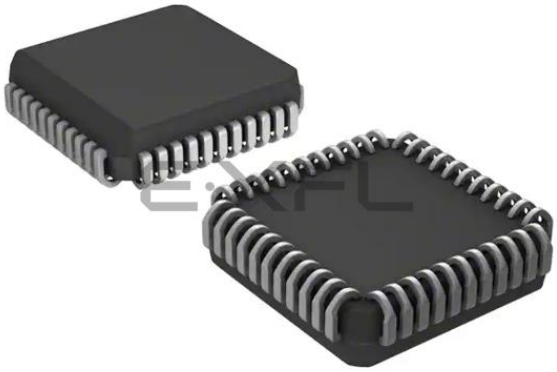


Welcome to E-XFL.COM

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	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	33MHz
Connectivity	I²C, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	32
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.25V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	44-PLCC (16.59x16.59)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/p89c668hfa-00-512

MNEMONIC	PIN NUMBER		TYPE	NAME AND FUNCTION
	PLCC	LQFP		
$\overline{EA}/V_{PP}$	35	29	I	**External Access Enable/Programming Supply Voltage:** $\overline{EA}$ must be externally held low to enable the device to fetch code from external program memory locations. If $\overline{EA}$ is held high, the device executes from internal program memory. The value on the $\overline{EA}$ pin is latched when RST is released and any subsequent changes have no effect. This pin also receives the programming supply voltage (V_{PP}) during Flash programming.
XTAL1	21	15	I	**Crystal 1:** Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	20	14	O	**Crystal 2:** Output from the inverting oscillator amplifier.

NOTE:
To avoid “latch-up” effect at power-on, the voltage on any pin (other than V_{PP}) must not be higher than V_{CC} + 0.5 V or less than V_{SS} – 0.5 V.

More Information on SIO1 Operating Modes

The four operating modes are:
– Master Transmitter
– Master Receiver
– Slave Receiver
– Slave Transmitter

Data transfers in each mode of operation are shown in Figures 8-11. These figures contain the following abbreviations:

Abbreviation	Explanation
S	Start condition
SLA	7-bit slave address
R	Read bit (high level at SDA)
W	Write bit (low level at SDA)
A	Acknowledge bit (low level at SDA)
$\overline{A}$	Not acknowledge bit (high level at SDA)
Data	8-bit data byte
P	Stop condition

In Figures 8-11, circles are used to indicate when the serial interrupt flag is set. The numbers in the circles show the status code held in the S1STA register. At these points, a service routine must be executed to continue or complete the serial transfer. These service routines are not critical since the serial transfer is suspended until the serial interrupt flag is cleared by software.

When a serial interrupt routine is entered, the status code in S1STA is used to branch to the appropriate service routine. For each status code, the required software action and details of the following serial transfer are given in Tables 4-8.

Master Transmitter mode
In the Master Transmitter mode, a number of data bytes are transmitted to a slave receiver (see Figure 8). Before the Master Transmitter mode can be entered, S1CON must be initialized as follows:

	7	6	5	4	3	2	1	0
S1CON (D8H)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
	bit rate	1	0	0	0	X	bit rate	

CR0, CR1, and CR2 define the serial bit rate. ENS1 must be set to logic 1 to enable SIO1. If the AA bit is reset, SIO1 will not acknowledge its own slave address or the general call address in the event of another device becoming master of the bus. In other words, if AA is reset, SIO0 cannot enter a Slave mode. STA, STO, and SI must be reset.

The Master Transmitter mode may now be entered by setting the STA bit using the SETB instruction. The SIO1 logic will now test the I^2C bus and generate a start condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SI) is set, and the status code in the status register (S1STA) will be 08H. This status code must be used to vector to an interrupt service routine that loads S1DAT with the slave address and the data direction bit (SLA+W). The SI bit in S1CON must then be reset before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in S1STA are possible. There are 18H, 20H, or 38H for the Master mode and also 68H, 78H, or B0H if the Slave mode was enabled (AA = logic 1). The appropriate action to be taken for each of these status codes is detailed in Table 4. After a repeated start condition (state 10H). SIO1 may switch to the Master Receiver mode by loading S1DAT with SLA+R).

Master Receiver mode
In the Master Receiver mode, a number of data bytes are received from a slave transmitter (see Figure 9). The transfer is initialized as in the Master Transmitter mode. When the start condition has been transmitted, the interrupt service routine must load S1DAT with the 7-bit slave address and the data direction bit (SLA+R). The SI bit in S1CON must then be cleared before the serial transfer can continue.

When the slave address and the data direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in S1STA are possible. These are 40H, 48H, or 38H for the Master mode and also 68H, 78H, or B0H if the Slave mode was enabled (AA = logic 1). The appropriate action to be taken for each of these status codes is detailed in Table 5. ENS1, CR1, and CR0 are not affected by the serial transfer and are not referred to in Table 5. After a repeated start condition (state 10H), SIO1 may switch to the Master Transmitter mode by loading S1DAT with SLA+W.

Slave Receiver mode
In the Slave Receiver mode, a number of data bytes are received from a master transmitter (see Figure 10). To initiate the Slave Receiver mode, S1ADR and S1CON must be loaded as follows:

	7	6	5	4	3	2	1	0
S1ADR (DBH)	X	X	X	X	X	X	X	GC
	own slave address							

The upper 7 bits are the address to which SIO1 will respond when addressed by a master. If the LSB (GC) is set, SIO1 will respond to the general call address (00H); otherwise it ignores the general call address.

	7	6	5	4	3	2	1	0
S1CON (D8H)	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
	X	1	0	0	0	1	X	X

CR0, CR1, and CR2 do not affect SIO1 in the Slave mode. ENS1 must be set to logic 1 to enable SIO1. The AA bit must be set to enable SIO1 to acknowledge its own slave address or the general call address. STA, STO, and SI must be reset.

When S1ADR and S1CON have been initialized, SIO1 waits until it is addressed by its own slave address followed by the data direction bit which must be “0” (W) for SIO1 to operate in the Slave Receiver mode. After its own slave address and the W bit have been received, the serial interrupt flag (I) is set and a valid status code can be read from S1STA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status codes is detailed in Table 6. The Slave Receiver mode may also be entered if arbitration is lost while SIO1 is in the Master mode (see status 68H and 78H).

If the AA bit is reset during a transfer, SIO1 will return a not acknowledge (logic 1) to SDA after the next received data byte. While AA is reset, SIO1 does not respond to its own slave address or a general call address. However, the I^2C bus is still monitored and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate SIO1 from the I^2C bus.

Table 4. Master Transmitter mode

STATUS CODE (S1STA)	STATUS OF THE I^2C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
08H	A START condition has been transmitted	Load SLA+W	X	0	0	X	SLA+W will be transmitted; ACK bit will be received
10H	A repeated START condition has been transmitted	Load SLA+W or	X	0	0	X	As above
		Load SLA+R	X	0	0	X	SLA+W will be transmitted; SIO1 will be switched to MST/REC mode
18H	SLA+W has been transmitted; ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no S1DAT action or	1	0	0	X	Repeated START will be transmitted;
		no S1DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no S1DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
20H	SLA+W has been transmitted; NOT ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no S1DAT action or	1	0	0	X	Repeated START will be transmitted;
		no S1DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no S1DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
28H	Data byte in S1DAT has been transmitted; ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no S1DAT action or	1	0	0	X	Repeated START will be transmitted;
		no S1DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no S1DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
30H	Data byte in S1DAT has been transmitted; NOT ACK has been received	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received
		no S1DAT action or	1	0	0	X	Repeated START will be transmitted;
		no S1DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		no S1DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
38H	Arbitration lost in SLA+R/$\overline{W}$ or Data bytes	No S1DAT action or	0	0	0	X	I^2C bus will be released; not addressed slave will be entered
		No S1DAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free

Table 8. Miscellaneous States

STATUS CODE (S1STA)	STATUS OF THE I^2C BUS AND SIO1 HARDWARE	APPLICATION SOFTWARE RESPONSE					NEXT ACTION TAKEN BY SIO1 HARDWARE
		TO/FROM S1DAT	TO S1CON				
			STA	STO	SI	AA	
F8H	No relevant state information available; SI = 0	No S1DAT action	No S1CON action				Wait or proceed current transfer
00H	Bus error during MST or selected Slave modes, due to an illegal START or STOP condition. State 00H can also occur when interference causes SIO1 to enter an undefined state.	No S1DAT action	0	1	0	X	Only the internal hardware is affected in the MST or addressed SLV modes. In all cases, the bus is released and SIO1 is switched to the not addressed SLV mode. STO is reset.

Slave Transmitter mode
In the Slave Transmitter mode, a number of data bytes are transmitted to a master receiver (see Figure 11). Data transfer is initialized as in the Slave Receiver mode. When S1ADR and S1CON have been initialized, SIO1 waits until it is addressed by its own slave address followed by the data direction bit which must be "1" (R) for SIO1 to operate in the Slave Transmitter mode. After its own slave address and the R bit have been received, the serial interrupt flag (SI) is set and a valid status code can be read from S1STA. This status code is used to vector to an interrupt service routine, and the appropriate action to be taken for each of these status codes is detailed in Table 7. The Slave Transmitter mode may also be entered if arbitration is lost while SIO1 is in the Master mode (see state B0H).

If the AA bit is reset during a transfer, SIO1 will transmit the last byte of the transfer and enter state C0H or C8H. SIO1 is switched to the "not addressed" Slave mode and will ignore the master receiver if it continues the transfer. Thus the master receiver receives all 1s as serial data. While AA is reset, SIO1 does not respond to its own slave address or a general call address. However, the I^2C bus is still monitored, and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate SIO1 from the I^2C bus.

Miscellaneous States

There are two S1STA codes that do not correspond to a defined SIO1 hardware state (see Table 8). These are discussed below.

S1STA = F8H
This status code indicates that no relevant information is available because the serial interrupt flag, SI, is not yet set. This occurs between other states and when SIO1 is not involved in a serial transfer.

S1STA = 00H
This status code indicates that a bus error has occurred during an SIO1 serial transfer. A bus error is caused when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. A bus error may also be caused when external interference disturbs the internal SIO1 signals. When a bus error occurs, SI is set. To recover from a bus error, the STO flag must be set and SI must be cleared. This causes SIO1 to enter the "not addressed" Slave mode (a defined state) and to clear the STO flag (no other bits in S1CON are affected). The SDA and SCL lines are released (a STOP condition is not transmitted).

Some Special Cases

The SIO1 hardware has facilities to handle the following special cases that may occur during a serial transfer.

Simultaneous Repeated START Conditions from Two Masters
A repeated START condition may be generated in the Master Transmitter or Master Receiver modes. A special case occurs if another master simultaneously generates a repeated START condition (see Figure 12). Until this occurs, arbitration is not lost by either master since they were both transmitting the same data.

If the SIO1 hardware detects a repeated START condition on the I^2C bus before generating a repeated START condition itself, it will release the bus, and no interrupt request is generated. If another master frees the bus by generating a STOP condition, SIO1 will transmit a normal START condition (state 08H), and a retry of the total serial data transfer can commence.

Data Transfer After Loss of Arbitration
Arbitration may be lost in the Master Transmitter and Master Receiver modes (see Figure 4). Loss of arbitration is indicated by the following states in S1STA: 38H, 68H, 78H, and B0H (see Figures 8 and 9).

If the STA flag in S1CON is set by the routines which service these states, then, if the bus is free again, a START condition (state 08H) is transmitted without intervention by the CPU, and a retry of the total serial transfer can commence.

Forced Access to the I^2C Bus
In some applications, it may be possible for an uncontrolled source to cause a bus hang-up. In such situations, the problem may be caused by interference, temporary interruption of the bus or a temporary short-circuit between SDA and SCL.

If an uncontrolled source generates a superfluous START or masks a STOP condition, then the I^2C bus stays busy indefinitely. If the STA flag is set and bus access is not obtained within a reasonable amount of time, then a forced access to the I^2C bus is possible. This is achieved by setting the STO flag while the STA flag is still set. No STOP condition is transmitted. The SIO1 hardware behaves as if a STOP condition was received and is able to transmit a START condition. The ST0 flag is cleared by hardware (see Figure 13).

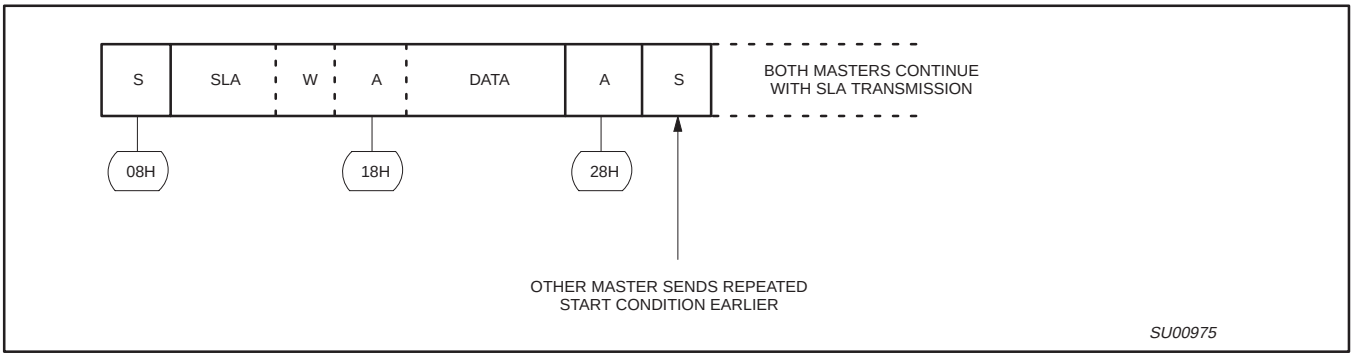


Figure 12. Simultaneous Repeated START Conditions from 2 Masters

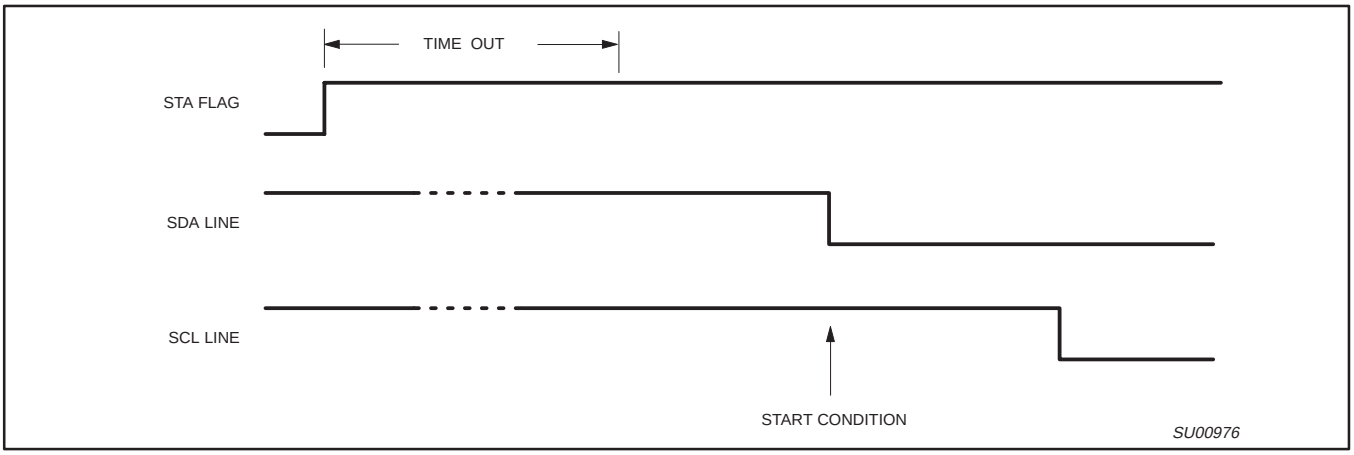


Figure 13. Forced Access to a Busy I²C Bus

I²C Bus Obstructed by a Low Level on SCL or SDA

An I²C bus hang-up occurs if SDA or SCL is pulled LOW by an uncontrolled source. If the SCL line is obstructed (pulled LOW) by a device on the bus, no further serial transfer is possible, and the SIO1 hardware cannot resolve this type of problem. When this occurs, the problem must be resolved by the device that is pulling the SCL bus line LOW.

If the SDA line is obstructed by another device on the bus (e.g., a slave device out of bit synchronization), the problem can be solved by transmitting additional clock pulses on the SCL line (see Figure 14). The SIO1 hardware transmits additional clock pulses when the STA flag is set, but no START condition can be generated because the SDA line is pulled LOW while the I²C bus is considered free. The SIO1 hardware attempts to generate a START condition after every two additional clock pulses on the SCL line. When the SDA line is eventually released, a normal START condition is transmitted, state 08H is entered, and the serial transfer continues.

If a forced bus access occurs or a repeated START condition is transmitted while SDA is obstructed (pulled LOW), the SIO1 hardware performs the same action as described above. In each case, state 08H is entered after a successful START condition is transmitted and normal serial transfer continues. Note that the CPU is not involved in solving these bus hang-up problems.

Bus Error

A bus error occurs when a START or STOP condition is present at an illegal position in the format frame. Examples of illegal positions are during the serial transfer of an address byte, a data, or an acknowledge bit.

The SIO1 hardware only reacts to a bus error when it is involved in a serial transfer either as a master or an addressed slave. When a bus error is detected, SIO1 immediately switches to the “not addressed” Slave mode, releases the SDA and SCL lines, sets the interrupt flag, and loads the status register with 00H. This status code may be used to vector to a service routine which either attempts the aborted serial transfer again or simply recovers from the error condition as shown in Table 8.

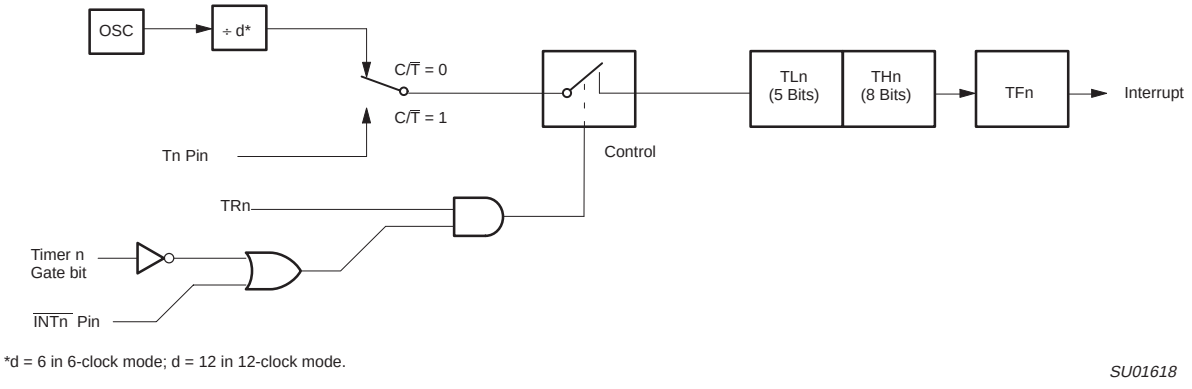


Figure 16. Timer/Counter 0/1 Mode 0: 13-Bit Timer/Counter

TCON Address = 88H Reset Value = 00H

Bit Addressable

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

BIT	SYMBOL	FUNCTION
TCON.7	TF1	Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine, or clearing the bit in software.
TCON.6	TR1	Timer 1 Run control bit. Set/cleared by software to turn Timer/Counter on/off.
TCON.5	TF0	Timer 0 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine, or by clearing the bit in software.
TCON.4	TR0	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter on/off.
TCON.3	IE1	Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TCON.2	IT1	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
TCON.1	IE0	Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TCON.0	IT0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

SU01516

Figure 17. Timer/Counter 0/1 Control (TCON) Register

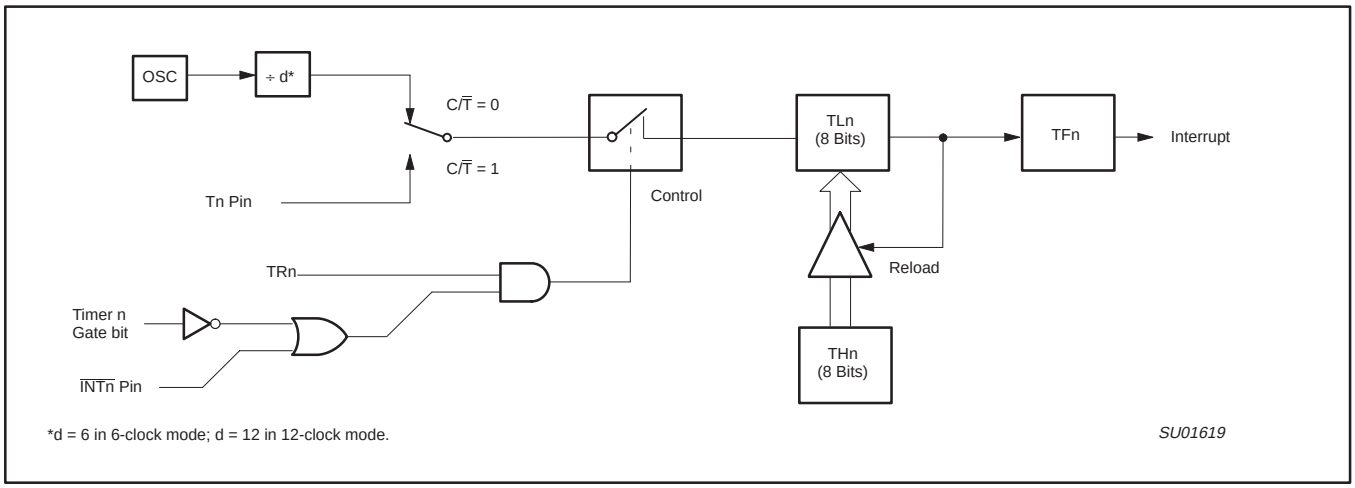


Figure 18. Timer/Counter 0/1 Mode 2: 8-Bit Auto-Reload

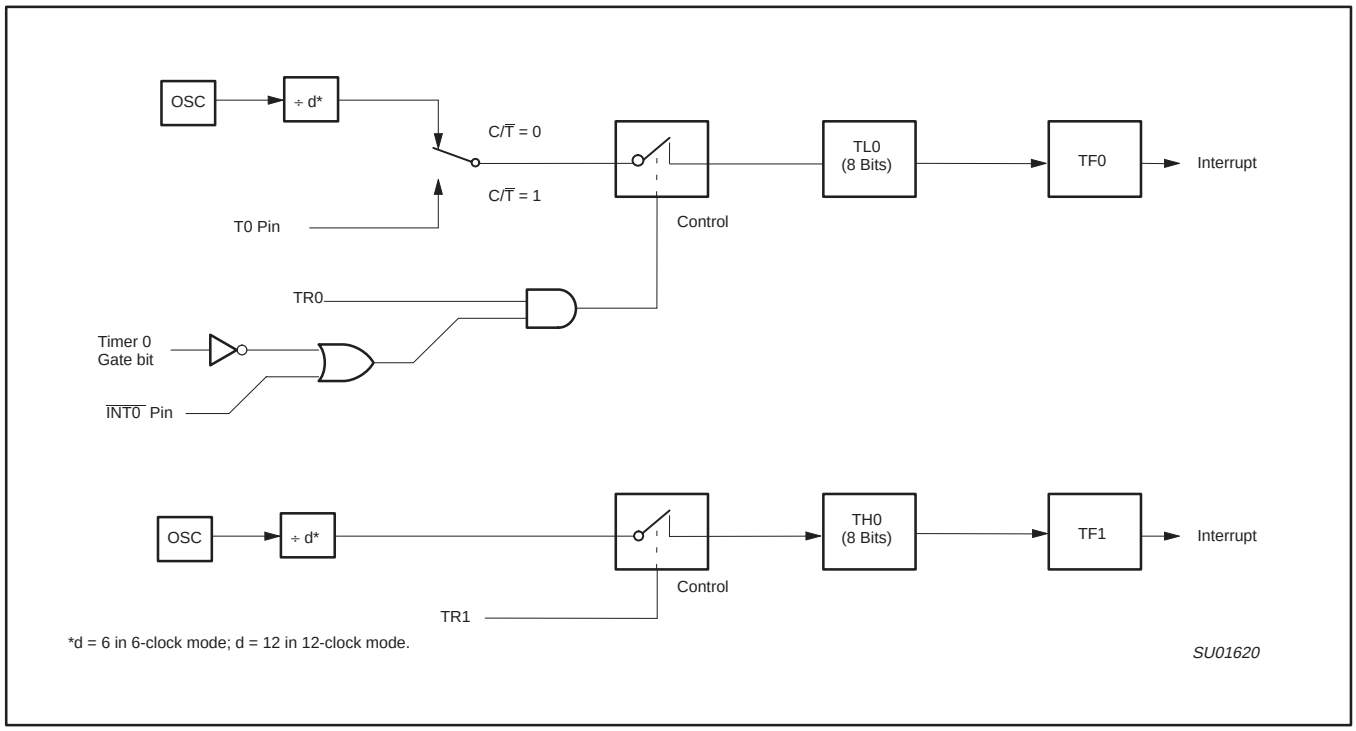
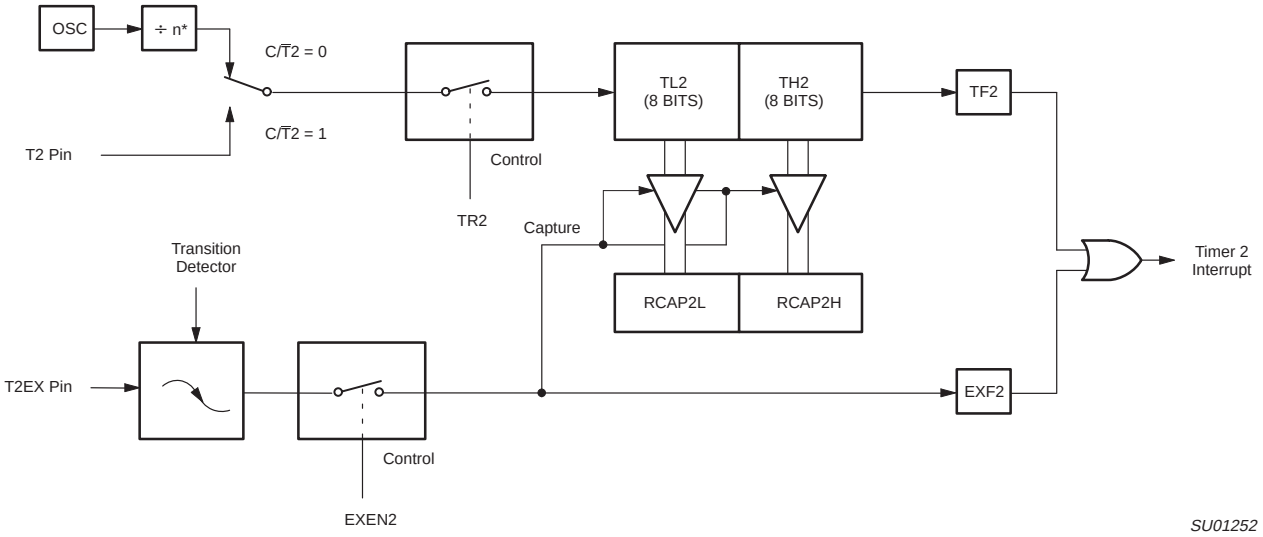


Figure 19. Timer/Counter 0 Mode 3: Two 8-Bit Counters

Table 9. Timer 2 Operating Modes

RCLK + TCLK	CP/$\overline{RL2}$	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud rate generator
X	X	0	(off)



* n = 6 in 6-clock mode, or 12 in 12-clock mode.

Figure 21. Timer 2 in Capture Mode

T2MOD Address = 0C9H Reset Value = XXXX XX00B

Not Bit Addressable

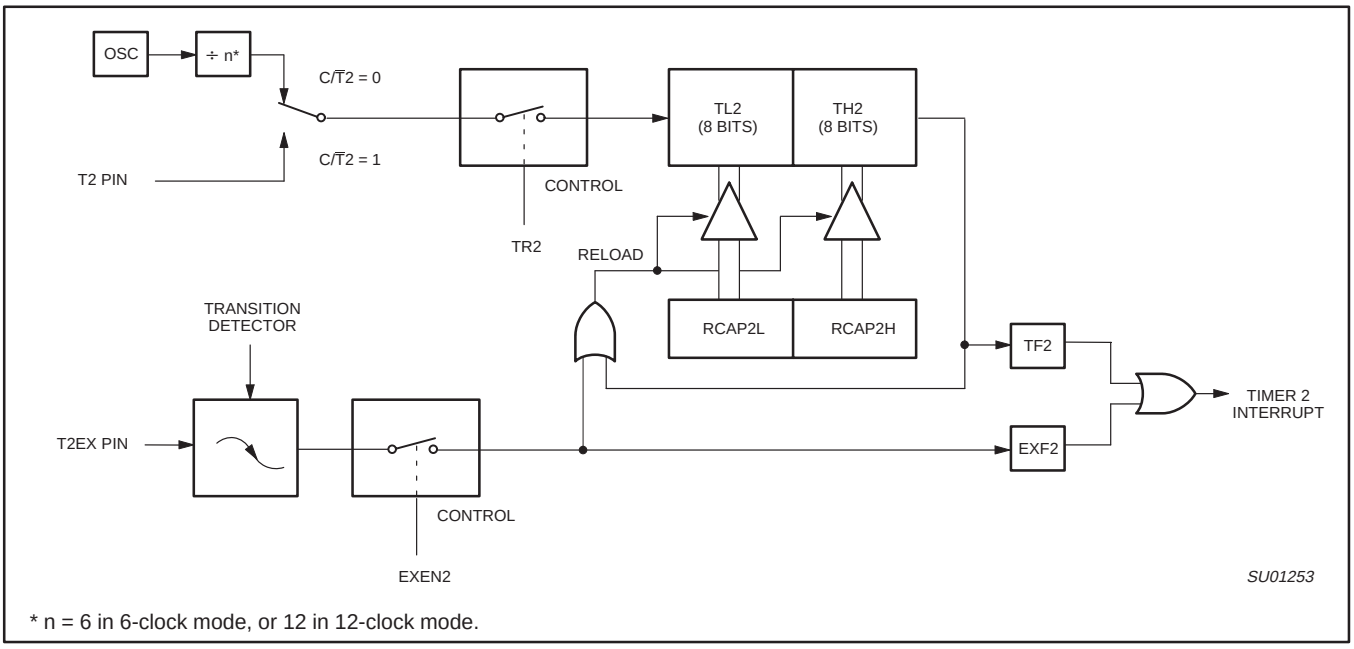
—	—	—	—	—	—	T2OE	DCEN
Bit 7	6	5	4	3	2	1	0

Symbol	Function
—	Not implemented, reserved for future use.*
T2OE	Timer 2 Output Enable bit.
DCEN	Down Count Enable bit. When set, this allows Timer 2 to be configured as an up/down counter depending on the value of the T2EX pin.

* User software should not write 1s to reserved bits. These bits may be used in future 8051 family products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1. The value read from a reserved bit is indeterminate.

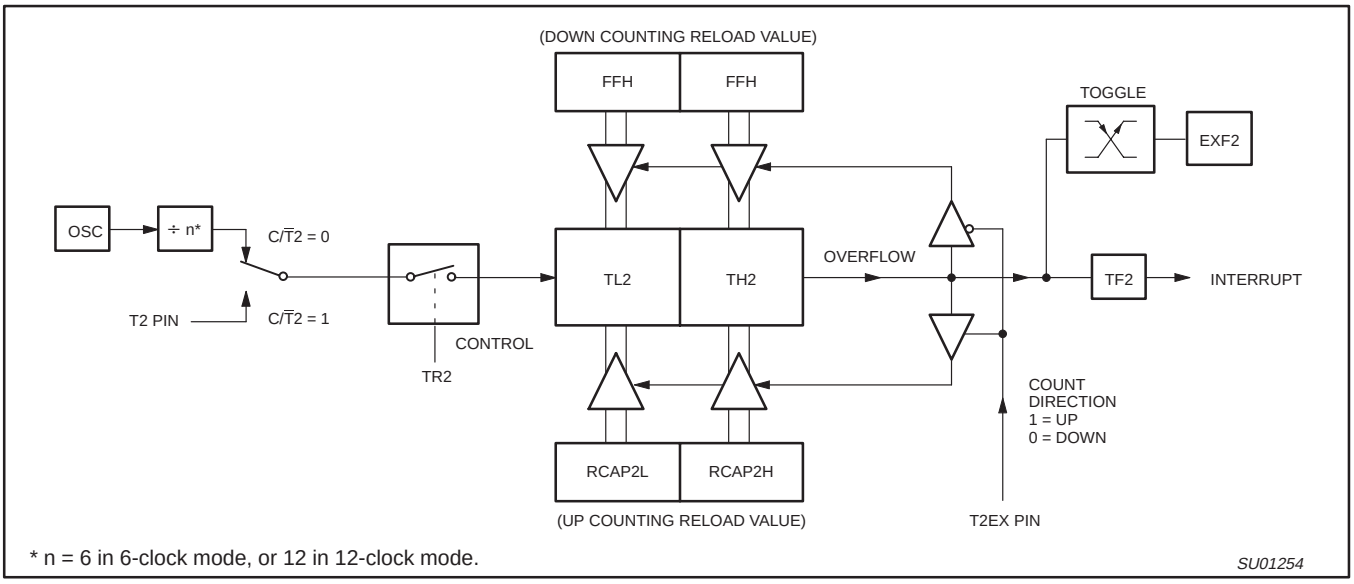
SU01714

Figure 22. Timer 2 Mode (T2MOD) Control Register



* n = 6 in 6-clock mode, or 12 in 12-clock mode.

Figure 23. Timer 2 in Auto-Reload Mode (DCEN = 0)



* n = 6 in 6-clock mode, or 12 in 12-clock mode.

Figure 24. Timer 2 Auto Reload Mode (DCEN = 1)

When Timer 2 is in the baud rate generator mode, one should not try to read or write TH2 and TL2. As a baud rate generator, Timer 2 is incremented every state time ($f_{OSC}/2$) or asynchronously from pin T2; under these conditions, a read or write of TH2 or TL2 may not be accurate. The RCAP2 registers may be read, but should not be written to, because a write might overlap a reload, and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing Timer 2 or RCAP2 registers.

Summary Of Baud Rate Equations: Timer 2 is in baud rate generating mode. If Timer 2 is being clocked through pin T2(P1.0) the baud rate is:

$$\text{Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

If Timer 2 is being clocked internally, the baud rate is:

$$\text{Baud Rate} = \frac{f_{OSC}}{[\,n^* \times [65536 - (\text{RCAP2H, RCAP2L})]]}$$

* n = 16 in 6 clock mode
32 in 12 clock mode

Where f_{OSC} = Oscillator Frequency

To obtain the reload value for RCAP2H and RCAP2L, the above equation can be rewritten as:

$$\text{RCAP2H, RCAP2L} = 65536 - \left(\frac{f_{OSC}}{n^* \times \text{Baud Rate}}\right)$$

Timer/Counter 2 Set-up
Except for the baud rate generator mode, the values given for T2CON do not include the setting of the TR2 bit. Therefore, bit TR2 must be set, separately, to turn the timer on. See Table 11 for set-up of Timer 2 as a timer. Also see Table 12 for set-up of Timer 2 as a counter.

Table 11. Timer 2 as a Timer

MODE	T2CON	
	INTERNAL CONTROL (Note 1)	EXTERNAL CONTROL (Note 2)
16-bit Auto-Reload	00H	08H
16-bit Capture	01H	09H
Baud rate generator receive and transmit same baud rate	34H	36H
Receive only	24H	26H
Transmit only	14H	16H

Table 12. Timer 2 as a Counter

MODE	TMOD	
	INTERNAL CONTROL (Note 1)	EXTERNAL CONTROL (Note 2)
16-bit	02H	0AH
Auto-Reload	03H	0BH

NOTES:
1. Capture/reload occurs only on timer/counter overflow.
2. Capture/reload occurs on timer/counter overflow and a 1-to-0 transition on T2EX (P1.1) pin except when Timer 2 is used in the baud rate generator mode.

shifted to the left by one position. The value that comes in, from the right, is the value that was sampled at the P3.0 pin at S5P2 of the same machine cycle.

As data bits come in from the right, 1s shift out to the left. When the 0 that was initially loaded into the rightmost position arrives at the leftmost position in the shift register, it flags the RX Control block to do one last shift and load SBUF. At S1P1 of the 10th machine cycle, after the write to SCON that cleared RI, RECEIVE is cleared as RI is set.

More About Mode 1

Ten bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in SCON. In the 80C51 the baud rate is determined by the Timer 1 or Timer 2 overflow rate.

Figure 29 shows a simplified functional diagram of the serial port in Mode 1, and associated timings for transmit receive.

Transmission is initiated by any instruction that uses SBUF as a destination register. The “write to SBUF” signal also loads a 1 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. (Thus, the bit times are synchronized to the divide-by-16 counter, not to the “write to SBUF” signal.)

The transmission begins with activation of SEND which puts the start bit at TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, zeros are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, the 1 that was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 10th divide-by-16 rollover after “write to SBUF.”

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its rollovers with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16ths. At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of RxD. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. This is to provide rejection of false start bits. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in mode 1 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

1. R1 = 0, and
2. Either SM2 = 0, or the received stop bit = 1.

If either of these two conditions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated. At this time, whether the above conditions are met or not, the unit goes back to looking for a 1-to-0 transition in RxD.

More About Modes 2 and 3

Eleven bits are transmitted (through TxD), or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8) can be assigned the value of 0 or 1. On receive, the 9th data bit goes into RB8 in SCON. The baud rate is programmable to either 1/32 or 1/64 (12-clock mode), or 1/16 or 1/32 (6-clock mode) of the oscillator frequency in Mode 2. Mode 3 may have a variable baud rate generated from Timer 1 or Timer 2.

Figures 30 and 31 show a functional diagram of the serial port in Modes 2 and 3. The receive portion is exactly the same as in Mode 1. The transmit portion differs from Mode 1 only in the 9th bit of the transmit shift register.

Transmission is initiated by any instruction that uses SBUF as a destination register. The “write to SBUF” signal also loads TB8 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter (thus, the bit times are synchronized to the divide-by-16 counter, not to the “write to SBUF” signal).

The transmission begins with activation of SEND, which puts the start bit at TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TxD. The first shift pulse occurs one bit time after that. The first shift clocks a 1 (the stop bit) into the 9th bit position of the shift register. Thereafter, only zeros are clocked in. Thus, as data bits shift out to the right, zeros are clocked in from the left. When TB8 is at the output position of the shift register, then the stop bit is just to the left of TB8, and all positions to the left of that contain zeros. This condition flags the TX Control unit to do one last shift and then deactivate SEND and set TI. This occurs at the 11th divide-by-16 rollover after “write to SBUF.”

Reception is initiated by a detected 1-to-0 transition at RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written to the input shift register.

At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of RxD. The value accepted is the value that was seen in at least 2 of the 3 samples. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit goes back to looking for another 1-to-0 transition. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in Modes 2 and 3 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI.

The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

1. RI = 0, and
2. Either SM2 = 0, or the received 9th data bit = 1.

If either of these conditions is not met, the received frame is irretrievably lost, and RI is not set. If both conditions are met, the received 9th data bit goes into RB8, and the first 8 data bits go into SBUF. One bit time later, whether the above conditions were met or not, the unit goes back to looking for a 1-to-0 transition at the RxD input.

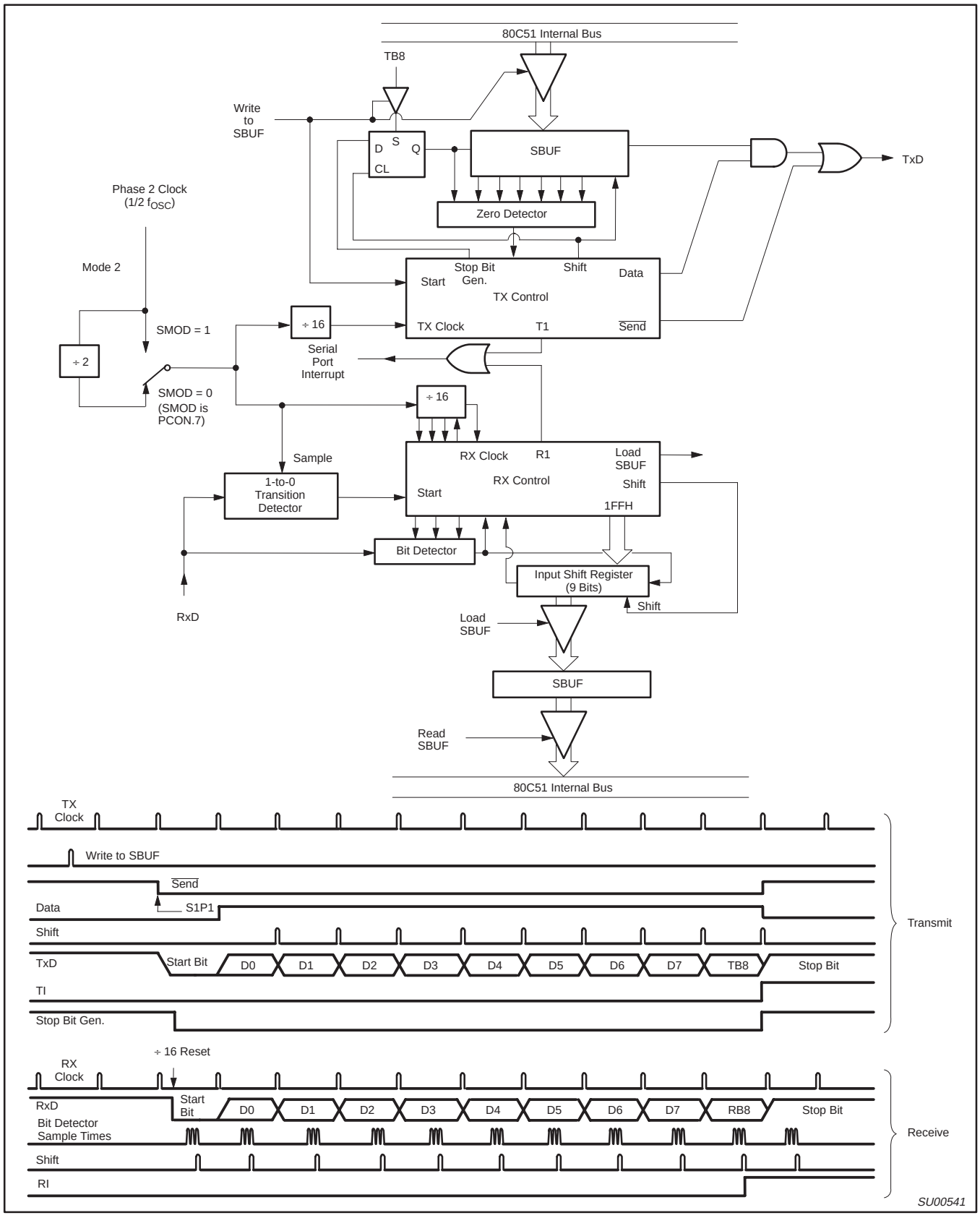


Figure 30. Serial Port Mode 2

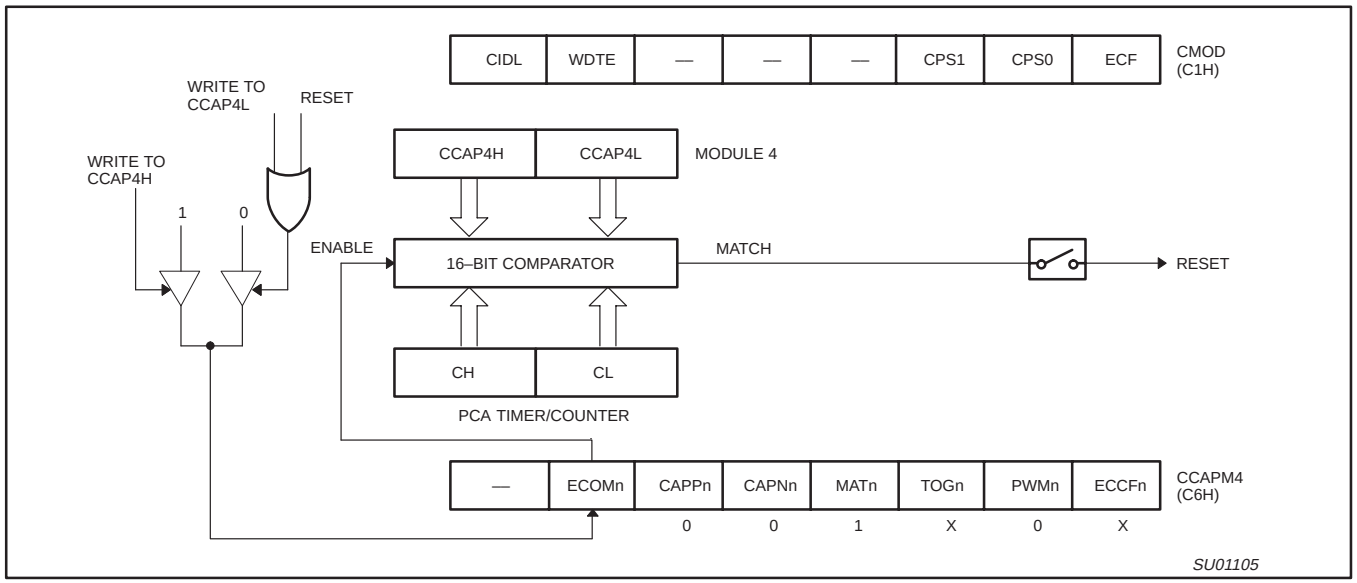


Figure 51. PCA Watchdog Timer m(Module 4 only)

PCA Watchdog Timer

An on-board watchdog timer is available with the PCA to improve the reliability of the system without increasing chip count. Watchdog timers are useful for systems that are susceptible to noise, power glitches, or electrostatic discharge. Module 4 is the only PCA module that can be programmed as a watchdog. However, this module can still be used for other modes if the watchdog is not needed.

Figure 51 shows a diagram of how the watchdog works. The user pre-loads a 16-bit value in the compare registers. Just like the other compare modes, this 16-bit value is compared to the PCA timer value. If a match is allowed to occur, an internal reset will be generated. This will not cause the RST pin to be driven high.

In order to hold off the reset, the user has three options:

1. periodically change the compare value so it will never match the PCA timer,
2. periodically change the PCA timer value so it will never match the compare values, or
3. disable the watchdog by clearing the WDTE bit before a match occurs and then re-enable it.

The first two options are more reliable because the watchdog timer is never disabled as in option #3. If the program counter ever goes astray, a match will eventually occur and cause an internal reset. The second option is also not recommended if other PCA modules are being used. Remember, the PCA timer is the time base for **all** modules; changing the time base for other modules would not be a good idea. Thus, in most applications the first solution is the best option.

Figure 52 shows the code for initializing the watchdog timer. Module 4 can be configured in either compare mode, and the WDTE bit in CMOD must also be set. The user's software must periodically change (CCAP4H,CCAP4L) to keep a match from occurring with the PCA timer (CH,CL). This code is given in the WATCHDOG routine in Figure 52.

This routine should not be part of an interrupt service routine, because if the program counter goes astray and gets stuck in an infinite loop, interrupts will still be serviced and the watchdog will keep getting reset. Thus, the purpose of the watchdog would be defeated. Instead, call this subroutine from the main program within 2^{16} count of the PCA timer.

Expanded Data RAM Addressing

The P89C660/662/664/668 has internal data memory that is mapped into four separate segments: the lower 128 bytes of RAM, upper 128 bytes of RAM, 128 bytes Special Function Register (SFR), and 256 bytes expanded RAM (ERAM) (256 bytes for the '660; 768 bytes for the '662; 1792 bytes for the '664; 7936 bytes for the '668).

The four segments are:

1. The Lower 128 bytes of RAM (addresses 00H to 7FH) are directly and indirectly addressable.
2. The Upper 128 bytes of RAM (addresses 80H to FFH) are indirectly addressable only.
3. The Special Function Registers, SFRs, (addresses 80H to FFH) are directly addressable only.
4. The 256/768/1792/7936-bytes expanded RAM (ERAM, 00H – XFFH/2FFH/6FFH/1FFFH) are indirectly accessed by move external instruction, MOVX, and with the EXTRAM bit cleared, see Figure 53.

The Lower 128 bytes can be accessed by either direct or indirect addressing. The Upper 128 bytes can be accessed by indirect addressing only. The Upper 128 bytes occupy the same address space as the SFR. That means they have the same address, but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the CPU knows whether the access is to the upper 128 bytes of data RAM, or to SFR space by the addressing mode used in the instruction. Instructions that use direct addressing, access SFR space. For example:

MOV 0A0H,A

accesses the SFR at location 0A0H (which is P2). Instructions that use indirect addressing, access the Upper 128 bytes of data RAM. For example:

MOV @R0,A

where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

The ERAM can be accessed by indirect addressing, with EXTRAM bit cleared and MOVX instructions. This part of memory is physically located on-chip, logically occupies the first 256 bytes (660), 768 (662), 1792 (664), 7936 (668) of external data memory.

With EXTRAM = 0, the ERAM is indirectly addressed, using the MOVX instruction in combination with any of the registers R0, R1 of the selected bank or DPTR. An access to ERAM will not affect ports P0, P3.6 (WR#) and P3.7 (RD#). P2 SFR is in output state during external addressing. For example, with EXTRAM = 0,

MOVX @R0,A

where R0 contains 0A0H, access the ERAM at address 0A0H rather than external memory. An access to external data memory locations higher than the ERAM will be performed with the MOVX DPTR instructions in the same way as in the standard 80C51 (with P0 and P2 as data/address bus, and P3.6 and P3.7 as write and read timing signals. Refer to Figure 54).

With EXTRAM = 1, MOVX @Ri and MOVX @DPTR will be similar to the standard 80C51. MOVX @ Ri will provide an 8-bit address multiplexed with data on Port 0 and any output port pins can be used to output higher order address bits. This is to provide the external paging capability. MOVX @DPTR will generate a 16-bit address. Port 2 outputs the high-order eight address bits (the contents of DPH) while Port 0 multiplexes the low-order eight address bits (the contents of DPL) with data. MOVX @Ri and MOVX @DPTR will generate either read or write signals on P3.6 ($\overline{WR}$) and P3.7 ($\overline{RD}$).

The stack pointer (SP) may be located anywhere in the 256 bytes RAM (lower and upper RAM) internal data memory. The stack may not be located in the ERAM.

AUXR Address = 8EH Reset Value = xxxx xx10B

Not Bit Addressable

	—	—	—	—	—	—	**EXTRAM**	**AO**
Bit:	7	6	5	4	3	2	1	0

Symbol	Function
AO	Disable/Enable ALE
	AO **Operating Mode**
	0 ALE is emitted at a constant rate of $^1/_3$ the oscillator frequency (6 clock mode; $^1/_6$ f_{OSC} in 12 clock mode)
	1 ALE is active only during off-chip memory access.
EXTRAM	Internal/External RAM access using MOVX @Ri/@DPTR
	EXTRAM **Operating Mode**
	0 Internal ERAM access using MOVX @Ri/@DPTR
	1 External data memory access.
—	Not implemented, reserved for future use*.

NOTE:
*User software should not write 1s to reserved bits. These bits may be used in future 8051 family products to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1. The value read from a reserved bit is indeterminate.

SU01711

Figure 53. AUXR: Auxiliary Register

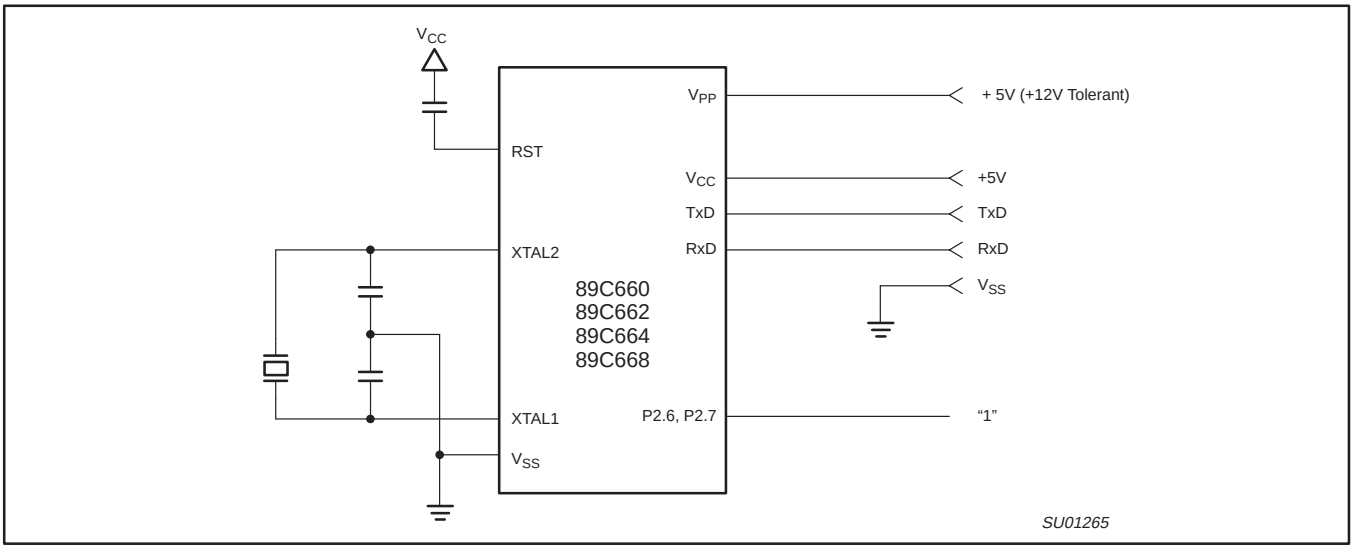


Figure 56. In-System Programming with a Minimum of Pins

In-System Programming (ISP)

The In-System Programming (ISP) is performed without removing the microcontroller from the system. The In-System Programming (ISP) facility consists of a series of internal hardware resources coupled with internal firmware to facilitate remote programming of the P89C660/662/664/668 through the serial port. This firmware is provided by Philips and embedded within each P89C660/662/664/668 device.

The Philips In-System Programming (ISP) facility has made in-circuit programming in an embedded application possible with a minimum of additional expense in components and circuit board area.

The ISP function uses five pins: TxD, RxD, V_{SS}, V_{CC}, and V_{PP} (see Figure 56). Only a small connector needs to be available to interface your application to an external circuit in order to use this feature. The V_{PP} supply should be adequately decoupled and V_{PP} not allowed to exceed datasheet limits.

Free ISP software is available on the Philips web site: "WinISP"

1. Direct your browser to the following page: **http://www.semiconductors.philips.com/products/standard/microcontrollers/download/80c51/flash/**
2. Download "WinISP.exe"
3. Execute WinISP.exe to install the software

Free ISP software is also available from the Embedded Systems Academy: "FlashMagic"

1. Direct your browser to the following page: **http://www.esacademy.com/software/flashmagic/**
2. Download Flashmagic
3. Execute "flashmagic.exe" to install the software

Using the In-System Programming (ISP)

The ISP feature allows for a wide range of baud rates to be used in your application, independent of the oscillator frequency. It is also adaptable to a wide range of oscillator frequencies. This is accomplished by measuring the bit-time of a single bit in a received character. This information is then used to program the baud rate in terms of timer counts based on the oscillator frequency. The ISP feature requires that an initial character (an uppercase U) be sent to the P89C660/662/664/668 to establish the baud rate. The ISP firmware provides auto-echo of received characters.

Once baud rate initialization has been performed, the ISP firmware will only accept Intel Hex-type records. Intel Hex records consist of ASCII characters used to represent hexadecimal values and are summarized below:

:NNAAAARRDD..DDCC<crlf>

In the Intel Hex record, the "NN" represents the number of data bytes in the record. The P89C660/662/664/668 will accept up to 16 (10H) data bytes. The "AAAA" string represents the address of the first byte in the record. If there are zero bytes in the record, this field is often set to 0000. The "RR" string indicates the record type. A record type of "00" is a data record. A record type of "01" indicates the end-of-file mark. In this application, additional record types will be added to indicate either commands or data for the ISP facility. The maximum number of data bytes in a record is limited to 16 (decimal). ISP commands are summarized in Table 14.

As a record is received by the P89C660/662/664/668, the information in the record is stored internally and a checksum calculation is performed. The operation indicated by the record type is not performed until the entire record has been received. Should an error occur in the checksum, the P89C660/662/664/668 will send an "X" out the serial port indicating a checksum error. If the checksum calculation is found to match the checksum in the record, then the command will be executed. In most cases, successful reception of the record will be indicated by transmitting a "." character out the serial port (displaying the contents of the internal program memory is an exceptions).

In the case of a Data Record (record type 00), an additional check is made. A "." character will NOT be sent unless the record checksum matched the calculated checksum and all of the bytes in the record were successfully programmed. For a data record, an "X" indicates that the checksum failed to match, and an "R" indicates that one of the bytes did not properly program. It is necessary to send a type 02 record (specify oscillator frequency) to the P89C660/662/664/668 before programming data.

In Application Programming Method

Several In Application Programming (IAP) calls are available for use by an application program to permit selective erasing and programming of Flash sectors. All calls are made through a common interface, PGM_MTP. The programming functions are selected by setting up the microcontroller's registers before making a call to PGM_MTP at FFF0H. The oscillator frequency is an integer number rounded down to the nearest megahertz. For example, set R0 to 11 for 11.0592 MHz. Results are returned in the registers. The IAP calls are shown in Table 15.

Using the Watchdog Timer (WDT)

The 89C66x devices support the use of the WDT in IAP. The user specifies that the WDT is to be fed by setting the most significant bit of the function parameter passed in R1 prior to calling PGM_MTP. The WDT function is only supported for Block Erase when using the Quick Block Erase. The Quick Block Erase is specified by performing a Block Erase with register R0 = 0. Requesting a WDT feed during IAP should only be performed in applications that use the WDT since the process of feeding the WDT will start the WDT if the WDT was not working.

Table 15. IAP calls

IAP CALL	PARAMETER

PROGRAM DATA BYTE	Input Parameters:
	R0 = osc freq (integer)
	R1 = 02h
	R1 = 82h (WDT feed, Rx2 & 66x only)
	DPTR = address of byte to program
	ACC = byte to program
	Return Parameter
	ACC = 00 if pass, !00 if fail
	Sample routine:
	;***** Program Device Data (DData) *****
	;***** ACC holds data to write
	;***** DPTR holds address of byte to write *****
	;***** Returns with ACC = 00h if successful, else ACC NEQ 00h
	WRData:
	MOV AUXR1,#20H ;set the ENBOOT bit
	MOV R0, #11 ;FOSC
	MOV R1,#02H ;program data function
	MOV A,Mydata ;data to write
	MOV DPTR,Address ;specify address of byte to read
	CALL PGM_MTP ;execute the function
	RET

ERASE BLOCK	Input Parameters:
	R0 = osc freq (integer)
	R0 = 0 (QUICK ERASE, Rx2 & 66x only)
	R1 = 01h
	R1 = 81h (WDT feed, Rx2 & 66x only; can only be used with Quick Erase)
	DPH = block code as shown below:
	block 0, 0k to 8k, 00H
	block 1, 8k to 16k, 20H
	block 2, 16k to 32k, 40H
	block 3, 32k to 48k, 80H
	block 4, 48k to 64k, C0H
	
	DPL = 00h
	Return Parameter
	none
	Sample routine:
	;***** Erase Code Memory Block *****
	;***** DPH (7:5) indicates which of the 5 blocks to erase
	;***** DPTR values for the blocks are:
	; 0000h = block 0
	; 2000h = block 1
	; 4000h = block 2
	; 8000h = block 3
	; C000h = block 4
	
	ERSBLK:
	MOV AUXR1,#20H ;set the ENBOOT bit
	MOV R0, #11 ;FOSC
	MOV R1,#01H ;erase block
	MOV DPTR,#BLk_NUM ;specify which block
	CALL PGM_MTP ;execute the function
	RET

ABSOLUTE MAXIMUM RATINGS[1, 2, 3]

PARAMETER	RATING	UNIT
Operating temperature under bias	0 to +70 or –40 to +85	°C
Storage temperature range	–65 to +150	°C
Voltage on $\overline{EA}/V_{PP}$ pin to V_{SS}	0 to +13.0	V
Voltage on any other pin to V_{SS}	–0.5 to +6.5	V
Maximum I_{OL} per I/O pin	15	mA
Power dissipation (based on package heat transfer limitations, not device power consumption)	1.5	W

NOTES:
1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any conditions other than those described in the AC and DC Electrical Characteristics section of this specification is not implied.
2. This product includes circuitry specifically designed for the protection of its internal devices from the damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying greater than the rated maximum.
3. Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to V_{SS} unless otherwise noted.

AC ELECTRICAL CHARACTERISTICS (6 CLOCK MODE)

T_{amb} = 0 °C to +70 °C, V_{CC} = 5 V ± 10% or –40 °C to +85 °C, V_{CC} = 5 V ±5%, V_{SS} = 0 V[1, 2, 3]

SYMBOL	FIGURE	PARAMETER	VARIABLE CLOCK[4]		20 MHz CLOCK[4]		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	57	Oscillator frequency	0	20	–	–	MHz
t_{LHLL}	57	ALE pulse width	t_{CLCL}–40	–	10	–	ns
t_{AVLL}	57	Address valid to ALE low	$0.5t_{CLCL}$–20	–	5	–	ns
t_{LLAX}	57	Address hold after ALE low	$0.5t_{CLCL}$–20	–	5	–	ns
t_{LLIV}	57	ALE low to valid instruction in	–	$2t_{CLCL}$–65	–	35	ns
t_{LLPL}	57	ALE low to $\overline{PSEN}$ low	$0.5t_{CLCL}$–20	–	5	–	ns
t_{PLPH}	57	$\overline{PSEN}$ pulse width	$1.5t_{CLCL}$–45	–	30	–	ns
t_{PLIV}	57	$\overline{PSEN}$ low to valid instruction in	–	$1.5t_{CLCL}$–60	–	15	ns
t_{PXIX}	57	Input instruction hold after $\overline{PSEN}$	0	–	0	–	ns
t_{PXIZ}	57	Input instruction float after $\overline{PSEN}$	–	$0.5t_{CLCL}$–20	–	5	ns
t_{AVIV}	57	Address to valid instruction in	–	$2.5t_{CLCL}$–80	–	45	ns
t_{PLAZ}	57	$\overline{PSEN}$ low to address float	–	10	–	10	ns
Data Memory							
t_{RLRH}	58, 59	$\overline{RD}$ pulse width	$3t_{CLCL}$–100	–	50	–	ns
t_{WLWH}	58, 59	$\overline{WR}$ pulse width	$3t_{CLCL}$–100	–	50	–	ns
t_{RLDV}	58, 59	$\overline{RD}$ low to valid data in	–	$2.5t_{CLCL}$–90	–	35	ns
t_{RHDX}	58, 59	Data hold after $\overline{RD}$	0	–	0	–	ns
t_{RHDZ}	58, 59	Data float after $\overline{RD}$	–	t_{CLCL}–20	–	5	ns
t_{LLDV}	58, 59	ALE low to valid data in	–	$4t_{CLCL}$–150	–	50	ns
t_{AVDV}	58, 59	Address to valid data in	–	$4.5t_{CLCL}$–165	–	60	ns
t_{LLWL}	58, 59	ALE low to $\overline{RD}$ or $\overline{WR}$ low	$1.5t_{CLCL}$–50	$1.5t_{CLCL}$+50	25	125	ns
t_{AVWL}	58, 59	Address valid to $\overline{WR}$ low or $\overline{RD}$ low	$2t_{CLCL}$–75	–	25	–	ns
t_{QVWX}	58, 59	Data valid to $\overline{WR}$ transition	$0.5t_{CLCL}$–25	–	0	–	ns
t_{WHQX}	58, 59	Data hold after $\overline{WR}$	$0.5t_{CLCL}$–20	–	5	–	ns
t_{QVWH}	59	Data valid to $\overline{WR}$ high	$3.5t_{CLCL}$–130	–	45	–	ns
t_{RLAZ}	58, 59	$\overline{RD}$ low to address float	–	0	–	0	ns
t_{WHLH}	58, 59	$\overline{RD}$ or $\overline{WR}$ high to ALE high	$0.5t_{CLCL}$–20	$0.5t_{CLCL}$+20	5	45	ns
External Clock							
t_{CHCX}	61	High time	20	t_{CLCL}–t_{CLCX}	–	–	ns
t_{CLCX}	61	Low time	20	t_{CLCL}–t_{CHCX}	–	–	ns
t_{CLCH}	61	Rise time	–	5	–	–	ns
t_{CHCL}	61	Fall time	–	5	–	–	ns
Shift Register							
t_{XLXL}	60	Serial port clock cycle time	$6t_{CLCL}$	–	300	–	ns
t_{QVXH}	60	Output data setup to clock rising edge	$5t_{CLCL}$–133	–	117	–	ns
t_{XHQX}	60	Output data hold after clock rising edge	t_{CLCL}–30	–	20	–	ns
t_{XHDX}	60	Input data hold after clock rising edge	0	–	0	–	ns
t_{XHDV}	60	Clock rising edge to input data valid	–	$5t_{CLCL}$–133	–	117	ns

NOTES:
1. Parameters are valid over operating temperature range unless otherwise specified.
2. Load capacitance for port 0, ALE, and $\overline{PSEN}$ = 100 pF, load capacitance for all other outputs = 80 pF.
3. Interfacing the microcontroller to devices with float times up to 45ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.
4. Parts are tested to 2 MHz, but are guaranteed to operate down to 0 Hz.

AC ELECTRICAL CHARACTERISTICS (6 CLOCK MODE) (Continued)

T_{amb} = 0 °C to +70 °C, V_{CC} = 5 V ± 10% or –40 °C to +85 °C, V_{CC} = 5 V ± 5%, V_{SS} = 0 V[1, 2]

SYMBOL	PARAMETER	INPUT	OUTPUT
I²C Interface			
$t_{HD;STA}$	START condition hold time	≥ 7 t_{CLCL}	> 4.0 µs [4]
t_{LOW}	SCL low time	≥ 8 t_{CLCL}	> 4.7 µs [46]
t_{HIGH}	SCL high time	≥ 7 t_{CLCL}	> 4.0 µs [4]
t_{RC}	SCL rise time	≤ 1 µs	– [5]
t_{FC}	SCL fall time	≤ 0.3 µs	< 0.3 µs [6]
$t_{SU;DAT1}$	Data set-up time	≥ 250 ns	> 10 t_{CLCL} – t_{RD}
$t_{SU;DAT2}$	SDA set-up time (before rep. START cond.)	≥ 250 ns	> 1 µs [4]
$t_{SU;DAT3}$	SDA set-up time (before STOP cond.)	≥ 250 ns	> 4 t_{CLCL}
$t_{HD;DAT}$	Data hold time	≥ 0 ns	> 4 t_{CLCL} – t_{FC}
$t_{SU;STA}$	Repeated START set-up time	≥ 7 t_{CLCL} [4]	> 4.7 µs [4]
$t_{SU;STO}$	STOP condition set-up time	≥ 7 t_{CLCL} [4]	> 4.0 µs [4]
t_{BUF}	Bus free time	≥ 7 t_{CLCL} [4]	> 4.7 µs [4]
t_{RD}	SDA rise time	≤ 1 µs[7]	– [5]
t_{FD}	SDA fall time	≤ 300 ns[7]	< 0.3 µs [6]

NOTES:
1. Parameters are valid over operating temperature range and voltage range unless otherwise specified.
2. Load capacitance for port 0, ALE, and $\overline{PSEN}$ = 100 pF, load capacitance for all other outputs = 80 pF.
3. These values are characterized but not 100% production tested.
4. At 100 kbit/s. At other bit rates this value is inversely proportional to the bit-rate of 100 kbit/s.
5. Determined by the external bus-line capacitance and the external bus-line pull-resistor, this must be < 1 µs.
6. Spikes on the SDA and SCL lines with a duration of less than 3 t_{CLCL} will be filtered out. Maximum capacitance on bus-lines SDA and SCL = 400 pF.
7. $t_{CLCL} = 1/f_{OSC}$ = one oscillator clock period at pin XTAL1.

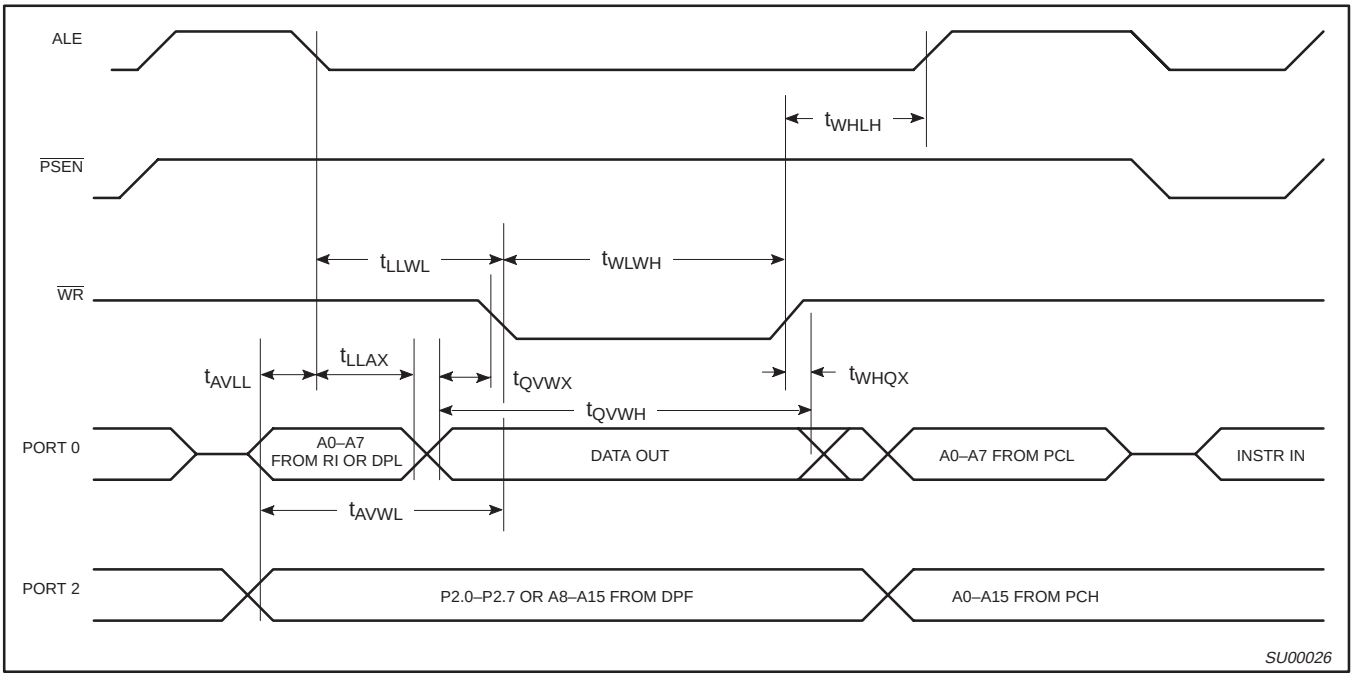


Figure 59. External Data Memory Write Cycle

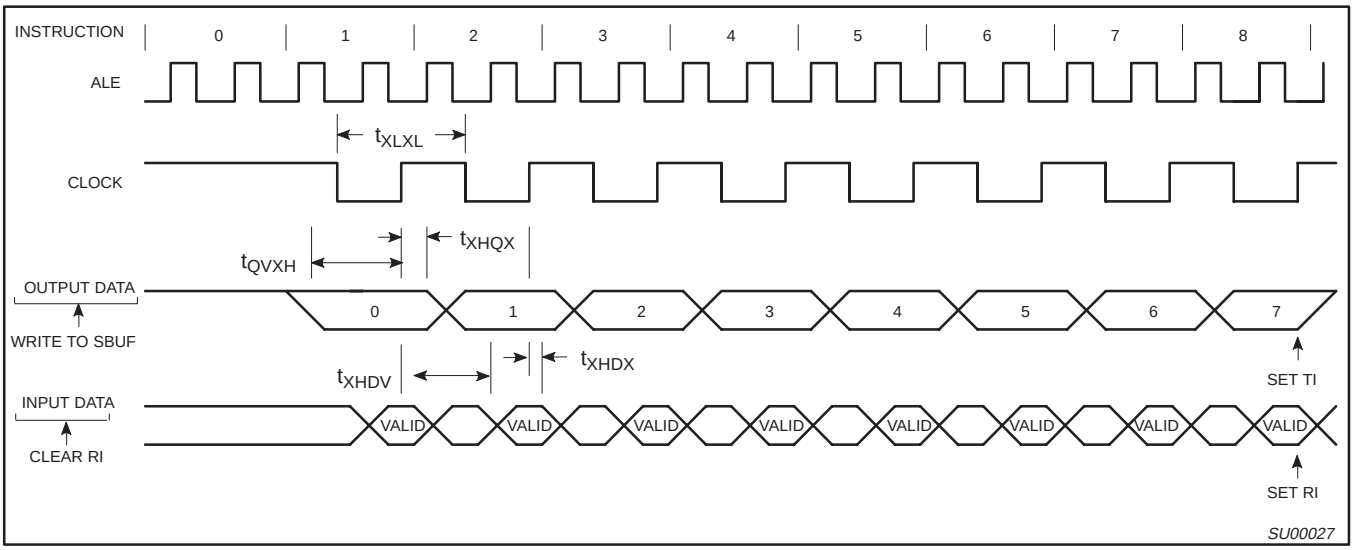


Figure 60. Shift Register Mode Timing

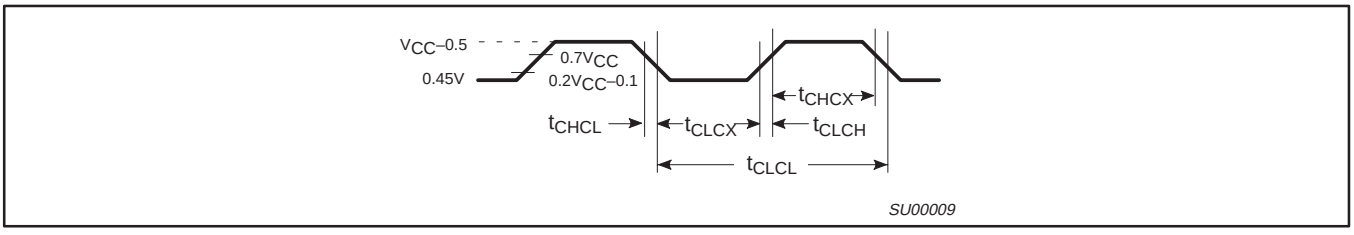


Figure 61. External Clock Drive