

Welcome to [E-XFL.COM](https://www.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	25MHz
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, POR, PWM, 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	-
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/c8051f349-gqr

C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

Table 17.2. Minimum SDA Setup and Hold Times	193
Table 17.3. Sources for Hardware Changes to SMB0CN	197
Table 17.4. SMBus Status Decoding	203
18. UART0	
Table 18.1. Timer Settings for Standard Baud Rates Using the Internal Oscillator	212
19. UART1 (C8051F340/1/4/5/8/A/B/C Only)	
Table 19.1. Baud Rate Generator Settings for Standard Baud Rates	214
20. Enhanced Serial Peripheral Interface (SPI0)	
Table 20.1. SPI Slave Timing Parameters	234
22. Programmable Counter Array (PCA0)	
Table 22.1. PCA Timebase Input Options	256
Table 22.2. PCA0CPM Register Settings for PCA Capture/Compare Modules	257
Table 22.3. Watchdog Timer Timeout Intervals1	265

C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

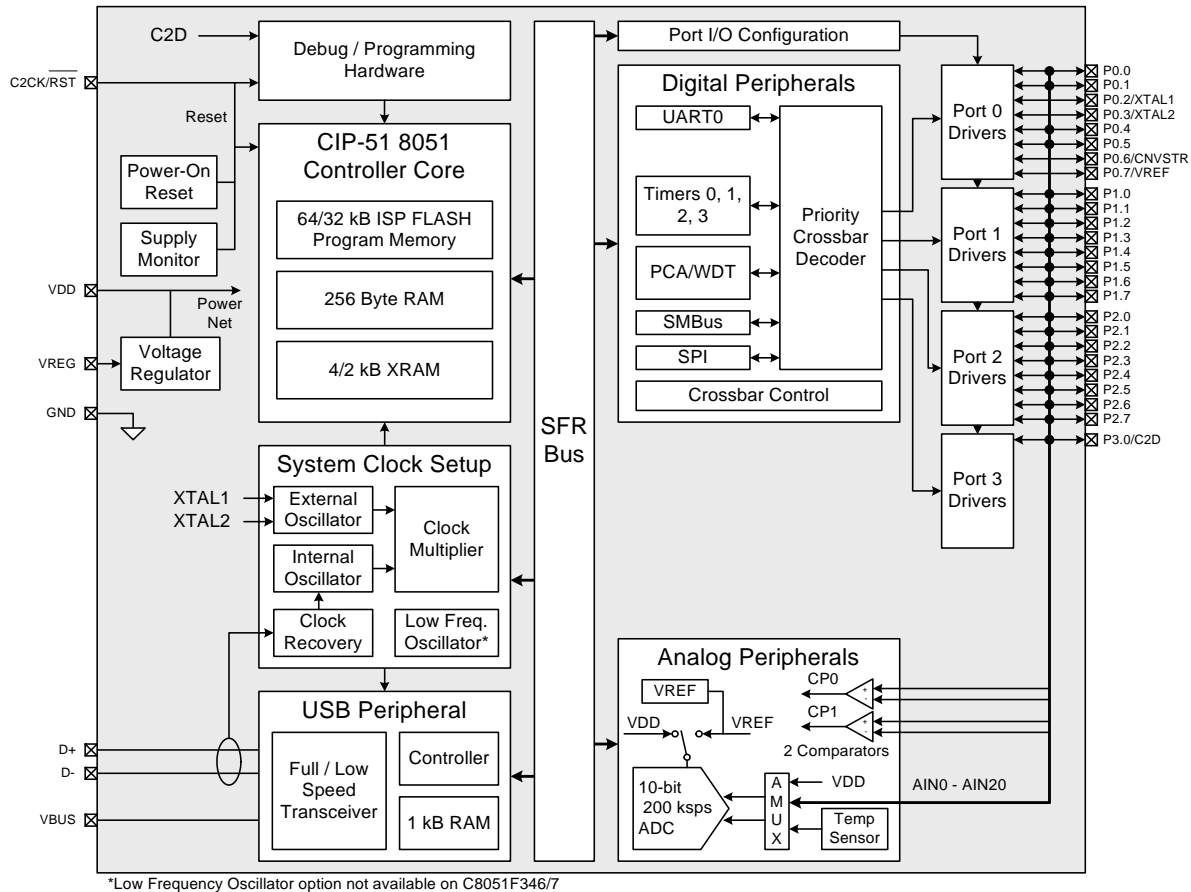


Figure 1.2. C8051F342/3/6/7 Block Diagram

C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

SFR Definition 7.4. CPT1CN: Comparator1 Control

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP1EN	CP1OUT	CP1RIF	CP1FIF	CP1HYP1	CP1HYP0	CP1HYN1	CP1HYN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9A
<p>Bit7: CP1EN: Comparator1 Enable Bit. 0: Comparator1 Disabled. 1: Comparator1 Enabled.</p> <p>Bit6: CP1OUT: Comparator1 Output State Flag. 0: Voltage on CP1+ < CP1−. 1: Voltage on CP1+ > CP1−.</p> <p>Bit5: CP1RIF: Comparator1 Rising-Edge Flag. 0: No Comparator1 Rising Edge has occurred since this flag was last cleared. 1: Comparator1 Rising Edge has occurred.</p> <p>Bit4: CP1FIF: Comparator1 Falling-Edge Flag. 0: No Comparator1 Falling-Edge has occurred since this flag was last cleared. 1: Comparator1 Falling-Edge has occurred.</p> <p>Bits3–2: CP1HYP1–0: Comparator1 Positive Hysteresis Control Bits. 00: Positive Hysteresis Disabled. 01: Positive Hysteresis = 5 mV. 10: Positive Hysteresis = 10 mV. 11: Positive Hysteresis = 20 mV.</p> <p>Bits1–0: CP1HYN1–0: Comparator1 Negative Hysteresis Control Bits. 00: Negative Hysteresis Disabled. 01: Negative Hysteresis = 5 mV. 10: Negative Hysteresis = 10 mV. 11: Negative Hysteresis = 20 mV.</p>								

C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

9.2.6. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFRs). The SFRs provide control and data exchange with the CIP-51's resources and peripherals. The CIP-51 duplicates the SFRs found in a typical 8051 implementation as well as implementing additional SFRs used to configure and access the sub-systems unique to the MCU. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 9.2 lists the SFRs implemented in the CIP-51 System Controller.

The SFR registers are accessed anytime the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFRs with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, SCON0, IE, etc.) are bit-addressable as well as byte-addressable. All other SFRs are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided. Refer to the corresponding pages of the datasheet, as indicated in Table 9.3, for a detailed description of each register.

Table 9.2. Special Function Register (SFR) Memory Map

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	ACC	XBR0	XBR1	XBR2	IT01CF	SMOD1	EIE1	EIE2
D8	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	PCA0CPM3	PCA0CPM4	P3SKIP
D0	PSW	REF0CN	SCON1	SBUF1	P0SKIP	P1SKIP	P2SKIP	USB0XCN
C8	TMR2CN	REG0CN	TMR2RLL	TMR2RLH	TMR2L	TMR2H	-	-
C0	SMB0CN	SMB0CF	SMB0DAT	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	P4
B8	IP	CLKMUL	AMX0N	AMX0P	ADC0CF	ADC0L	ADC0H	-
B0	P3	OSCXCN	OSCICN	OSCICL	SBRL1	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	P1	TMR3CN	TMR3RLL	TMR3RLH	TMR3L	TMR3H	USB0ADR	USB0DAT
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)

(bit addressable)

C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

Table 9.3. Special Function Registers (Continued)

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

Register	Address	Description	Page
P1MDIN	0xF2	Port 1 Input Mode Configuration	152
P1MDOUT	0xA5	Port 1 Output Mode Configuration	152
P1SKIP	0xD5	Port 1 Skip	153
P2	0xA0	Port 2 Latch	153
P2MDIN	0xF3	Port 2 Input Mode Configuration	153
P2MDOUT	0xA6	Port 2 Output Mode Configuration	154
P2SKIP	0xD6	Port 2 Skip	154
P3	0xB0	Port 3 Latch	155
P3MDIN	0xF4	Port 3 Input Mode Configuration	155
P3MDOUT	0xA7	Port 3 Output Mode Configuration	155
P3SKIP	0xDF	Port 3Skip	156
P4	0xC7	Port 4 Latch	156
P4MDIN	0xF5	Port 4 Input Mode Configuration	157
P4MDOUT	0xAE	Port 4 Output Mode Configuration	157
PCA0CN	0xD8	PCA Control	266
PCA0CPH0	0xFC	PCA Capture 0 High	270
PCA0CPH1	0xEA	PCA Capture 1 High	270
PCA0CPH2	0xEC	PCA Capture 2 High	270
PCA0CPH3	0xEE	PCA Capture 3High	270
PCA0CPH4	0xFE	PCA Capture 4 High	270
PCA0CPL0	0xFB	PCA Capture 0 Low	269
PCA0CPL1	0xE9	PCA Capture 1 Low	269
PCA0CPL2	0xEB	PCA Capture 2 Low	269
PCA0CPL3	0xED	PCA Capture 3 Low	269
PCA0CPL4	0xFD	PCA Capture 4 Low	269
PCA0CPM0	0xDA	PCA Module 0 Mode Register	268
PCA0CPM1	0xDB	PCA Module 1 Mode Register	268
PCA0CPM2	0xDC	PCA Module 2 Mode Register	268
PCA0CPM3	0xDD	PCA Module 3 Mode Register	268
PCA0CPM4	0xDE	PCA Module 4 Mode Register	268
PCA0H	0xFA	PCA Counter High	269
PCA0L	0xF9	PCA Counter Low	269
PCA0MD	0xD9	PCA Mode	267
PCON	0x87	Power Control	98
PFE0CN	0xAF	Prefetch Engine Control	99
PSCTL	0x8F	Program Store R/W Control	112
PSW	0xD0	Program Status Word	87
REF0CN	0xD1	Voltage Reference Control	58
REG0CN	0xC9	Voltage Regulator Control	72
RSTSRC	0xEF	Reset Source Configuration/Status	105
SBCON1	0xAC	UART1 Baud Rate Generator Control	220
SBRLH1	0xB5	UART1 Baud Rate Generator High	221
SBRL1	0xB4	UART1 Baud Rate Generator Low	221
SBUF1	0xD3	UART1 Data Buffer	220
SCON1	0xD2	UART1 Control	218

IT0	IN0PL	INT0 Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

IT1	IN1PL	INT1 Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

$\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ are assigned to Port pins as defined in the IT01CF register (see SFR Definition 9.13). Note that $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ Port pin assignments are independent of any Crossbar assignments. $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ will monitor their assigned Port pins without disturbing the peripheral that was assigned the Port pin via the Crossbar. To assign a Port pin only to $\overline{\text{INT0}}$ and/or $\overline{\text{INT1}}$, configure the Crossbar to skip the selected pin(s). This is accomplished by setting the associated bit in register XBR0 (see **Section “15.1. Priority Crossbar Decoder” on page 144** for complete details on configuring the Crossbar). In the typical configuration, the external interrupt pin should be skipped in the crossbar and configured as open-drain with the pin latch set to '1'.

IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flags for the $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ external interrupts, respectively. If an $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag remains logic 1 while the input is active as defined by the corresponding polarity bit (IN0PL or IN1PL); the flag remains logic 0 while the input is inactive. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

9.3.3. 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 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 9.4.

9.3.4. 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/erase operations and USB FIFO MOVX accesses (see **Section “13.2. Accessing USB FIFO Space” on page 115**). 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.

11.3. External Reset

The external $\overline{\text{RST}}$ pin provides a means for external circuitry to force the device into a reset state. Asserting an active-low signal on the $\overline{\text{RST}}$ pin generates a reset; an external pull-up and/or decoupling of the $\overline{\text{RST}}$ pin may be necessary to avoid erroneous noise-induced resets. See Table 11.1 for complete $\overline{\text{RST}}$ pin specifications. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

11.4. Missing Clock Detector Reset

The Missing Clock Detector (MCD) is a one-shot circuit that is triggered by the system clock. If more than 100 μs pass between rising edges on the system clock, the one-shot will time out and generate a reset. After a MCD reset, the MCDRSF flag (RSTSRC.2) will read '1', signifying the MCD as the reset source; otherwise, this bit reads '0'. Writing a '1' to the MCDRSF bit enables the Missing Clock Detector; writing a '0' disables it. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

11.5. Comparator0 Reset

Comparator0 can be configured as a reset source by writing a '1' to the CORSEF flag (RSTSRC.5). Comparator0 should be enabled and allowed to settle prior to writing to CORSEF to prevent any turn-on chatter on the output from generating an unwanted reset. The Comparator0 reset is active-low: if the non-inverting input voltage (on CP0+) is less than the inverting input voltage (on CP0-), a system reset is generated. After a Comparator0 reset, the CORSEF flag (RSTSRC.5) will read '1' signifying Comparator0 as the reset source; otherwise, this bit reads '0'. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

11.6. PCA Watchdog Timer Reset

The programmable Watchdog Timer (WDT) function of the Programmable Counter Array (PCA) can be used to prevent software from running out of control during a system malfunction. The PCA WDT function can be enabled or disabled by software as described in **Section “22.3. Watchdog Timer Mode” on page 264**; the WDT is enabled and clocked by SYSCLK / 12 following any reset. If a system malfunction prevents user software from updating the WDT, a reset is generated and the WDTRSF bit (RSTSRC.5) is set to '1'. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

11.7. Flash Error Reset

If a Flash read/write/erase or program read targets an illegal address, a system reset is generated. This may occur due to any of the following:

- A Flash write or erase is attempted above user code space. This occurs when PSWE is set to “1”, and a MOVX write operation is attempted above address 0x7FFF (32 kB Flash devices) or 0xFBFF (64 kB Flash devices).
- A Flash read is attempted above user code space. This occurs when a MOVC operation is attempted above address 0x7FFF (32 kB Flash devices) or 0xFBFF (64 kB Flash devices).
- A Program read is attempted above user code space. This occurs when user code attempts to branch to an address above 0x7FFF (32 kB Flash devices) or 0xFBFF (64 kB Flash devices).
- A Flash read, write or erase attempt is restricted due to a Flash security setting (see **Section “12.3. Security Options” on page 109**).
- A Flash Write or Erase is attempted when the V_{DD} monitor is not enabled.

The FERROR bit (RSTSRC.6) is set following a Flash error reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

13. External Data Memory Interface and On-Chip XRAM

4k Bytes (C8051F340/2/4/6/A/C/D) or 2k Bytes (C8051F341/3/5/7/8/9/B) of RAM are included on-chip, and mapped into the external data memory space (XRAM). The 1k Bytes of USB FIFO space can also be mapped into XRAM address space for additional general-purpose data storage. Additionally, an External Memory Interface (EMIF) is available on the C8051F340/1/4/5/8/C devices, which can be used to access off-chip data memories and memory-mapped devices connected to the GPIO ports. The external memory space may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using the MOVX indirect addressing mode using R0 or R1. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the External Memory Interface Control Register (EMI0CN, shown in SFR Definition 13.1). Note: the MOVX instruction can also be used for writing to the FLASH memory. See **Section “12. Flash Memory” on page 107** for details. The MOVX instruction accesses XRAM by default.

13.1. Accessing XRAM

The XRAM memory space is accessed using the MOVX instruction. The MOVX instruction has two forms, both of which use an indirect addressing method. The first method uses the Data Pointer, DPTR, a 16-bit register which contains the effective address of the XRAM location to be read from or written to. The second method uses R0 or R1 in combination with the EMI0CN register to generate the effective XRAM address. Examples of both of these methods are given below.

13.1.1. 16-Bit MOVX Example

The 16-bit form of the MOVX instruction accesses the memory location pointed to by the contents of the DPTR register. The following series of instructions reads the value of the byte at address 0x1234 into the accumulator A:

```
MOV    DPTR, #1234h      ; load DPTR with 16-bit address to read (0x1234)
MOVX   A, @DPTR          ; load contents of 0x1234 into accumulator A
```

The above example uses the 16-bit immediate MOV instruction to set the contents of DPTR. Alternately, the DPTR can be accessed through the SFR registers DPH, which contains the upper 8-bits of DPTR, and DPL, which contains the lower 8-bits of DPTR.

13.1.2. 8-Bit MOVX Example

The 8-bit form of the MOVX instruction uses the contents of the EMI0CN SFR to determine the upper 8-bits of the effective address to be accessed and the contents of R0 or R1 to determine the lower 8-bits of the effective address to be accessed. The following series of instructions read the contents of the byte at address 0x1234 into the accumulator A.

```
MOV    EMI0CN, #12h      ; load high byte of address into EMI0CN
MOV    R0, #34h          ; load low byte of address into R0 (or R1)
MOVX   a, @R0            ; load contents of 0x1234 into accumulator A
```

13.5.2. Non-multiplexed Configuration

In Non-multiplexed mode, the Data Bus and the Address Bus pins are not shared. An example of a Non-multiplexed Configuration is shown in Figure 13.3. See **Section “13.7.1. Non-multiplexed Mode” on page 124** for more information about Non-multiplexed operation.

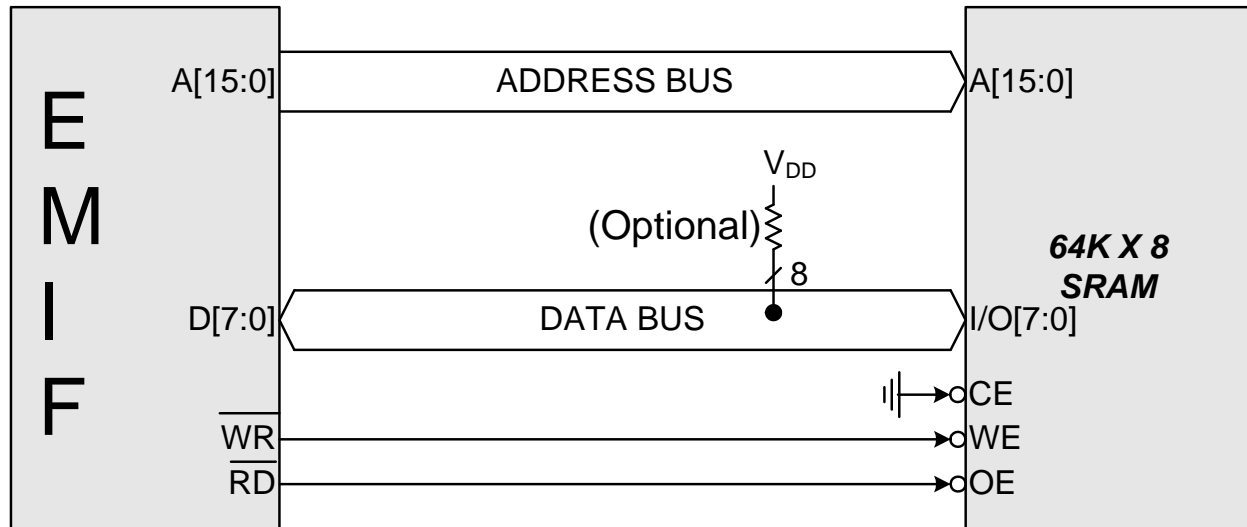


Figure 13.3. Non-multiplexed Configuration Example

13.6. Memory Mode Selection

The external data memory space can be configured in one of four modes, shown in Figure 13.4, based on the EMIOCF Mode bits in the EMI0CF register (SFR Definition 13.2). These modes are summarized below. More information about the different modes can be found in **Section “13.7. Timing” on page 122**.

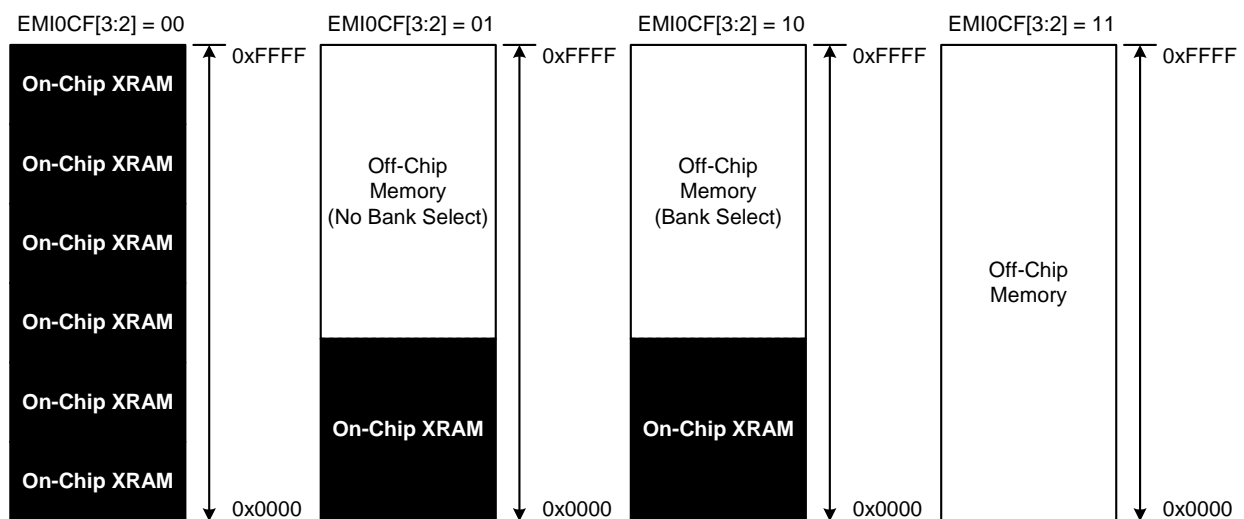


Figure 13.4. EMIF Operating Modes

13.7.1. Non-multiplexed Mode

13.7.1.1. 16-bit MOVX: EMI0CF[4:2] = '101', '110', or '111'.

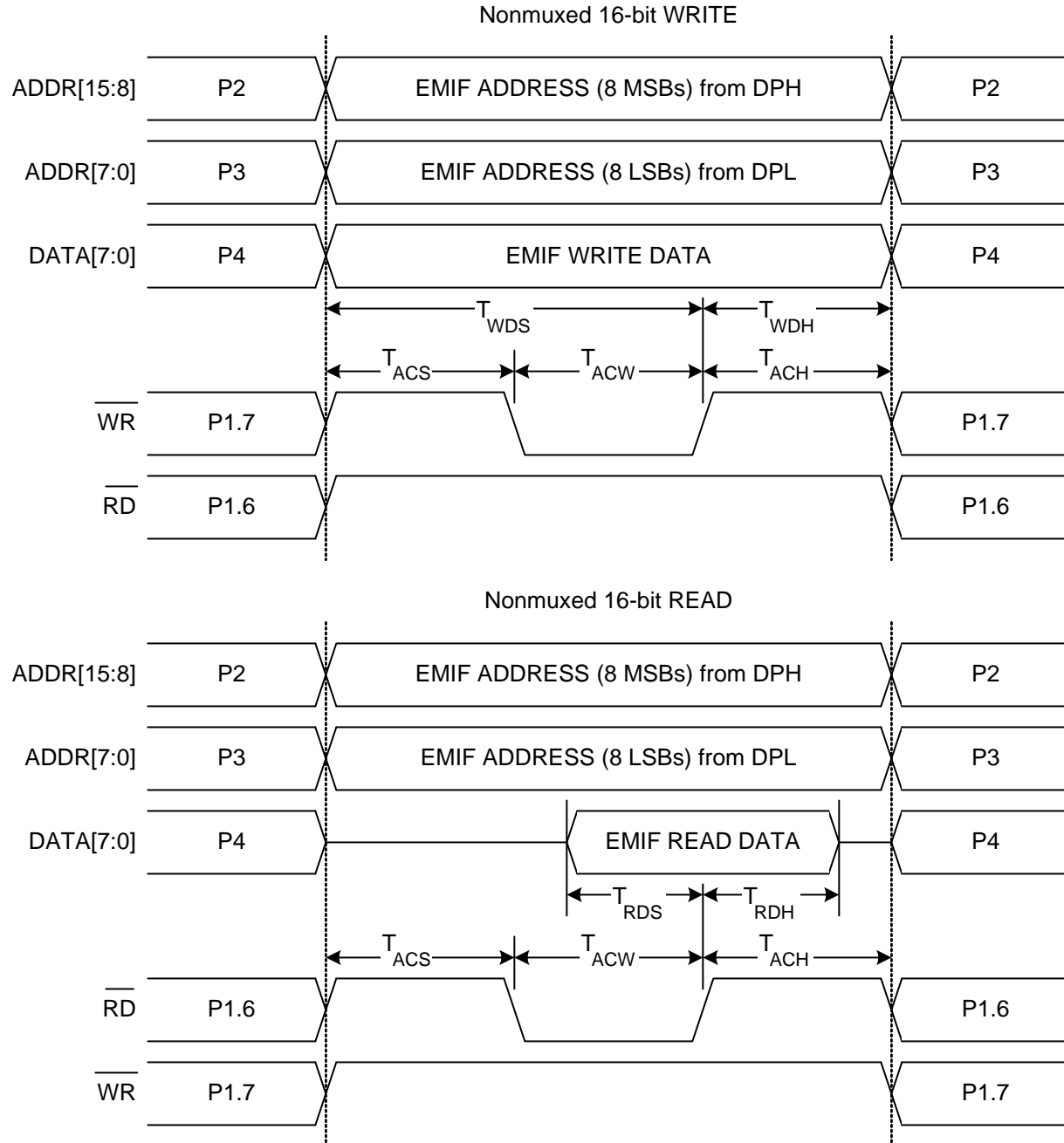


Figure 13.5. Non-multiplexed 16-bit MOVX Timing

13.7.2.3.8-bit MOVX with Bank Select: EMI0CF[4:2] = '010'.

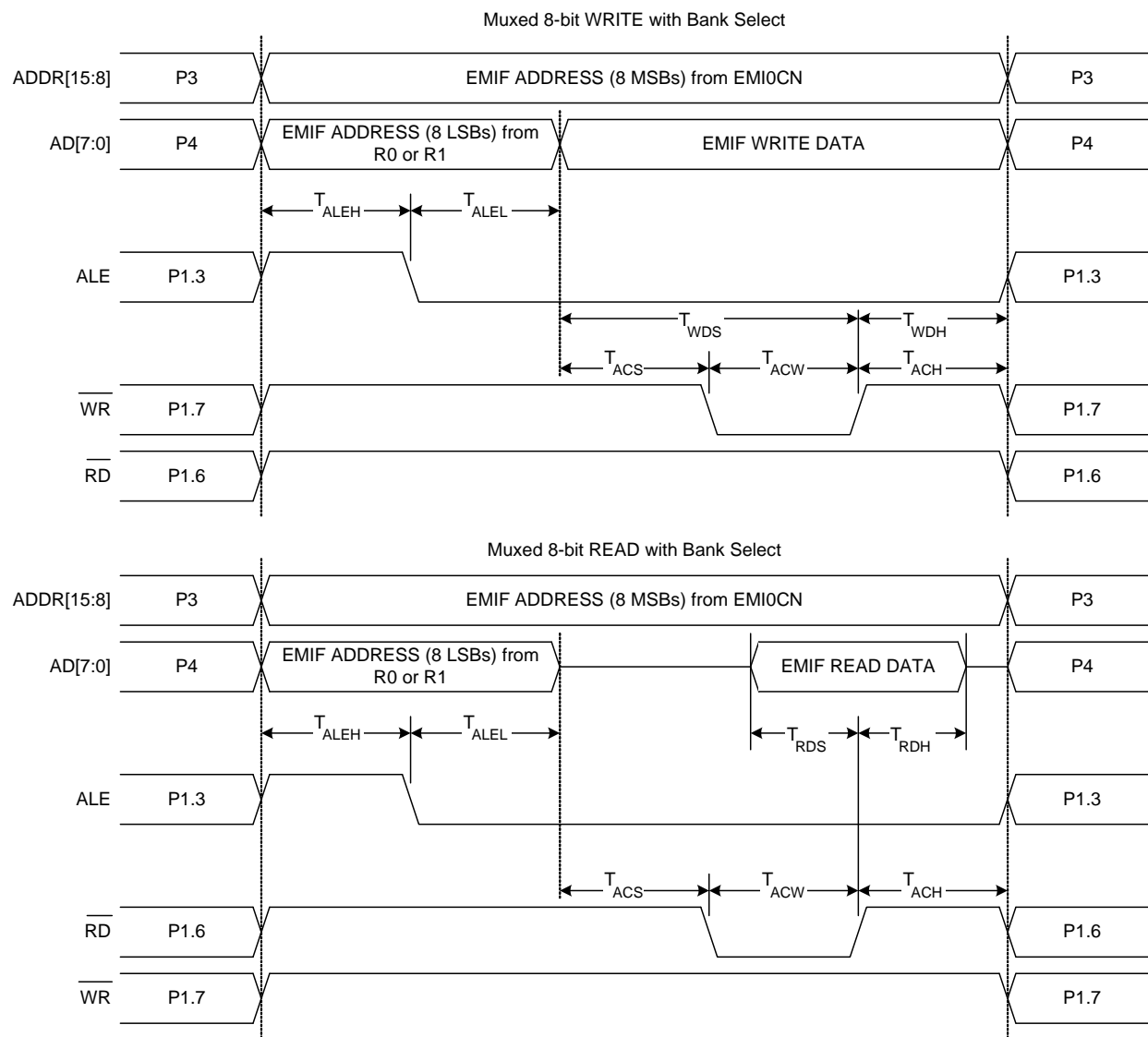


Figure 13.10. Multiplexed 8-bit MOVX with Bank Select Timing

16.5.2. FIFO Double Buffering

FIFO slots for Endpoints1-3 can be configured for double-buffered mode. In this mode, the maximum packet size is halved and the FIFO may contain two packets at a time. This mode is available for Endpoints1-3. When an endpoint is configured for Split Mode, double buffering may be enabled for the IN Endpoint and/or the OUT endpoint. When Split Mode is not enabled, double-buffering may be enabled for the entire endpoint FIFO. See Table 16.3 for a list of maximum packet sizes for each FIFO configuration.

Table 16.3. FIFO Configurations

Endpoint Number	Split Mode Enabled?	Maximum IN Packet Size (Double Buffer Disabled / Enabled)	Maximum OUT Packet Size (Double Buffer Disabled / Enabled)
0	N/A	64	
1	N	128 / 64	
	Y	64 / 32	64 / 32
2	N	256 / 128	
	Y	128 / 64	128 / 64
3	N	512 / 256	
	Y	256 / 128	256 / 128

16.5.1. FIFO Access

Each endpoint FIFO is accessed through a corresponding FIFOn register. A read of an endpoint FIFOn register unloads one byte from the FIFO; a write of an endpoint FIFOn register loads one byte into the endpoint FIFO. When an endpoint FIFO is configured for Split Mode, a read of the endpoint FIFOn register unloads one byte from the OUT endpoint FIFO; a write of the endpoint FIFOn register loads one byte into the IN endpoint FIFO.

USB Register Definition 16.6. FIFOn: USB0 Endpoint FIFO Access

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
FIFODATA								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x20 - 0x23

USB Addresses 0x20–0x23 provide access to the 4 pairs of endpoint FIFOs:

IN/OUT Endpoint FIFO	USB Address
0	0x20
1	0x21
2	0x22
3	0x23

Writing to the FIFO address loads data into the IN FIFO for the corresponding endpoint. Reading from the FIFO address unloads data from the OUT FIFO for the corresponding endpoint.

C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

Table 16.4. USB Transceiver Electrical Characteristics

$V_{DD} = 3.0$ to 3.6 V, -40 to $+85$ °C unless otherwise specified

Parameters	Symbol	Conditions	Min	Typ	Max	Units
Transmitter						
Output High Voltage	V_{OH}		2.8			V
Output Low Voltage	V_{OL}				0.8	V
Output Crossover Point	V_{CRS}		1.3		2.0	V
Output Impedance	Z_{DRV}	Driving High Driving Low		38 38		Ω
Pull-up Resistance	R_{PU}	Full Speed (D+ Pull-up) Low Speed (D– Pull-up)	1.425	1.5	1.575	k Ω
Output Rise Time	T_R	Low Speed Full Speed	75 4		300 20	ns
Output Fall Time	T_F	Low Speed Full Speed	75 4		300 20	ns
Receiver						
Differential Input Sensitivity	V_{DI}	$ (D+) - (D-) $	0.2			V
Differential Input Common Mode Range	V_{CM}		0.8		2.5	V
Input Leakage Current	I_L	Pullups Disabled		<1.0		μA

Note: Refer to the USB Specification for timing diagrams and symbol definitions.

17.3.2. Clock Low Extension

SMBus provides a clock synchronization mechanism, similar to I2C, which allows devices with different speed capabilities to coexist on the bus. A clock-low extension is used during a transfer in order to allow slower slave devices to communicate with faster masters. The slave may temporarily hold the SCL line LOW to extend the clock low period, effectively decreasing the serial clock frequency.

17.3.3. SCL Low Timeout

If the SCL line is held low by a slave device on the bus, no further communication is possible. Furthermore, the master cannot force the SCL line high to correct the error condition. To solve this problem, the SMBus protocol specifies that devices participating in a transfer must detect any clock cycle held low longer than 25 ms as a “timeout” condition. Devices that have detected the timeout condition must reset the communication no later than 10 ms after detecting the timeout condition.

When the SMBTOE bit in SMB0CF is set, Timer 3 is used to detect SCL low timeouts. Timer 3 is forced to reload when SCL is high, and allowed to count when SCL is low. With Timer 3 enabled and configured to overflow after 25 ms (and SMBTOE set), the Timer 3 interrupt service routine can be used to reset (disable and re-enable) the SMBus in the event of an SCL low timeout.

17.3.4. SCL High (SMBus Free) Timeout

The SMBus specification stipulates that if the SCL and SDA lines remain high for more than 50 μ s, the bus is designated as free. When the SMBFTE bit in SMB0CF is set, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods. If the SMBus is waiting to generate a Master START, the START will be generated following this timeout. Note that a clock source is required for free timeout detection, even in a slave-only implementation.

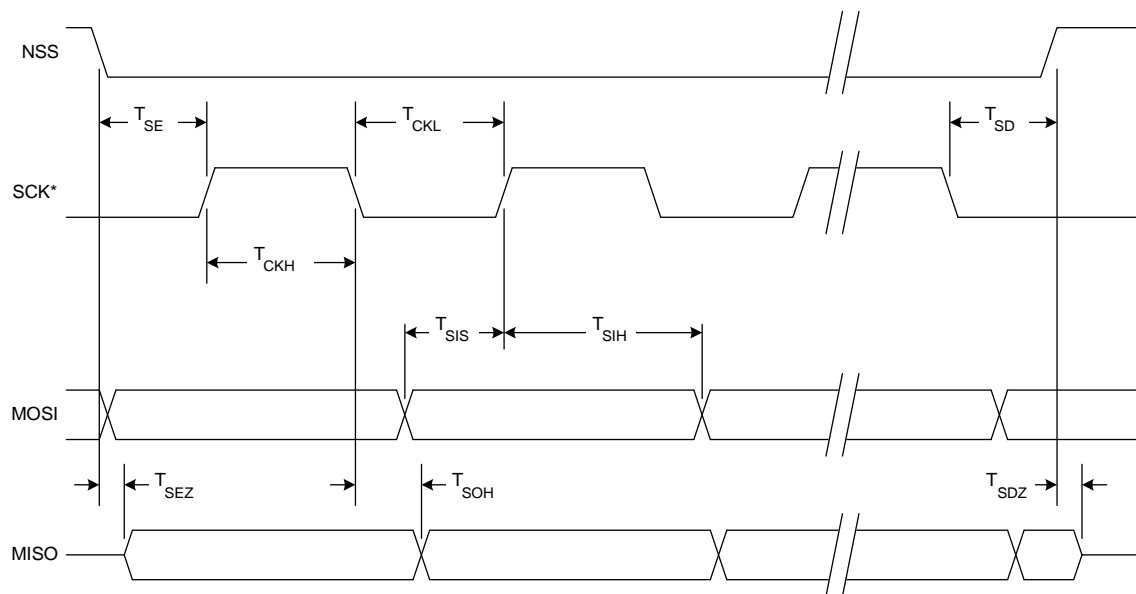
17.4. Using the SMBus

The SMBus can operate in both Master and Slave modes. The interface provides timing and shifting control for serial transfers; higher level protocol is determined by user software. The SMBus interface provides the following application-independent features:

- Byte-wise serial data transfers
- Clock signal generation on SCL (Master Mode only) and SDA data synchronization
- Timeout/bus error recognition, as defined by the SMB0CF configuration register
- START/STOP timing, detection, and generation
- Bus arbitration
- Interrupt generation
- Status information

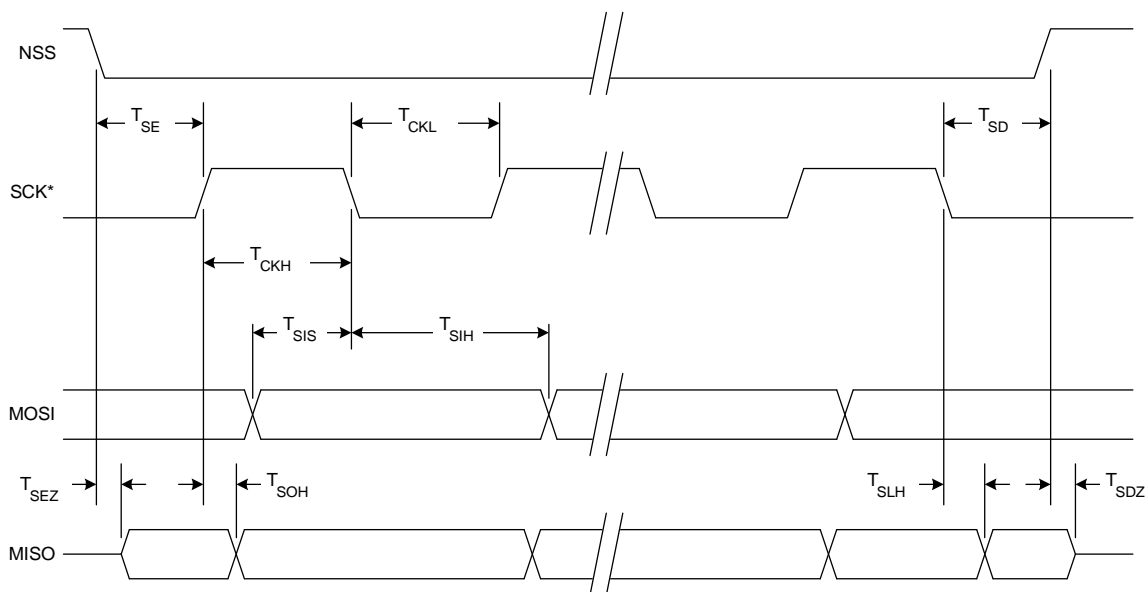
SMBus interrupts are generated for each data byte or slave address that is transferred. When transmitting, this interrupt is generated after the ACK cycle so that software may read the received ACK value; when receiving data, this interrupt is generated before the ACK cycle so that software may define the outgoing ACK value. See **Section “17.5. SMBus Transfer Modes” on page 198** for more details on transmission sequences.

Interrupts are also generated to indicate the beginning of a transfer when a master (START generated), or the end of a transfer when a slave (STOP detected). Software should read the SMB0CN (SMBus Control register) to find the cause of the SMBus interrupt. The SMB0CN register is described in **Section “17.4.2. SMB0CN Control Register” on page 195**; Table 17.4 provides a quick SMB0CN decoding reference.



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 20.10. SPI Slave Timing (CKPHA = 0)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 20.11. SPI Slave Timing (CKPHA = 1)

21.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 (TCON.5) is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or when the input signal INT0 is active as defined by bit IN0PL in register INT01CF (see **Section “9.3.2. External Interrupts”** on **page 88** for details on the external input signals INT0 and INT1).

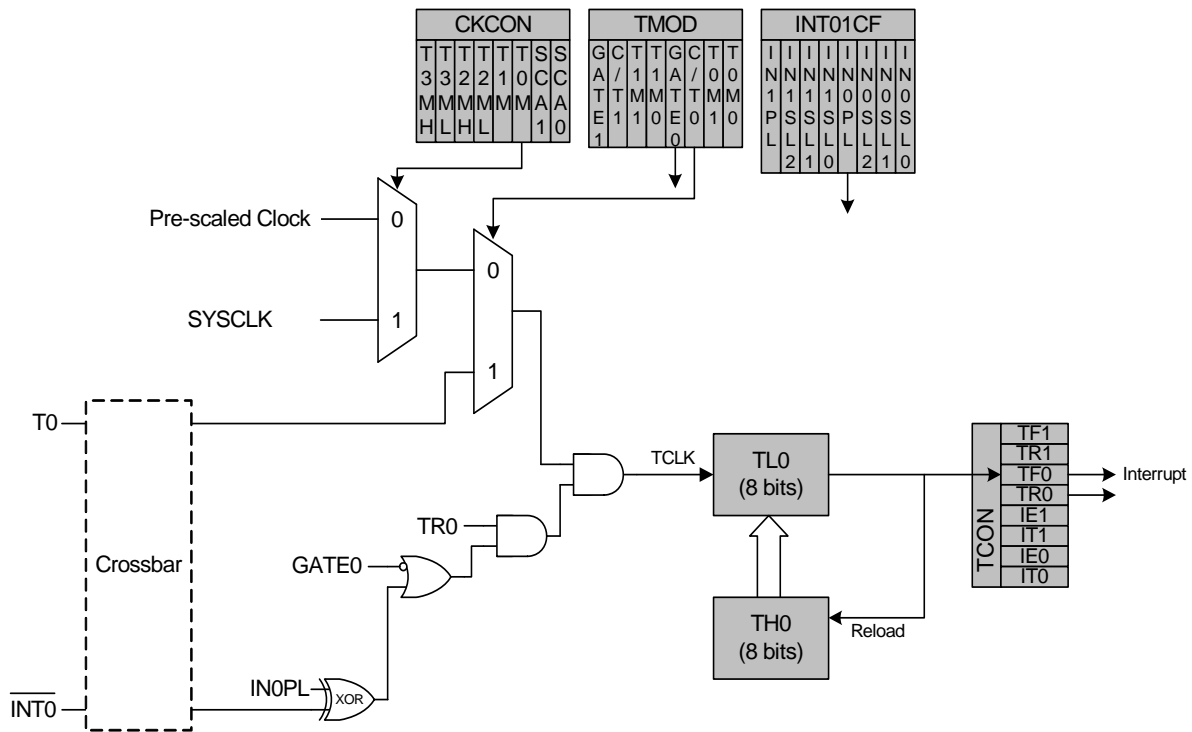


Figure 21.2. T0 Mode 2 Block Diagram

21.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3. When Timer 0 is operating in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate baud rates for the SMBus and/or UART, and/or initiate ADC conversions. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

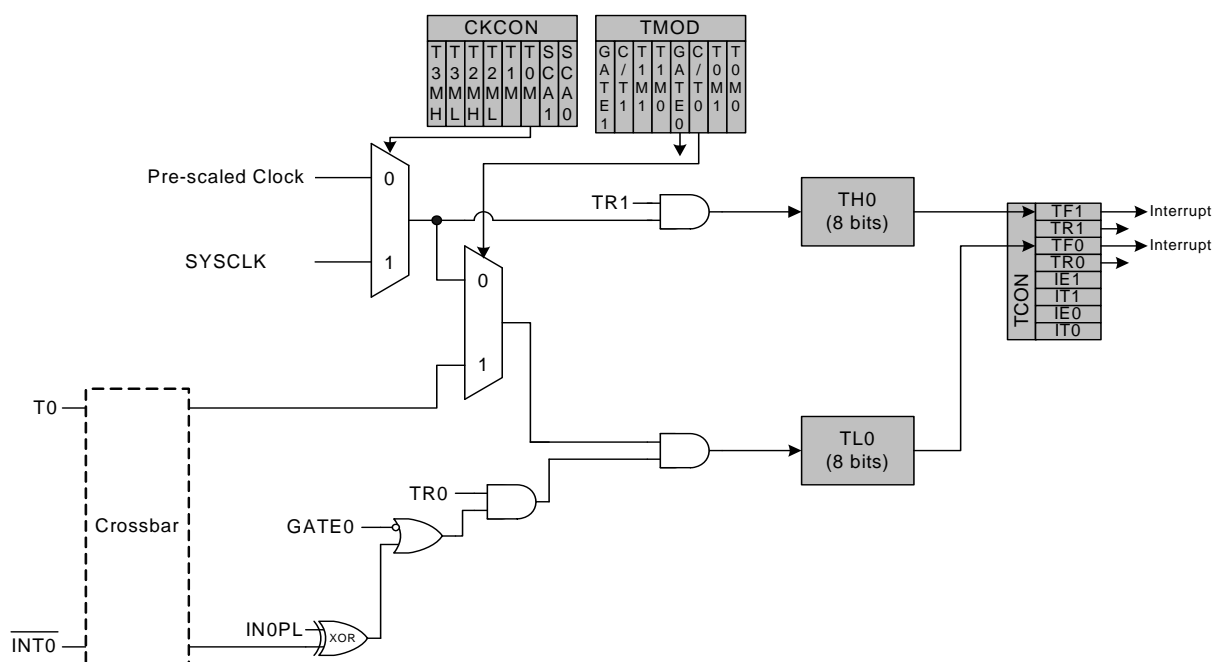


Figure 21.3. T0 Mode 3 Block Diagram

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, (split) 8-bit auto-reload mode, USB Start-of-Frame (SOF) capture mode, or Low-Frequency Oscillator (LFO) Rising Edge capture mode. The Timer 3 operation mode is defined by the T3SPLIT (TMR3CN.3), T3CE (TMR3CN.4) bits, and T3CSS (TMR3CN.1) bits.

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 '0' and T3CE = '0', 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 21.4, and the Timer 3 High Byte Overflow Flag (TMR3CN.7) is set. If Timer 3 interrupts are enabled, 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.

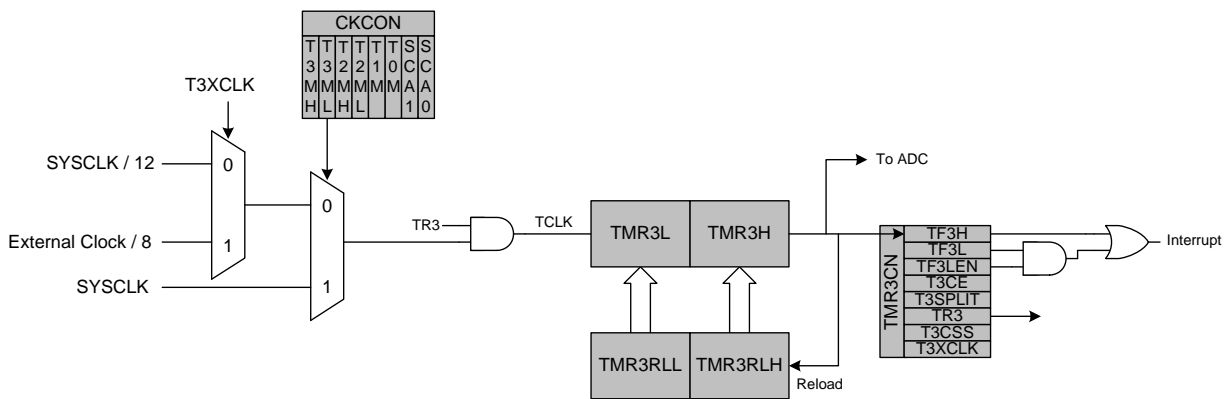


Figure 21.8. Timer 3 16-Bit Mode Block Diagram

SFR Definition 21.13. TMR3CN: Timer 3 Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF3H	TF3L	TF3LEN	T3CE	T3SPLIT	TR3	T3CSS	T3XCLK	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x91
<p>Bit7: TF3H: Timer 3 High Byte Overflow Flag. Set by hardware when the Timer 3 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 3 overflows from 0xFFFF to 0x0000. When the Timer 3 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 3 interrupt service routine. TF3H is not automatically cleared by hardware and must be cleared by software.</p> <p>Bit6: TF3L: Timer 3 Low Byte Overflow Flag. Set by hardware when the Timer 3 low byte overflows from 0xFF to 0x00. When this bit is set, an interrupt will be generated if TF3LEN is set and Timer 3 interrupts are enabled. TF3L will set when the low byte overflows regardless of the Timer 3 mode. This bit is not automatically cleared by hardware.</p> <p>Bit5: TF3LEN: Timer 3 Low Byte Interrupt Enable. This bit enables/disables Timer 3 Low Byte interrupts. If TF3LEN is set and Timer 3 interrupts are enabled, an interrupt will be generated when the low byte of Timer 3 overflows. This bit should be cleared when operating Timer 3 in 16-bit mode. 0: Timer 3 Low Byte interrupts disabled. 1: Timer 3 Low Byte interrupts enabled.</p> <p>Bit4: T3CE: Timer 3 Capture Enable 0: Capture function disabled. 1: Capture function enabled. The timer is in capture mode, with the capture event selected by bit T3CSS. Each time a capture event is received, the contents of the Timer 3 registers (TMR3H and TMR3L) are latched into the Timer 3 reload registers (TMR3RLH and TMR3RLH), and a Timer 3 interrupt is generated (if enabled).</p> <p>Bit3: T3SPLIT: Timer 3 Split Mode Enable. When this bit is set, Timer 3 operates as two 8-bit timers with auto-reload. 0: Timer 3 operates in 16-bit auto-reload mode. 1: Timer 3 operates as two 8-bit auto-reload timers.</p> <p>Bit2: TR3: Timer 3 Run Control. This bit enables/disables Timer 3. In 8-bit mode, this bit enables/disables TMR3H only; TMR3L is always enabled in this mode. 0: Timer 3 disabled. 1: Timer 3 enabled.</p> <p>Bit1: T3CSS: Timer 3 Capture Source Select. This bit selects the source of a capture event when bit T3CE is set to '1'. 0: Capture source is USB SOF event. 1: Capture source is rising edge of Low-Frequency Oscillator.</p> <p>Bit0: T3XCLK: Timer 3 External Clock Select. This bit selects the external clock source for Timer 3. If Timer 3 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 3 Clock Select bits (T3MH and T3ML in register CKCON) may still be used to select between the external clock and the system clock for either timer. 0: Timer 3 external clock selection is the system clock divided by 12. 1: Timer 3 external clock selection is the external clock divided by 8. Note that the external oscillator source divided by 8 is synchronized with the system clock.</p>								

22.2.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA counter/timer value is compared to 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 and an interrupt request is generated if CCF interrupts are 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 ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

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'.

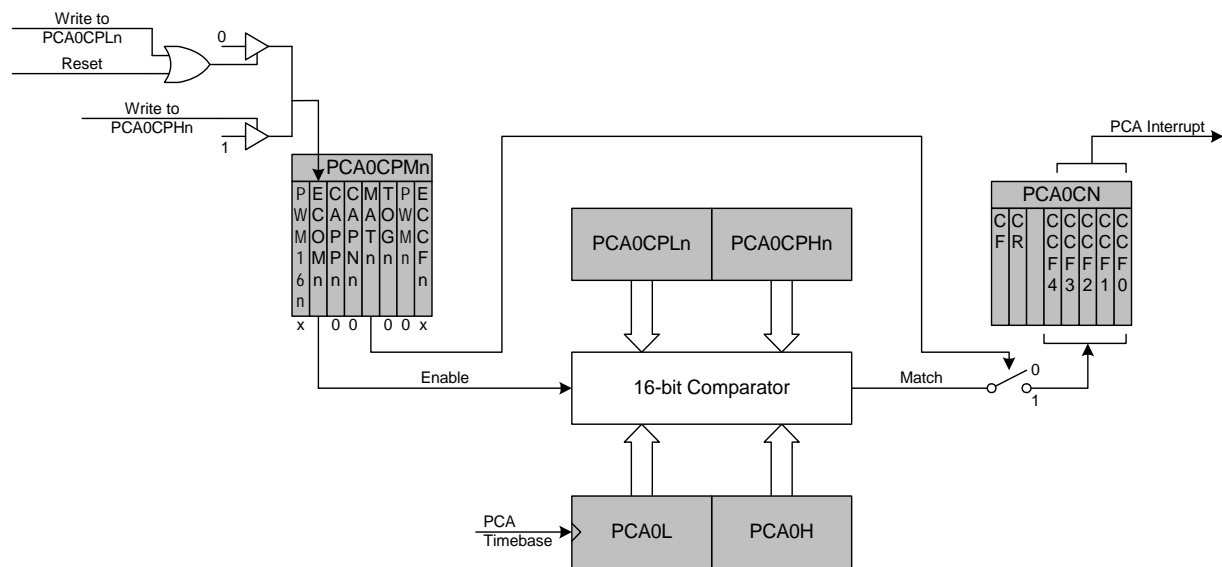


Figure 22.5. PCA Software Timer Mode Diagram