

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	25MHz
Connectivity	SMBus (2-Wire/I ² C), UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	8
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 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	10-VDFN Exposed Pad
Supplier Device Package	11-QFN (3x3)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f301-gmr

C8051F300/1/2/3/4/5

10. Flash Memory	
Figure 10.1. Flash Program Memory Map	91
11. Oscillators	
Figure 11.1. Oscillator Diagram	97
Figure 11.2. 32.768 kHz External Crystal Example	101
12. Port Input/Output	
Figure 12.1. Port I/O Functional Block Diagram	103
Figure 12.2. Port I/O Cell Block Diagram	103
Figure 12.3. Crossbar Priority Decoder with XBR0 = 0x00	104
Figure 12.4. Crossbar Priority Decoder with XBR0 = 0x44	105
13. SMBus	
Figure 13.1. SMBus Block Diagram	111
Figure 13.2. Typical SMBus Configuration	112
Figure 13.3. SMBus Transaction	113
Figure 13.4. Typical SMBus SCL Generation	117
Figure 13.5. Typical Master Transmitter Sequence	123
Figure 13.6. Typical Master Receiver Sequence	124
Figure 13.7. Typical Slave Receiver Sequence	125
Figure 13.8. Typical Slave Transmitter Sequence	126
14. UART0	
Figure 14.1. UART0 Block Diagram	131
Figure 14.2. UART0 Baud Rate Logic	132
Figure 14.3. UART Interconnect Diagram	133
Figure 14.4. 8-Bit UART Timing Diagram	133
Figure 14.5. 9-Bit UART Timing Diagram	134
Figure 14.6. UART Multi-Processor Mode Interconnect Diagram	135
15. Timers	
Figure 15.1. T0 Mode 0 Block Diagram	144
Figure 15.2. T0 Mode 2 Block Diagram	145
Figure 15.3. T0 Mode 3 Block Diagram	146
Figure 15.4. Timer 2 16-Bit Mode Block Diagram	151
Figure 15.5. Timer 2 8-Bit Mode Block Diagram	152
16. Programmable Counter Array	
Figure 16.1. PCA Block Diagram	155
Figure 16.2. PCA Counter/Timer Block Diagram	156
Figure 16.3. PCA Interrupt Block Diagram	157
Figure 16.4. PCA Capture Mode Diagram	158
Figure 16.5. PCA Software Timer Mode Diagram	159
Figure 16.6. PCA High Speed Output Mode Diagram	160
Figure 16.7. PCA Frequency Output Mode	161
Figure 16.8. PCA 8-Bit PWM Mode Diagram	162
Figure 16.9. PCA 16-Bit PWM Mode	163
Figure 16.10. PCA Module 2 with Watchdog Timer Enabled	164
17. C2 Interface	
Figure 17.1. Typical C2 Pin Sharing	175

7. Comparator0

C8051F300/1/2/3/4/5 devices include an on-chip programmable voltage comparator, which is shown in Figure 7.1. Comparator0 offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a synchronous “latched” output (CP0), or an asynchronous “raw” output (CP0A). The asynchronous CP0A signal is available even when the system clock is not active. This allows Comparator0 to operate and generate an output with the device in STOP mode. When assigned to a Port pin, the Comparator0 output may be configured as open drain or push-pull (see **Section “12.2. Port I/O Initialization” on page 106**). Comparator0 may also be used as a reset source (see **Section “9.5. Comparator0 Reset” on page 85**).

The inputs for Comparator0 are selected in the CPT0MX register (SFR Definition 7.2). The CMX0P1-CMX0P0 bits select the Comparator0 positive input; the CMX0N1-CMX0N0 bits select the Comparator0 negative input.

Important Note About Comparator Inputs: The Port pins selected as comparator inputs should be configured as analog inputs in their associated Port configuration register, and configured to be skipped by the Crossbar (for details on Port configuration, see **Section “12.3. General Purpose Port I/O” on page 108**).

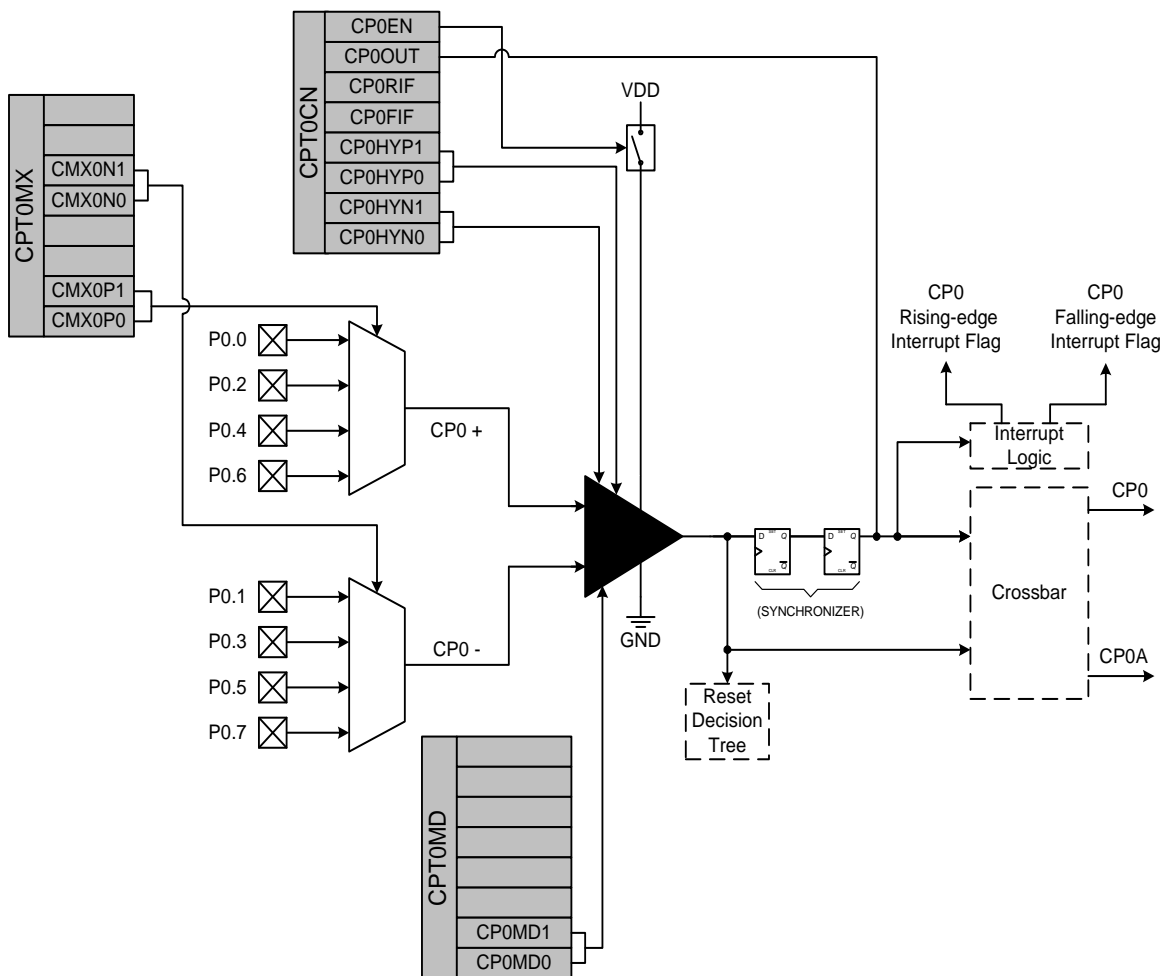


Figure 7.1. Comparator0 Functional Block Diagram

C8051F300/1/2/3/4/5

The output of Comparator0 can be polled in software, used as an interrupt source, and/or routed to a Port pin. When routed to a Port pin, the Comparator0 output is available asynchronous or synchronous to the system clock; the asynchronous output is available even in STOP mode (with no system clock active). When disabled, the Comparator0 output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, and its supply current falls to less than 100 nA. See **Section “12.1. Priority Crossbar Decoder” on page 104** for details on configuring the Comparator0 output via the digital Crossbar. Comparator0 inputs can be externally driven from -0.25 to $(V_{DD}) + 0.25$ V without damage or upset. The complete electrical specifications for Comparator0 are given in Table 7.1.

The Comparator0 response time may be configured in software via the CP0MD1-0 bits in register CPT0MD (see SFR Definition 7.3). Selecting a longer response time reduces the amount of power consumed by Comparator0. See Table 7.1 for complete timing and power consumption specifications.

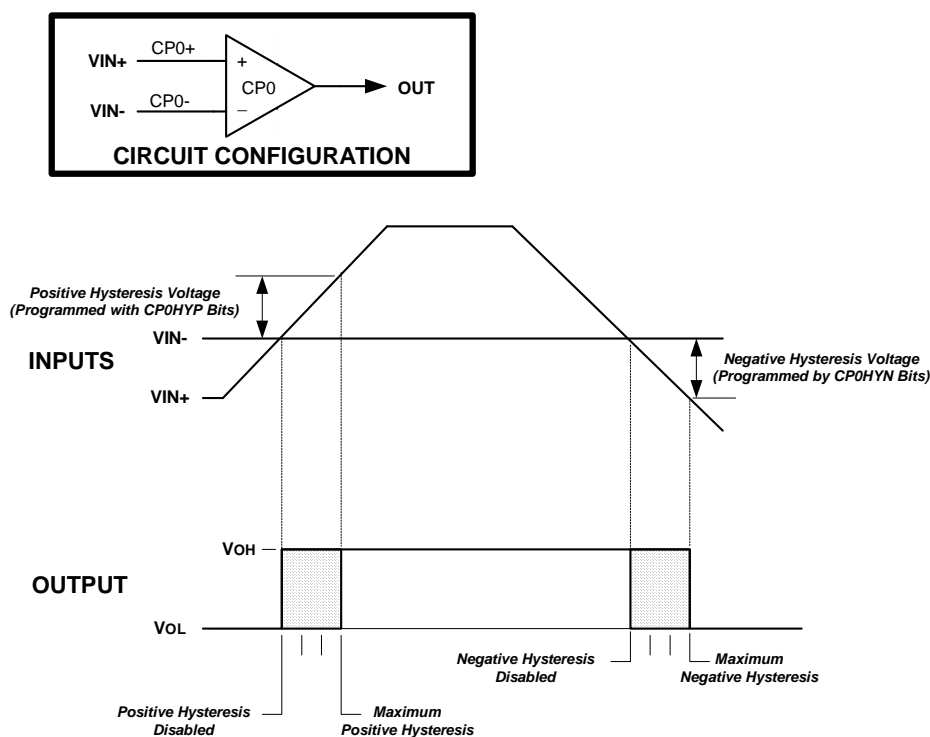


Figure 7.2. Comparator Hysteresis Plot

The hysteresis of Comparator0 is software-programmable via its Comparator0 Control register (CPT0CN). The user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage.

The Comparator0 hysteresis is programmed using Bits3–0 in the Comparator0 Control Register CPT0CN (shown in SFR Definition 7.1). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN bits. As shown in Figure 7.2, settings of 20, 10 or 5 mV of negative hysteresis can be programmed, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CP0HYP bits.

CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

8.1.2. MOVX Instruction and Program Memory

The MOVX instruction is typically used to access external data memory (Note: the C8051F300/1/2/3/4/5 does not support external data or program memory). In the CIP-51, the MOVX instruction accesses the on-chip program memory space implemented as re-programmable Flash memory. This feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to **Section “10. Flash Memory” on page 89** for further details.

Table 8.1. CIP-51 Instruction Set Summary

Mnemonic	Description	Bytes	Clock Cycles
Arithmetic Operations			
ADD A, Rn	Add register to A	1	1
ADD A, direct	Add direct byte to A	2	2
ADD A, @Ri	Add indirect RAM to A	1	2
ADD A, #data	Add immediate to A	2	2
ADDC A, Rn	Add register to A with carry	1	1
ADDC A, direct	Add direct byte to A with carry	2	2
ADDC A, @Ri	Add indirect RAM to A with carry	1	2
ADDC A, #data	Add immediate to A with carry	2	2
SUBB A, Rn	Subtract register from A with borrow	1	1
SUBB A, direct	Subtract direct byte from A with borrow	2	2
SUBB A, @Ri	Subtract indirect RAM from A with borrow	1	2
SUBB A, #data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	2
INC @Ri	Increment indirect RAM	1	2
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC direct	Decrement direct byte	2	2
DEC @Ri	Decrement indirect RAM	1	2
INC DPTR	Increment Data Pointer	1	1
MUL AB	Multiply A and B	1	4
DIV AB	Divide A by B	1	8
DA A	Decimal adjust A	1	1
Logical Operations			
ANL A, Rn	AND Register to A	1	1
ANL A, direct	AND direct byte to A	2	2
ANL A, @Ri	AND indirect RAM to A	1	2
ANL A, #data	AND immediate to A	2	2

Table 8.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
MOVC A, @A+PC	Move code byte relative PC to A	1	3
MOVX A, @Ri	Move external data (8-bit address) to A	1	3
MOVX @Ri, A	Move A to external data (8-bit address)	1	3
MOVX A, @DPTR	Move external data (16-bit address) to A	1	3
MOVX @DPTR, A	Move A to external data (16-bit address)	1	3
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange Register with A	1	1
XCH A, direct	Exchange direct byte with A	2	2
XCH A, @Ri	Exchange indirect RAM with A	1	2
XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	2
Boolean Manipulation			
CLR C	Clear Carry	1	1
CLR bit	Clear direct bit	2	2
SETB C	Set Carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement Carry	1	1
CPL bit	Complement direct bit	2	2
ANL C, bit	AND direct bit to Carry	2	2
ANL C, /bit	AND complement of direct bit to Carry	2	2
ORL C, bit	OR direct bit to carry	2	2
ORL C, /bit	OR complement of direct bit to Carry	2	2
MOV C, bit	Move direct bit to Carry	2	2
MOV bit, C	Move Carry to direct bit	2	2
JC rel	Jump if Carry is set	2	2/3
JNC rel	Jump if Carry is not set	2	2/3
JB bit, rel	Jump if direct bit is set	3	3/4
JNB bit, rel	Jump if direct bit is not set	3	3/4
JBC bit, rel	Jump if direct bit is set and clear bit	3	3/4
Program Branching			
ACALL addr11	Absolute subroutine call	2	3
LCALL addr16	Long subroutine call	3	4
RET	Return from subroutine	1	5
RETI	Return from interrupt	1	5
AJMP addr11	Absolute jump	2	3
LJMP addr16	Long jump	3	4
SJMP rel	Short jump (relative address)	2	3
JMP @A+DPTR	Jump indirect relative to DPTR	1	3
JZ rel	Jump if A equals zero	2	2/3

SFR Definition 8.2. DPH: Data Pointer High Byte

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: 0x83

Bits7–0: DPH: Data Pointer High.
The DPH register is the high byte of the 16-bit DPTR. DPTR is used to access indirectly addressed Flash memory.

SFR Definition 8.3. SP: Stack Pointer

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x81

Bits7–0: SP: Stack Pointer.
The Stack Pointer holds the location of the top of the stack. The stack pointer is incremented before every PUSH operation. The SP register defaults to 0x07 after reset.

Table 8.4. Interrupt Summary

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit addressable?	Cleared by HW?	Enable Flag	Priority Control
Reset	0x0000	Top	None	N/A	N/A	Always Enabled	Always Highest
External Interrupt 0 (/INT0)	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)	PX0 (IP.0)
Timer 0 Overflow	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)	PT0 (IP.1)
External Interrupt 1 (/INT1)	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)	PX1 (IP.2)
Timer 1 Overflow	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)	PT1 (IP.3)
UART0	0x0023	4	RI0 (SCON0.0) TI0 (SCON0.1)	Y	N	ES0 (IE.4)	PS0 (IP.4)
Timer 2 Overflow	0x002B	5	TF2H (TMR2CN.7) TF2L (TMR2CN.6)	Y	N	ET2 (IE.5)	PT2 (IP.5)
SMBus Interface	0x0033	6	SI (SMB0CN.0)	Y	N	ESMB0 (EIE1.0)	PSMB0 (EIP1.0)
ADC0 Window Compare	0x003B	7	AD0WINT (ADC0CN.3)	Y	N	EWADC0 (EIE1.1)	PWADC0 (EIP1.1)
ADC0 Conversion Complete	0x0043	8	AD0INT (ADC0CN.5)	Y	N	EADC0C (EIE1.2)	PADC0C (EIP1.2)
Programmable Counter Array	0x004B	9	CF (PCA0CN.7) CCFn (PCA0CN.n)	Y	N	EPCA0 (EIE1.3)	PPCA0 (EIP1.3)
Comparator0 Falling Edge	0x0053	10	CP0FIF (CPT0CN.4)	N	N	ECP0F (EIE1.4)	PCP0F (EIP1.4)
Comparator0 Rising Edge	0x005B	11	CP0RIF (CPT0CN.5)	N	N	ECP0R (EIE1.5)	PCP0R (EIP1.5)

bit in register RSTSRC. See Figure 9.2 for V_{DD} monitor timing; note that the reset delay is not incurred after a V_{DD} monitor reset. See Table 9.2 for electrical characteristics of the V_{DD} monitor.

Important Note: Enabling the V_{DD} monitor will immediately generate a system reset. The device will then return from the reset state with the V_{DD} monitor enabled. **Writing a logic '1' to the PORSF flag when the V_{DD} monitor is enabled does not cause a system reset.**

9.3. External Reset

The external \overline{RST} pin provides a means for external circuitry to force the device into a reset state. Asserting an active-low signal on the \overline{RST} pin generates a reset; an external pullup and/or decoupling of the \overline{RST} pin may be necessary to avoid erroneous noise-induced resets. See Table 9.2 for complete \overline{RST} pin specifications. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

9.4. Missing Clock Detector Reset

The Missing Clock Detector (MCD) is a one-shot circuit that is triggered by the system clock. If the system clock remains high or low for more than 100 μ s, 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{RST} pin is unaffected by this reset.

9.5. 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-), the device is put into the reset state. 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{RST} pin is unaffected by this reset.

9.6. 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 "16.3. Watchdog Timer Mode" on page 164**; 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{RST} pin is unaffected by this reset.

C8051F300/1/2/3/4/5

Accessing Flash **from user firmware** executing from an **unlocked page**:

1. Any unlocked page except the page containing the Lock Byte may be read, written, or erased.
2. Locked pages cannot be read, written, or erased. An erase attempt on the page containing the Lock Byte will result in a Flash Error device reset.
3. **The page containing the Lock Byte cannot be erased.** It may be read or written only if it is unlocked. An erase attempt on the page containing the Lock Byte will result in a Flash Error device reset.
4. Reading the contents of the Lock Byte is always permitted.
5. Locking additional pages (changing '1's to '0's in the Lock Byte) is not permitted.
6. Unlocking Flash pages (changing '0's to '1's in the Lock Byte) is not permitted.
7. The Reserved Area cannot be read, written, or erased. Any attempt to access the reserved area, or any other locked page, will result in a Flash Error device reset.

Accessing Flash **from user firmware** executing from a **locked page**:

1. Any unlocked page except the page containing the Lock Byte may be read, written, or erased.
2. Any locked page except the page containing the Lock Byte may be read, written, or erased. An erase attempt on the page containing the Lock Byte will result in a Flash Error device reset.
3. **The page containing the Lock Byte cannot be erased.** It may only be read or written. An erase attempt on the page containing the Lock Byte will result in a Flash Error device reset.
4. Reading the contents of the Lock Byte is always permitted.
5. Locking additional pages (changing '1's to '0's in the Lock Byte) is not permitted.
6. Unlocking Flash pages (changing '0's to '1's in the Lock Byte) is not permitted.
7. The Reserved Area cannot be read, written, or erased. Any attempt to access the reserved area, or any other locked page, will result in a Flash Error device reset.

SFR Definition 10.1. PSCTL: Program Store R/W Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	—	—	—	—	PSEE	PSWE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x8F

Bits7–2: UNUSED: Read = 000000b, Write = don't care.

Bit1: PSEE: Program Store Erase Enable
Setting this bit (in combination with PSWE) allows an entire page of Flash program memory to be erased. If this bit is logic 1 and Flash writes are enabled (PSWE is logic 1), a write to Flash memory using the MOVX instruction will erase the entire page that contains the location addressed by the MOVX instruction. The value of the data byte written does not matter.
0: Flash program memory erasure disabled.
1: Flash program memory erasure enabled.

Bit0: PSWE: Program Store Write Enable
Setting this bit allows writing a byte of data to the Flash program memory using the MOVX instruction. The Flash location should be erased before writing data.
0: Writes to Flash program memory disabled.
1: Writes to Flash program memory enabled; the MOVX instruction targets Flash memory.

11.4. External Crystal Example

If a crystal or ceramic resonator is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 11.1, Option 1. The External Oscillator Frequency Control value (XFCN) should be chosen from the Crystal column of the table in SFR Definition 11.3 (OSCXCN register). For example, an 11.0592 MHz crystal requires an XFCN setting of 111b.

When the crystal oscillator is first enabled, the oscillator amplitude detection circuit requires a settling time to achieve proper bias. Introducing a delay of 1 ms between enabling the oscillator and checking the XTLVLD bit will prevent a premature switch to the external oscillator as the system clock. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure is:

- Step 1. Force the XTAL1 and XTAL2 pins low by writing 0's to the port latch.
- Step 2. Configure XTAL1 and XTAL2 as analog inputs.
- Step 3. Enable the external oscillator.
- Step 4. Wait at least 1 ms.
- Step 5. Poll for XTLVLD => '1'.
- Step 6. Switch the system clock to the external oscillator.

Note: Tuning-fork crystals may require additional settling time before XTLVLD returns a valid result.

The capacitors shown in the external crystal configuration provide the load capacitance required by the crystal for correct oscillation. These capacitors are “in series” as seen by the crystal and “in parallel” with the stray capacitance of the XTAL1 and XTAL2 pins.

Note: The load capacitance depends upon the crystal and the manufacturer. Please refer to the crystal data sheet when completing these calculations.

For example, a tuning-fork crystal of 32.768 kHz with a recommended load capacitance of 12.5 pF should use the configuration shown in Figure 12.1, Option 1. The total value of the capacitors and the stray capacitance of the XTAL pins should equal 25 pF. With a stray capacitance of 3 pF per pin, the 22 pF capacitors yield an equivalent capacitance of 12.5 pF across the crystal, as shown in Figure 11.2.

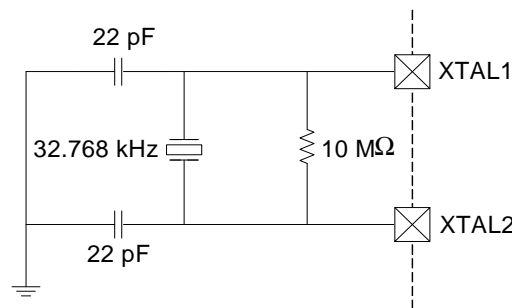


Figure 11.2. 32.768 kHz External Crystal Example

13.4. 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 “13.5. SMBus Transfer Modes” on page 123** 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 “13.4.2. SMB0CN Control Register” on page 119**; Table 13.4 provides a quick SMB0CN decoding reference.

SMBus configuration options include:

- Timeout detection (SCL Low Timeout and/or Bus Free Timeout)
- SDA setup and hold time extensions
- Slave event enable/disable
- Clock source selection

These options are selected in the SMB0CF register, as described in **Section “13.4.1. SMBus Configuration Register” on page 116**.

13.5.3. Slave Receiver Mode

Serial data is received on SDA and the clock is received on SCL. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode when a START followed by a slave address and direction bit (WRITE in this case) is received. Upon entering Slave Receiver Mode, an interrupt is generated and the ACKRQ bit is set. Software responds to the received slave address with an ACK, or ignores the received slave address with a NACK. If the received slave address is ignored, slave interrupts will be inhibited until the next START is detected. If the received slave address is acknowledged, zero or more data bytes are received. Software must write the ACK bit after each received byte to ACK or NACK the received byte. The interface exits Slave Receiver Mode after receiving a STOP. Note that the interface will switch to Slave Transmitter Mode if SMB0DAT is written while an active Slave Receiver; see **Section 13.5.4** for details on this procedure. Figure 13.7 shows a typical Slave Receiver sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the 'data byte transferred' interrupts occur **before** the ACK cycle in this mode.

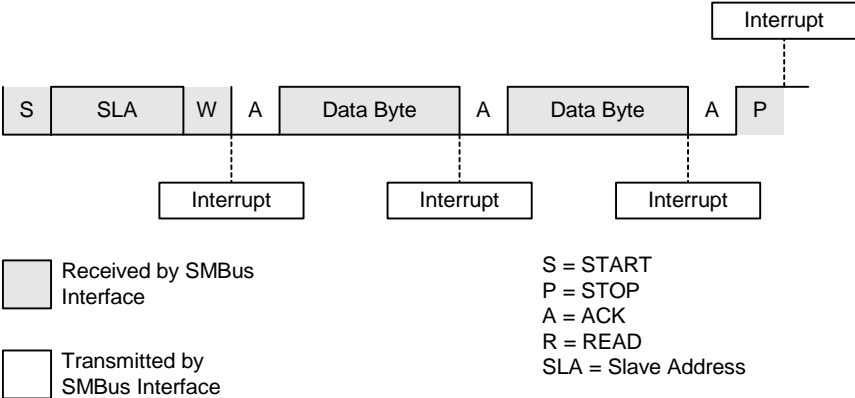


Figure 13.7. Typical Slave Receiver Sequence

Table 13.4. SMBus Status Decoding (Continued)

Mode	Values Read			Current SMBus State	Typical Response Options	Values Written						
	Status Vector	ACKRQ	ARBLOST			ACK	STA	STO	ACK			
SLAVE RECEIVER	0010	1	0	X	A slave address was received; ACK requested.	Acknowledge received address (received slave address match, R/W bit = READ).	0	0	1			
					Do not acknowledge received address.	0	0	0				
					Acknowledge received address, and switch to transmitter mode (received slave address match, R/W bit = WRITE); see Section 13.5.4 for procedure.	0	0	1				
		1	1	X	Lost arbitration as master; slave address received; ACK requested.	Acknowledge received address (received slave address match, R/W bit = READ).	0	0	1			
					Do not acknowledge received address.	0	0	0				
					Acknowledge received address, and switch to transmitter mode (received slave address match, R/W bit = WRITE); see Section 13.5.4 for procedure.	0	0	1				
	0010	0	1	X	Lost arbitration while attempting a repeated START.	Abort failed transfer.	0	0	X			
					Reschedule failed transfer.	1	0	X				
	0001	1	1	X	Lost arbitration while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0			
					0	0	X	A STOP was detected while addressed as a Slave Transmitter or Slave Receiver.	Clear STO.	0	0	X
					0	1	X	Lost arbitration due to a detected STOP.	Abort transfer.	0	0	X
	Reschedule failed transfer.	1	0	X								
	0000	1	0	X	A slave byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1			
					Do not acknowledge received byte.	0	0	0				
		1	1	X	Lost arbitration while transmitting a data byte as master.	Abort failed transfer.	0	0	0			
	Reschedule failed transfer.				1	0	0					

C8051F300/1/2/3/4/5

NOTES:

14. UART0

UART0 is an asynchronous, full duplex serial port offering modes 1 and 3 of the standard 8051 UART. Enhanced baud rate support allows a wide range of clock sources to generate standard baud rates (details in **Section “14.1. Enhanced Baud Rate Generation” on page 132**). Received data buffering allows UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte.

UART0 has two associated SFRs: Serial Control Register 0 (SCON0) and Serial Data Buffer 0 (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. Reading SBUF0 accesses the buffered Receive register; writing SBUF0 accesses the Transmit register.

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

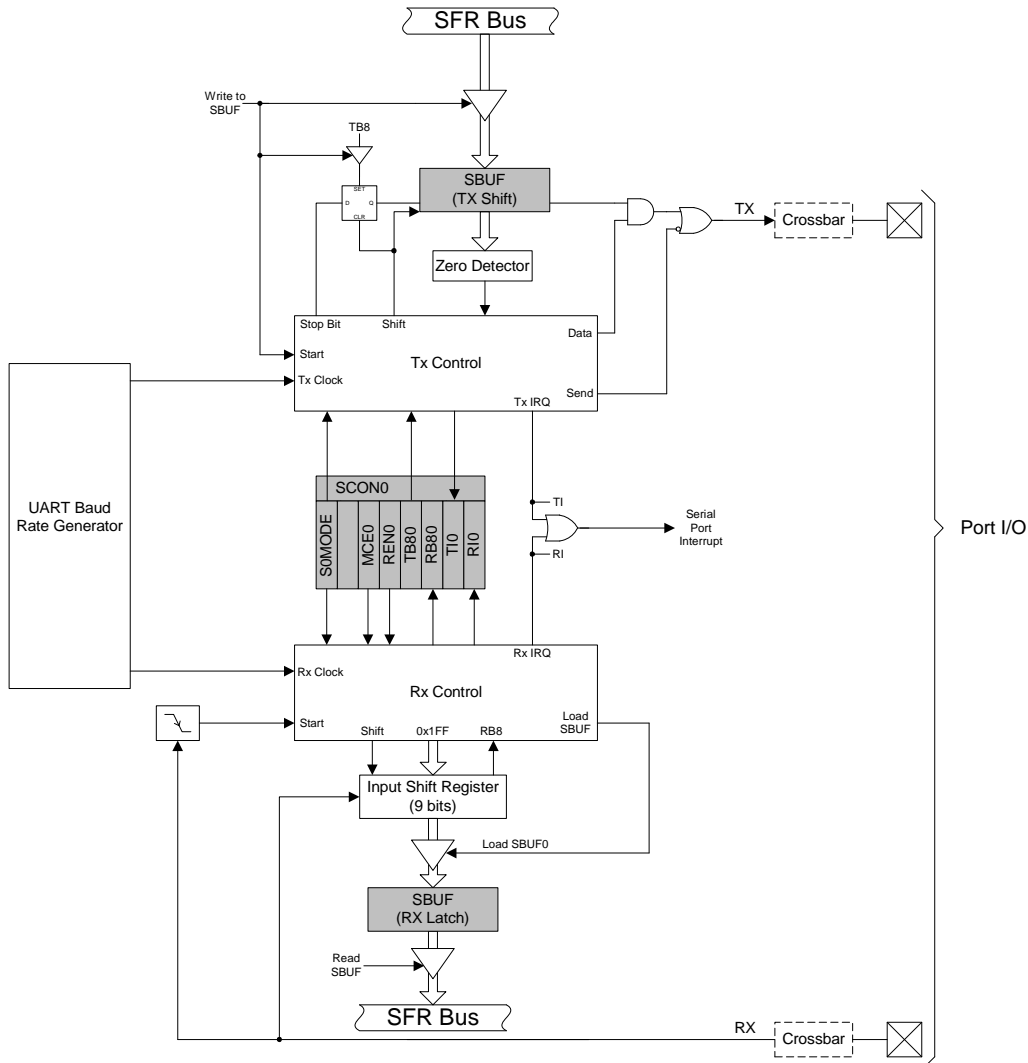


Figure 14.1. UART0 Block Diagram

C8051F300/1/2/3/4/5

14.2.2. 9-Bit UART

9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB80 (SCON0.3), which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB80 (SCON0.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The T10 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to '1'. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: (1) RI0 must be logic 0, and (2) if MCE0 is logic 1, the 9th bit must be logic 1 (when MCE0 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80, and the RI0 flag is set to '1'. If the above conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set to '1'. A UART0 interrupt will occur if enabled when either T10 or RI0 is set to '1'.

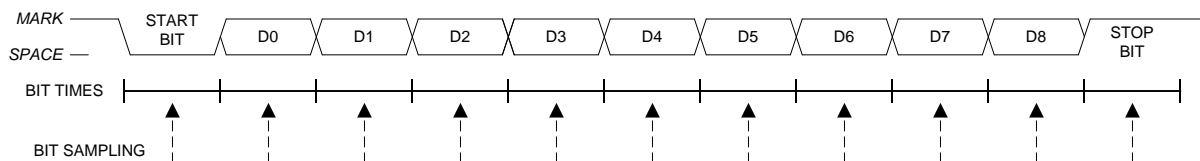


Figure 14.5. 9-Bit UART Timing Diagram

C8051F300/1/2/3/4/5

Table 14.1. Timer Settings for Standard Baud Rates Using The Internal 24.5 MHz Oscillator

Frequency: 24.5 MHz							
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select) ¹	T1M ¹	Timer 1 Reload Value (hex)
SYSCLK from Internal Osc.	230400	–0.32%	106	SYSCLK	XX ²	1	0xCB
	115200	–0.32%	212	SYSCLK	XX ²	1	0x96
	57600	0.15%	426	SYSCLK	XX ²	1	0x2B
	28800	–0.32%	848	SYSCLK / 4	01	0	0x96
	14400	0.15%	1704	SYSCLK / 12	00	0	0xB9
	9600	–0.32%	2544	SYSCLK / 12	00	0	0x96
	2400	–0.32%	10176	SYSCLK / 48	10	0	0x96
	1200	0.15%	20448	SYSCLK / 48	10	0	0x2B

Notes:

1. SCA1–SCA0 and T1M bit definitions can be found in **Section 15.1**.
2. X = Don't care.

Table 14.2. Timer Settings for Standard Baud Rates Using an External 25 MHz Oscillator

Frequency: 25.0 MHz							
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select) ¹	T1M ¹	Timer 1 Reload Value (hex)
SYSCLK from External Osc.	230400	–0.47%	108	SYSCLK	XX ²	1	0xCA
	115200	0.45%	218	SYSCLK	XX ²	1	0x93
	57600	–0.01%	434	SYSCLK	XX ²	1	0x27
	28800	0.45%	872	SYSCLK / 4	01	0	0x93
	14400	–0.01%	1736	SYSCLK / 4	01	0	0x27
	9600	0.15%	2608	EXTCLK / 8	11	0	0x5D
	2400	0.45%	10464	SYSCLK / 48	10	0	0x93
	1200	–0.01%	20832	SYSCLK / 48	10	0	0x27
SYSCLK from Internal Osc.	57600	–0.47%	432	EXTCLK / 8	11	0	0xE5
	28800	–0.47%	864	EXTCLK / 8	11	0	0xCA
	14400	0.45%	1744	EXTCLK / 8	11	0	0x93
	9600	0.15%	2608	EXTCLK / 8	11	0	0x5D

Notes:

1. SCA1–SCA0 and T1M bit definitions can be found in **Section 15.1**.
2. X = Don't care

15.1.2. 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.

15.1.3. 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 INOPL in register IT01CF (see **Section “8.3.2. External Interrupts” on page 73** for details on the external input signals /INT0 and /INT1).

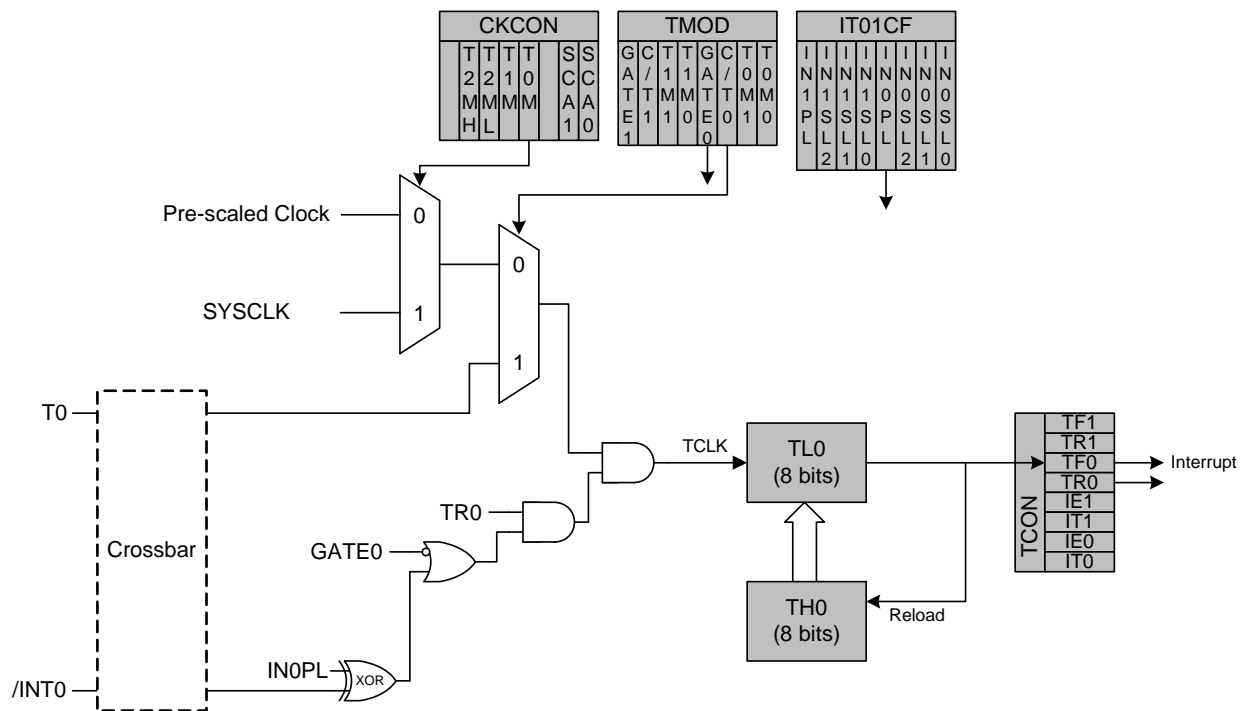
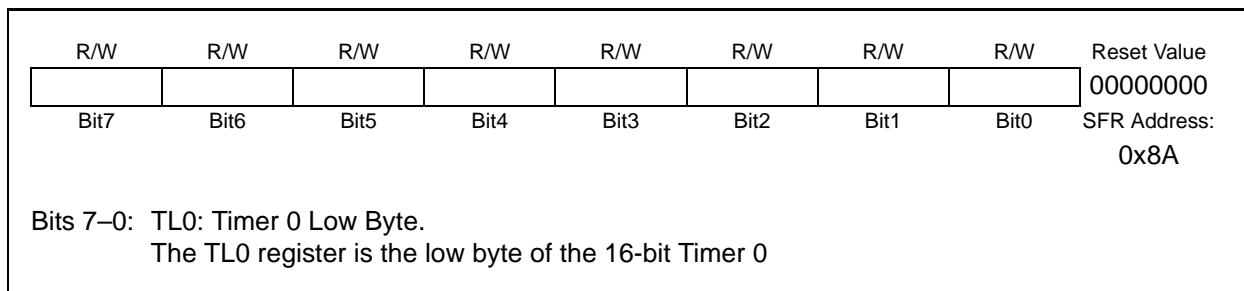


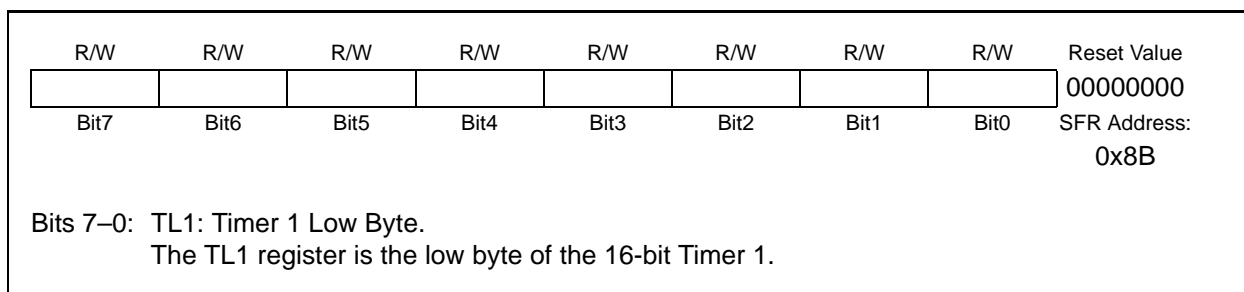
Figure 15.2. T0 Mode 2 Block Diagram

C8051F300/1/2/3/4/5

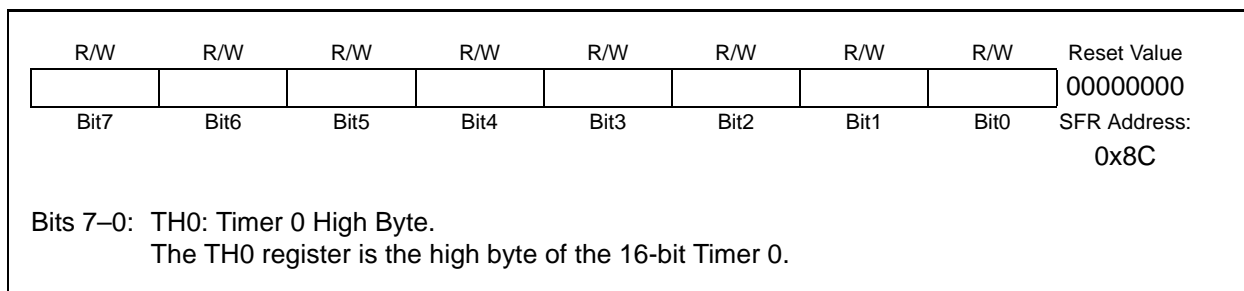
SFR Definition 15.4. TL0: Timer 0 Low Byte



SFR Definition 15.5. TL1: Timer 1 Low Byte



SFR Definition 15.6. TH0: Timer 0 High Byte



SFR Definition 15.7. TH1: Timer 1 High Byte

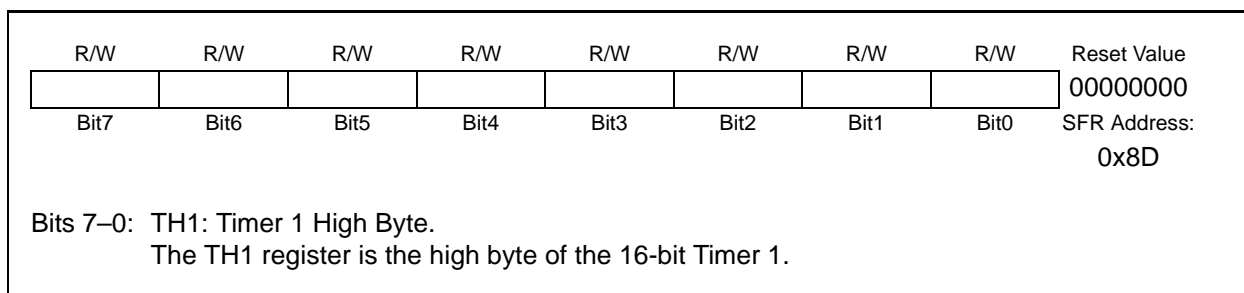


Table 16.3. Watchdog Timer Timeout Intervals¹

System Clock (Hz)	PCA0CPL2	Timeout Interval (ms)
24,500,000	255	32.1
24,500,000	128	16.2
24,500,000	32	4.1
18,432,000	255	42.7
18,432,000	128	21.5
18,432,000	32	5.5
11,059,200	255	71.1
11,059,200	128	35.8
11,059,200	32	9.2
3,062,500 ²	255	257
3,062,500 ²	128	129.5
3,062,500 ²	32	33.1
32,000	255	24576
32,000	128	12384
32,000	32	3168

Notes:

1. Assumes SYSCLK / 12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.
2. Internal oscillator reset frequency for devices with a calibrated internal oscillator. The reset system clock for devices with an uncalibrated internal oscillator will vary.