot 0 for the MOVC information (excluding cache push operations). The CHFIXM bit (CCH0TN.2) controls this behavior.

Further cache control can be implemented by disabling all cache writes. Cache writes can be disabled by clearing the CHWREN bit (CCH0CN.7) to ‘0’. Although normal cache writes (such as those after a cache miss) are disabled, data can still be written to the cache with a cache push operation. Disabling cache writes can be used to prevent a non-critical section of code from changing the cache contents. Note that regardless of the value of CHWREN, a Flash write or erase operation automatically removes the affected bytes from the cache. Cache reads and the prefetch engine can also be individually disabled. Disabling cache reads forces all instructions data to execute from Flash memory or from the prefetch engine. To disable cache reads, the CHRDEN bit (CCH0CN.6) can be cleared to ‘0’. Note that when cache reads are disabled, cache writes will still occur (if CHWREN is set to ‘1’). Disabling the prefetch engine is accomplished using the CHPFEN bit (CCH0CN.5). When this bit is cleared to ‘0’, the prefetch engine will be disabled. If both CHPFEN and CHRDEN are ‘0’, code will execute at a fixed rate, as instructions become available from the Flash memory.

Cache locations can also be pre-loaded and locked with time-critical branch destinations. For example, in a system with an ISR that must respond as fast as possible, the entry point for the ISR can be locked into a cache location to minimize the response latency of the ISR. Up to 30 locations can be locked into the cache at one time. Instructions are locked into cache by enabling cache push operations with the CHPUSH bit (CCH0LC.7). When CHPUSH is set to ‘1’, a MOVC instruction will cause the four-byte segment containing the data byte to be written to the cache slot location indicated by CHSLOT (CCH0LC.4-0). CHSLOT is them decremented to point to the next lockable cache location. This process is called a cache push operation. Cache locations that are above CHSLOT are “locked”, and cannot be changed by the processor core, as shown in Figure 14.3. Cache locations can be unlocked by using a cache pop operation. A cache pop is performed by writing a ‘1’ to the CHPOP bit (CCH0LC.6). When a cache pop is initiated, the value of CHSLOT is incremented. This unlocks the most recently locked cache location, but does not remove the information from the cache. Note that a cache pop should not be initiated if CHSLOT is equal to 11110b. Doing so may have an adverse effect on cache performance. **Important: Although locking cache location 1 is not explicitly disabled by hardware, the entire cache will be unlocked when CHSLOT is equal to 00000b. Therefore, cache locations 1 and 0 must remain unlocked at all times.**

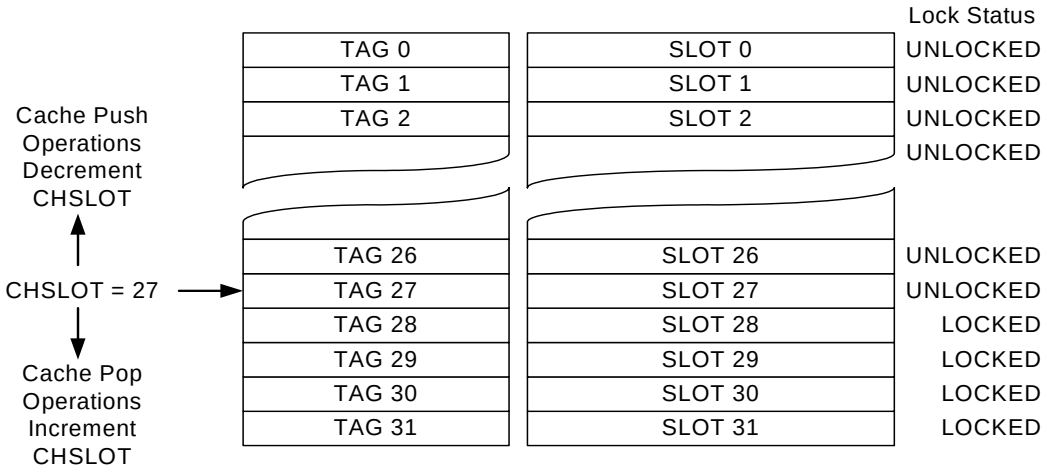

Figure 14.3. Cache Lock Operation

15.6.2.2.8-bit MOVX without Bank Select: EMI0CF[4:2] = ‘001’ or ‘011’.

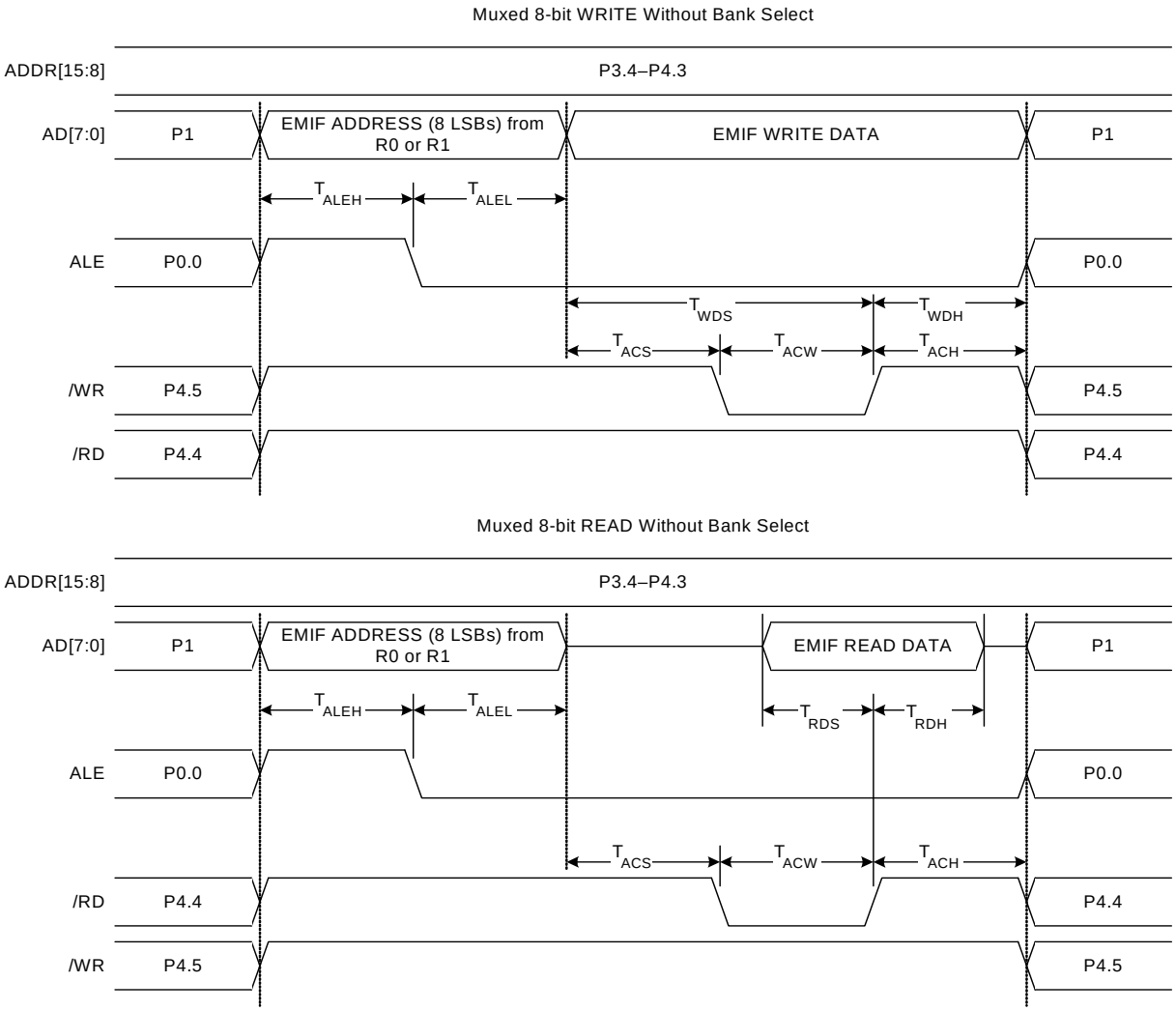

Figure 15.8. Multiplexed 8-bit MOVX without Bank Select Timing

17.1. Priority Crossbar Decoder

The Priority Crossbar Decoder (Figure 17.3) assigns a priority to each I/O function, starting at the top with UART0. When a digital resource is selected, the least-significant unassigned Port pin is assigned to that resource (excluding UART0, which will be assigned to specific port pins (P0.1 and P0.2 in the C8051F360/3 devices, P0.4 and P0.5 in the C8051F361/2/4/5/6/7/8/9 devices). If a Port pin is assigned, the Crossbar skips that pin when assigning the next selected resource. Additionally, the Crossbar will skip Port pins whose associated bits in the PnSKIP registers are set. The PnSKIP registers allow software to skip Port pins that are to be used for analog input, dedicated functions, or GPIO.

Important Note on Crossbar Configuration: If a Port pin is claimed by a peripheral without use of the Crossbar, its corresponding PnSKIP bit should be set. This applies to the port pins associated with the external oscillator, V_{REF}, external CNVSTR signal, IDA0, and any selected ADC or comparator inputs. The Crossbar skips selected pins as if they were already assigned, and moves to the next unassigned pin. Figure 17.3 shows the Crossbar Decoder priority with no Port pins skipped (P0SKIP, P1SKIP, P2SKIP, P3SKIP = 0x00); Figure 17.4 shows the Crossbar Decoder priority with the P1.0 and P1.1 pins skipped (P1SKIP = 0x03).

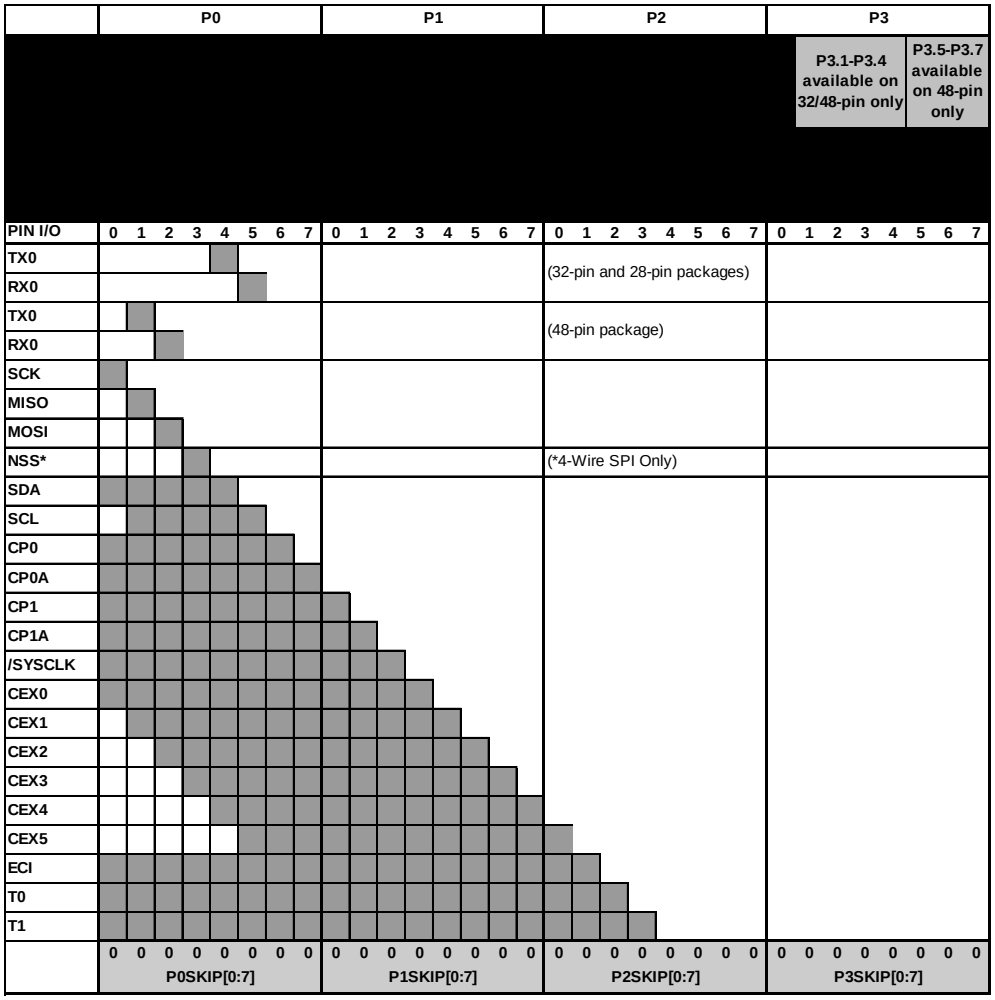

Figure 17.3. Crossbar Priority Decoder with No Pins Skipped

SFR Definition 17.6. P0SKIP: Port0 Skip

SFR Page: F
SFR Address: 0xD4

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	

Bits 7–0: P0SKIP[7:0]: Port0 Crossbar Skip Enable Bits.
These bits select Port pins to be skipped by the Crossbar Decoder. Port pins used as analog inputs (for ADC or Comparator) or used as special functions (V_{REF} input, external oscillator circuit, CNVSTR input) should be skipped by the Crossbar.
0: Corresponding P0.n pin is not skipped by the Crossbar.
1: Corresponding P0.n pin is skipped by the Crossbar.

SFR Definition 17.7. P0MAT: Port0 Match

SFR Page: 0
SFR Address: 0xF3

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	

Bits 7–0: P0MAT[7:0]: Port0 Match Value.
These bits control the value that unmasked P0 Port pins are compared against. A Port Match event is generated if (P0 & P0MASK) does not equal (P0MAT & P0MASK).

SFR Definition 17.8. P0MASK: Port0 Mask

SFR Page: 0
SFR Address: 0xF4

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	

Bits 7–0: P0MASK[7:0]: Port0 Mask Value.
These bits select which Port pins will be compared to the value stored in P0MAT.
0: Corresponding P0.n pin is ignored and cannot cause a Port Match event.
1: Corresponding P0.n pin is compared to the corresponding bit in P0MAT.

Figure 18.4 shows the typical SCL generation described by Equation 18.2. Notice that T_{HIGH} is typically twice as large as T_{LOW}. The actual SCL output may vary due to other devices on the bus (SCL may be extended low by slower slave devices, or driven low by contending master devices). The bit rate when operating as a master will never exceed the limits defined by equation Equation 18.1.

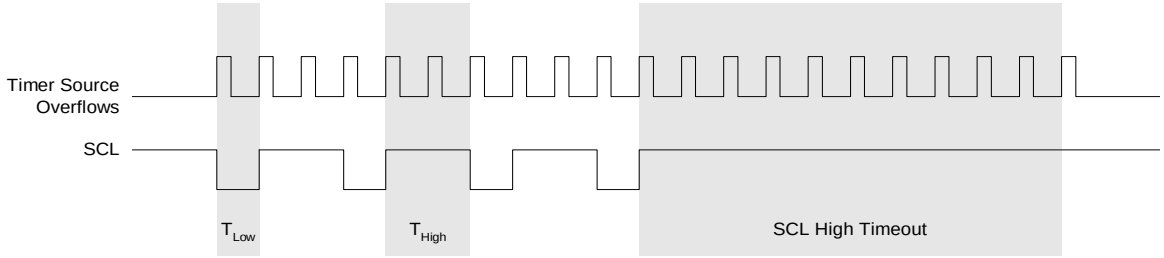

Figure 18.4. Typical SMBus SCL Generation

Setting the EXTHOLD bit extends the minimum setup and hold times for the SDA line. The minimum SDA setup time defines the absolute minimum time that SDA is stable before SCL transitions from low-to-high. The minimum SDA hold time defines the absolute minimum time that the current SDA value remains stable after SCL transitions from high-to-low. EXTHOLD should be set so that the minimum setup and hold times meet the SMBus Specification requirements of 250 ns and 300 ns, respectively. Table 18.2 shows the minimum setup and hold times for the two EXTHOLD settings. Setup and hold time extensions are typically necessary when SYSCLK is above 10 MHz.

Table 18.2. Minimum SDA Setup and Hold Times

EXTHOLD	Minimum SDA Setup Time	Minimum SDA Hold Time
0	T_{low} – 4 system clocks or 1 system clock + s/w delay*	3 system clocks
1	11 system clocks	12 system clocks
***Note:** Setup Time for ACK bit transmissions and the MSB of all data transfers. The s/w delay occurs between the time SMB0DAT or ACK is written and when SI is cleared. Note that if SI is cleared in the same write that defines the outgoing ACK value, s/w delay is zero.		

With the SMBTOE bit set, Timer 3 should be configured to overflow after 25 ms in order to detect SCL low timeouts (see Section “18.3.3. SCL Low Timeout” on page 202). The SMBus interface will force Timer 3 to reload while SCL is high, and allow Timer 3 to count when SCL is low. The Timer 3 interrupt service routine should be used to reset SMBus communication by disabling and re-enabling the SMBus.

SMBus Free Timeout detection can be enabled by setting the SMBFTE bit. When this bit is set, the bus will be considered free if SDA and SCL remain high for more than 10 SMBus clock source periods (see Figure 18.4). When a Free Timeout is detected, the interface will respond as if a STOP was detected (an interrupt will be generated, and STO will be set).

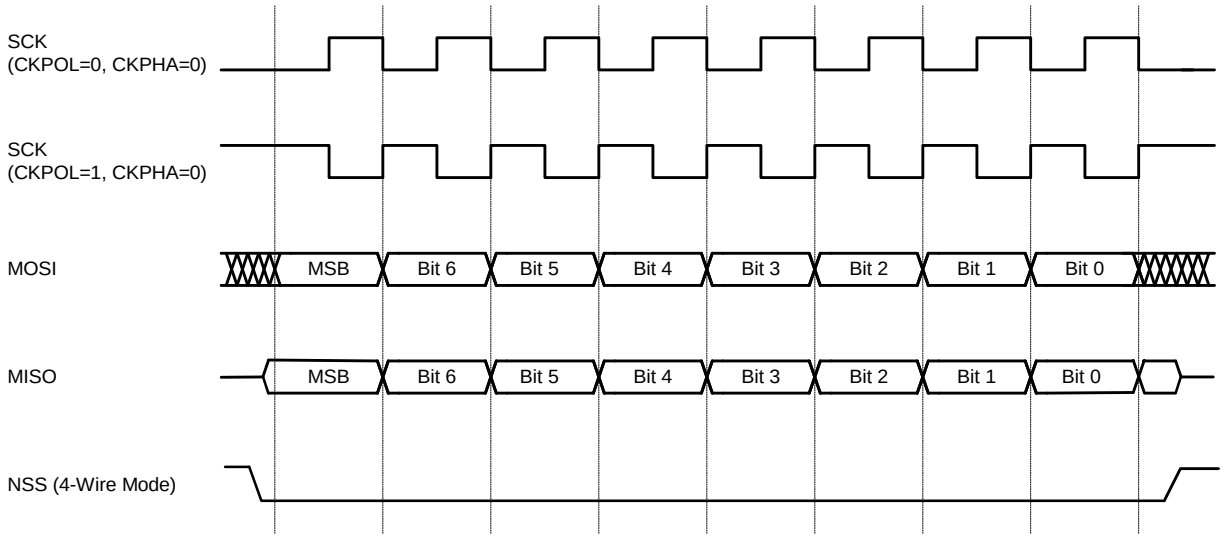

Figure 20.6. Slave Mode Data/Clock Timing (CKPHA = 0)

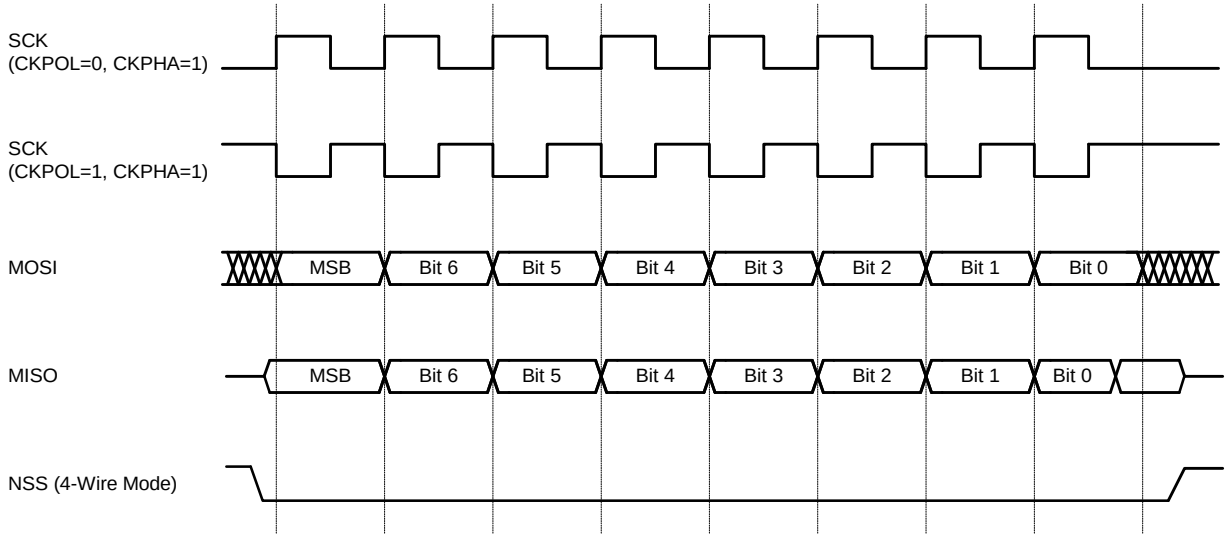

Figure 20.7. Slave Mode Data/Clock Timing (CKPHA = 1)

SFR Definition 21.3. CKCON: Clock Control

SFR Page: all pages
SFR Address: 0x8E

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
T3MH	T3ML	T2MH	T2ML	T1M	T0M	SCA1	SCA0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: T3MH: Timer 3 High Byte Clock Select.
This bit selects the clock supplied to the Timer 3 high byte if Timer 3 is configured in split 8-bit timer mode. T3MH is ignored if Time 3 is in any other mode.
0: Timer 3 high byte uses the clock defined by the T3XCLK bit in TMR3CN.
1: Timer 3 high byte uses the system clock.

Bit 6: T3ML: Timer 3 Low Byte Clock Select.
This bit selects the clock supplied to Timer 3. If Timer 3 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer.
0: Timer 3 low byte uses the clock defined by the T3XCLK bit in TMR3CN.
1: Timer 3 low byte uses the system clock.

Bit 5: T2MH: Timer 2 High Byte Clock Select.
This bit selects the clock supplied to the Timer 2 high byte if Timer 2 is configured in split 8-bit timer mode. T2MH is ignored if Timer 2 is in any other mode.
0: Timer 2 high byte uses the clock defined by the T2XCLK bit in TMR2CN.
1: Timer 2 high byte uses the system clock.

Bit 4: T2ML: Timer 2 Low Byte Clock Select.
This bit selects the clock supplied to Timer 2. If Timer 2 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer.
0: Timer 2 low byte uses the clock defined by the T2XCLK bit in TMR2CN.
1: Timer 2 low byte uses the system clock.

Bit 3: T1M: Timer 1 Clock Select.
This select the clock source supplied to Timer 1. T1M is ignored when C/T1 is set to logic ‘1’.
0: Timer 1 uses the clock defined by the prescale bits, SCA1–SCA0.
1: Timer 1 uses the system clock.

Bit 2: T0M: Timer 0 Clock Select.
This bit selects the clock source supplied to Timer 0. T0M is ignored when C/T0 is set to logic ‘1’.
0: Counter/Timer 0 uses the clock defined by the prescale bits, SCA1–SCA0.
1: Counter/Timer 0 uses the system clock.

Bits 1–0: SCA1–SCA0: Timer 0/1 Prescale Bits.
These bits control the division of the clock supplied to Timer 0 and/or Timer 1 if configured to use prescaled clock inputs.

SCA1	SCA0	Prescaled Clock
0	0	System clock divided by 12
0	1	System clock divided by 4
1	0	System clock divided by 48
1	1	External clock divided by 8

Note: External clock divided by 8 is synchronized with the system clock.

21.3. Timer 3

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

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

21.3.1. 16-bit Timer with Auto-Reload

When T3SPLIT (TMR3CN.3) is zero, Timer 3 operates as a 16-bit timer with auto-reload. Timer 3 can be clocked by SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 3 reload registers (TMR3RLH and TMR3RLL) is loaded into the Timer 3 register as shown in Figure 21.6, and the Timer 3 High Byte Overflow Flag (TMR3CN.7) is set. If Timer 3 interrupts are enabled (if EIE1.7 is set), an interrupt will be generated on each Timer 3 overflow. Additionally, if Timer 3 interrupts are enabled and the TF3LEN bit is set (TMR3CN.5), an interrupt will be generated each time the lower 8 bits (TMR3L) overflow from 0xFF to 0x00.

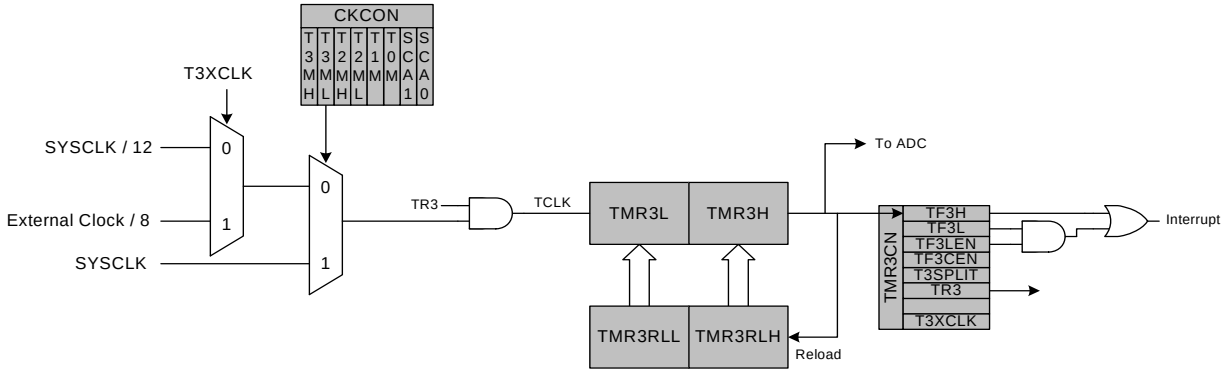

Figure 21.6. Timer 3 16-Bit Mode Block Diagram

22.2. Capture/Compare Modules

Each module can be configured to operate independently in one of six operation modes: Edge-triggered Capture, Software Timer, High Speed Output, Frequency Output, 8-Bit Pulse Width Modulator, or 16-Bit Pulse Width Modulator. Each module has Special Function Registers (SFRs) associated with it in the CIP-51 system controller. These registers are used to exchange data with a module and configure the module's mode of operation.

Table 22.2 summarizes the bit settings in the PCA0CPMn registers used to select the PCA0 capture/compare module's operating modes. Setting the ECCFn bit in a PCA0CPMn register enables the module's CCFn interrupt. Note: PCA0 interrupts must be globally enabled before individual CCFn interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit (IE.7) and the EPCA0 bit (EIE1.3) to logic '1'. See Figure 22.3 for details on the PCA interrupt configuration.

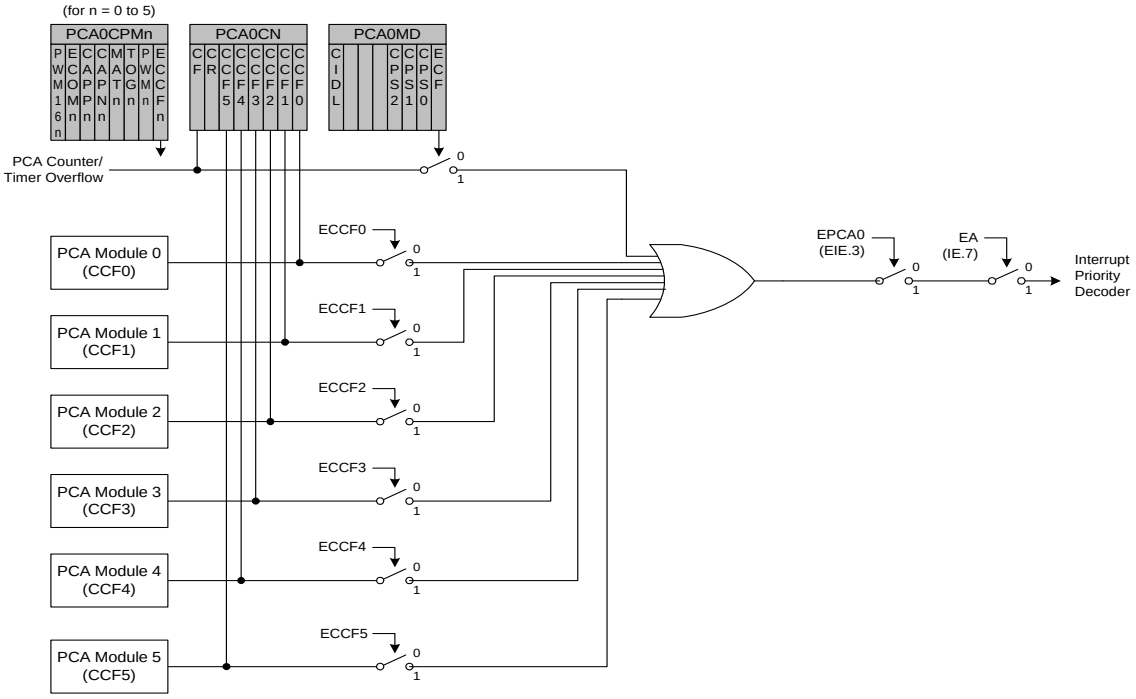

Figure 22.3. PCA Interrupt Block Diagram