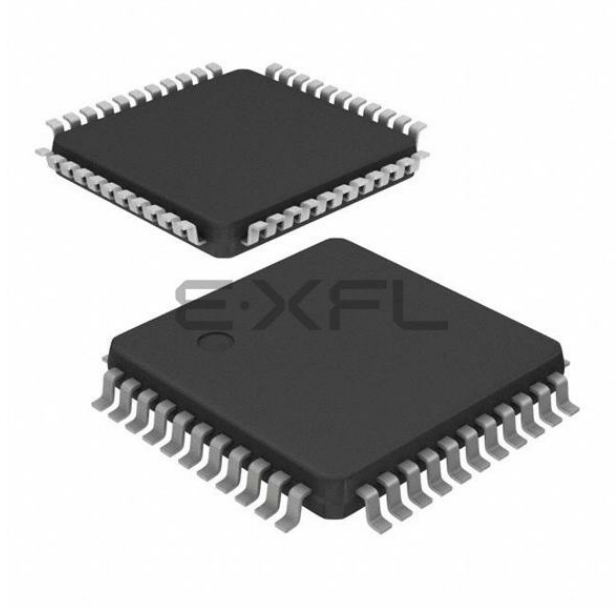


Welcome to E-XFL.COM

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	EBI/EMI, I²C, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	32
Program Memory Size	16KB (16K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/p87c661x2bbd-157

CLOCK CONTROL REGISTER (CKCON)

This device allows control of the 6-clock/12-clock mode by means of both an SFR bit (X2) and an OTP bit. The OTP clock control bit OX2, when programmed by a parallel programmer (6-clock mode), supersedes the X2 bit (CKCON.0). The CKCON register is shown below in Figure 1.

CKCON Address = 8Fh Reset Value = x0000000B

Not Bit Addressable

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	X2

BIT	SYMBOL	FUNCTION
CKCON.7	–	Reserved.
CKCON.6		Reserved.
CKCON.5		Reserved.
CKCON.4		Reserved.
CKCON.3		Reserved.
CKCON.2		Reserved.
CKCON.1		Reserved.
CKCON.0	X2	CPU clock; 1 = 6 clocks for each machine cycle, 0 = 12 clocks for each machine cycle

SU01689

Figure 1. Clock control (CKCON) register

Also please note that the clock divider applies to the serial port for modes 0 & 2 (fixed baud rate modes). This is because modes 1 & 3 (variable baud rate modes) use either Timer 1 or Timer 2.

Below is the truth table for the CPU clock mode.

Table 1.

OX2 clock mode bit (can only be set by parallel programmer)	X2 bit (CKCON.0)	CPU clock mode
erased	0	12-clock mode (default)
erased	1	6-clock mode
programmed	X	6-clock mode

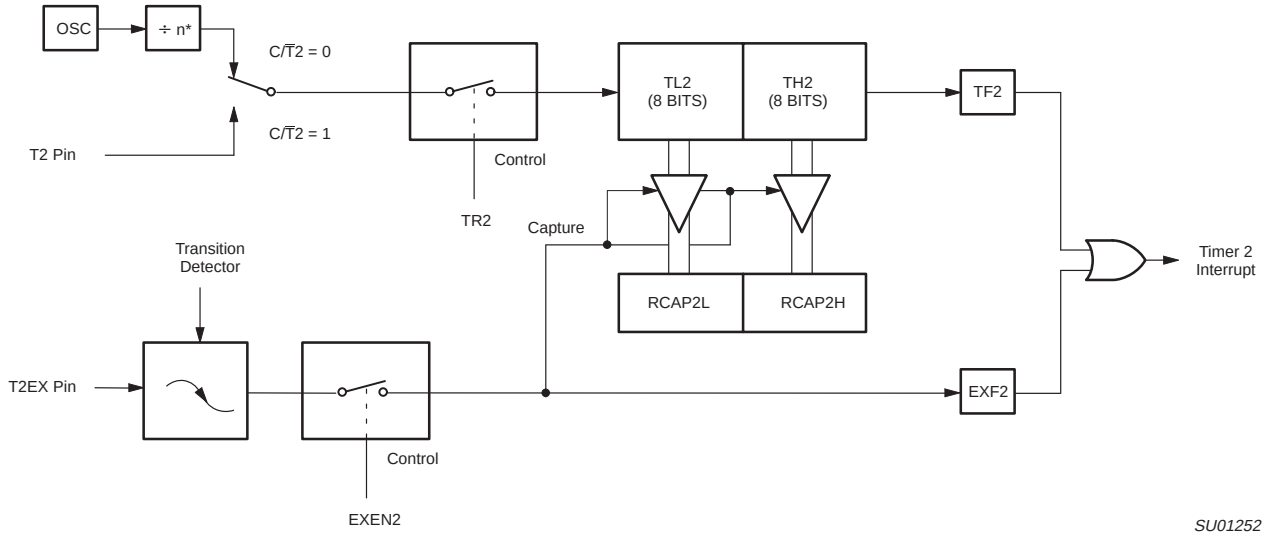
RESET

A reset is accomplished by holding the RST pin HIGH for at least two machine cycles (12 oscillator periods in 6-clock mode, or 24 oscillator periods in 12-clock mode), while the oscillator is running. To ensure a good power-on reset, the RST pin must be HIGH long enough to allow the oscillator time to start up (normally a few milliseconds) plus two machine cycles. At power-on, the voltage on V_{CC} and RST must come up at the same time for a proper start-up. Ports 1, 2, and 3 will asynchronously be driven to their reset condition when a voltage above V_{IH1} (min.) is applied to RST.

The value on the $\overline{EA}$ pin is latched when RST is deasserted and has no further effect.

Table 3. 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 2. 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.

* 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.

SU00729

Figure 3. Timer 2 Mode (T2MOD) Control Register

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 (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 the Timer 2 or RCAP2 registers.

Table 4 shows commonly used baud rates and how they can be obtained from Timer 2.

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:

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

* 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:

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

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 5 for set-up of Timer 2 as a timer. Also see Table 6 for set-up of Timer 2 as a counter.

Table 5. 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 6. 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.

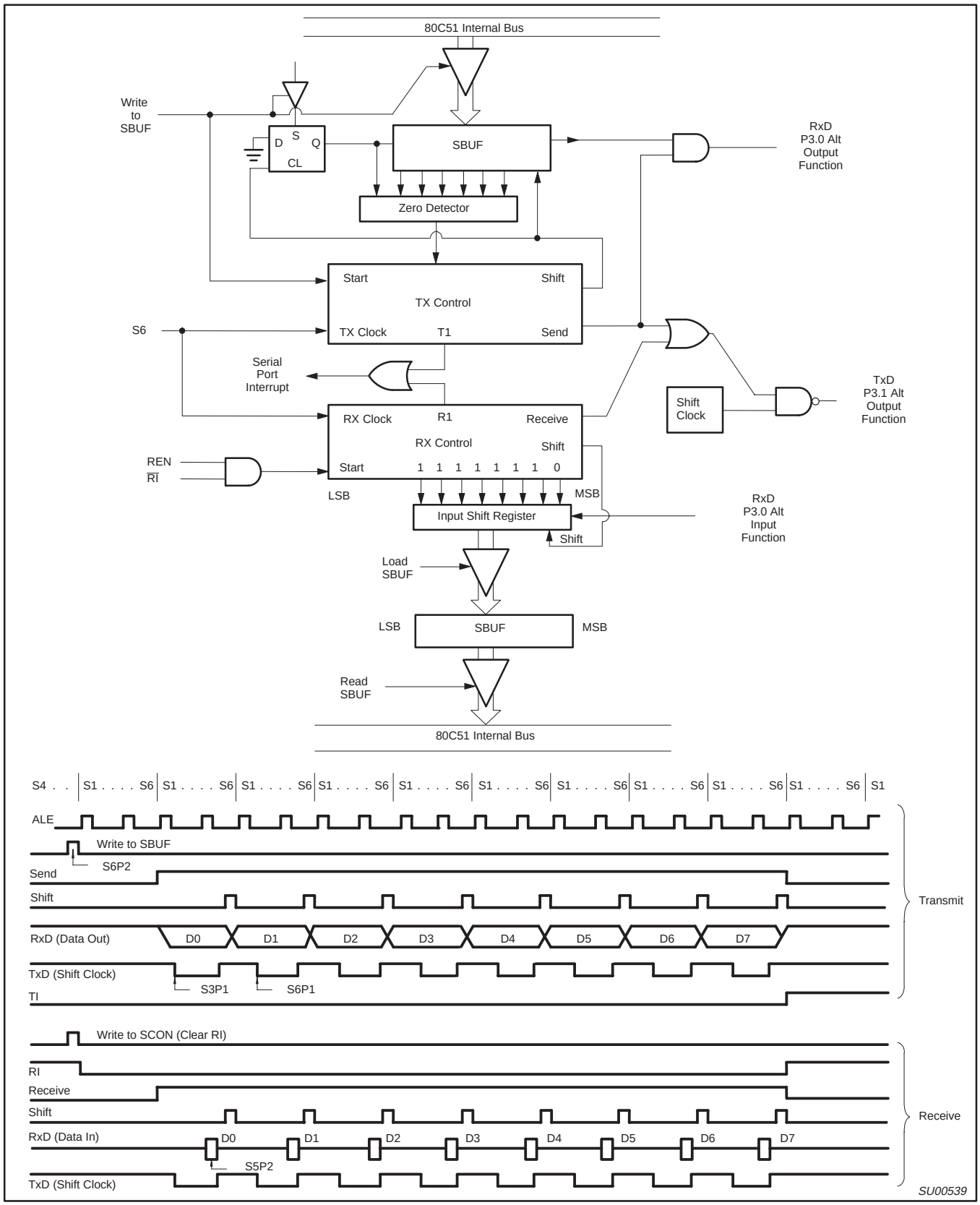


Figure 9. Serial Port Mode 0

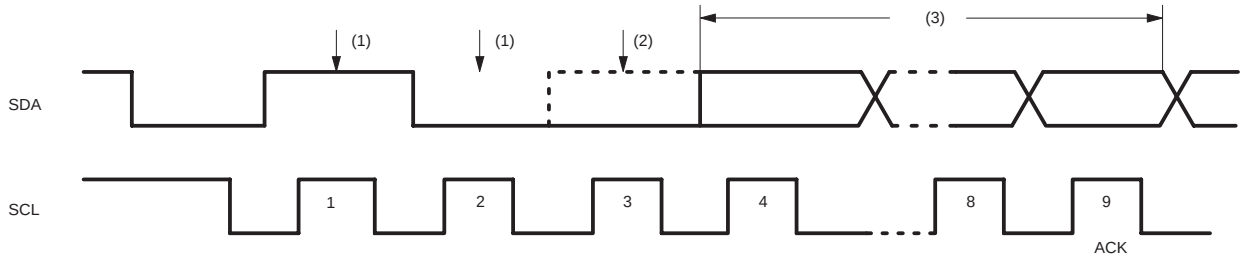
Arbitration and Synchronization Logic

In the master transmitter mode, the arbitration logic checks that every transmitted logic 1 actually appears as a logic 1 on the I²C bus. If another device on the bus overrules a logic 1 and pulls the SDA line LOW, arbitration is lost, and SIO1 immediately changes from master transmitter to slave receiver. SIO1 will continue to output clock pulses (on SCL) until transmission of the current serial byte is complete.

Arbitration may also be lost in the master receiver mode. Loss of arbitration in this mode can only occur while SIO1 is returning a “not acknowledge: (logic 1) to the bus. Arbitration is lost when another device on the bus pulls this signal LOW. Since this can occur only at the end of a serial byte, SIO1 generates no further clock pulses. Figure 18 shows the arbitration procedure.

The synchronization logic will synchronize the serial clock generator with the clock pulses on the SCL line from another device. If two or more master devices generate clock pulses, the “mark” duration is determined by the device that generates the shortest “marks,” and the “space” duration is determined by the device that generates the longest “spaces.” Figure 19 shows the synchronization procedure.

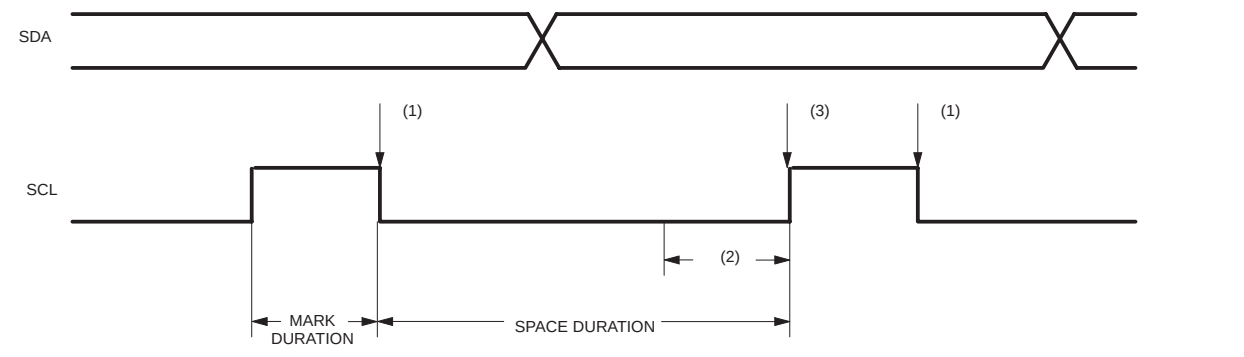
A slave may stretch the space duration to slow down the bus master. The space duration may also be stretched for handshaking purposes. This can be done after each bit or after a complete byte transfer. SIO1 will stretch the SCL space duration after a byte has been transmitted or received and the acknowledge bit has been transferred. The serial interrupt flag (SI) is set, and the stretching continues until the serial interrupt flag is cleared.



1. Another device transmits identical serial data.
2. Another device overrules a logic 1 (dotted line) transmitted by SIO1 (master) by pulling the SDA line low. Arbitration is lost, and SIO1 enters the slave receiver mode.
3. SIO1 is in the slave receiver mode but still generates clock pulses until the current byte has been transmitted. SIO1 will not generate clock pulses for the next byte. Data on SDA originates from the new master once it has won arbitration.

SU00967

Figure 18. Arbitration Procedure



1. Another service pulls the SCL line low before the SIO1 “mark” duration is complete. The serial clock generator is immediately reset and commences with the “space” duration by pulling SCL low.
2. Another device still pulls the SCL line low after SIO1 releases SCL. The serial clock generator is forced into the wait state until the SCL line is released.
3. The SCL line is released, and the serial clock generator commences with the mark duration.

SU00968

Figure 19. Serial Clock Synchronization

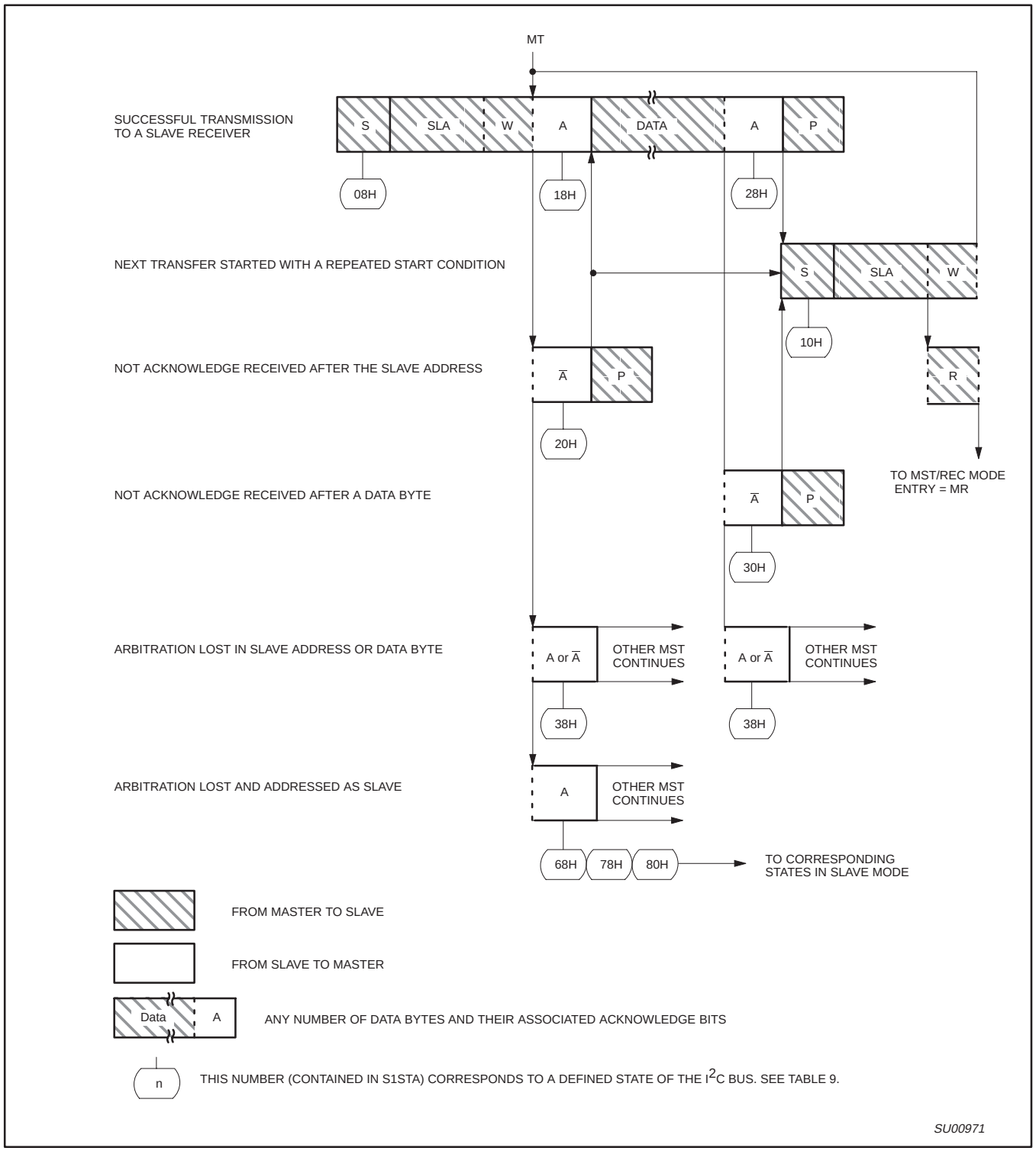


Figure 22. Format and States in the Master Transmitter Mode

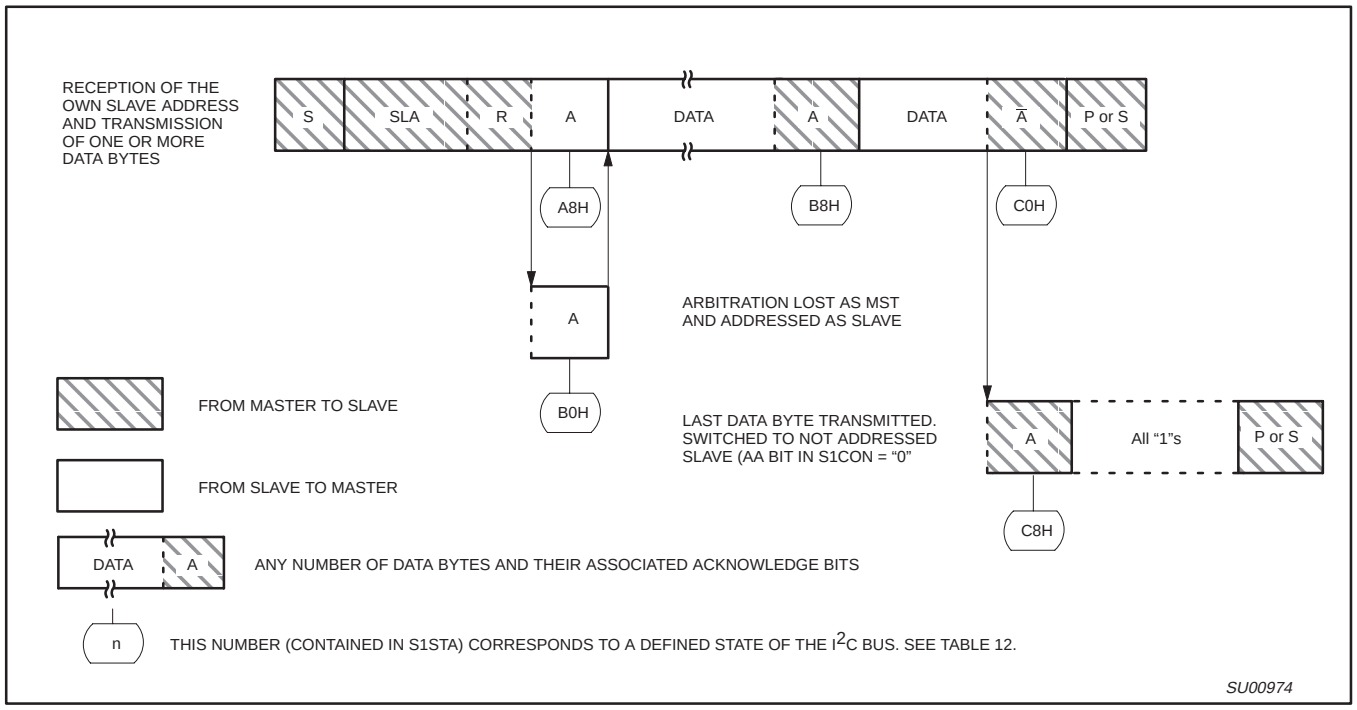


Figure 25. Format and States of the Slave Transmitter Mode

Table 9. 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 10. Master Receiver Mode

STATUS CODE (S1STA)	STATUS OF THE I²C 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+R	X	0	0	X	SLA+R will be transmitted; ACK bit will be received
10H	A repeated START condition has been transmitted	Load SLA+R or	X	0	0	X	As above
		Load SLA+W	X	0	0	X	SLA+W will be transmitted; SIO1 will be switched to MST/TRX mode
38H	Arbitration lost in NOT ACK bit	No S1DAT action or	0	0	0	X	I²C bus will be released; SIO1 will enter a slave mode
		No S1DAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free
40H	SLA+R has been transmitted; ACK has been received	No S1DAT action or	0	0	0	0	Data byte will be received; NOT ACK bit will be returned
		no S1DAT action	0	0	0	1	Data byte will be received; ACK bit will be returned
48H	SLA+R has been transmitted; NOT ACK has been received	No S1DAT action or	1	0	0	X	Repeated START condition 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
50H	Data byte has been received; ACK has been returned	Read data byte or	0	0	0	0	Data byte will be received; NOT ACK bit will be returned
		read data byte	0	0	0	1	Data byte will be received; ACK bit will be returned
58H	Data byte has been received; NOT ACK has been returned	Read data byte or	1	0	0	X	Repeated START condition will be transmitted
		read data byte or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
		read data byte	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset

Table 13. Miscellaneous States

STATUS CODE (S1STA)	STATUS OF THE I²C 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 25). 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 12. 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 I²C 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 I²C bus.

Miscellaneous States: There are two S1STA codes that do not correspond to a defined SIO1 hardware state (see Table 13). 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 26). 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 I²C 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 18). Loss of arbitration is indicated by the following states in S1STA; 38H, 68H, 78H, and B0H (see Figures 22 and 23).

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 I²C 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 I²C 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 I²C 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 STO flag is cleared by hardware (see Figure 27).

Programmable Counter Array (PCA)

The Programmable Counter Array available on the P8xC66xX2 is a special 16-bit Timer that has five 16-bit capture/compare modules associated with it. Each of the modules can be programmed to operate in one of four modes: rising and/or falling edge capture, software timer, high-speed output, or pulse width modulator. Each module has a pin associated with it in port 1. Module 0 is connected to P1.3 (CEX0), module 1 to P1.4 (CEX1), etc. The basic PCA configuration is shown in Figure 35.

The PCA timer is a common time base for all five modules and can be programmed to run at: 1/6 the oscillator frequency, 1/2 the oscillator frequency, the Timer 0 overflow, or the input on the ECI pin (P1.2). The timer count source is determined from the CPS1 and CPS0 bits in the CMOD SFR as follows (see Figure 38):

CPS1	CPS0	PCA Timer Count Source
0	0	1/6 oscillator frequency (6-clock mode); 1/12 oscillator frequency (12-clock mode)
0	1	1/2 oscillator frequency (6-clock mode); 1/4 oscillator frequency (12-clock mode)
1	0	Timer 0 overflow
1	1	External Input at ECI pin

In the CMOD SFR are three additional bits associated with the PCA. They are CIDL which allows the PCA to stop during idle mode, WDTE which enables or disables the watchdog function on module 4, and ECF which when set causes an interrupt and the PCA overflow flag CF (in the CCON SFR) to be set when the PCA timer overflows. These functions are shown in Figure 36.

The watchdog timer function is implemented in module 4 (see Figure 45).

The CCON SFR contains the run control bit for the PCA and the flags for the PCA timer (CF) and each module (refer to Figure 39). To run the PCA the CR bit (CCON.6) must be set by software. The PCA is shut off by clearing this bit. The CF bit (CCON.7) is set when the PCA counter overflows and an interrupt will be generated if the ECF bit in the CMOD register is set, The CF bit can only be cleared by software. Bits 0 through 4 of the CCON register are the flags for the modules (bit 0 for module 0, bit 1 for module 1, etc.) and are set by hardware when either a match or a capture occurs. These flags also can only be cleared by software. The PCA interrupt system shown in Figure 37.

Each module in the PCA has a special function register associated with it. These registers are: CCAPM0 for module 0, CCAPM1 for module 1, etc. (see Figure 40). The registers contain the bits that control the mode that each module will operate in. The ECCF bit (CCAPMn.0 where n=0, 1, 2, 3, or 4 depending on the module) enables the CCF flag in the CCON SFR to generate an interrupt when a match or compare occurs in the associated module. PWM (CCAPMn.1) enables the pulse width modulation mode. The TOG bit (CCAPMn.2) when set causes the CEX output associated with the module to toggle when there is a match between the PCA counter and the module's capture/compare register. The match bit MAT (CCAPMn.3) when set will cause the CCFn bit in the CCON register to be set when there is a match between the PCA counter and the module's capture/compare register.

The next two bits CAPN (CCAPMn.4) and CAPP (CCAPMn.5) determine the edge that a capture input will be active on. The CAPN bit enables the negative edge, and the CAPP bit enables the positive edge. If both bits are set both edges will be enabled and a capture will occur for either transition. The last bit in the register ECOM (CCAPMn.6) when set enables the comparator function. Figure 41 shows the CCAPMn settings for the various PCA functions.

There are two additional registers associated with each of the PCA modules. They are CCAPnH and CCAPnL and these are the registers that store the 16-bit count when a capture occurs or a compare should occur. When a module is used in the PWM mode these registers are used to control the duty cycle of the output.

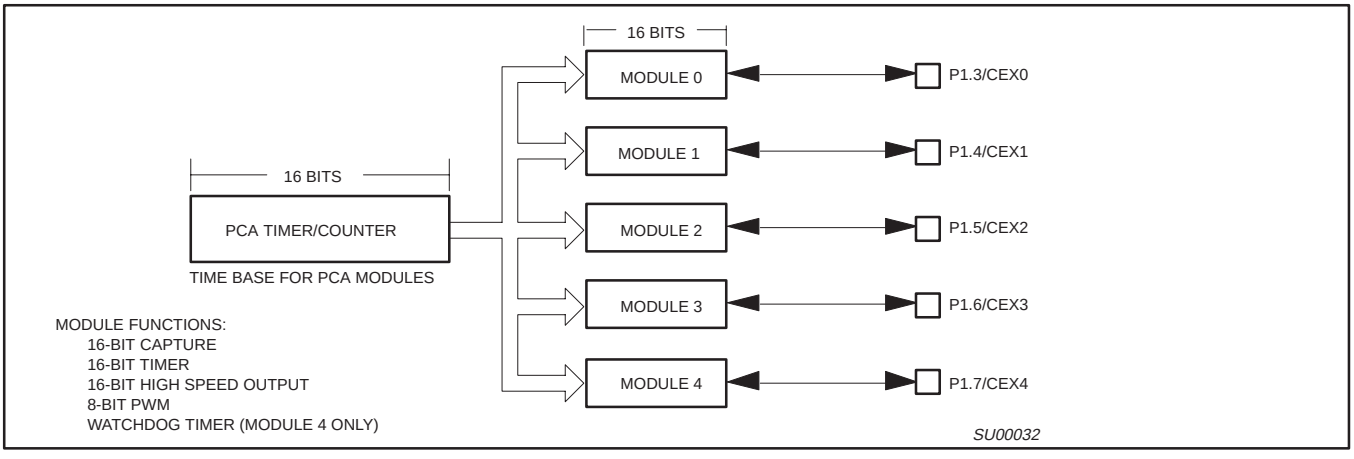


Figure 35. Programmable Counter Array (PCA)

HARDWARE WATCHDOG TIMER (ONE-TIME ENABLED WITH RESET-OUT FOR P8XC66xX2)

The WDT is intended as a recovery method in situations where the CPU may be subjected to software upset. The WDT consists of a 14-bit counter and the WatchDog Timer reset (WDTRST) SFR. The WDT is disabled at reset. To enable the WDT, the user must write 01EH and 0E1H in sequence to the WDTRST, SFR location 0A6H. When the WDT is enabled, it will increment every machine cycle while the oscillator is running and there is no way to disable the WDT except through reset (either hardware reset or WDT overflow reset). When the WDT overflows, it will drive an output reset HIGH pulse at the RST-pin (see the note below).

Using the WDT

To enable the WDT, the user must write 01EH and 0E1H in sequence to the WDTRST, SFR location 0A6H. When the WDT is enabled, the user needs to service it by writing 01EH and 0E1H to WDTRST to avoid a WDT overflow. The 14-bit counter overflows when it reaches 16383 (3FFFH) and this will reset the device. When the WDT is enabled, it will increment every machine cycle while the oscillator is running. This means the user must reset the WDT at least every 16383 machine cycles. To reset the WDT, the user must write 01EH and 0E1H to WDTRST. WDTRST is a write only register. The WDT counter cannot be read or written. When the WDT overflows, it will generate an output RESET pulse at the reset pin (see note below). The RESET pulse duration is $98 \times T_{OSC}$ (6-clock mode; 196 in 12-clock mode), where $T_{OSC} = 1/f_{OSC}$. To make the best use of the WDT, it should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset.

AC ELECTRICAL CHARACTERISTICS (12-CLOCK MODE, 5 V ±10% OPERATION)

T_{amb} = 0 °C to +70 °C or –40 °C to +85 °C ; V_{CC} = 5 V ±10%, V_{SS} = 0 V[1,2,3,4]

Symbol	Figure	Parameter	Limits		16 MHz Clock		Unit
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	55	Oscillator frequency	0	33			MHz
t_{LHLL}	50	ALE pulse width	2 t_{CLCL}–8		117		ns
t_{AVLL}	50	Address valid to ALE LOW	t_{CLCL} –13		49.5		ns
t_{LLAX}	50	Address hold after ALE LOW	t_{CLCL} –20		42.5		ns
t_{LLIV}	50	ALE LOW to valid instruction in		4 t_{CLCL} –35		215	ns
t_{LLPL}	50	ALE LOW to $\overline{PSEN}$ LOW	t_{CLCL} –10		52.5		ns
t_{PLPH}	50	$\overline{PSEN}$ pulse width	3 t_{CLCL} –10		177.5		ns
t_{PLIV}	50	$\overline{PSEN}$ LOW to valid instruction in		3 t_{CLCL} –35		152.5	ns
t_{PXIX}	50	Input instruction hold after $\overline{PSEN}$	0		0		ns
t_{PXIZ}	50	Input instruction float after $\overline{PSEN}$		t_{CLCL} –10		52.5	ns
t_{AVIV}	50	Address to valid instruction in		5 t_{CLCL} –35		277.5	ns
t_{PLAZ}	50	$\overline{PSEN}$ LOW to address float		10		10	ns
Data Memory							
t_{RLRH}	51	$\overline{RD}$ pulse width	6 t_{CLCL} –20		355		ns
t_{WLWH}	52	$\overline{WR}$ pulse width	6 t_{CLCL} –20		355		ns
t_{RLDV}	51	$\overline{RD}$ LOW to valid data in		5 t_{CLCL} –35		277.5	ns
t_{RHDX}	51	Data hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	51	Data float after $\overline{RD}$		2 t_{CLCL} –10		115	ns
t_{LLDV}	51	ALE LOW to valid data in		8 t_{CLCL} –35		465	ns
t_{AVDV}	51	Address to valid data in		9 t_{CLCL} –35		527.5	ns
t_{LLWL}	51, 52	ALE LOW to $\overline{RD}$ or $\overline{WR}$ LOW	3 t_{CLCL} –15	3 t_{CLCL} +15	172.5	202.5	ns
t_{AVWL}	51, 52	Address valid to $\overline{WR}$ LOW or $\overline{RD}$ LOW	4 t_{CLCL} –15		235		ns
t_{QVWX}	52	Data valid to $\overline{WR}$ transition	t_{CLCL} –25		37.5		ns
t_{WHQX}	52	Data hold after $\overline{WR}$	t_{CLCL} –15		47.5		ns
t_{QVWH}	52	Data valid to $\overline{WR}$ HIGH	7 t_{CLCL} –5		432.5		ns
t_{RLAZ}	51	$\overline{RD}$ LOW to address float		0		0	ns
t_{WHLH}	51, 52	$\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	t_{CLCL} –10	t_{CLCL} +10	52.5	72.5	ns
External Clock							
t_{CHCX}	55	High time	0.32 t_{CLCL}	t_{CLCL} – t_{CLCX}			ns
t_{CLCX}	55	Low time	0.32 t_{CLCL}	t_{CLCL} – t_{CHCX}			ns
t_{CLCH}	55	Rise time		5			ns
t_{CHCL}	55	Fall time		5			ns
Shift register							
t_{XLXL}	54	Serial port clock cycle time	12 t_{CLCL}		750		ns
t_{QVXH}	54	Output data setup to clock rising edge	10 t_{CLCL} –25		600		ns
t_{XHQX}	54	Output data hold after clock rising edge	2 t_{CLCL} –15		110		ns
t_{XHDX}	54	Input data hold after clock rising edge	0		0		ns
t_{XHDV}	54	Clock rising edge to input data valid[5]		10 t_{CLCL} –133		492	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 outputs = 80 pF
3. Interfacing the microcontroller to devices with float time up to 45 ns is permitted. This limited bus contention will not cause damage to port 0 drivers.
4. Parts are guaranteed by design to operate down to 0 Hz.
5. Below 16 MHz this parameter is 8 t_{CLCL} – 133.

AC ELECTRICAL CHARACTERISTICS (6-CLOCK MODE, 5 V ±10% OPERATION)

T_{amb} = 0 °C to +70 °C or –40 °C to +85 °C ; V_{CC} = 5 V ±10%, V_{SS} = 0 V[1,2,3,4,5]

Symbol	Figure	Parameter	Limits		16 MHz Clock		Unit
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	55	Oscillator frequency	0	30			MHz
t_{LHLL}	50	ALE pulse width	t_{CLCL}–8		54.5		ns
t_{AVLL}	50	Address valid to ALE LOW	0.5 t_{CLCL} –13		18.25		ns
t_{LLAX}	50	Address hold after ALE LOW	0.5 t_{CLCL} –20		11.25		ns
t_{LLIV}	50	ALE LOW to valid instruction in		2 t_{CLCL} –35		90	ns
t_{LLPL}	50	ALE LOW to $\overline{PSEN}$ LOW	0.5 t_{CLCL} –10		21.25		ns
t_{PLPH}	50	$\overline{PSEN}$ pulse width	1.5 t_{CLCL} –10		83.75		ns
t_{PLIV}	50	$\overline{PSEN}$ LOW to valid instruction in		1.5 t_{CLCL} –35		58.75	ns
t_{PXIX}	50	Input instruction hold after $\overline{PSEN}$	0		0		ns
t_{PXIZ}	50	Input instruction float after $\overline{PSEN}$		0.5 t_{CLCL} –10		21.25	ns
t_{AVIV}	50	Address to valid instruction in		2.5 t_{CLCL} –35		121.25	ns
t_{PLAZ}	50	$\overline{PSEN}$ LOW to address float		10		10	ns
Data Memory							
t_{RLRH}	51	$\overline{RD}$ pulse width	3 t_{CLCL} –20		167.5		ns
t_{WLWH}	52	$\overline{WR}$ pulse width	3 t_{CLCL} –20		167.5		ns
t_{RLDV}	51	$\overline{RD}$ LOW to valid data in		2.5 t_{CLCL} –35		121.25	ns
t_{RHDX}	51	Data hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	51	Data float after $\overline{RD}$		t_{CLCL} –10		52.5	ns
t_{LLDV}	51	ALE LOW to valid data in		4 t_{CLCL} –35		215	ns
t_{AVDV}	51	Address to valid data in		4.5 t_{CLCL} –35		246.25	ns
t_{LLWL}	51, 52	ALE LOW to $\overline{RD}$ or $\overline{WR}$ LOW	1.5 t_{CLCL} –15	1.5 t_{CLCL} +15	78.75	108.75	ns
t_{AVWL}	51, 52	Address valid to $\overline{WR}$ LOW or $\overline{RD}$ LOW	2 t_{CLCL} –15		110		ns
t_{QVWX}	52	Data valid to $\overline{WR}$ transition	0.5 t_{CLCL} –25		6.25		ns
t_{WHQX}	52	Data hold after $\overline{WR}$	0.5 t_{CLCL} –15		16.25		ns
t_{QVWH}	52	Data valid to $\overline{WR}$ HIGH	3.5 t_{CLCL} –5		213.75		ns
t_{RLAZ}	51	$\overline{RD}$ LOW to address float		0		0	ns
t_{WHLH}	51, 52	$\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	0.5 t_{CLCL} –10	0.5 t_{CLCL} +10	21.25	41.25	ns
External Clock							
t_{CHCX}	55	High time	0.4 t_{CLCL}	$t_{CLCL} - t_{CLCX}$			ns
t_{CLCX}	55	Low time	0.4 t_{CLCL}	$t_{CLCL} - t_{CHCX}$			ns
t_{CLCH}	55	Rise time		5			ns
t_{CHCL}	55	Fall time		5			ns
Shift register							
t_{XLXL}	54	Serial port clock cycle time	6 t_{CLCL}		375		ns
t_{QVXH}	54	Output data setup to clock rising edge	5 t_{CLCL} –25		287.5		ns
t_{XHQX}	54	Output data hold after clock rising edge	t_{CLCL} –15		47.5		ns
t_{XHDX}	54	Input data hold after clock rising edge	0		0		ns
t_{XHDV}	54	Clock rising edge to input data valid[6]		5 t_{CLCL} –133		179.5	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 outputs = 80 pF
3. Interfacing the microcontroller to devices with float time up to 45ns is permitted. This limited bus contention will not cause damage to port 0 drivers.
4. Parts are guaranteed by design to operate down to 0 Hz.
5. Data shown in the table are the best mathematical models for the set of measured values obtained in tests. If a particular parameter calculated at a customer specified frequency has a negative value, it should be considered equal to zero.
6. Below 16 MHz this parameter is 4 t_{CLCL} – 133

AC ELECTRICAL CHARACTERISTICS (6-CLOCK MODE, 2.7 V TO 5.5 V OPERATION)

T_{amb} = 0 °C to +70 °C or –40 °C to +85 °C ; V_{CC}=2.7 V to 5.5 V, V_{SS} = 0 V[1,2,3,4,5]

Symbol	Figure	Parameter	Limits		16 MHz Clock		Unit
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	55	Oscillator frequency	0	16			MHz
t_{LHLL}	50	ALE pulse width	t_{CLCL}–10		52.5		ns
t_{AVLL}	50	Address valid to ALE LOW	0.5 t_{CLCL} –15		16.25		ns
t_{LLAX}	50	Address hold after ALE LOW	0.5 t_{CLCL} –25		6.25		ns
t_{LLIV}	50	ALE LOW to valid instruction in		2 t_{CLCL} –55		70	ns
t_{LLPL}	50	ALE LOW to $\overline{PSEN}$ LOW	0.5 t_{CLCL} –15		16.25		ns
t_{PLPH}	50	$\overline{PSEN}$ pulse width	1.5 t_{CLCL} –15		78.75		ns
t_{PLIV}	50	$\overline{PSEN}$ LOW to valid instruction in		1.5 t_{CLCL} –55		38.75	ns
t_{PXIX}	50	Input instruction hold after $\overline{PSEN}$	0		0		ns
t_{PXIZ}	50	Input instruction float after $\overline{PSEN}$		0.5 t_{CLCL} –10		21.25	ns
t_{AVIV}	50	Address to valid instruction in		2.5 t_{CLCL} –50		101.25	ns
t_{PLAZ}	50	$\overline{PSEN}$ LOW to address float		10		10	ns
Data Memory							
t_{RLRH}	51	$\overline{RD}$ pulse width	3 t_{CLCL} –25		162.5		ns
t_{WLWH}	52	$\overline{WR}$ pulse width	3 t_{CLCL} –25		162.5		ns
t_{RLDV}	51	$\overline{RD}$ LOW to valid data in		2.5 t_{CLCL} –50		106.25	ns
t_{RHDX}	51	Data hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	51	Data float after $\overline{RD}$		t_{CLCL} –20		42.5	ns
t_{LLDV}	51	ALE LOW to valid data in		4 t_{CLCL} –55		195	ns
t_{AVDV}	51	Address to valid data in		4.5 t_{CLCL} –50		231.25	ns
t_{LLWL}	51, 52	ALE LOW to $\overline{RD}$ or $\overline{WR}$ LOW	1.5 t_{CLCL} –20	1.5 t_{CLCL} +20	73.75	113.75	ns
t_{AVWL}	51, 52	Address valid to $\overline{WR}$ LOW or $\overline{RD}$ LOW	2 t_{CLCL} –20		105		ns
t_{QVWX}	52	Data valid to $\overline{WR}$ transition	0.5 t_{CLCL} –30		1.25		ns
t_{WHQX}	52	Data hold after $\overline{WR}$	0.5 t_{CLCL} –20		11.25		ns
t_{QVWH}	52	Data valid to $\overline{WR}$ HIGH	3.5 t_{CLCL} –10		208.75		ns
t_{RLAZ}	51	$\overline{RD}$ LOW to address float		0		0	ns
t_{WHLH}	51, 52	$\overline{RD}$ or $\overline{WR}$ HIGH to ALE HIGH	0.5 t_{CLCL} –15	0.5 t_{CLCL} +15	16.25	46.25	ns
External Clock							
t_{CHCX}	55	High time	0.4 t_{CLCL}	t_{CLCL} – t_{CLCX}			ns
t_{CLCX}	55	Low time	0.4 t_{CLCL}	t_{CLCL} – t_{CHCX}			ns
t_{CLCH}	55	Rise time		5			ns
t_{CHCL}	55	Fall time		5			ns
Shift register							
t_{XLXL}	54	Serial port clock cycle time	6 t_{CLCL}		375		ns
t_{QVXH}	54	Output data setup to clock rising edge	5 t_{CLCL} –25		287.5		ns
t_{XHQX}	54	Output data hold after clock rising edge	t_{CLCL} –15		47.5		ns
t_{XHDX}	54	Input data hold after clock rising edge	0		0		ns
t_{XHDV}	54	Clock rising edge to input data valid[6]		5 t_{CLCL} –133		179.5	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 outputs = 80 pF
3. Interfacing the microcontroller to devices with float time up to 45ns is permitted. This limited bus contention will not cause damage to port 0 drivers.
4. Parts are guaranteed by design to operate down to 0 Hz.
5. Data shown in the table are the best mathematical models for the set of measured values obtained in tests. If a particular parameter calculated at a customer specified frequency has a negative value, it should be considered equal to zero.
6. Below 16 MHz this parameter is 4 t_{CLCL} – 133

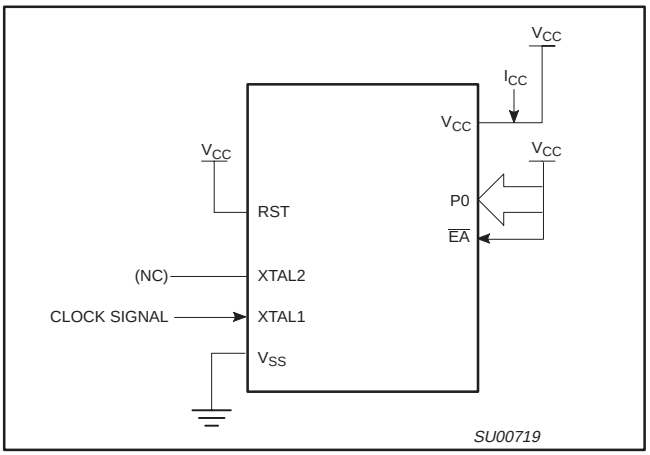


Figure 60. I_{CC} Test Condition, Active Mode
All other pins are disconnected

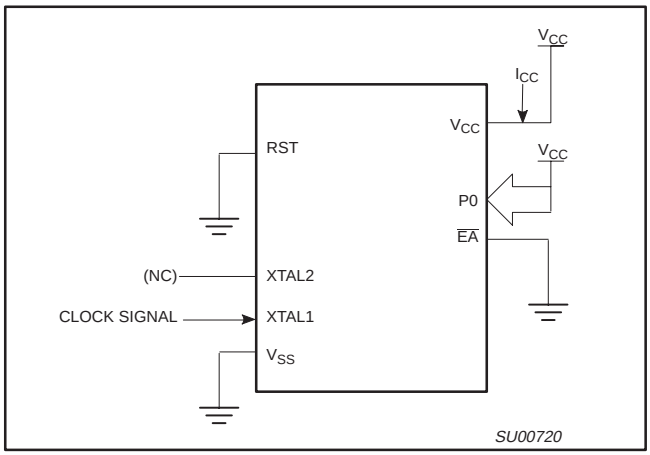


Figure 61. I_{CC} Test Condition, Idle Mode
All other pins are disconnected

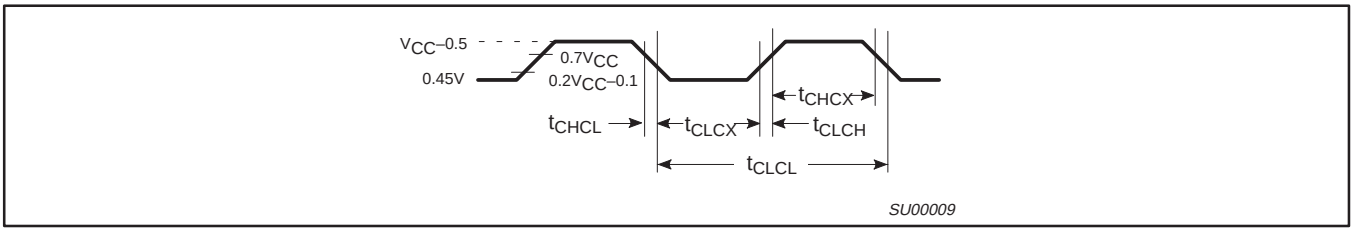


Figure 62. Clock Signal Waveform for I_{CC} Tests in Active and Idle Modes
$t_{CLCH} = t_{CHCL} = 5ns$

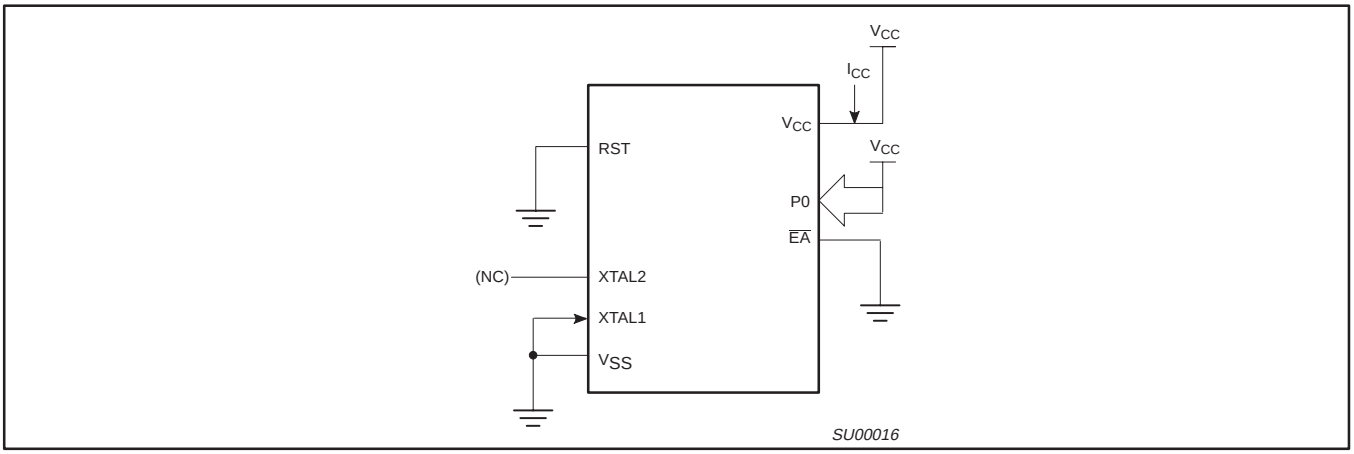


Figure 63. I_{CC} Test Condition, Power Down Mode
All other pins are disconnected. V_{CC} = 2 V to 5.5 V

Table 17. EPROM Programming Modes

MODE	RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}/V_{PP}$	P2.7	P2.6	P3.7	P3.6	P3.3
Read signature	1	0	1	1	0	0	0	0	X
Program code data	1	0	0*	V_{PP}	1	0	1	1	X
Verify code data	1	0	1	1	0	0	1	1	X
Pgm encryption table	1	0	0*	V_{PP}	1	0	1	0	X
Pgm security bit 1	1	0	0*	V_{PP}	1	1	1	1	X
Pgm security bit 2	1	0	0*	V_{PP}	1	1	0	0	X
Pgm security bit 3	1	0	0*	V_{PP}	0	1	0	1	X
Program to 6-clock mode	1	0	0*	V_{PP}	0	0	1	0	0
Verify 6-clock[4]	1	0	1	1	e	0	0	1	1
Verify security bits[5]	1	0	1	1	e	0	1	0	X

NOTES:
1. '0' = Valid LOW for that pin, '1' = valid HIGH for that pin.
2. V_{PP} = 12.75 V ±0.25 V.
3. V_{CC} = 5 V±10% during programming and verification.
4. Bit is output on P0.4 (1 = 12x, 0 = 6x).
5. Security bit one is output on P0.7.
   Security bit two is output on P0.6.
   Security bit three is output on P0.3.

* ALE/$\overline{PROG}$ receives 5 programming pulses for code data (also for user array; 5 pulses for encryption or security bits) while V_{PP} is held at 12.75 V. Each programming pulse is LOW for 100 µs (±10 µs) and HIGH for a minimum of 10 µs.

Table 18. Program Security Bits for EPROM Devices

PROGRAM LOCK BITS[1, 2]				PROTECTION DESCRIPTION
	SB1	SB2	SB3	
1	U	U	U	No Program Security features enabled. (Code verify will still be encrypted by the Encryption Array if programmed.)
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on Reset, and further programming of the EPROM is disabled.
3	P	P	U	Same as 2, also verify is disabled.
4	P	P	P	Same as 3, external execution is disabled. Internal data RAM is not accessible.

NOTES:
1. P – programmed. U – unprogrammed.
2. Any other combination of the security bits is not defined.

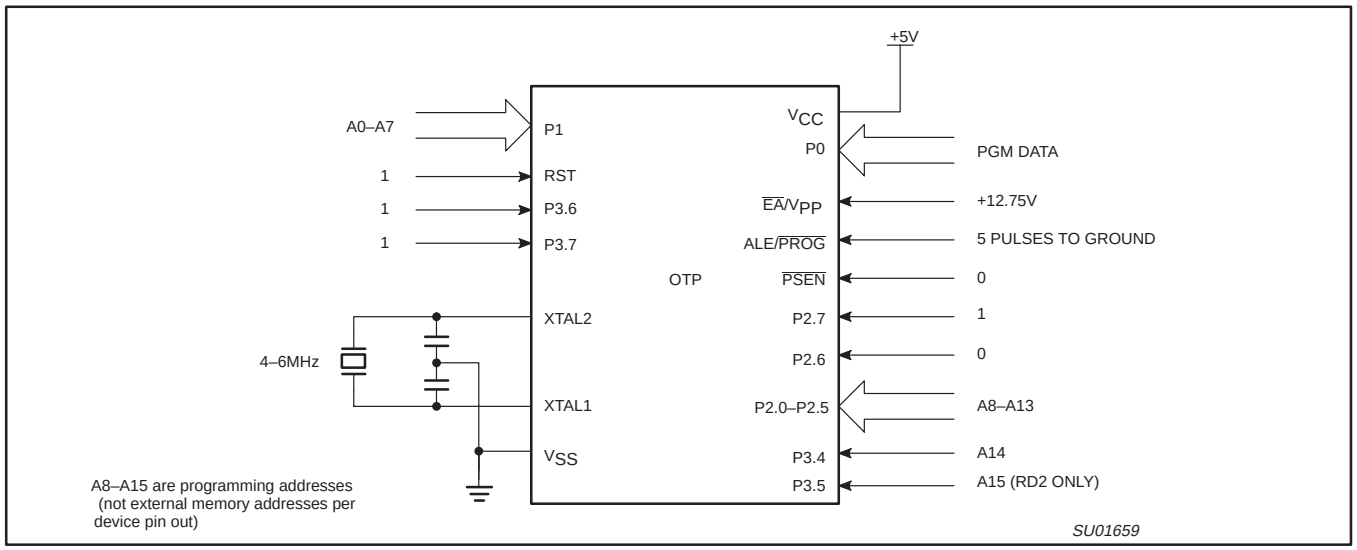


Figure 64. Programming Configuration

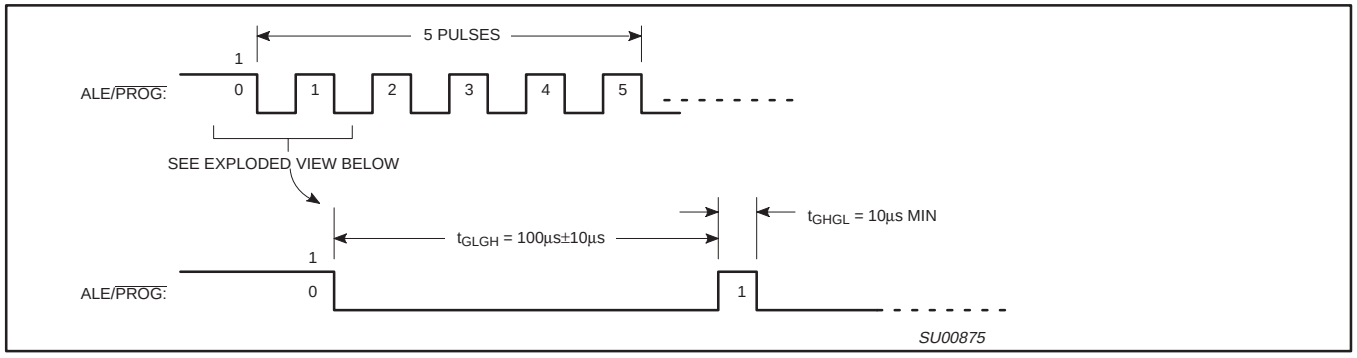


Figure 65. PROG Waveform

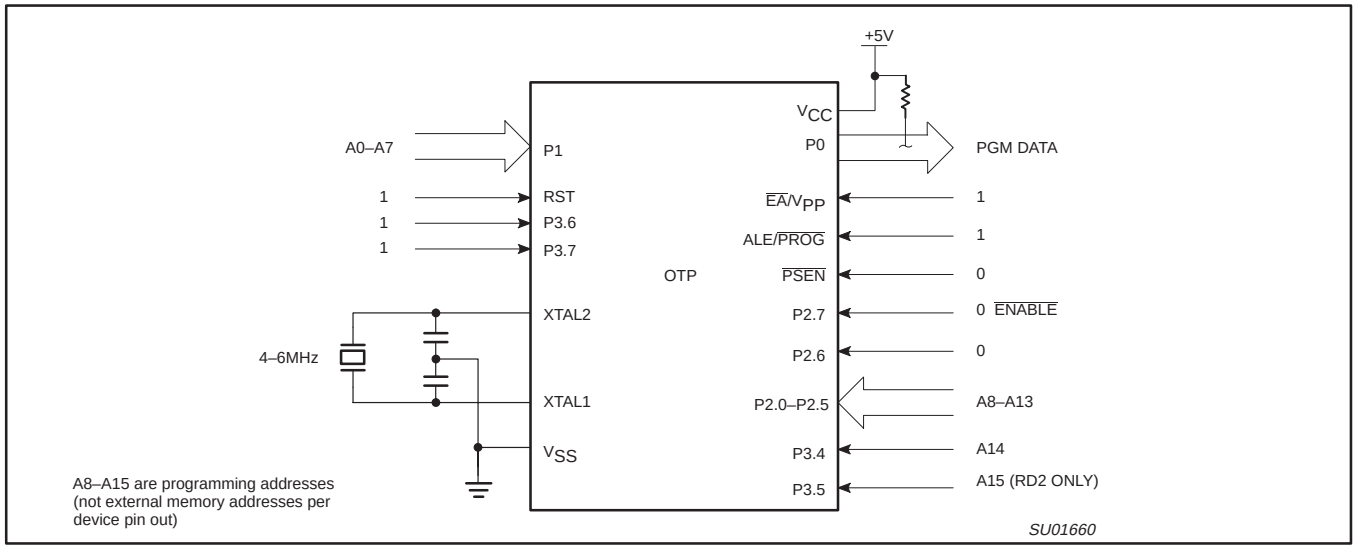


Figure 66. Program Verification

MASK ROM DEVICES

Security Bits

With none of the security bits programmed the code in the program memory can be verified. If the encryption table is programmed, the code will be encrypted when verified. When only security bit 1 (see Table 19) is programmed, MOVC instructions executed from external program memory are disabled from fetching code bytes from the internal memory, $\overline{EA}$ is latched on Reset and all further programming of the EPROM is disabled. When security bits 1 and 2 are programmed, in addition to the above, verify mode is disabled.

Encryption Array

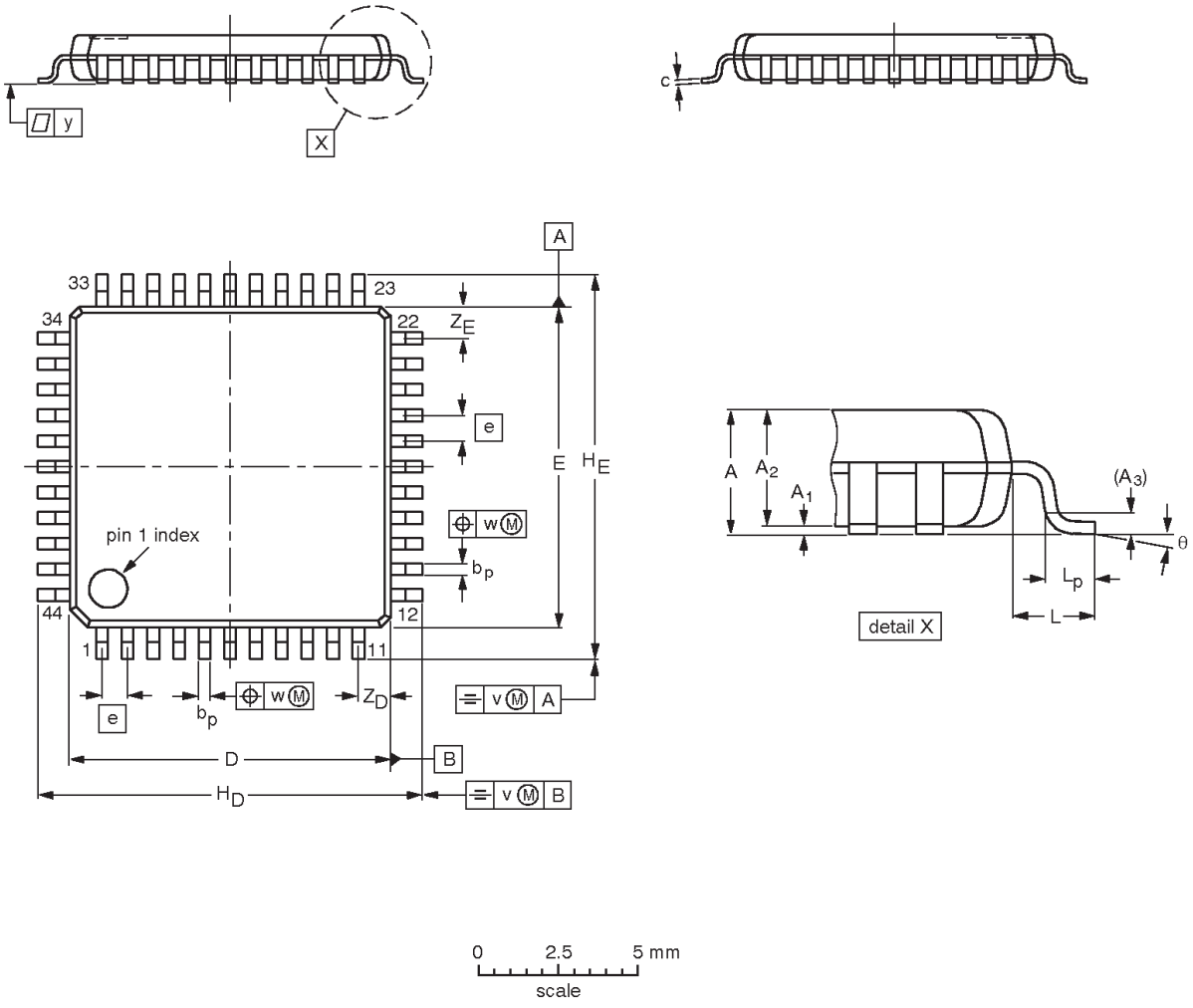
64 bytes of encryption array are initially unprogrammed (all 1s).

Table 19. Program Security Bits

PROGRAM LOCK BITS[1, 2]			
	SB1	SB2	PROTECTION DESCRIPTION
1	U	U	No Program Security features enabled. (Code verify will still be encrypted by the Encryption Array if programmed.)
2	P	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on Reset, and further programming of the EPROM is disabled.

NOTES:
1. P – programmed. U – unprogrammed.
2. Any other combination of the security bits is not defined.

LQFP44: plastic low profile quad flat package; 44 leads; body 10 x 10 x 1.4 mm **SOT389-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_D	H_E	L	L_p	v	w	y	$Z_D^{(1)}$	$Z_E^{(1)}$	θ
mm	1.6	0.15 0.05	1.45 1.35	0.25	0.45 0.30	0.20 0.12	10.1 9.9	10.1 9.9	0.8	12.15 11.85	12.15 11.85	1	0.75 0.45	0.2	0.2	0.1	1.14 0.85	1.14 0.85	7^o 0^o

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT389-1	136E08	MS-026				~~00-01-19~~ 02-06-07