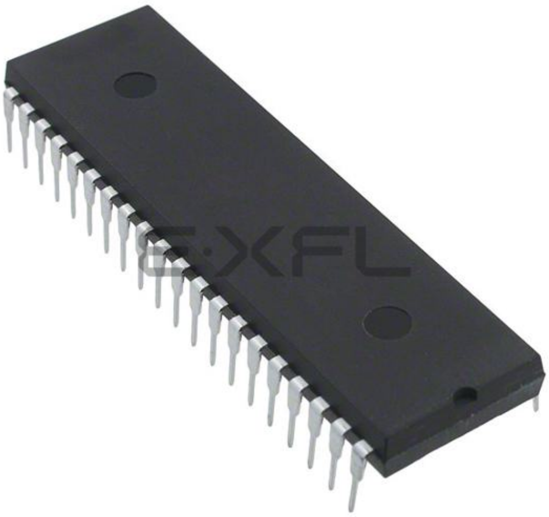


What is "Embedded - Microcontrollers"?

"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	UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	32
Program Memory Size	32KB (32K 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	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-DIP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/p89c51rc2bn-01-112

DESCRIPTION

The P89C51RA2/RB2/RC2/RD2xx contains a non-volatile 8KB/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.

The device supports 6-clock/12-clock mode selection by programming a Flash bit using parallel programming or In-System Programming. In addition, an SFR bit (X2) in the clock control register (CKCON) also selects between 6-clock/12-clock mode.

Additionally, when in 6-clock mode, peripherals may use either 6 clocks per machine cycle or 12 clocks per machine cycle. This choice is available individually for each peripheral and is selected by bits in the CKCON register.

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

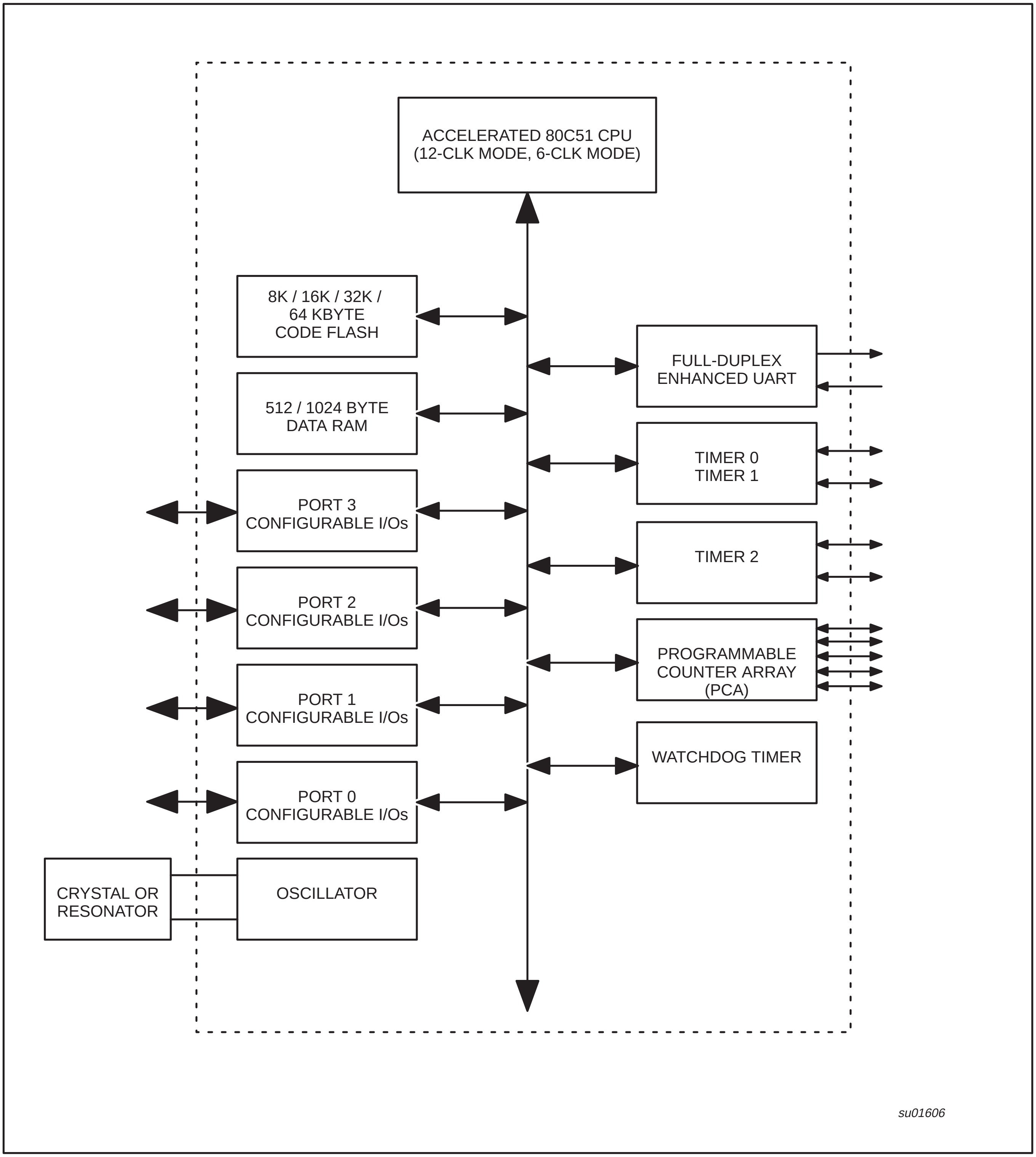
The device also has four 8-bit I/O ports, three 16-bit timer/event counters, a multi-source, four-priority-level, nested interrupt structure, an enhanced UART and on-chip oscillator and timing circuits.

The added features of the P89C51RA2/RB2/RC2/RD2xx make 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
- Supports 6-clock/12-clock mode via parallel programmer (default clock mode after ChipErase is 12-clock)
- 6-clock/12-clock mode Flash bit erasable and programmable via ISP
- 6-clock/12-clock mode programmable “on-the-fly” by SFR bit
- Peripherals (PCA, timers, UART) may use either 6-clock or 12-clock mode while the CPU is in 6-clock mode
- 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 expandable externally to 64 kbytes
- Four interrupt priority levels
- Seven 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 pin
- Second DPTR register
- Asynchronous port reset
- Low EMI (inhibit ALE)
- Programmable Counter Array (PCA)
  - PWM
  - Capture/compare

BLOCK DIAGRAM 1



PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	PDIP	PLCC	LQFP		
V_{SS}	20	22	16	I	**Ground:** 0 V reference.
V_{CC}	40	44	38	I	**Power Supply:** This is the power supply voltage for normal, idle, and power-down operation.
P0.0–0.7	39–32	43–36	37–30	I/O	**Port 0:** Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s.
P1.0–P1.7	1–8	2–9	40–44, 1–3	I/O	**Port 1:** Port 1 is an 8-bit bidirectional I/O port with internal pull-ups on all pins. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I_{IL}).
					Alternate functions for P89C51RA2/RB2/RC2/RD2xx Port 1 include:
	1	2	40	I/O	**T2 (P1.0):** Timer/Counter 2 external count input/Clockout (see Programmable Clock-Out)
	2	3	41	I	**T2EX (P1.1):** Timer/Counter 2 Reload/Capture/Direction Control
	3	4	42	I	**ECI (P1.2):** External Clock Input to the PCA
	4	5	43	I/O	**CEX0 (P1.3):** Capture/Compare External I/O for PCA module 0
	5	6	44	I/O	**CEX1 (P1.4):** Capture/Compare External I/O for PCA module 1
	6	7	1	I/O	**CEX2 (P1.5):** Capture/Compare External I/O for PCA module 2
	7	8	2	I/O	**CEX3 (P1.6):** Capture/Compare External I/O for PCA module 3
	8	9	3	I/O	**CEX4 (P1.7):** Capture/Compare External I/O for PCA module 4
P2.0–P2.7	21–28	24–31	18–25	I/O	**Port 2:** Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: I_{IL}). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register.
P3.0–P3.7	10–17	11, 13–19	5, 7–13	I/O	**Port 3:** Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: I_{IL}). Port 3 also serves the special features of the P89C51RA2/RB2/RC2/RD2xx, as listed below:
	10	11	5	I	**RxD (P3.0):** Serial input port
	11	13	7	O	**TxD (P3.1):** Serial output port
	12	14	8	I	$\overline{\textbf{INT0}}$ **(P3.2):** External interrupt
	13	15	9	I	$\overline{\textbf{INT1}}$ **(P3.3):** External interrupt
	14	16	10	I	**T0 (P3.4):** Timer 0 external input
	15	17	11	I	**T1 (P3.5):** Timer 1 external input
	16	18	12	O	$\overline{\textbf{WR}}$ **(P3.6):** External data memory write strobe
	17	19	13	O	$\overline{\textbf{RD}}$ **(P3.7):** External data memory read strobe
RST	9	10	4	I	**Reset:** A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal resistor to V_{SS} permits a power-on reset using only an external capacitor to V_{CC}.
ALE	30	33	27	O	**Address Latch Enable:** Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted twice every machine cycle, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. ALE can be disabled by setting SFR auxiliary.0. With this bit set, ALE will be active only during a MOVX instruction.

CLOCK CONTROL REGISTER (CKCON)

This device provides control of the 6-clock/12-clock mode by means of both an SFR bit (X2) and a Flash bit (FX2, located in the Security Block). The Flash clock control bit, FX2, when programmed (6-clock mode) supercedes the X2 bit (CKCON.0).

The CKCON register also provides individual control of the clock rates for the peripherals devices. When running in 6-clock mode each peripheral may be individually clocked from either fosc/6 or fosc/12. When in 12-clock mode, all peripheral devices will use fosc/12. The CKCON register is shown below.

CKCON Address = 8Fh Reset Value = x0000000B

Not Bit Addressable

7	6	5	4	3	2	1	0
–	WDX2	PCAX2	SIX2	T2X2	T1X2	T0X2	X2

BIT	**SYMBOL**	**FUNCTION**
CKCON.7	–	Reserved.
CKCON.6	WDX2	Watchdog clock; 0 = 6 clocks for each WDT clock, 1 = 12 clocks for each WDT clock
CKCON.5	PCAX2	PCA clock; 0 = 6 clocks for each PCA clock, 1 = 12 clocks for each PCA clock
CKCON.4	SIX2	UART clock; 0 = 6 clocks for each UART clock, 1 = 12 clocks for each UART clock
CKCON.3	T2X2	Timer2 clock; 0 = 6 clocks for each Timer2 clock, 1 = 12 clocks for each Timer2 clock
CKCON.2	T1X2	Timer1 clock; 0 = 6 clocks for each Timer1 clock, 1 = 12 clocks for each Timer1 clock
CKCON.1	T0X2	Timer0 clock; 0 = 6 clocks for each Timer0 clock, 1 = 12 clocks for each Timer0 clock
CKCON.0	X2	CPU clock; 1 = 6 clocks for each machine cycle, 0 = 12 clocks for each machine cycle

SU01607

Bits 1 through 6 only apply if 6 clocks per machine cycle is chosen (i.e.– Bit 0 = 1). If Bit 0 = 0 (12 clocks per machine cycle) then all peripherals will have 12 clocks per machine cycle as their clock source.

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 peripheral input clock sources.

FX2 clock mode bit	X2	Peripheral clock mode bit (e.g., T0X2)	CPU MODE	Peripheral Clock Rate
erased	0	x	12-clock (default)	12-clock (default)
erased	1	0	6-clock	6-clock
erased	1	1	6-clock	12-clock
programmed	x	0	6-clock	6-clock
programmed	x	1	6-clock	12-clock

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.

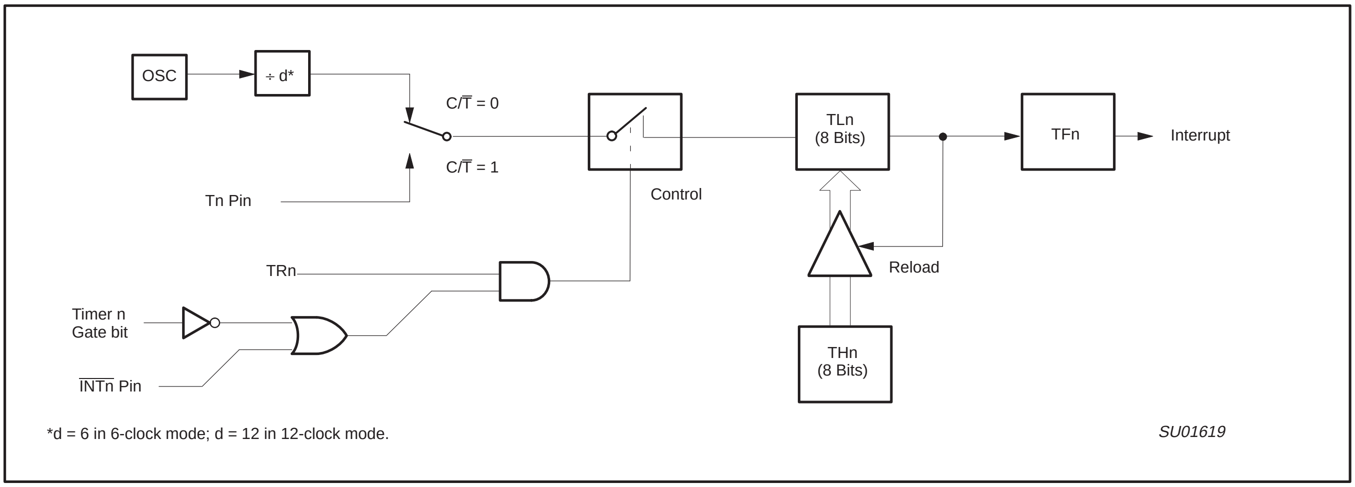


Figure 4. Timer/Counter 0/1 Mode 2: 8-Bit Auto-Reload

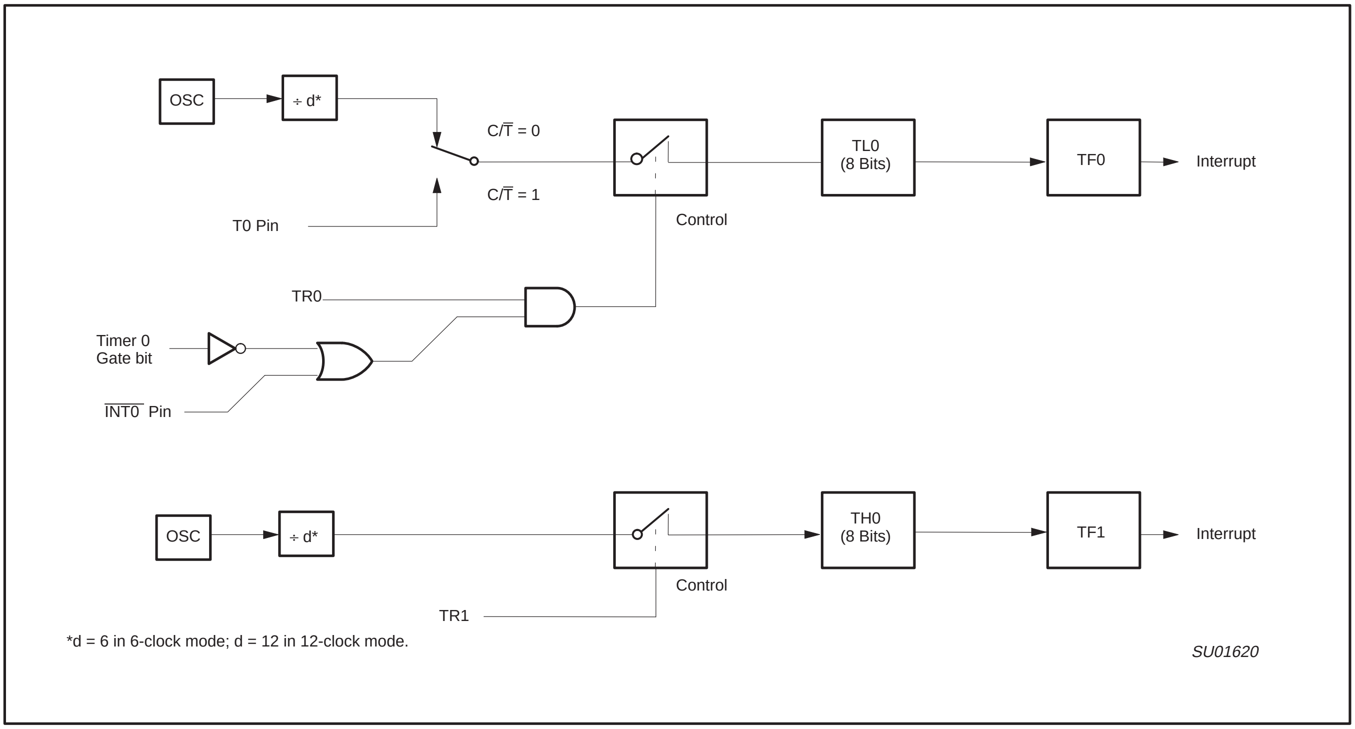
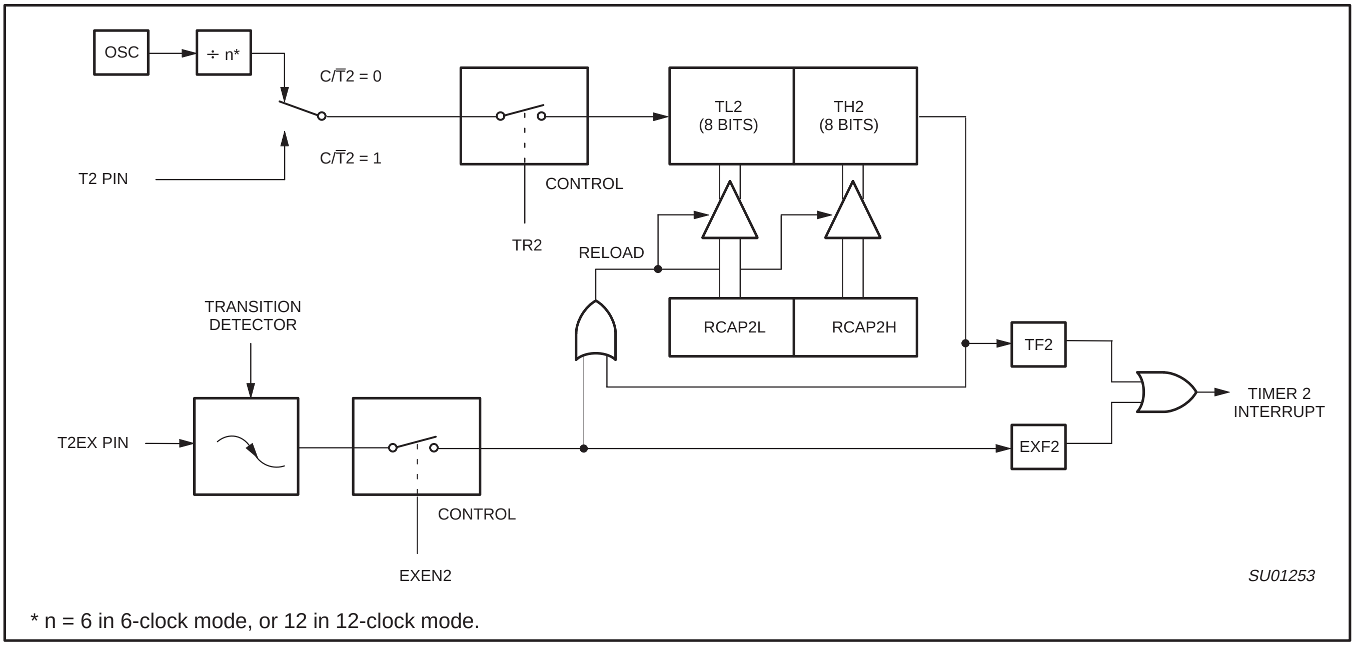
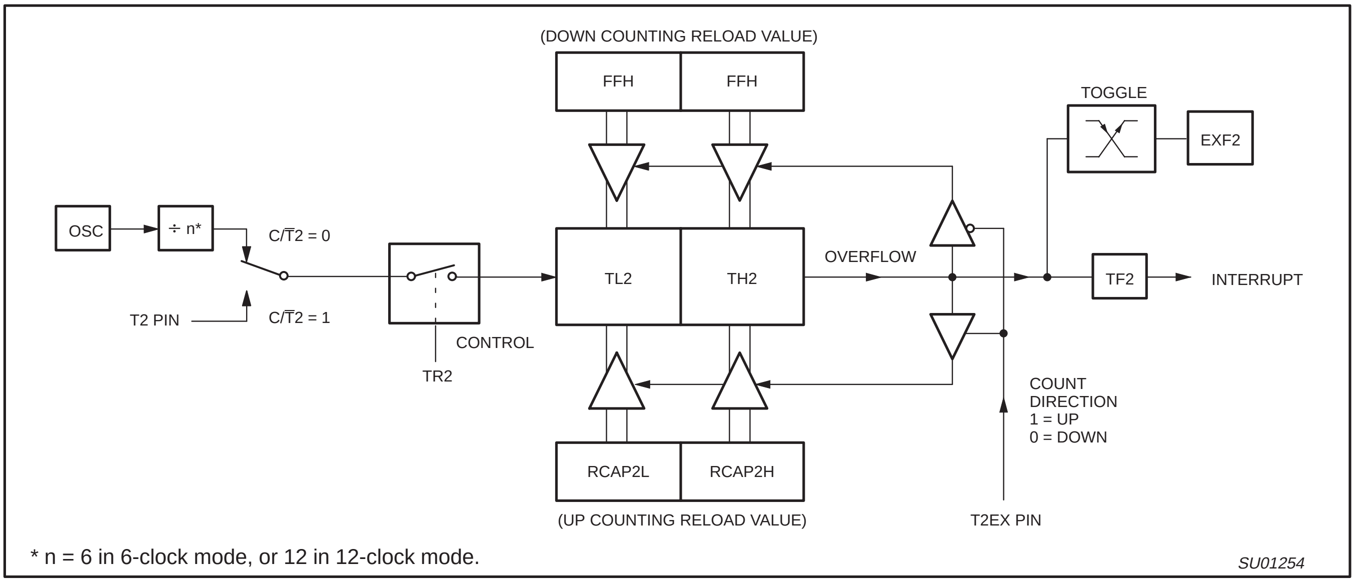


Figure 5. Timer/Counter 0 Mode 3: Two 8-Bit Counters



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

Figure 9. Timer 2 in Auto-Reload Mode (DCEN = 0)



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

Figure 10. Timer 2 Auto Reload Mode (DCEN = 1)

FULL-DUPLEX ENHANCED UART

Standard UART operation

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the register. (However, if the first byte still hasn't been read by the time reception of the second byte is complete, one of the bytes will be lost.) The serial port receive and transmit registers are both accessed at Special Function Register SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

The serial port can operate in 4 modes:

Mode 0: Serial data enters and exits through RxD. TxD outputs the shift clock. 8 bits are transmitted/received (LSB first). The baud rate is fixed at 1/12 the oscillator frequency in 12-clock mode or 1/6 the oscillator frequency in 6-clock mode.

Mode 1: 10 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 Special Function Register SCON. The baud rate is variable.

Mode 2: 11 bits are transmitted (through TxD) or received (through RxD): 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 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) could be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Function Register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency in 12-clock mode or 1/16 or 1/32 the oscillator frequency in 6-clock mode.

Mode 3: 11 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). In fact, Mode 3 is the same as Mode 2 in all respects except baud rate. The baud rate in Mode 3 is variable.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received. The 9th one goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. A way to use this feature in multiprocessor systems is as follows:

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. The slaves that weren't being addressed leave their SM2s set and go on about their business, ignoring the coming data bytes.

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit. In a Mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

Serial Port Control Register

The serial port control and status register is the Special Function Register SCON, shown in Figure 12. This register contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (TI and RI).

Baud Rates

The baud rate in Mode 0 is fixed: Mode 0 Baud Rate = Oscillator Frequency / 12 (12-clock mode) or / 6 (6-clock mode). The baud rate in Mode 2 depends on the value of bit SMOD in Special Function Register PCON. If SMOD = 0 (which is the value on reset), and the port pins in 12-clock mode, the baud rate is 1/64 the oscillator frequency. If SMOD = 1, the baud rate is 1/32 the oscillator frequency. In 6-clock mode, the baud rate is 1/32 or 1/16 the oscillator frequency, respectively.

Mode 2 Baud Rate =

$$\frac{2^{SMOD}}{n} \times (\text{Oscillator Frequency})$$

Where:

n = 64 in 12-clock mode, 32 in 6-clock mode

The baud rates in Modes 1 and 3 are determined by the Timer 1 or Timer 2 overflow rate.

Using Timer 1 to Generate Baud Rates

When Timer 1 is used as the baud rate generator (T2CON.RCLK = 0, T2CON.TCLK = 0), the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD as follows:

Mode 1, 3 Baud Rate =

$$\frac{2^{SMOD}}{n} \times (\text{Timer 1 Overflow Rate})$$

Where:

n = 32 in 12-clock mode, 16 in 6-clock mode

The Timer 1 interrupt should be disabled in this application. The Timer itself can be configured for either "timer" or "counter" operation, and in any of its 3 running modes. In the most typical applications, it is configured for "timer" operation, in the auto-reload mode (high nibble of TMOD = 0010B). In that case the baud rate is given by the formula:

Mode 1, 3 Baud Rate =

$$\frac{2^{SMOD}}{n} \times \frac{\text{Oscillator Frequency}}{12 \times [256–(TH1)]}$$

Where:

n = 32 in 12-clock mode, 16 in 6-clock mode

One can achieve very low baud rates with Timer 1 by leaving the Timer 1 interrupt enabled, and configuring the Timer to run as a 16-bit timer (high nibble of TMOD = 0001B), and using the Timer 1 interrupt to do a 16-bit software reload. Figure 13 lists various commonly used baud rates and how they can be obtained from Timer 1.

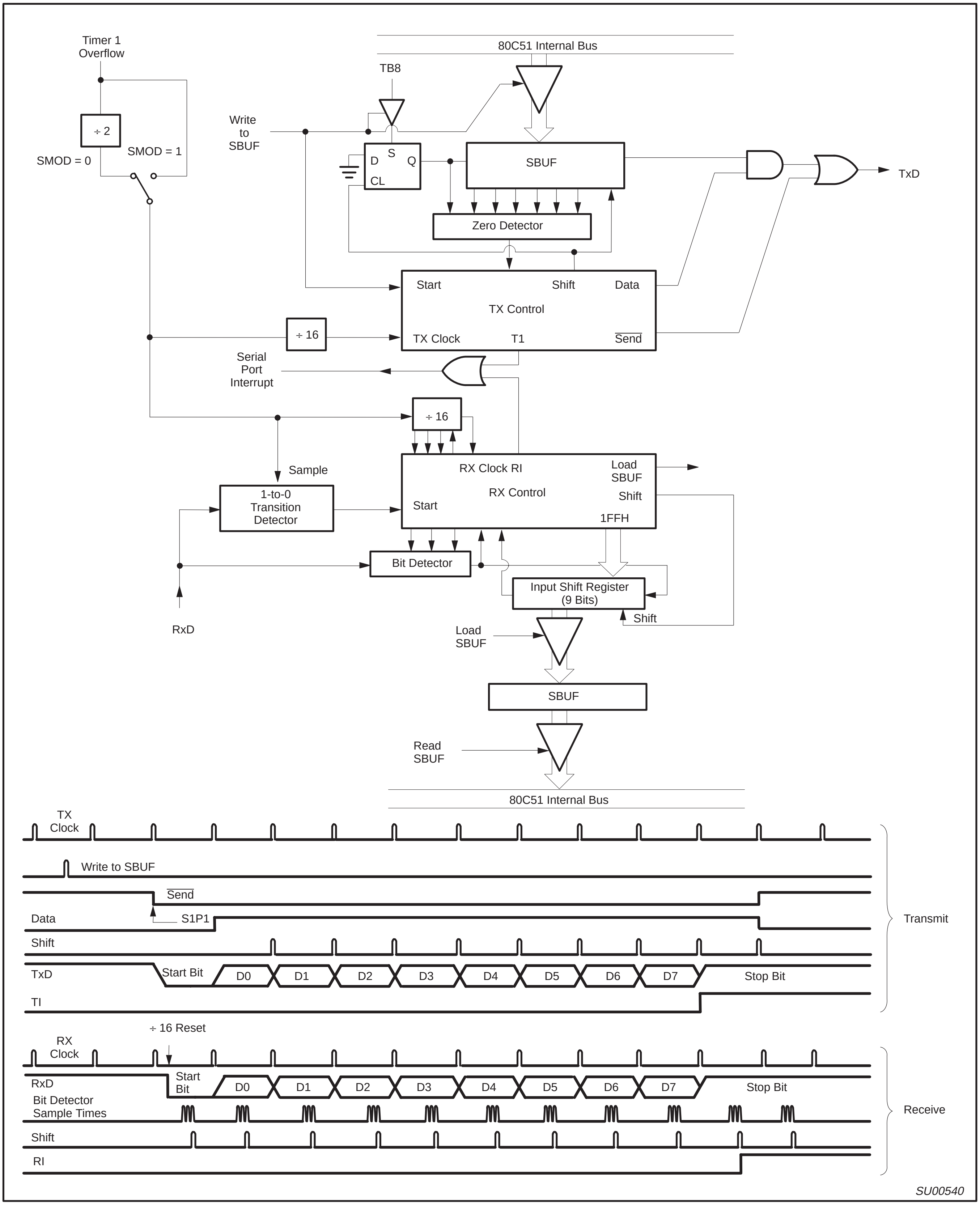


Figure 15. Serial Port Mode 1

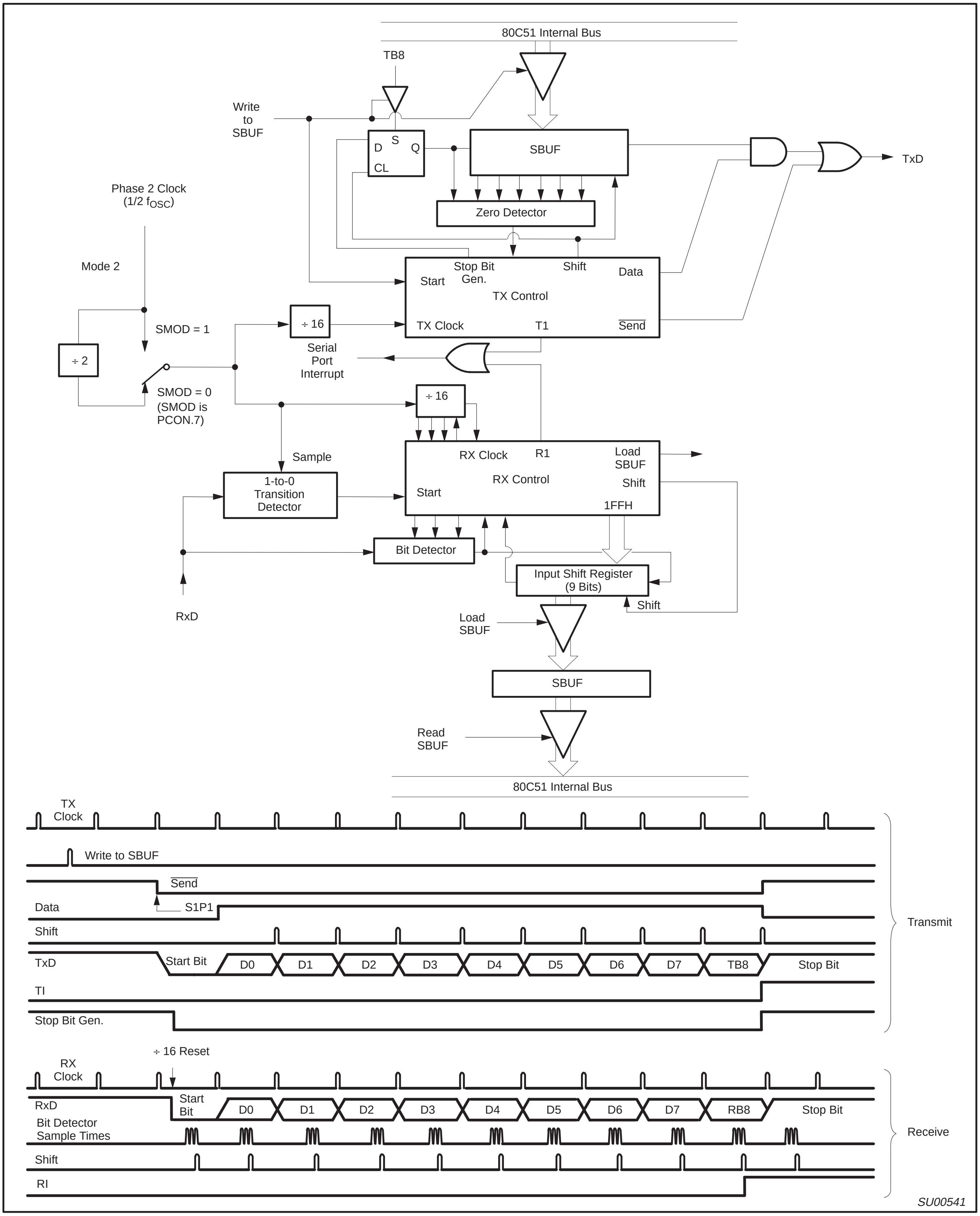


Figure 16. Serial Port Mode 2

SCON Address = 98H Reset Value = 0000 0000B

Bit Addressable

	SM0/FE	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**
Bit:	7	6	5	4	3	2	1	0
	(SMOD0 = 0/1)*							

Symbol	**Function**
FE	Framing Error bit. This bit is set by the receiver when an invalid stop bit is detected. The FE bit is not cleared by valid frames but should be cleared by software. The SMOD0 bit must be set to enable access to the FE bit.
SM0	Serial Port Mode Bit 0, (SMOD0 must = 0 to access bit SM0)
SM1	Serial Port Mode Bit 1

SM0	**SM1**	**Mode**	**Description**	**Baud Rate****
0	0	0	shift register	f_{OSC}/6 (6-clock mode) or f_{OSC}/12 (12-clock mode)
0	1	1	8-bit UART	variable
1	0	2	9-bit UART	f_{OSC}/32 or f_{OSC}/16 (6-clock mode) or f_{OSC}/64 or f_{OSC}/32 (12-clock mode)
1	1	3	9-bit UART	variable

SM2	Enables the Automatic Address Recognition feature in Modes 2 or 3. If SM2 = 1 then RI will not be set unless the received 9th data bit (RB8) is 1, indicating an address, and the received byte is a Given or Broadcast Address. In Mode 1, if SM2 = 1 then RI will not be activated unless a valid stop bit was received, and the received byte is a Given or Broadcast Address. In Mode 0, SM2 should be 0.
REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.
TB8	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.
RB8	In modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.
TI	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
RI	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

NOTE:
*SMOD0 is located at PCON6.
**f_{OSC} = oscillator frequency

SU01255

Figure 18. SCON: Serial Port Control Register

```
INIT_WATCHDOG:
   MOV CCAPM4, #4CH         ; Module 4 in compare mode
   MOV CCAP4L, #0FFH        ; Write to low byte first
   MOV CCAP4H, #0FFH        ; Before PCA timer counts up to
                            ; FFFF Hex, these compare values
                            ; must be changed
   ORL CMOD, #40H           ; Set the WDTE bit to enable the
                            ; watchdog timer without changing
                            ; the other bits in CMOD
;
;********************************************************************
;
; Main program goes here, but CALL WATCHDOG periodically.
;
;********************************************************************
;
WATCHDOG:
   CLR EA                   ; Hold off interrupts
   MOV CCAP4L, #00          ; Next compare value is within
   MOV CCAP4H, CH           ; 255 counts of the current PCA
   SETB EA                  ; timer value
   RET
```

Figure 37. PCA Watchdog Timer Initialization Code

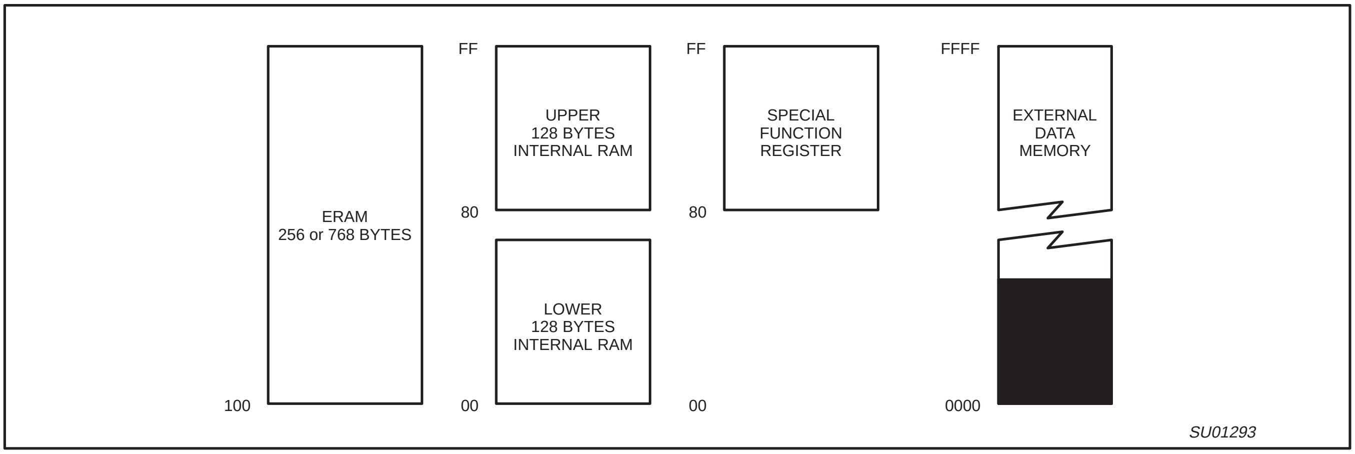


Figure 39. Internal and External Data Memory Address Space with EXTRAM = 0

HARDWARE WATCHDOG TIMER (ONE-TIME ENABLED WITH RESET-OUT FOR P89C51RA2/RB2/RC2/RD2xx)

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.

Hardware Activation of the Boot Loader

The boot loader can also be executed by holding $\overline{\text{PSEN}}$ LOW, $\overline{\text{EA}}$ greater than V_{IH} (such as +5 V), and ALE HIGH (or not connected) at the falling edge of RESET. This is the same effect as having a non-zero status byte. This allows an application to be built that will normally execute the end user's code but can be manually forced into ISP operation.

If the factory default setting for the Boot Vector (0FCH) is changed, it will no longer point to the ISP masked-ROM boot loader code. If this happens, the only way it is possible to change the contents of the Boot Vector is through the parallel programming method, provided that the end user application does not contain a customized loader that provides for erasing and reprogramming of the Boot Vector and Status Byte.

After programming the Flash, the status byte should be programmed to zero in order to allow execution of the user's application code beginning at address 0000H.

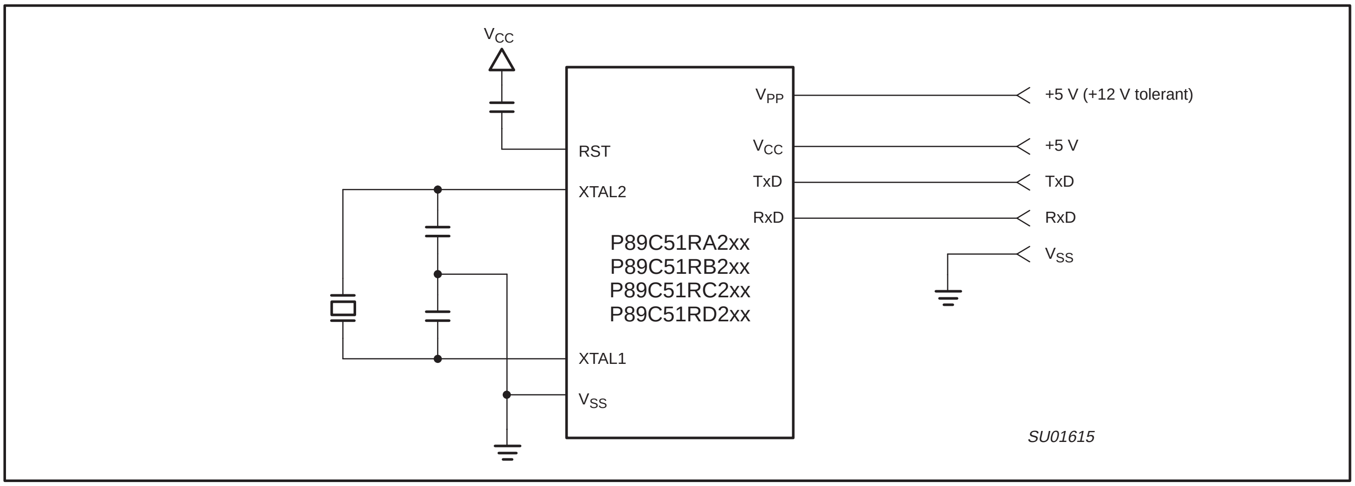


Figure 41. In-System Programming with a Minimum of Pins

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 P89C51RA2/RB2/RC2/RD2xx through the serial port. This firmware is provided by Philips and embedded within each P89C51RA2/RB2/RC2/RD2xx 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 41). 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 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 P89C51RA2/RB2/RC2/RD2xx 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 P89C51RA2/RB2/RC2/RD2xx 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 9.

As a record is received by the P89C51RA2/RB2/RC2/RD2xx, 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 P89C51RA2/RB2/RC2/RD2xx 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 exception).

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” character indicates that one of the bytes did not properly program. It is necessary to send a type 02 record (specify oscillator frequency) to the P89C51RA2/RB2/RC2/RD2xx before programming data.

The ISP facility was designed to that specific crystal frequencies were not required in order to generate baud rates or time the programming pulses. The user thus needs to provide the P89C51RA2/RB2/RC2/RD2xx with information required to generate the proper timing. Record type 02 is provided for this purpose.

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 10.

Using the Watchdog Timer (WDT)

The P89C51Rx2 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 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 running.

Table 10. IAP calls

IAP CALL	PARAMETER

PROGRAM BYTE	Input Parameter:
	R0 = osc freq (integer)
	R1 = 02h or R1= 82h (WDT feed)
	DPTR = address of byte to program
	ACC = byte to program
	Return Parameter:
	ACC = 00 if pass, !=00 if fail

ERASE 4K CODE BLOCK
(New function)
	Input Parameter:
	R0 = osc freq (integer)
	R1 = 0Ch or R1 = 8Ch (WDT feed)
	DPH = address of 4k code block
	DPH = 00H , 4k block 0, 0k~4k
	DPH = 10H , 4k block 1, 4k~8k
	DPH = 20H , 4k block 2, 8k~12k
	DPH = 30H , 4k block 3, 12k~16k
	DPH = 40H , 4k block 4, 16k~20k
	DPH = 50H , 4k block 5, 20k~24k
	DPH = 60H , 4k block 6, 24k~28k
	DPH = 70H , 4k block 7, 28k~32k
	DPH = 80H , 4k block 8, 32k~36k
	DPH = 90H , 4k block 9, 36k~40k
	DPH = A0H , 4k block 10, 40k~44k
	DPH = B0H , 4k block 11, 44k~48k
	DPH = C0H , 4k block 12, 48k~52k
	DPH = D0H , 4k block 13, 52k~56k
	DPH = E0H , 4k block 14, 56k~60k
	DPH = F0H , 4k block 15, 60k~64k
	DPL = 00h
	Return Parameter:
	ACC = 00 if pass, !=00 if fail

ERASE 8K / 16K CODE BLOCK	Input Parameter:
	R0 = osc freq (integer)
	R1 = 01h or R1 = 81h (WDT feed)
	DPH = address of code block
	DPH = 00H , block 0 , 0k~8k
	DPH = 20H , block 1 , 8k~16k
	DPH = 40H , block 2 , 16~32k
	DPH = 80H , block 3 , 32k~48k
	DPH = C0H , block 4 , 48k~64k
	DPL = 00h
	Return Parameter:
	ACC = 00 if pass , !=0 if fail

ERASE STATUS BYTE &
BOOT VECTOR
	Input Parameter:
	R0 = osc freq (integer)
	R1 = 04h or R1 = 84h (WDT feed)
	DPH = 00h
	DPL = don't care
	Return Parameter:
	ACC = 00 if pass , !=0 if fail

Security

The security feature protects against software piracy and prevents the contents of the Flash from being read. The Security Lock bits are located in Flash. The P89C51RA2/RB2/RC2/RD2xx has three programmable security lock bits that will provide different levels of protection for the on-chip code and data (see Table 11).

Table 11.

	SECURITY LOCK BITS[1]			PROTECTION DESCRIPTION
LEVEL	LB1	LB2	LB3	
1	0	0	0	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory.
2	1	0	0	Block erase is disabled. Erase or programming of the status byte or boot vector is disabled.
3	1	1	0	Verify of code memory is disabled.
4	1	1	1	External execution is disabled.

NOTE:

1. Security bits are independent of each other. Full-chip erase may be performed regardless of the state of the security bits.
2. Any other combination of lock bits is undefined.
3. Setting LBx doesn’t prevent programming of unprogrammed bits.

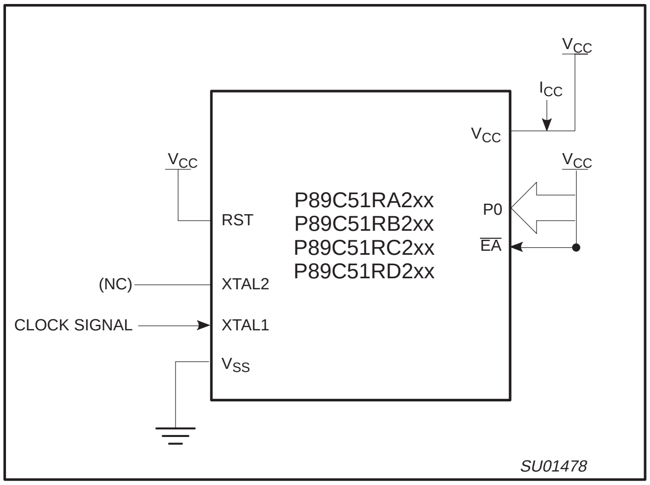


Figure 52. I_{CC} Test Condition, Active Mode, T_{amb} = 25 °C. All other pins are disconnected

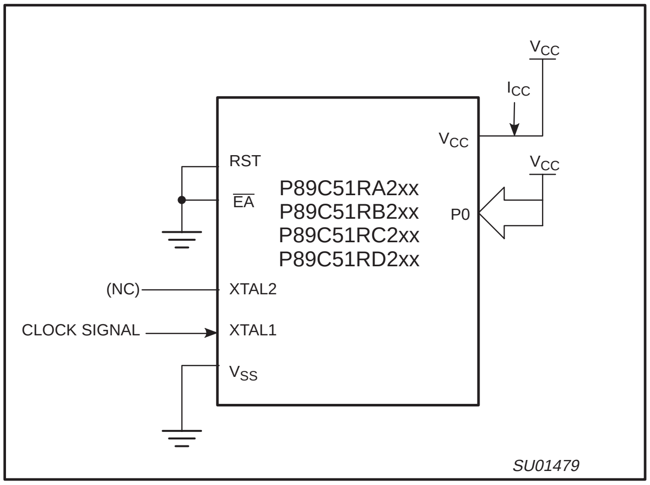


Figure 53. I_{CC} Test Condition, Idle Mode, T_{amb} = 25 °C. All other pins are disconnected

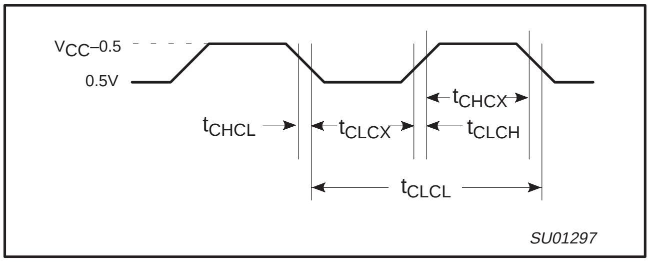


Figure 54. Clock Signal Waveform for I_{CC} Tests in Active and Idle Modes. $t_{CLCL} = t_{CHCL}$ = 10 ns

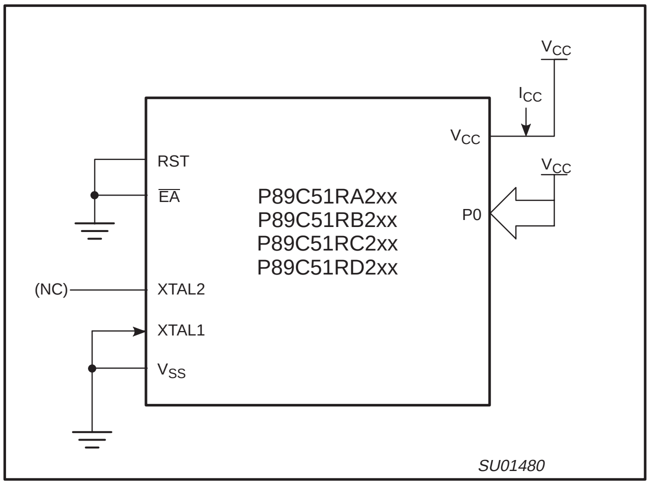
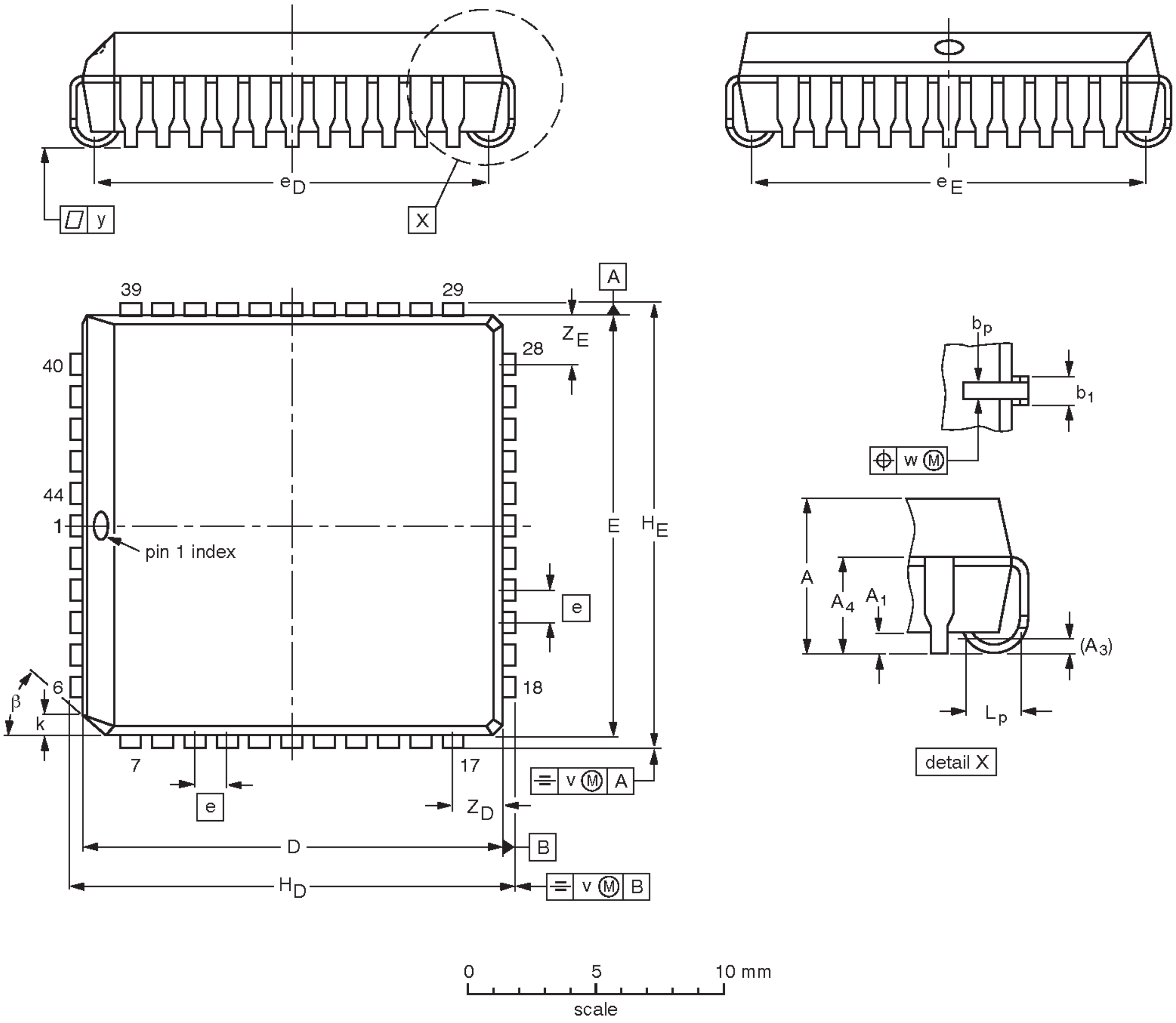


Figure 55. I_{CC} Test Condition, Power Down Mode. All other pins are disconnected; V_{CC} = 2 V to 5.5 V

PLCC44: plastic leaded chip carrier; 44 leads SOT187-2



DIMENSIONS (mm dimensions are derived from the original inch dimensions)

UNIT	A	A_1 min.	A_3	A_4 max.	b_p	b_1	$D^{(1)}$	$E^{(1)}$	e	e_D	e_E	H_D	H_E	k	L_p	v	w	y	$Z_D^{(1)}$ max.	$Z_E^{(1)}$ max.	β
mm	4.57 4.19	0.51	0.25	3.05	0.53 0.33	0.81 0.66	16.66 16.51	16.66 16.51	1.27	16.00 14.99	16.00 14.99	17.65 17.40	17.65 17.40	1.22 1.07	1.44 1.02	0.18	0.18	0.1	2.16	2.16	45°
inches	0.180 0.165	0.02	0.01	0.12	0.021 0.013	0.032 0.026	0.656 0.650	0.656 0.650	0.05	0.63 0.59	0.63 0.59	0.695 0.685	0.695 0.685	0.048 0.042	0.057 0.040	0.007	0.007	0.004	0.085	0.085	

Note

1. Plastic or metal protrusions of 0.25 mm (0.01 inch) maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT187-2	112E10	MS-018	EDR-7319			~~99-12-27~~ 01-11-14

REVISION HISTORY

Date	CPCN	Description
2002 July 18	9397 750 10129	Modified ordering information table
2002 May 20	9397 750 09843	Initial release

Data sheet status

Data sheet status [1]	Product status [2]	Definitions
Objective data	Development	This data sheet contains data from the objective specification for product development. Philips Semiconductors reserves the right to change the specification in any manner without notice.
Preliminary data	Qualification	This data sheet contains data from the preliminary specification. Supplementary data will be published at a later date. Philips Semiconductors reserves the right to change the specification without notice, in order to improve the design and supply the best possible product.
Product data	Production	This data sheet contains data from the product specification. Philips Semiconductors reserves the right to make changes at any time in order to improve the design, manufacturing and supply. Changes will be communicated according to the Customer Product/Process Change Notification (CPCN) procedure SNW-SQ-650A.

[1] Please consult the most recently issued data sheet before initiating or completing a design.

[2] The product status of the device(s) described in this data sheet may have changed since this data sheet was published. The latest information is available on the Internet at URL http://www.semiconductors.philips.com.

Definitions

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

Limiting values definition — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 60134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Disclaimers

Life support — These products are not designed for use in life support appliances, devices or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Contact information

For additional information please visit
http://www.semiconductors.philips.com. Fax: **+31 40 27 24825**

For sales offices addresses send e-mail to:
sales.addresses@www.semiconductors.philips.com.

Date of release: 07-02

Document order number: 9397 750 10129

Let's make things better.

**Philips
Semiconductors**

PHILIPS