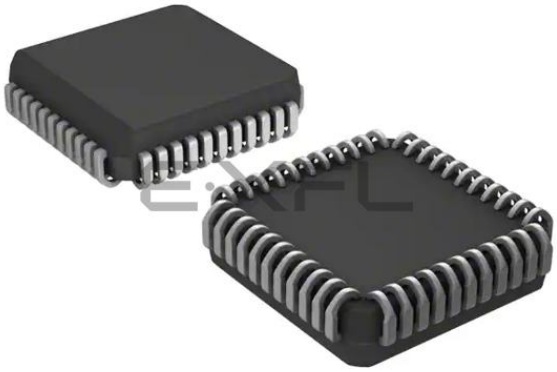


Welcome to E-XFL.COM

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	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°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/p89c660hba-00-512

80C51 8-bit Flash microcontroller family
16KB/32KB/64KB ISP/IAP Flash with 512B/1KB/2KB/8KB RAM

P89C660/P89C662/P89C664/ P89C668

DESCRIPTION

The P89C660/662/664/668 device contains a non-volatile 16KB/32KB/64KB Flash program memory that is both parallel programmable and serial In-System and In-Application Programmable. In-System Programming (ISP) allows the user to download new code while the microcontroller sits in the application. In-Application Programming (IAP) means that the microcontroller fetches new program code and reprograms itself while in the system. This allows for remote programming over a modem link. A default serial loader (boot loader) program in ROM allows serial In-System Programming of the Flash memory via the UART without the need for a loader in the Flash code. For In-Application Programming, the user program erases and reprograms the Flash memory by use of standard routines contained in ROM.

This device executes one instruction in 6 clock cycles, hence providing twice the speed of a conventional 80C51. An OTP configuration bit gives the user the option to select conventional 12-clock timing.

This device is a Single-Chip 8-Bit Microcontroller manufactured in advanced CMOS process and is a derivative of the 80C51 microcontroller family. The instruction set is 100% executing and timing compatible with the 80C51 instruction set.

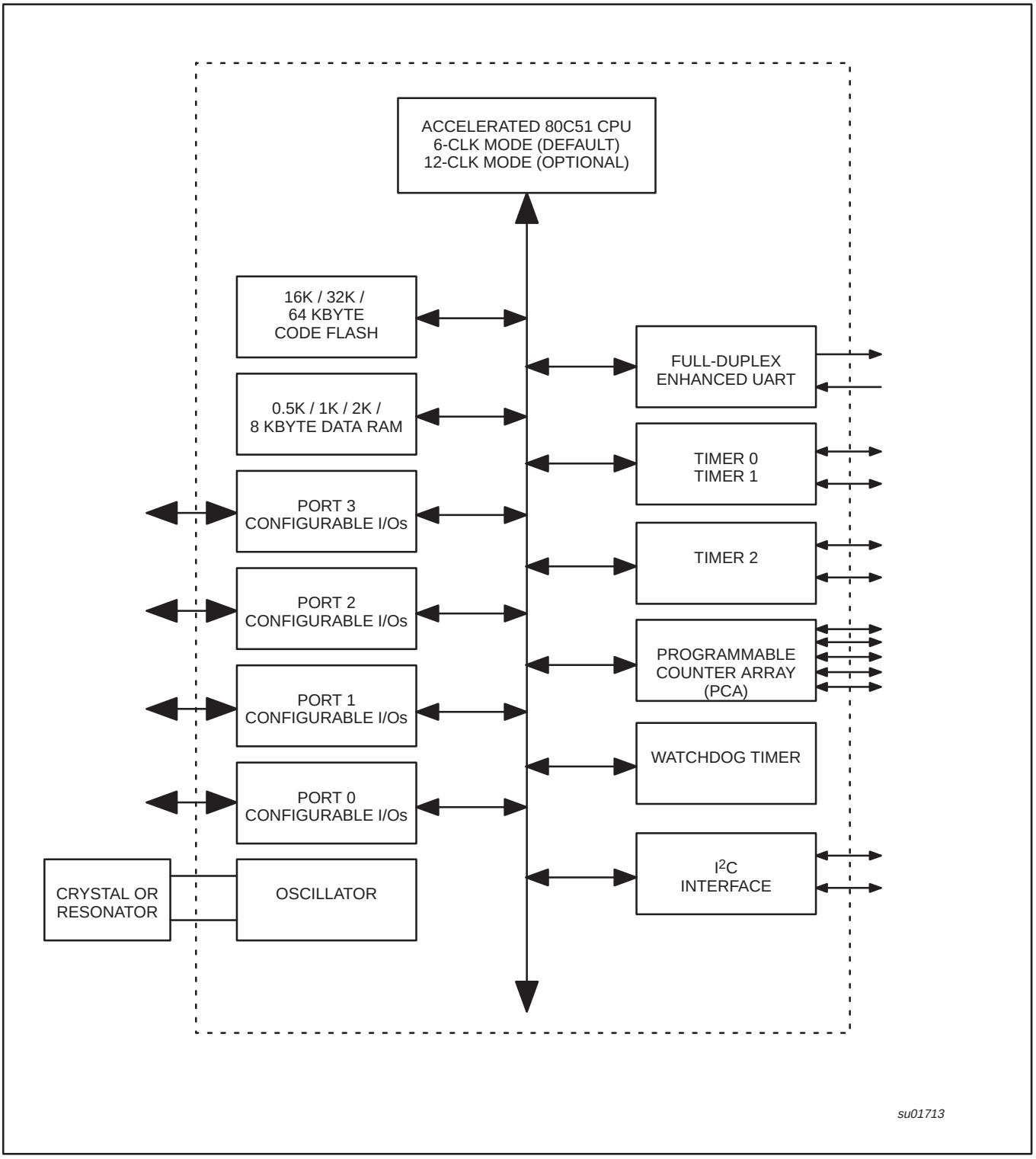
The device also has four 8-bit I/O ports, three 16-bit timer/event counters, a multi-source, four-priority-level, nested interrupt structure, an enhanced UART and on-chip oscillator and timing circuits.

The added features of the P89C660/662/664/668 makes it a powerful microcontroller for applications that require pulse width modulation, high-speed I/O and up/down counting capabilities such as motor control.

FEATURES

- 80C51 Central Processing Unit
- On-chip Flash program memory with In-System Programming (ISP) and In-Application Programming (IAP) capability
- Boot ROM contains low level Flash programming routines for downloading via the UART
- Can be programmed by the end-user application (IAP)
- Parallel programming with 87C51 compatible hardware interface to programmer
- Six clocks per machine cycle operation (standard)
- 12 clocks per machine cycle operation (optional)
- Speed up to 20 MHz with 6 clock cycles per machine cycle (40 MHz equivalent performance); up to 33 MHz with 12 clocks per machine cycle
- Fully static operation
- RAM externally expandable to 64 kbytes
- Four interrupt priority levels
- Eight interrupt sources
- Four 8-bit I/O ports
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Power control modes
  - Clock can be stopped and resumed
  - Idle mode
  - Power-Down mode
- Programmable clock out
- Second DPTR register
- Asynchronous port reset
- Low EMI (inhibit ALE)
- I^2C serial interface
- Programmable Counter Array (PCA)
  - PWM
  - Capture/compare
- Well-suited for IPMI applications

BLOCK DIAGRAM 1



Serial Clock Generator
This programmable clock pulse generator provides the SCL clock pulses when SIO1 is in the Master Transmitter or Master Receiver mode. It is switched off when SIO1 is in a Slave mode. The programmable output clock frequencies are: $f_{OSC}/120$, $f_{OSC}/9600$ (12-clock mode) or $f_{OSC}/60$, $f_{OSC}/4800$ (6-clock mode) and the Timer 1 overflow rate divided by eight. The output clock pulses have a 50% duty cycle unless the clock generator is synchronized with other SCL clock sources as described above.

Timing and Control
The timing and control logic generates the timing and control signals for serial byte handling. This logic block provides the shift pulses for S1DAT, enables the comparator, generates and detects start and stop conditions, receives and transmits acknowledge bits, controls the master and Slave modes, contains interrupt request logic, and monitors the I^2C bus status.

Control Register, S1CON
This 7-bit special function register is used by the microcontroller to control the following SIO1 functions: start and restart of a serial transfer, termination of a serial transfer, bit rate, address recognition, and acknowledgment.

Status Decoder and Status Register
The status decoder takes all of the internal status bits and compresses them into a 5-bit code. This code is unique for each I^2C bus status. The 5-bit code may be used to generate vector addresses for fast processing of the various service routines. Each service routine processes a particular bus status. There are 26 possible bus states if all four modes of SIO1 are used. The 5-bit status code is latched into the five most significant bits of the status register when the serial interrupt flag is set (by hardware) and remains stable until the interrupt flag is cleared by software. The three least significant bits of the status register are always zero. If the status code is used as a vector to service routines, then the routines are displaced by eight address locations. Eight bytes of code is sufficient for most of the service routines.

The Four SIO1 Special Function Registers

The microcontroller interfaces to SIO1 via four special function registers. These four SFRs (S1ADR, S1DAT, S1CON, and S1STA) are described individually in the following sections.

The Address Register, S1ADR
The CPU can read from and write to this 8-bit, directly addressable SFR. S1ADR is not affected by the SIO1 hardware. The contents of this register are irrelevant when SIO1 is in a Master mode. In the Slave modes, the seven most significant bits must be loaded with the microcontroller's own slave address, and, if the least significant bit is set, the general call address (00H) is recognized; otherwise it is ignored.

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

The most significant bit corresponds to the first bit received from the I^2C bus after a start condition. A logic 1 in S1ADR corresponds to a high level on the I^2C bus, and a logic 0 corresponds to a low level on the bus.

The Data Register, S1DAT
S1DAT contains a byte of serial data to be transmitted or a byte which has just been received. The CPU can read from and write to this 8-bit, directly addressable SFR while it is not in the process of shifting a byte. This occurs when SIO1 is in a defined state and the serial interrupt flag is set. Data in S1DAT remains stable as long as SI is set. Data in S1DAT is always shifted from right to left: the first bit to be transmitted is the MSB (bit 7), and, after a byte has been received, the first bit of received data is located at the MSB of S1DAT. While data is being shifted out, data on the bus is simultaneously being shifted in; S1DAT always contains the last data byte present on the bus. Thus, in the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in S1DAT.

	7	6	5	4	3	2	1	0
S1DAT (DAH)	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
	shift direction							

SD7 - SD0:
Eight bits to be transmitted or just received. A logic 1 in S1DAT corresponds to a high level on the I^2C bus, and a logic 0 corresponds to a low level on the bus. Serial data shifts through S1DAT from right to left. Figure 6 shows how data in S1DAT is serially transferred to and from the SDA line.

S1DAT and the ACK flag form a 9-bit shift register which shifts in or shifts out an 8-bit byte, followed by an acknowledge bit. The ACK flag is controlled by the SIO1 hardware and cannot be accessed by the CPU. Serial data is shifted through the ACK flag into S1DAT on the rising edges of serial clock pulses on the SCL line. When a byte has been shifted into S1DAT, the serial data is available in S1DAT, and the acknowledge bit is returned by the control logic during the ninth clock pulse. Serial data is shifted out from S1DAT via a buffer (BSD7) on the falling edges of clock pulses on the SCL line.

When the CPU writes to S1DAT, BSD7 is loaded with the content of S1DAT.7, which is the first bit to be transmitted to the SDA line (see Figure 7). After nine serial clock pulses, the eight bits in S1DAT will have been transmitted to the SDA line, and the acknowledge bit will be present in ACK. Note that the eight transmitted bits are shifted back into S1DAT.

The Control Register, S1CON
The CPU can read from and write to this 8-bit, directly addressable SFR. Two bits are affected by the SIO1 hardware: the SI bit is set when a serial interrupt is requested, and the STO bit is cleared when a STOP condition is present on the I^2C bus. The STO bit is also cleared when ENS1 = "0".

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

ENS1, the SIO1 Enable Bit: ENS1 = "0": When ENS1 is "0", the SDA and SCL outputs are in a high impedance state. SDA and SCL input signals are ignored, SIO1 is in the "not addressed" slave state, and the STO bit in S1CON is forced to "0". No other bits are affected. P1.6 and P1.7 may be used as open drain I/O ports.

ENS1 = "1": When ENS1 is "1", SIO1 is enabled. The P1.6 and P1.7 port latches must be set to logic 1.

ENS1 should not be used to temporarily release SIO1 from the I2C bus since, when ENS1 is reset, the I2C bus status is lost. The AA flag should be used instead (see description of the AA flag in the following text).

If the STA and STO bits are both set, the a STOP condition is transmitted to the I^2C bus if SIO1 is in a Master mode (in a Slave mode, SIO1 generates an internal STOP condition which is not transmitted). SIO1 then transmits a START condition.

STO = “0”: When the STO bit is reset, no STOP condition will be generated.

The Serial Interrupt Flag, SI: SI = “1”: When the SI flag is set, then, if the EA and ES1 (interrupt enable register) bits are also set, a serial interrupt is requested. SI is set by hardware when one of 25 of the 26 possible SIO1 states is entered. The only state that does not cause SI to be set is state F8H, which indicates that no relevant state information is available.

While SI is set, the low period of the serial clock on the SCL line is stretched, and the serial transfer is suspended. A high level on the SCL line is unaffected by the serial interrupt flag. SI must be reset by software.

SI = “0”: When the SI flag is reset, no serial interrupt is requested, and there is no stretching of the serial clock on the SCL line.

The Assert Acknowledge Flag, AA: AA = “1”: If the AA flag is set, an acknowledge (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when:
– The “own slave address” has been received
– The general call address has been received while the general call bit (GC) in S1ADR is set
– A data byte has been received while SIO1 is in the Master Receiver mode
– A data byte has been received while SIO1 is in the addressed Slave Receiver mode

AA = “0”: if the AA flag is reset, a not acknowledge (high level to SDA) will be returned during the acknowledge clock pulse on SCL when:
– A data has been received while SIO1 is in the Master Receiver mode
– A data byte has been received while SIO1 is in the addressed Slave Receiver mode

When SIO1 is in the addressed Slave Transmitter mode, state C8H will be entered after the last serial is transmitted (see Figure 11).

When SI is cleared, SIO1 leaves state C8H, enters the not addressed Slave Receiver mode, and the SDA line remains at a high level. In state C8H, the AA flag can be set again for future address recognition.

When SIO1 is in the not addressed Slave mode, its own slave address and the general call address are ignored. Consequently, no acknowledge is returned, and a serial interrupt is not requested. Thus, SIO1 can be temporarily released from the I^2C bus while the bus status is monitored. While SIO1 is released from the bus, START and STOP conditions are detected, and serial data is shifted in. Address recognition can be resumed at any time by setting the AA flag. If the AA flag is set when the part’s own Slave address or the general call address has been partly received, the address will be recognized at the end of the byte transmission.

The Clock Rate Bits CR0, CR1, and CR2: These three bits determine the serial clock frequency when SIO1 is in a Master mode. The various serial rates are shown in Table 3.

A 12.5 kHz bit rate may be used by devices that interface to the I^2C bus via standard I/O port lines which are software driven and slow. 100 kHz is usually the maximum bit rate and can be derived from a 16 MHz, 12 MHz, or a 6 MHz oscillator. A variable bit rate (0.5 kHz to 62.5 kHz) may also be used if Timer 1 is not required for any other purpose while SIO1 is in a Master mode.

The frequencies shown in Table 3 are unimportant when SIO1 is in a Slave mode. In the Slave modes, SIO1 will automatically synchronize with any clock frequency up to 100 kHz.

The Status Register, S1STA

S1STA is an 8-bit read-only special function register. The three least significant bits are always zero. The five most significant bits contain the status code. There are 26 possible status codes. When S1STA contains F8H, no relevant state information is available and no serial interrupt is requested. All other S1STA values correspond to defined SIO1 states. When each of these states is entered, a serial interrupt is requested (SI = “1”). A valid status code is present in S1STA one machine cycle after SI is set by hardware and is still present one machine cycle after SI has been reset by software.

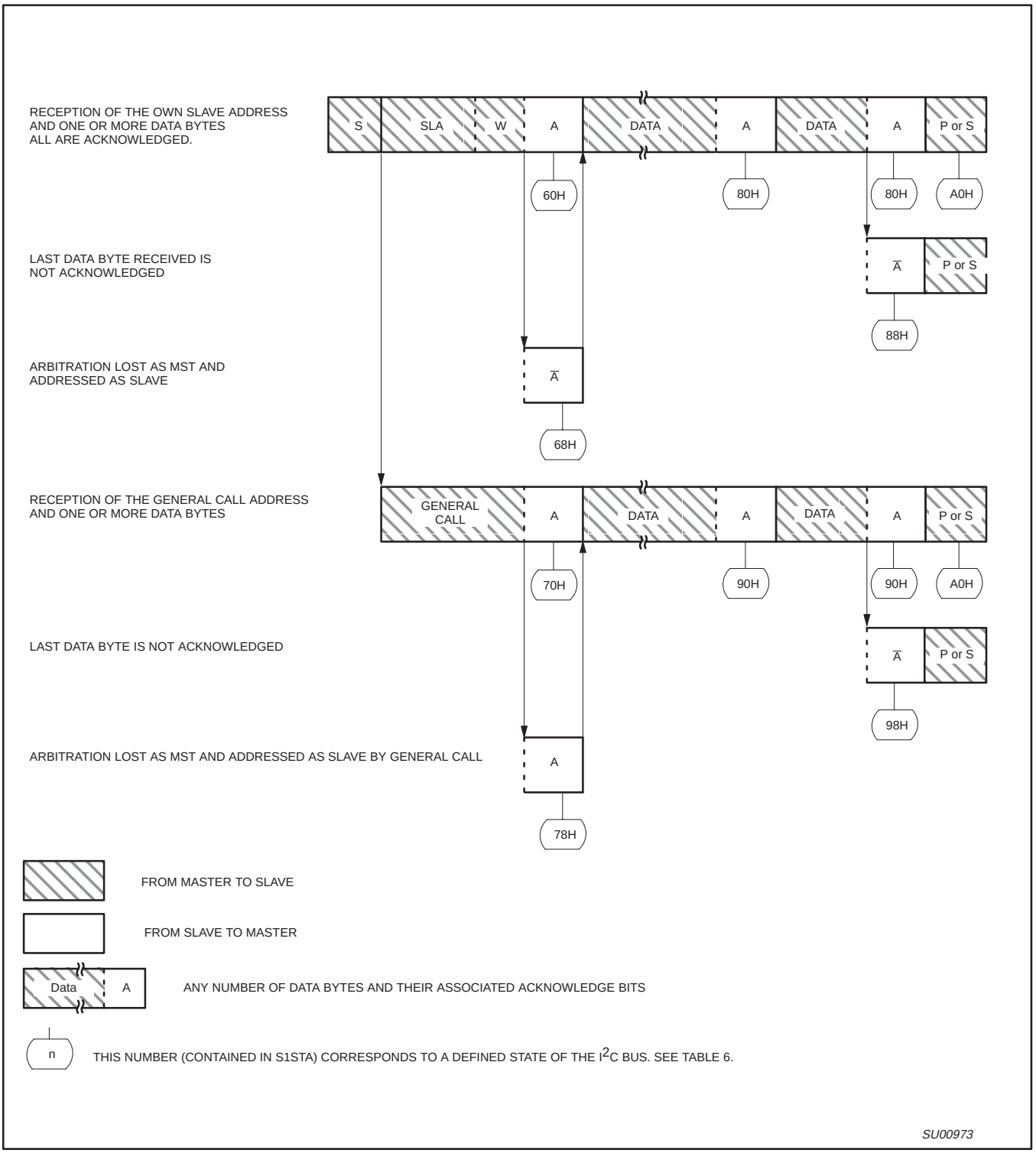


Figure 10. Format and States in the Slave Receiver mode

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 5. Master Receiver 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+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	I^2C 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 6. Slave Receiver mode (Continued)

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	
A0H	A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX	No STDAT action or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address
		No STDAT action or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1
		No STDAT action or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free
		No STDAT action	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.

Table 7. Slave 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	
A8H	Own SLA+R has been received; ACK has been returned	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received
		load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received
B0H	Arbitration lost in SLA+R/$\overline{W}$ as master; Own SLA+R has been received, ACK has been returned	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received
		load data byte	X	0	0	1	Data byte will be transmitted; ACK bit will be received
B8H	Data byte in S1DAT has been transmitted; ACK has been received	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received
		load data byte	X	0	0	1	Data byte will be transmitted; ACK bit will be received
C0H	Data byte in S1DAT has been transmitted; NOT ACK has been received	No S1DAT action or	0	0	0	01	Switched to not addressed SLV mode; no recognition of own SLA or General call address
		no S1DAT action or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1
		no S1DAT action or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free
		no S1DAT action	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.
C8H	Last data byte in S1DAT has been transmitted (AA = 0); ACK has been received	No S1DAT action or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address
		no S1DAT action or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1
		no S1DAT action or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free
		no S1DAT action	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if S1ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.

TIMER 0 AND TIMER 1 OPERATION

Timer 0 and Timer 1

The "Timer" or "Counter" function is selected by control bits C/$\overline{T}$ in the Special Function Register TMOD (see Figure 15). These two Timer/Counters have four operating modes, which are selected by bit-pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both Timers/Counters. Mode 3 is different. The four operating modes are described in the following text.

Mode 0

Putting either Timer into Mode 0 makes it behave as an 8048 Timer, which is an 8-bit Counter with a divide-by-32 prescaler. Figure 16 shows the Mode 0 operation.

In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the Timer interrupt flag TFn. The counted input is enabled to the Timer when TRn = 1 and either GATE = 0 or $\overline{INTn}$ = 1. TRn is a control bit in the Special Function Register TCON (Figure 17). (Setting GATE = 1 allows the Timer to be controlled by external input $\overline{INTn}$, to facilitate pulse width measurements).

The 13-bit register consists of all 8 bits of THn and the lower 5 bits of TLn. The upper 3 bits of TLn are indeterminate and should be ignored. Setting the run flag (TRn) does not clear the registers.

Mode 0 operation is the same for Timer 0 as for Timer 1. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

Mode 1

Mode 1 is the same as Mode 0, except that the Timer register is being run with all 16 bits.

Mode 2

Mode 2 configures the Timer register as an 8-bit Counter (TLn) with automatic reload, as shown in Figure 18. Overflow from TLn not only sets TFn, but also reloads TLn with the contents of THn, which is preset by software. The reload leaves THn unchanged.

Mode 2 operation is the same for Timer 0 as for Timer 1.

Mode 3

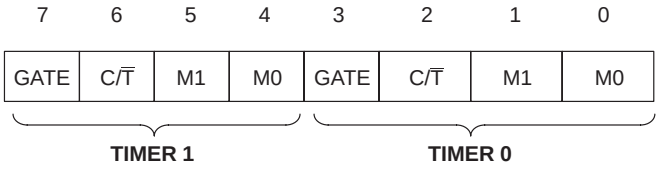
Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0.

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in Figure 19. TL0 uses the Timer 0 control bits: C/$\overline{T}$, GATE, TR0, and TF0 as well as pin $\overline{INT0}$. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the "Timer 1" interrupt.

Mode 3 is provided for applications requiring an extra 8-bit timer on the counter. Putting Timer 0 in Mode 3 allows an 80C51 to have three Timer/Counters. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3, or can still be used by the serial port as a baud rate generator, or in fact, in any application not requiring an interrupt.

TMOD Address = 89H Reset Value = 00H

Not Bit Addressable



BIT	SYMBOL	FUNCTION
TMOD.3/ TMOD.7	GATE	Gating control when set. Timer/Counter "n" is enabled only while "$\overline{INTn}$" pin is high and "TRn" control pin is set. when cleared Timer "n" is enabled whenever "TRn" control bit is set.
TMOD.2/ TMOD.6	C/$\overline{T}$	Timer or Counter Selector cleared for Timer operation (input from internal system clock.) Set for Counter operation (input from "Tn" input pin).

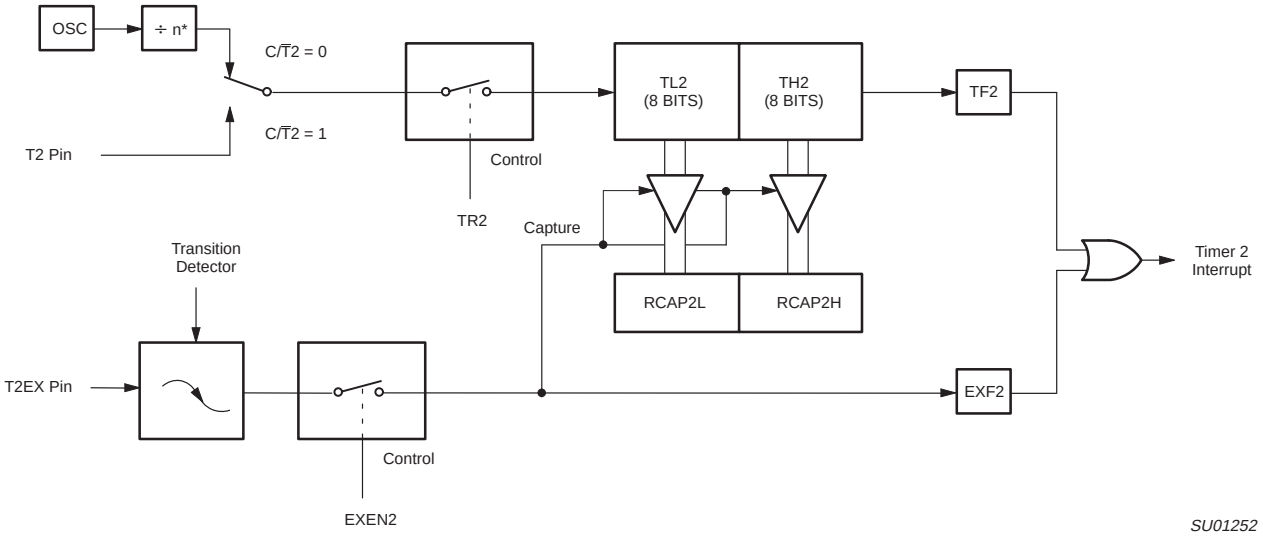
M1	M0	OPERATING
0	0	8048 Timer: "TLn" serves as 5-bit prescaler.
0	1	16-bit Timer/Counter: "THn" and "TLn" are cascaded; there is no prescaler.
1	0	8-bit auto-reload Timer/Counter: "THn" holds a value which is to be reloaded into "TLn" each time it overflows.
1	1	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits.
1	1	(Timer 1) Timer/Counter 1 stopped.

SU01580

Figure 15. Timer/Counter 0/1 Mode Control (TMOD) Register

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

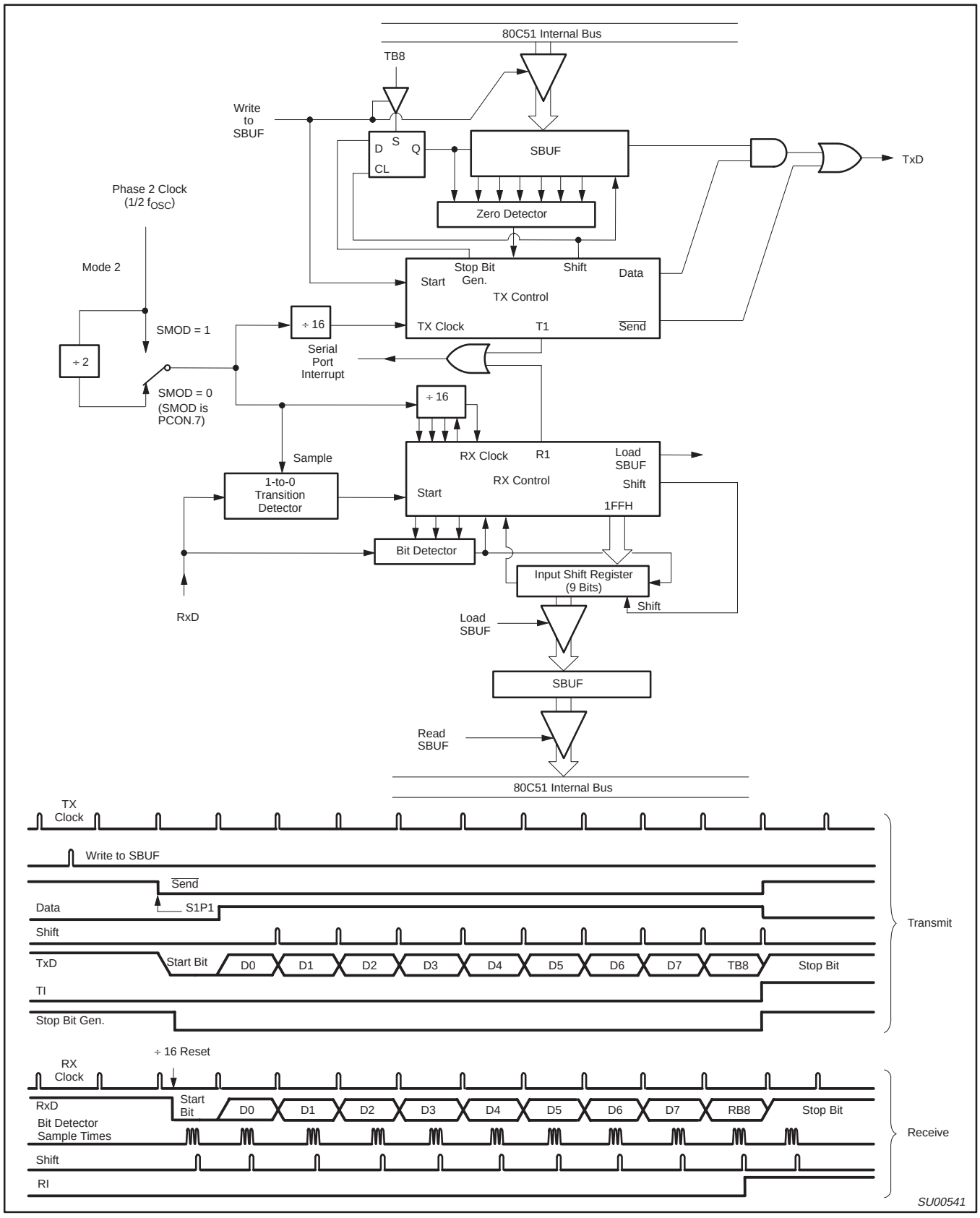


Figure 30. Serial Port Mode 2

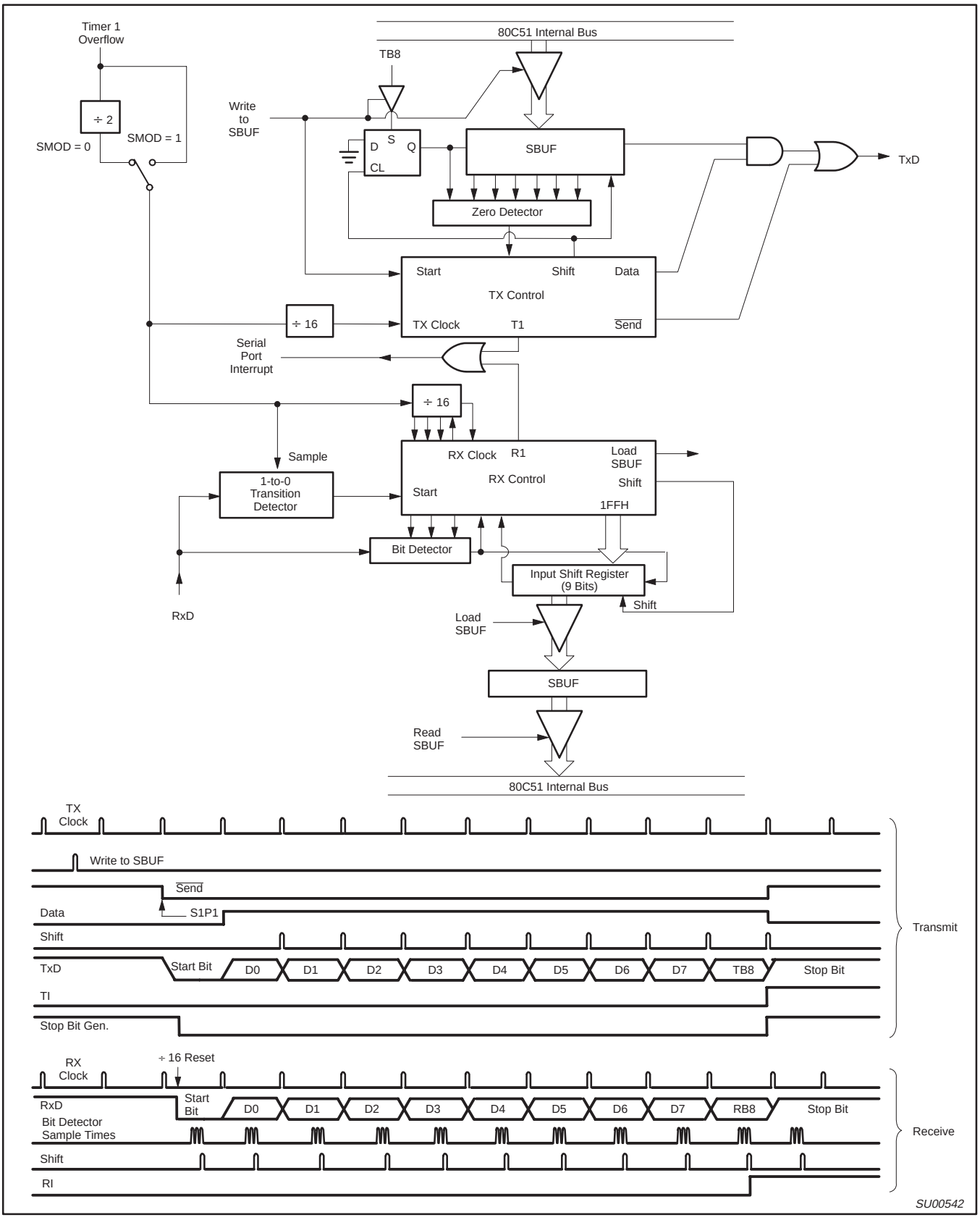


Figure 31. Serial Port Mode 3

Enhanced UART

In addition to the standard operation, the UART can perform framing error detect by looking for missing stop bits, and automatic address recognition. The UART also fully supports multiprocessor communication as does the standard 80C51 UART.

When used for framing error detect, the UART looks for missing stop bits in the communication. A missing bit will set the FE bit in the S0CON register. The FE bit shares the S0CON.7 bit with SM0, and the function of S0CON.7 is determined by PCON.6 (SMOD0) (see Figure 32). If SMOD0 is set then S0CON.7 functions as FE. S0CON.7 functions as SM0 when SMOD0 is cleared. When used as FE, S0CON.7 can only be cleared by software (refer to Figure 33).

Automatic Address Recognition

Automatic Address Recognition is a feature which allows the UART to recognize certain addresses in the serial bit stream by using hardware to make the comparisons. This feature saves a great deal of software overhead by eliminating the need for the software to examine every serial address which passes by the serial port. This feature is enabled by setting the SM2 bit in S0CON. In the 9-bit UART modes (mode 2 and mode 3), the Receive Interrupt flag (RI) will be automatically set when the received byte contains either the “Given” address or the “Broadcast” address. The 9-bit mode requires that the 9th information bit is a 1 to indicate that the received information is an address and not data. Automatic address recognition is shown in Figure 34.

The 8-bit mode is called Mode 1. In this mode, the RI flag will be set if SM2 is enabled and the information received has a valid stop bit following the 8 address bits, and the information is either a Given or Broadcast address.

Mode 0 is the Shift Register mode and SM2 is ignored.

Using the Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the Given slave address or addresses. All of the slaves may be contacted by using the Broadcast address. Two special Function Registers are used to define the slave’s address, SADDR, and the address mask, SADEN. SADEN is used to define which bits in the SADDR are to be used and which bits are “don’t care”. The SADEN mask can be logically ANDed with the SADDR to create the “Given” address which the master will use for addressing each of the slaves. Use of the Given address allows multiple slaves to be recognized while excluding others. The following examples will help to show the versatility of this scheme:

Slave 0	SADDR	=	1100 0000
	SADEN	=	1111 1101
	Given	=	1100 00X0
Slave 1	SADDR	=	1100 0000
	SADEN	=	1111 1110
	Given	=	1100 000X

In the above example SADDR is the same and the SADEN data is used to differentiate between the two slaves. Slave 0 requires a 0 in bit 0 and it ignores bit 1. Slave 1 requires a 0 in bit 1 and bit 0 is ignored. A unique address for Slave 0 would be 1100 0010 since slave 1 requires a 0 in bit 1. A unique address for slave 1 would be 1100 0001 since a 1 in bit 0 will exclude slave 0. Both slaves can be selected at the same time by an address which has bit 0 = 0 (for slave 0) and bit 1 = 0 (for slave 1). Thus, both could be addressed with 1100 0000.

In a more complex system the following could be used to select slaves 1 and 2 while excluding slave 0:

Slave 0	SADDR	=	1100 0000
	SADEN	=	1111 1001
	Given	=	1100 0XX0
Slave 1	SADDR	=	1110 0000
	SADEN	=	1111 1010
	Given	=	1110 0X0X
Slave 2	SADDR	=	1110 0000
	SADEN	=	1111 1100
	Given	=	1110 00XX

In the above example the differentiation among the 3 slaves is in the lower 3 address bits. Slave 0 requires that bit 0 = 0 and it can be uniquely addressed by 1110 0110. Slave 1 requires that bit 1 = 0 and it can be uniquely addressed by 1110 and 0101. Slave 2 requires that bit 2 = 0 and its unique address is 1110 0011. To select Slaves 0 and 1 and exclude Slave 2 use address 1110 0100, since it is necessary to make bit 2 = 1 to exclude slave 2.

The Broadcast Address for each slave is created by taking the logical OR of SADDR and SADEN. Zeros in this result are trended as don’t-cares. In most cases, interpreting the don’t-cares as ones, the broadcast address will be FF hexadecimal.

Upon reset, SADDR (SFR address 0A9H) and SADEN (SFR address 0B9H) are leaded with 0s. This produces a given address of all “don’t cares” as well as a Broadcast address of all “don’t cares”. This effectively disables the Automatic Addressing mode and allows the microcontroller to use standard 80C51 type UART drivers which do not make use of this feature.

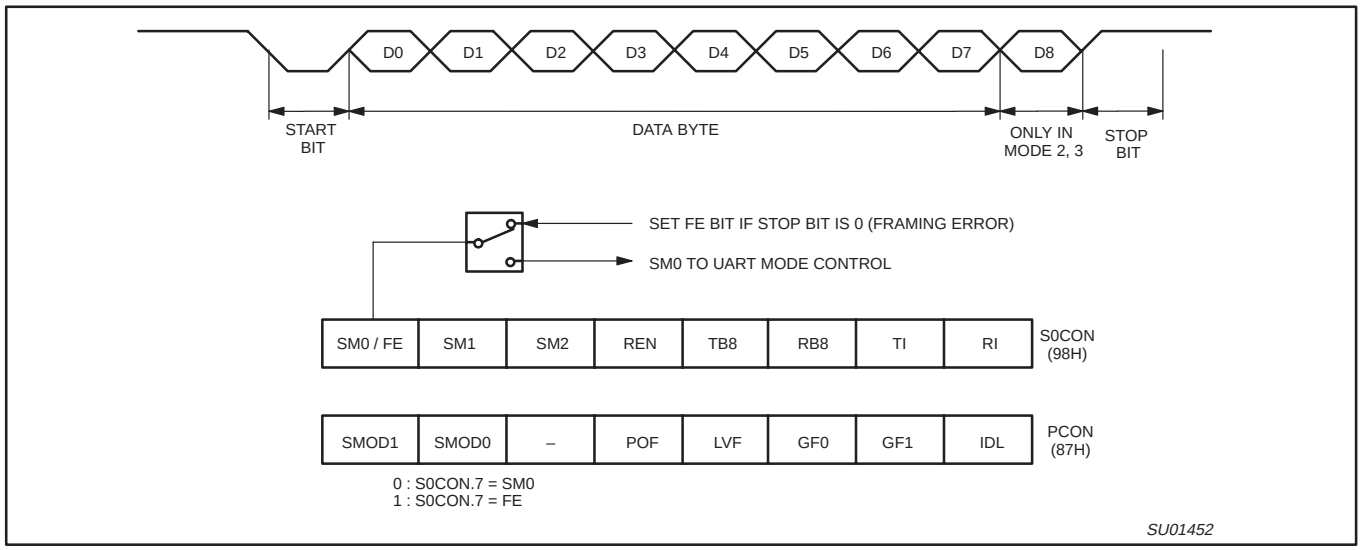


Figure 33. UART Framing Error Detection

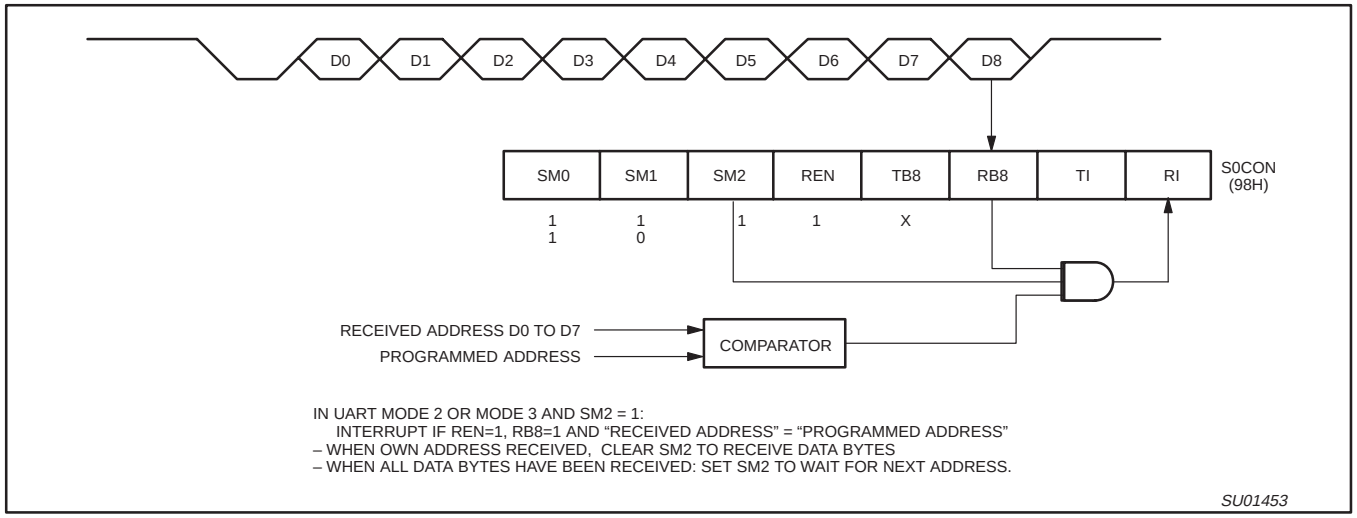


Figure 34. UART Multiprocessor Communication, Automatic Address Recognition

Interrupt Priority Structure

The P89C660/662/664/668 has an 8 source four-level interrupt structure (see Table 13).

There are 4 SFRs associated with the four-level interrupt. They are the IE, IP, IEN1, and IPH (see Figures 35, 36, 37, and 38). The IPH (Interrupt Priority High) register makes the four-level interrupt structure possible. The IPH is located at SFR address B7H. The structure of the IPH register and a description of its bits is shown in Figure 37.

The function of the IPH SFR, when combined with the IP SFR, determines the priority of each interrupt. The priority of each interrupt is determined as shown in the following table:

PRIORITY BITS		INTERRUPT PRIORITY LEVEL
IPH.x	IP.x	
0	0	Level 0 (lowest priority)
0	1	Level 1
1	0	Level 2
1	1	Level 3 (highest priority)

The priority scheme for servicing the interrupts is the same as that for the 80C51, except that there are four interrupt levels rather than two (as on the 80C51). An interrupt will be serviced as long as an interrupt of equal or higher priority is not already being serviced. If an interrupt of equal or higher level priority is being serviced, the new interrupt will wait until it is finished before being serviced. If a lower priority level interrupt is being serviced, it will be stopped and the new interrupt serviced. When the new interrupt is finished, the lower priority level interrupt that was stopped will be completed.

Table 13. Interrupt Table

SOURCE	POLLING PRIORITY	REQUEST BITS	HARDWARE CLEAR?	VECTOR ADDRESS
X0	1	IE0	N (L)[1] Y (T)[2]	03H
SI01 (I^2C)	2	—	N	2BH
T0	3	TP0	Y	0BH
X1	4	IE1	N (L) Y (T)	13H
T1	5	TF1	Y	1BH
SP	6	RI, TI	N	23H
T2	7	TF2, EXF2	N	3BH
PCA	8	CF, CCFn n = 0–4	N	33H

NOTES:
1. L = Level activated
2. T = Transition activated

	7	6	5	4	3	2	1	0
IEN0 (0A8H)	EA	EC	ES1	ES0	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.
Enable Bit = 0 disables it.

BIT	SYMBOL	FUNCTION
IEN0.7	EA	Global disable bit. If EA = 0, all interrupts are disabled. If EA = 1, each interrupt can be individually enabled or disabled by setting or clearing its enable bit.
IEN0.6	EC	PCA interrupt enable bit
IEN0.5	ES1	I^2C interrupt enable bit.
IEN0.4	ES0	Serial Port interrupt enable bit.
IEN0.3	ET1	Timer 1 interrupt enable bit.
IEN0.2	EX1	External interrupt 1 enable bit.
IEN0.1	ET0	Timer 0 interrupt enable bit.
IEN0.0	EX0	External interrupt 0 enable bit.

SU01454

Figure 35. IE Registers

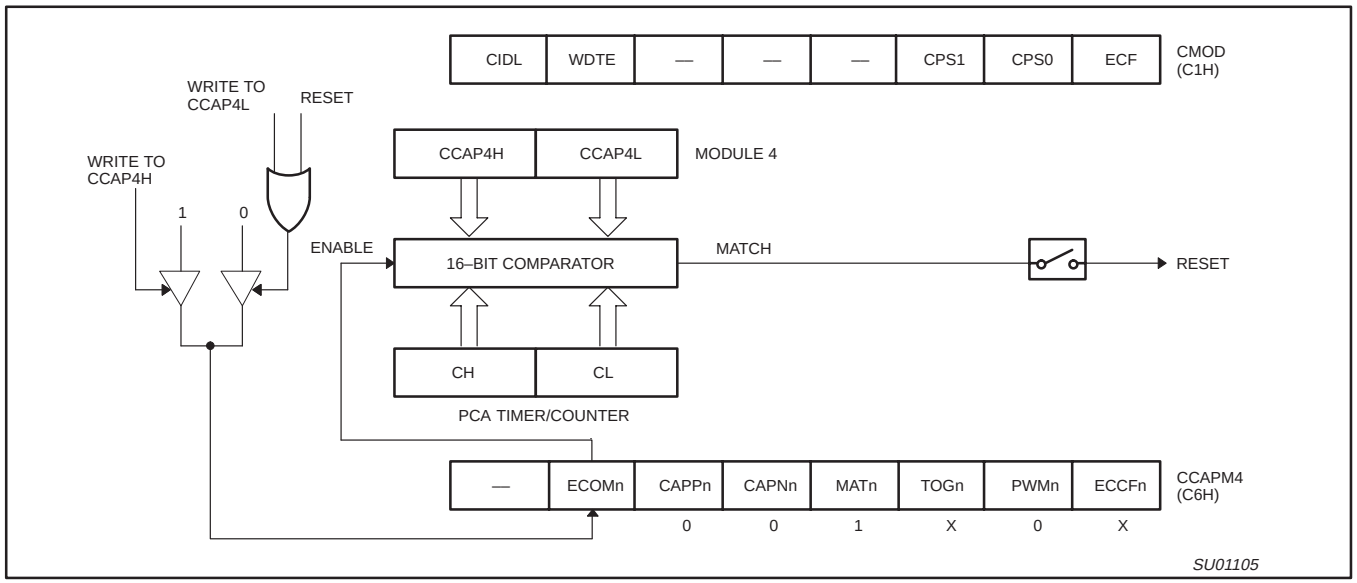


Figure 51. PCA Watchdog Timer m(Module 4 only)

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.

RECORD TYPE	COMMAND/DATA FUNCTION

03	Miscellaneous Write Functions
	`:nnxxxx03ffssddcc`
	Where:
	`nn = number of bytes (hex) in record`
	`xxxx = required field, but value is a “don’t care”`
	`03 = Write Function`
	`ff = subfunction code`
	`ss = selection code`
	`dd = data input (as needed)`
	`cc = checksum`
	
	Subfunction Code = 01 (Erase Blocks)
	`ff = 01`
	`ss = 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`
	Example:
	`:0200000301C03C erase block 4`
	
	Subfunction Code = 04 (Erase Boot Vector and Status Byte)
	`ff = 04`
	`ss = don’t care`
	Example:
	`:020000030400F7 erase boot vector and status byte`
	
	Subfunction Code = 05 (Program Security Bits)
	`ff = 05`
	`ss = 00 program security bit 1 (inhibit writing to Flash)`
	`01 program security bit 2 (inhibit Flash verify)`
	`02 program security bit 3 (disable external memory)`
	Example:
	`:020000030501F5 program security bit 2`
	
	Subfunction Code = 06 (Program Status Byte or Boot Vector)
	`ff = 06`
	`ss = 00 program status byte`
	`01 program boot vector`
	Example:
	`:030000030601FCF7 program boot vector with 0FCH`
	
	Subfunction Code = 07 (Full Chip Erase)
	Erases all blocks, security bits, and sets status and boot vector to default values
	`ff = 07`
	`ss = don’t care`
	`dd = don’t care`
	Example:
	`:0100000307F5 full chip erase`

04	Display Device Data or Blank Check – Record type 04 causes the contents of the entire Flash array to be sent out the serial port in a formatted display. This display consists of an address and the contents of 16 bytes starting with that address. No display of the device contents will occur if security bit 2 has been programmed. Data to the serial port is initiated by the reception of any character and terminated by the reception of any character.
	
	General Format of Function 04
	`:05xxxx04sssseeeeffcc`
	Where:
	`05 = number of bytes (hex) in record`
	`xxxx = required field, but value is a “don’t care”`
	`04 = “Display Device Data or Blank Check” function code`
	`ssss = starting address`
	`eeee = ending address`
	`ff = subfunction`
	`00 = display data`
	`01 = blank check`
	`cc = checksum`
	Example:
	`:0500000440004FFF0069 display 4000–4FFF`

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

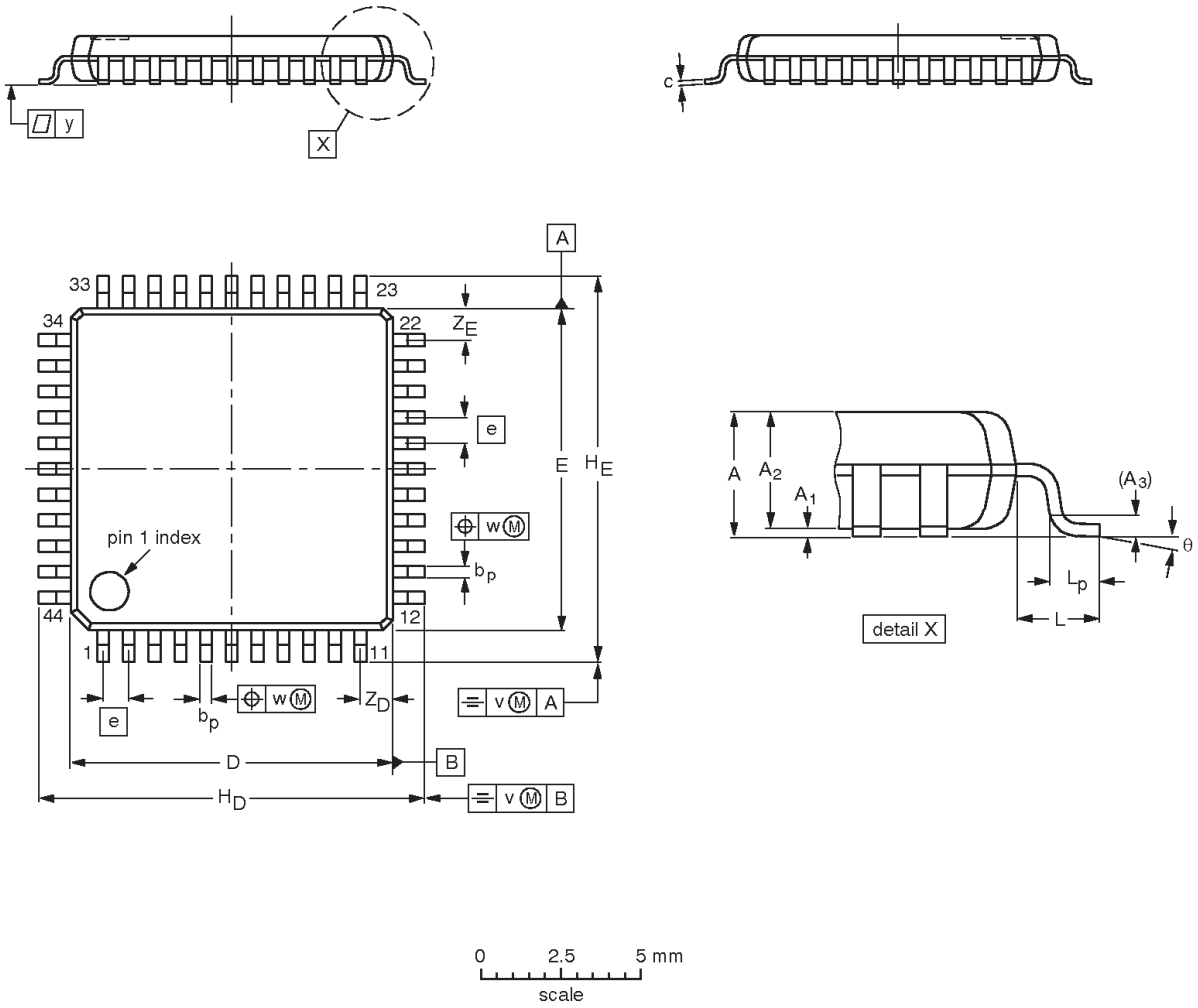
AC ELECTRICAL CHARACTERISTICS (6 CLOCK MODE)

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.

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