DATA SHEET

P89C51RA2xx/RB2xx/RC2xx/RD2xx

80C51 8-bit Flash microcontroller family

8KB/16KB/32KB/64KB ISP/IAP Flash with
512B/512B/512B/1KB RAM

Preliminary data

2002 May 20

Philips
Semiconductors

PHILIPS

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

SELECTION TABLE

Type	Memory				Timers				Serial Interfaces													
	RAM	ROM	OTP	Flash	# of Timers	PWM	PCA	WD	UART	I2C	CAN	SPI	ADC bits/ch.	I/O Pins	Interrupts (Ext.)/Levels	Program Security	Default Clock Rate[1]	Optional Clock Rate[1]	Reset active low/high?	Max. Freq. at 6-clk / 12-clk (MHz)	Freq. Range at 3V (MHz)	Freq. Range at 5V (MHz)
P89C51RD2xx[2]	1K	–	–	64K	4	√	√	√	√	–	–	–	–	32	7(2)/4	√	12-clk	6-clk	H	20/33	–	0-20/33
P89C51RC2xx	512B	–	–	32K	4	√	√	√	√	–	–	–	–	32	7(2)/4	√	12-clk	6-clk	H	20/33	–	0-20/33
P89C51RB2xx	512B	–	–	16K	4	√	√	√	√	–	–	–	–	32	7(2)/4	√	12-clk	6-clk	H	20/33	–	0-20/33
P89C51RA2xx	512B	–	–	8K	4	√	√	√	√	–	–	–	–	32	7(2)/4	√	12-clk	6-clk	H	20/33	–	0-20/33

NOTE:
1. P89C51Rx2Hxx devices have a 6-clk default clock rate (12-clk optional). Please also see Device Comparison Table.
2. P89C51RD2xx will be released shortly.

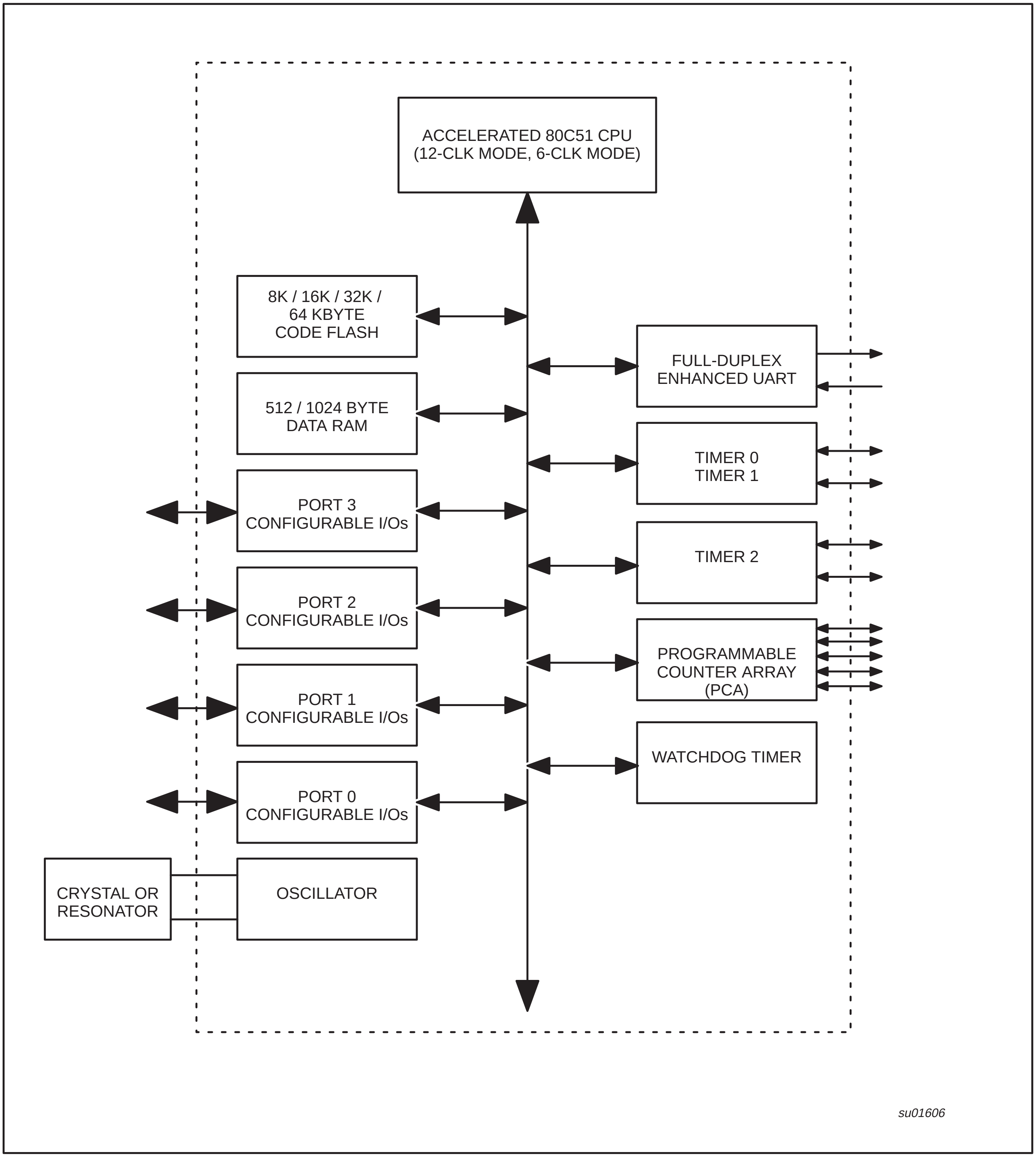
DEVICE COMPARISON TABLE

Item	1st generation of Rx2 devices	2nd generation of Rx2 devices	Difference
Type description	P89C51Rx2**H**xx(x)	P89C51Rx2xx(x)	No more letter 'H'
Programming algorithm	When using a parallel programmer, be sure to select P89C51Rx2**H**xx(x) devices	When using a parallel programmer, be sure to select P89C51Rx2xx(x) devices (no more letter 'H')	Different programming algorithm due to process change
Clock mode (I)	**6-clk default, OTP** configuration bit to program to **12-clk** mode using parallel programmer (**cannot** be programmed back to 6-clk)	**12-clk default, Flash** configuration bit to program to **6-clk** mode using parallel programmer or ISP (**can** be reprogrammed)	More flexibility for the end user, more compatibility to older P89C51Rx+ parts
Clock mode (II)	N/A	6-clock/12-clock mode programmable "on the fly" by SFR bit X2 (CKCON.0)	Clock mode can be changed by software
Peripheral clock modes	N/A	Peripherals can be run in 12-clk mode while CPU runs in 6-clk mode	More flexibility, lower power consumption
Flash block structure	Two 8-Kbyte blocks 1–3 16-Kbyte blocks	2–16 4-Kbyte blocks	More flexibility

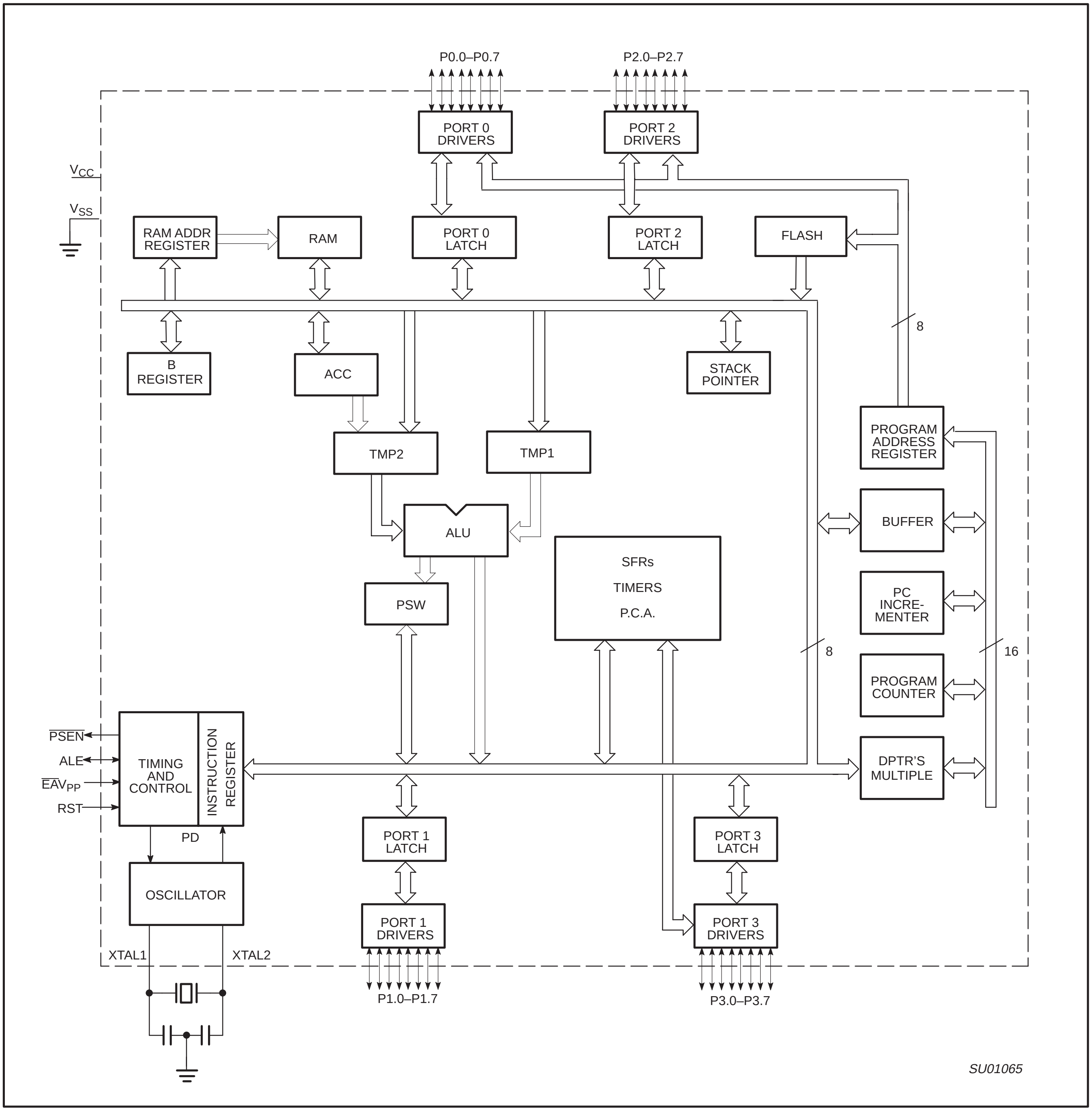
ORDERING INFORMATION

	PHILIPS (EXCEPT NORTH AMERICA) PART ORDER NUMBER PART MARKING	MEMORY		TEMPERATURE RANGE (°C) AND PACKAGE	VOLTAGE RANGE	FREQUENCY (MHz)		DWG #
		FLASH	RAM			6-CLOCK MODE	12-CLOCK MODE	
1.	P89C51RA2BA	8 KB	512 B	0 to +70, PLCC	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT187-2
2.	P89C51RA2BBD	8 KB	512 B	0 to +70, LQFP	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT389-1
3.	P89C51RB2BA	16 KB	512 B	0 to +70, PLCC	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT187-2
4.	P89C51RB2BBD	16 KB	512 B	0 to +70, LQFP	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT389-1
5.	P89C51RC2BN	32 KB	512 B	0 to +70, PDIP	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT129-1
6.	P89C51RC2BA	32 KB	512 B	0 to +70, PLCC	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT187-2
7.	P89C51RC2FA	32 KB	512 B	–40 to +85, PLCC	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT187-2
8.	P89C51RC2BBD	32 KB	512 B	0 to +70, LQFP	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT389-1
9.	P89C51RC2FBD	32 KB	512 B	–40 to +85, LQFP	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT389-1
10.	P89C51RD2BN	64 KB	1024 B	0 to +70, PDIP	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT129-1
11.	P89C51RD2BA	64 KB	1024 B	0 to +70, PLCC	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT187-2
12.	P89C51RD2BBD	64 KB	1024 B	0 to +70, LQFP	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT389-1
13.	P89C51RD2FA	64 KB	1024 B	–40 to +85, PLCC	4.5–5.5 V	0 to 20 MHz	0 to 33 MHz	SOT187-2

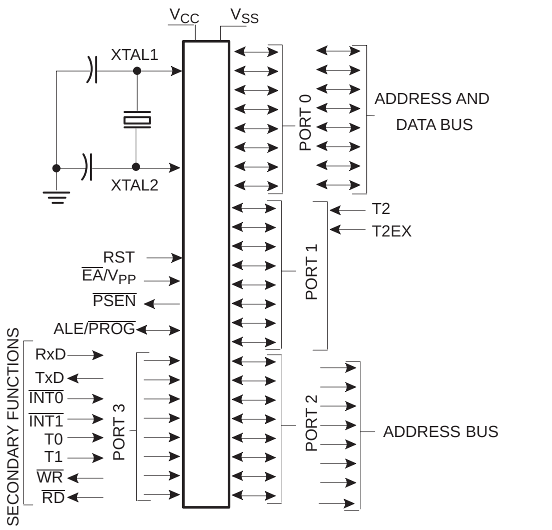
BLOCK DIAGRAM 1



BLOCK DIAGRAM – CPU ORIENTED

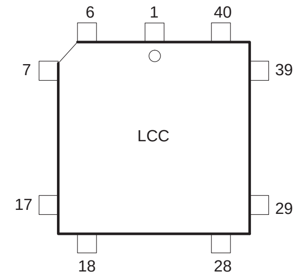


LOGIC SYMBOL



SU01302

Plastic Leaded Chip Carrier



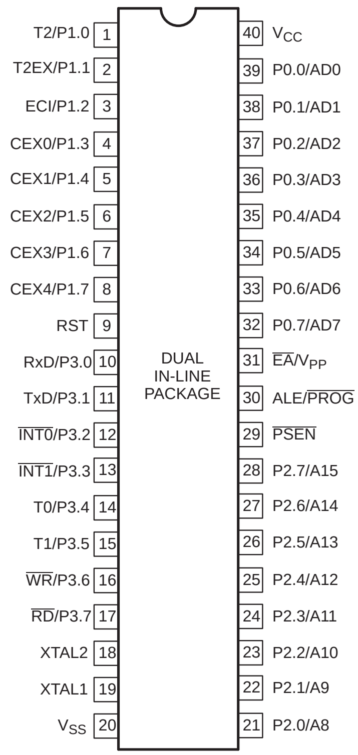
Pin	Function	Pin	Function	Pin	Function
1	NIC*	16	P3.4/T0	31	P2.7/A15
2	P1.0/T2	17	P3.5/T1	32	$\overline{PSEN}$
3	P1.1/T2EX	18	P3.6/$\overline{WR}$	33	ALE/$\overline{PROG}$
4	P1.2/ECI	19	P3.7/$\overline{RD}$	34	NIC*
5	P1.3/CEX0	20	XTAL2	35	$\overline{EA}/V_{PP}$
6	P1.4/CEX1	21	XTAL1	36	P0.7/AD7
7	P1.5/CEX2	22	V_{SS}	37	P0.6/AD6
8	P1.6/CEX3	23	NIC*	38	P0.5/AD5
9	P1.7/CEX4	24	P2.0/A8	39	P0.4/AD4
10	RST	25	P2.1/A9	40	P0.3/AD3
11	P3.0/RxD	26	P2.2/A10	41	P0.2/AD2
12	NIC*	27	P2.3/A11	42	P0.1/AD1
13	P3.1/TxD	28	P2.4/A12	43	P0.0/AD0
14	P3.2/$\overline{INT0}$	29	P2.5/A13	44	V_{CC}
15	P3.3/$\overline{INT1}$	30	P2.6/A14		

* NO INTERNAL CONNECTION

SU00023

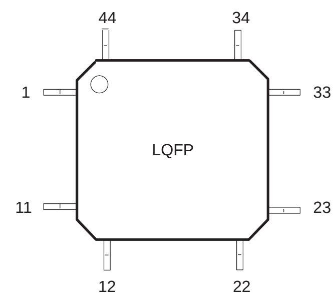
PINNING

Plastic Dual In-Line Package



SU00021

Plastic Quad Flat Pack



Pin	Function	Pin	Function	Pin	Function
1	P1.5/CEX2	16	V_{SS}	31	P0.6/AD6
2	P1.6/CEX3	17	NIC*	32	P0.5/AD5
3	P1.7/CEX4	18	P2.0/A8	33	P0.4/AD4
4	RST	19	P2.1/A9	34	P0.3/AD3
5	P3.0/RxD	20	P2.2/A10	35	P0.2/AD2
6	NIC*	21	P2.3/A11	36	P0.1/AD1
7	P3.1/TxD	22	P2.4/A12	37	P0.0/AD0
8	P3.2/$\overline{INT0}$	23	P2.5/A13	38	V_{CC}
9	P3.3/$\overline{INT1}$	24	P2.6/A14	39	NIC*
10	P3.4/T0	25	P2.7/A15	40	P1.0/T2
11	P3.5/T1	26	$\overline{PSEN}$	41	P1.1/T2EX
12	P3.6/$\overline{WR}$	27	ALE/$\overline{PROG}$	42	P1.2/ECI
13	P3.7/$\overline{RD}$	28	NIC*	43	P1.3/CEX0
14	XTAL2	29	$\overline{EA}/V_{PP}$	44	P1.4/CEX1
15	XTAL1	30	P0.7/AD7		

* NO INTERNAL CONNECTION

SU01400

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	**$\overline{\text{INT0}}$ (P3.2):** External interrupt
	13	15	9	I	**$\overline{\text{INT1}}$ (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	**$\overline{\text{WR}}$ (P3.6):** External data memory write strobe
	17	19	13	O	**$\overline{\text{RD}}$ (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.

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	PDIP	PLCC	LQFP		
$\overline{\text{PSEN}}$	29	32	26	O	**Program Store Enable:** The read strobe to external program memory. When executing code from the external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory. $\overline{\text{PSEN}}$ is not activated during fetches from internal program memory.
$\overline{\text{EA}}/V_{PP}$	31	35	29	I	**External Access Enable/Programming Supply Voltage:** $\overline{\text{EA}}$ must be externally held low to enable the device to fetch code from external program memory locations. If $\overline{\text{EA}}$ is held high, the device executes from internal program memory. The value on the $\overline{\text{EA}}$ pin is latched when RST is released and any subsequent changes have no effect. This pin also receives the programming supply voltage (V_{PP}) during Flash programming.
XTAL1	19	21	15	I	**Crystal 1:** Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	20	14	O	**Crystal 2:** Output from the inverting oscillator amplifier.

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

Table 1. Special Function Registers

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION MSB							LSB	RESET VALUE
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
AUXR#	Auxiliary	8EH	–	–	–	–	–	–	EXTRAM	AO	xxxxxx00B
AUXR1#	Auxiliary 1	A2H	–	–	ENBOOT	–	GF2	0	–	DPS	xxxxxxx0B
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
CCAP0H#	Module 0 Capture High	FAH									xxxxxxxxB
CCAP1H#	Module 1 Capture High	FBH									xxxxxxxxB
CCAP2H#	Module 2 Capture High	FCH									xxxxxxxxB
CCAP3H#	Module 3 Capture High	FDH									xxxxxxxxB
CCAP4H#	Module 4 Capture High	FEH									xxxxxxxxB
CCAP0L#	Module 0 Capture Low	EAH									xxxxxxxxB
CCAP1L#	Module 1 Capture Low	EBH									xxxxxxxxB
CCAP2L#	Module 2 Capture Low	ECH									xxxxxxxxB
CCAP3L#	Module 3 Capture Low	EDH									xxxxxxxxB
CCAP4L#	Module 4 Capture Low	EEH									xxxxxxxxB
CCAPM0#	Module 0 Mode	DAH	–	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM1#	Module 1 Mode	DBH	–	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM2#	Module 2 Mode	DCH	–	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM3#	Module 3 Mode	DDH	–	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
CCAPM4#	Module 4 Mode	DEH	–	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	x0000000B
			DF	DE	DD	DC	DB	DA	D9	D8	
CCON*#	PCA Counter Control	D8H	CF	CR	–	CCF4	CCF3	CCF2	CCF1	CCF0	00x00000B
CH#	PCA Counter High	F9H									00H
CKCON#	Clock control	8FH	–	WDX2	PCAX2	SIX2	T2X2	T1X2	T0X2	X2	x0000000B
CL#	PCA Counter Low	E9H									00H
CMOD#	PCA Counter Mode	D9H	CIDL	WDTE	–	–	–	CPS1	CPS0	ECF	00xxx000B
DPTR:	Data Pointer (2 bytes)										
DPH	Data Pointer High	83H									00H
DPL	Data Pointer Low	82H									00H
			AF	AE	AD	AC	AB	AA	A9	A8	
IE*	Interrupt Enable 0	A8H	EA	EC	ET2	ES	ET1	EX1	ET0	EX0	00H
			BF	BE	BD	BC	BB	BA	B9	B8	
IP*	Interrupt Priority	B8H	–	PPC	PT2	PS	PT1	PX1	PT0	PX0	x0000000B
IPH#	Interrupt Priority High	B7H	–	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	x0000000B
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	CEX4	CEX3	CEX2	CEX1	CEX0	ECI	T2EX	T2	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	$\overline{RD}$	$\overline{WR}$	T1	T0	$\overline{INT1}$	$\overline{INT0}$	TxD	RxD	FFH
PCON#[1]	Power Control	87H	SMOD1	SMOD0	–	POF	GF1	GF0	PD	IDL	00xxx000B

* SFRs are bit addressable.
# SFRs are modified from or added to the 80C51 SFRs.
– Reserved bits.
1. Reset value depends on reset source.

Table 1. Special Function Registers (Continued)

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION MSB							LSB	RESET VALUE
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program Status Word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00000000B
RCAP2H#	Timer 2 Capture High	CBH									00H
RCAP2L#	Timer 2 Capture Low	CAH									00H
SADDR#	Slave Address	A9H									00H
SADEN#	Slave Address Mask	B9H									00H
SBUF	Serial Data Buffer	99H									xxxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
SCON*	Serial Control	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SP	Stack Pointer	81H									07H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer Control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
			CF	CE	CD	CC	CB	CA	C9	C8	
T2CON*	Timer 2 Control	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$C/\overline{T}2$	$CP/\overline{RL}2$	00H
T2MOD#	Timer 2 Mode Control	C9H	–	–	–	–	–	–	T2OE	DCEN	xxxxxx00B
TH0	Timer High 0	8CH									00H
TH1	Timer High 1	8DH									00H
TH2#	Timer High 2	CDH									00H
TL0	Timer Low 0	8AH									00H
TL1	Timer Low 1	8BH									00H
TL2#	Timer Low 2	CCH									00H
TMOD	Timer Mode	89H	GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0	00H
WDTRST	Watchdog Timer Reset	A6H									

* SFRs are bit addressable.
\# SFRs are modified from or added to the 80C51 SFRs.
– Reserved bits.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier. The pins can be configured for use as an on-chip oscillator.

To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected. Minimum and maximum high and low times specified in the data sheet must be observed.

This device is configured at the factory to operate using 12 clock periods per machine cycle, referred to in this datasheet as “12-clock mode”. It may be optionally configured on commercially available Flash programming equipment or via ISP or via software to operate at 6 clocks per machine cycle, referred to in this datasheet as “6-clock mode”. (This yields performance equivalent to twice that of standard 80C51 family devices). Also see next page.

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.

LOW POWER MODES

Stop Clock Mode

The static design enables the clock speed to be reduced down to 0 MHz (stopped). When the oscillator is stopped, the RAM and Special Function Registers retain their values. This mode allows step-by-step utilization and permits reduced system power consumption by lowering the clock frequency down to any value. For lowest power consumption the Power Down mode is suggested.

Idle Mode

In the idle mode (see Table 2), the CPU puts itself to sleep while all of the on-chip peripherals stay active. The instruction to invoke the idle mode is the last instruction executed in the normal operating mode before the idle mode is activated. The CPU contents, the on-chip RAM, and all of the special function registers remain intact during this mode. The idle mode can be terminated either by any enabled interrupt (at which time the process is picked up at the interrupt service routine and continued), or by a hardware reset which starts the processor in the same manner as a power-on reset.

Power-Down Mode

To save even more power, a Power Down mode (see Table 2) can be invoked by software. In this mode, the oscillator is stopped and the instruction that invoked Power Down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values down to 2 V and care must be taken to return V_{CC} to the minimum specified operating voltages before the Power Down Mode is terminated.

Either a hardware reset or external interrupt can be used to exit from Power Down. Reset redefines all the SFRs but does not change the on-chip RAM. An external interrupt allows both the SFRs and the on-chip RAM to retain their values.

To properly terminate Power Down, the reset or external interrupt should not be executed before V_{CC} is restored to its normal operating level and must be held active long enough for the oscillator to restart and stabilize (normally less than 10 ms).

With an external interrupt, INT0 and INT1 must be enabled and configured as level-sensitive. Holding the pin low restarts the oscillator but bringing the pin back high completes the exit. Once the interrupt is serviced, the next instruction to be executed after RETI will be the one following the instruction that put the device into Power Down.

POWER-ON FLAG

The Power-On Flag (POF) is set by on-chip circuitry when the V_{CC} level on the P89C51RA2/RB2/RC2/RD2xx rises from 0 to 5 V. The POF bit can be set or cleared by software allowing a user to determine if the reset is the result of a power-on or a warm start after powerdown. The V_{CC} level must remain above 3 V for the POF to remain unaffected by the V_{CC} level.

Design Consideration

When the idle mode is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

ONCE™ Mode

The ONCE (“On-Circuit Emulation”) Mode facilitates testing and debugging of systems without the device having to be removed from the circuit. The ONCE Mode is invoked by:

1. Pull ALE low while the device is in reset and $\overline{\text{PSEN}}$ is high;
2. Hold ALE low as RST is deactivated.

While the device is in ONCE Mode, the Port 0 pins go into a float state, and the other port pins and ALE and $\overline{\text{PSEN}}$ are weakly pulled high. The oscillator circuit remains active. While the device is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.

Programmable Clock-Out

A 50% duty cycle clock can be programmed to come out on P1.0. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed:

1. to input the external clock for Timer/Counter 2, or
2. to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency in 12-clock mode (122 Hz to 8 MHz in 6-clock mode).

To configure the Timer/Counter 2 as a clock generator, bit C/$\overline{\text{T}}$2 (in T2CON) must be cleared and bit T20E in T2MOD must be set. Bit TR2 (T2CON.2) also must be set to start the timer.

The Clock-Out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L) as shown in this equation:

$$\frac{\text{Oscillator Frequency}}{n \times (65536 - \text{RCAP2H}, \text{RCAP2L})}$$

n = 2 in 6-clock mode
4 in 12-clock mode

Where (RCAP2H,RCAP2L) = the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

In the Clock-Out mode Timer 2 roll-overs will not generate an interrupt. This is similar to when it is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and the Clock-Out frequency will be the same.

Table 2. External Pin Status During Idle and Power-Down Mode

MODE	PROGRAM MEMORY	ALE	$\overline{\text{PSEN}}$	PORT 0	PORT 1	PORT 2	PORT 3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

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. 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 look like an 8048 Timer, which is an 8-bit Counter with a divide-by-32 prescaler. Figure 2 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. (Setting GATE = 1 allows the Timer to be controlled by external input $\overline{INTn}$, to facilitate pulse width measurements). TRn is a control bit in the Special Function Register TCON (Figure 3).

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 4. 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 5. 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. With Timer 0 in Mode 3, an 80C51 can look like it has 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

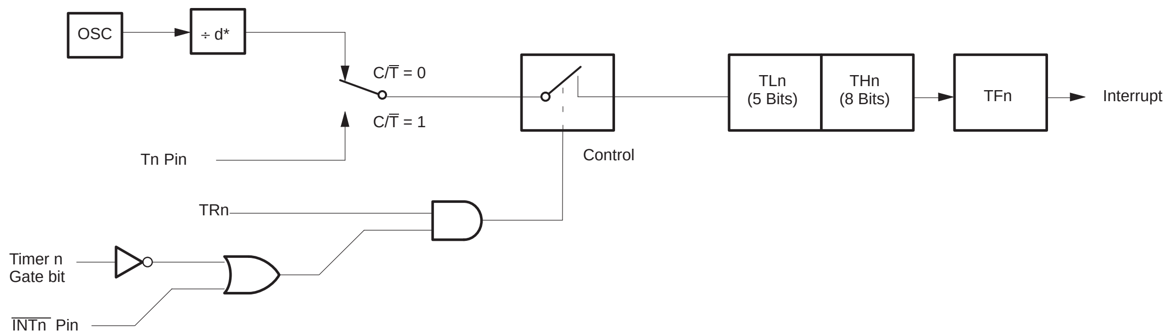
7	6	5	4	3	2	1	0
GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0
TIMER 1				TIMER 0			

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



*d = 6 in 6-clock mode; d = 12 in 12-clock mode.

SU01618

Figure 2. Timer/Counter 0/1 Mode 0: 13-Bit Timer/Counter

TCON Address = 88H Reset Value = 00H

Bit Addressable

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

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

SU01516

Figure 3. Timer/Counter 0/1 Control (TCON) Register

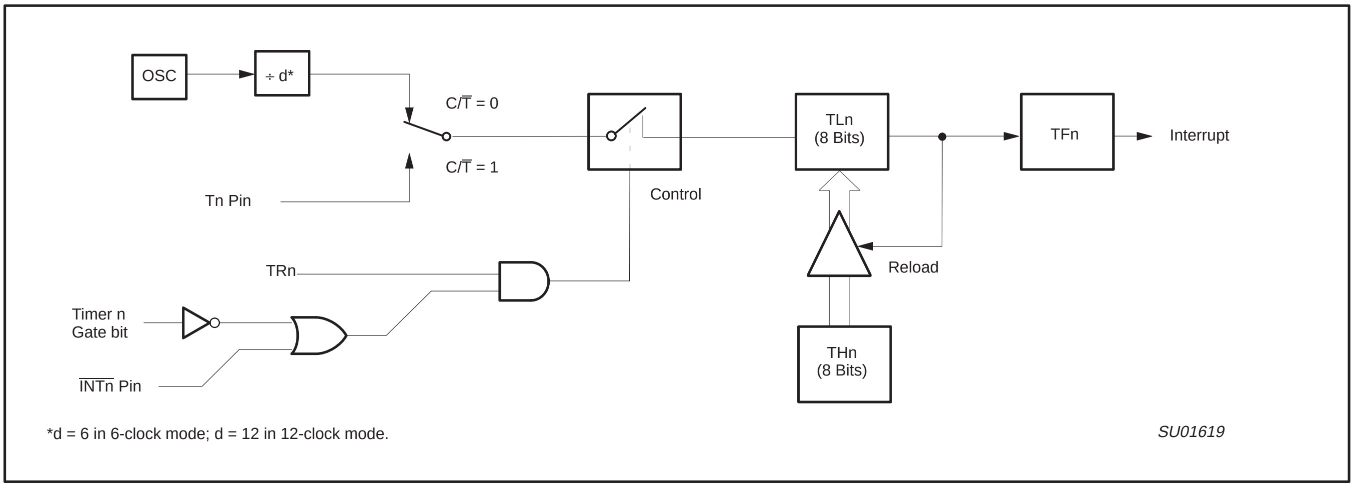


Figure 4. Timer/Counter 0/1 Mode 2: 8-Bit Auto-Reload

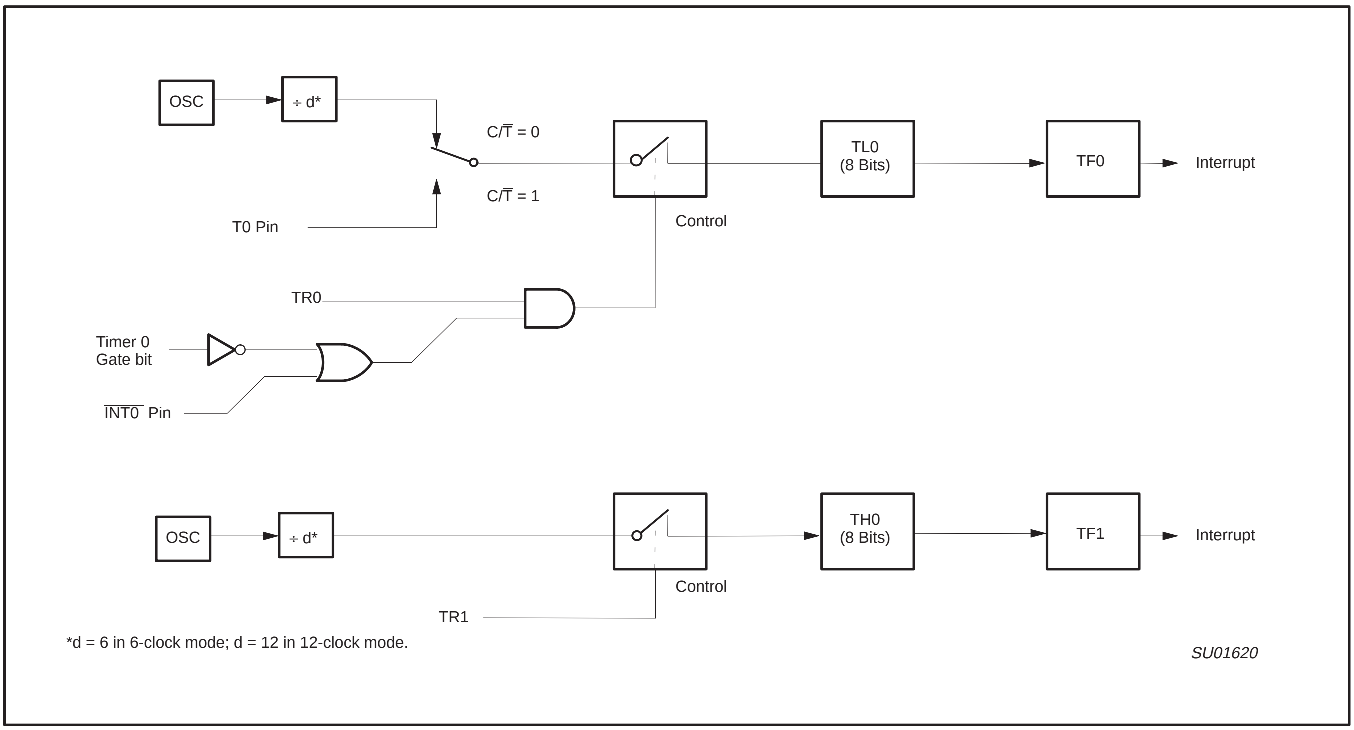


Figure 5. Timer/Counter 0 Mode 3: Two 8-Bit Counters

TIMER 2 OPERATION

Timer 2

Timer 2 is a 16-bit Timer/Counter which can operate as either an event timer or an event counter, as selected by $C/\overline{T}2$ in the special function register T2CON (see Figure 6). Timer 2 has three operating modes: Capture, Auto-reload (up or down counting), and Baud Rate Generator, which are selected by bits in the T2CON as shown in Table 3.

Capture Mode

In the capture mode there are two options which are selected by bit EXEN2 in T2CON. If EXEN2=0, then timer 2 is a 16-bit timer or counter (as selected by $C/\overline{T}2$ in T2CON) which, upon overflowing sets bit TF2, the timer 2 overflow bit. This bit can be used to generate an interrupt (by enabling the Timer 2 interrupt bit in the IE register). If EXEN2= 1, Timer 2 operates as described above, but with the added feature that a 1- to -0 transition at external input T2EX causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set, and EXF2 like TF2 can generate an interrupt (which vectors to the same location as Timer 2 overflow interrupt. The Timer 2 interrupt service routine can interrogate TF2 and EXF2 to determine which event caused the interrupt). The capture mode is illustrated in Figure 7 (There is no reload value for TL2 and TH2 in this mode. Even when a capture event occurs from T2EX, the counter keeps on counting T2EX pin transitions or osc/6 pulses (osc/12 in 12-clock mode).).

Auto-Reload Mode (Up or Down Counter)

In the 16-bit auto-reload mode, Timer 2 can be configured (as either a timer or counter [$C/\overline{T}2$ in T2CON]) then programmed to count up or down. The counting direction is determined by bit DCEN (Down Counter Enable) which is located in the T2MOD register (see Figure 8). When reset is applied the DCEN=0 which means Timer 2 will default to counting up. If DCEN bit is set, Timer 2 can count up or down depending on the value of the T2EX pin.

Figure 9 shows Timer 2 which will count up automatically since DCEN=0. In this mode there are two options selected by bit EXEN2 in T2CON register. If EXEN2=0, then Timer 2 counts up to 0FFFFH and sets the TF2 (Overflow Flag) bit upon overflow. This causes the Timer 2 registers to be reloaded with the 16-bit value in RCAP2L and RCAP2H. The values in RCAP2L and RCAP2H are preset by software means.

If EXEN2=1, then a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at input T2EX. This transition also sets the EXF2 bit. The Timer 2 interrupt, if enabled, can be generated when either TF2 or EXF2 are 1.

In Figure 10 DCEN=1 which enables Timer 2 to count up or down. This mode allows pin T2EX to control the direction of count. When a logic 1 is applied at pin T2EX Timer 2 will count up. Timer 2 will overflow at 0FFFFH and set the TF2 flag, which can then generate an interrupt, if the interrupt is enabled. This timer overflow also causes the 16-bit value in RCAP2L and RCAP2H to be reloaded into the timer registers TL2 and TH2.

When a logic 0 is applied at pin T2EX this causes Timer 2 to count down. The timer will underflow when TL2 and TH2 become equal to the value stored in RCAP2L and RCAP2H. Timer 2 underflow sets the TF2 flag and causes 0FFFFH to be reloaded into the timer registers TL2 and TH2.

The external flag EXF2 toggles when Timer 2 underflows or overflows. This EXF2 bit can be used as a 17th bit of resolution if needed. The EXF2 flag does not generate an interrupt in this mode of operation.

(MSB)							(LSB)
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$C/\overline{T}2$	$CP/\overline{RL}2$

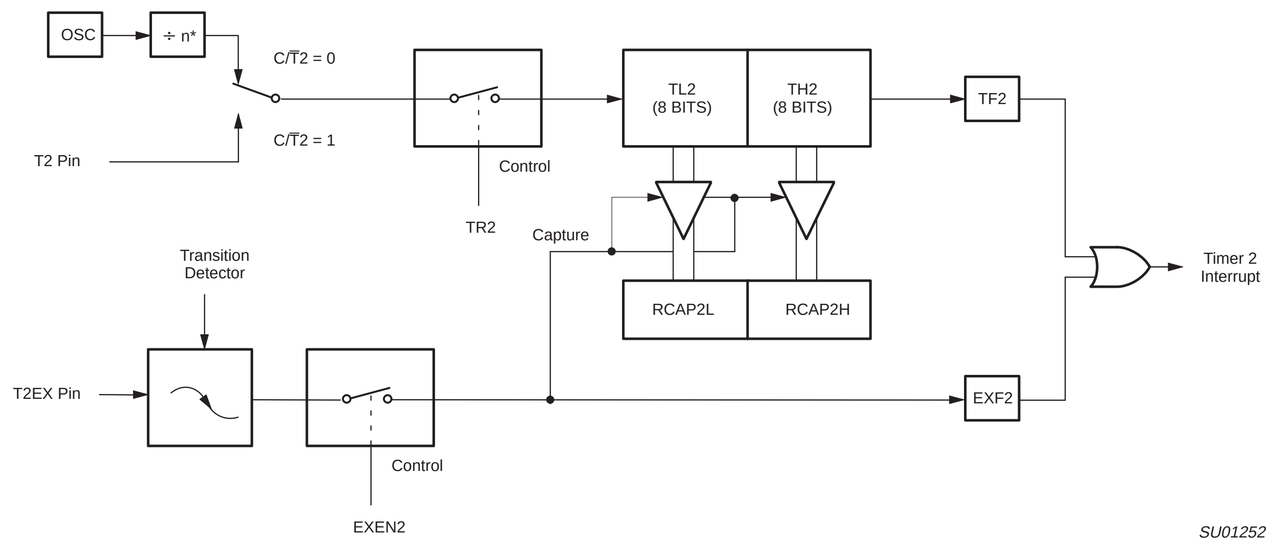
Symbol	Position	Name and Significance
TF2	T2CON.7	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK or TCLK = 1.
EXF2	T2CON.6	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	T2CON.5	Receive clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	T2CON.4	Transmit clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	T2CON.3	Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	T2CON.2	Start/stop control for Timer 2. A logic 1 starts the timer.
$C/\overline{T}2$	T2CON.1	Timer or counter select. (Timer 2) 0 = Internal timer (OSC/6 in 6-clock mode or OSC/12 in 12-clock mode) 1 = External event counter (falling edge triggered).
$CP/\overline{RL}2$	T2CON.0	Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2 = 1. When cleared, auto-reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

SU01251

Figure 6. Timer/Counter 2 (T2CON) Control Register

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 7. Timer 2 in Capture Mode

T2MOD Address = 0C9H Reset Value = XXXX XX00B

Not Bit Addressable

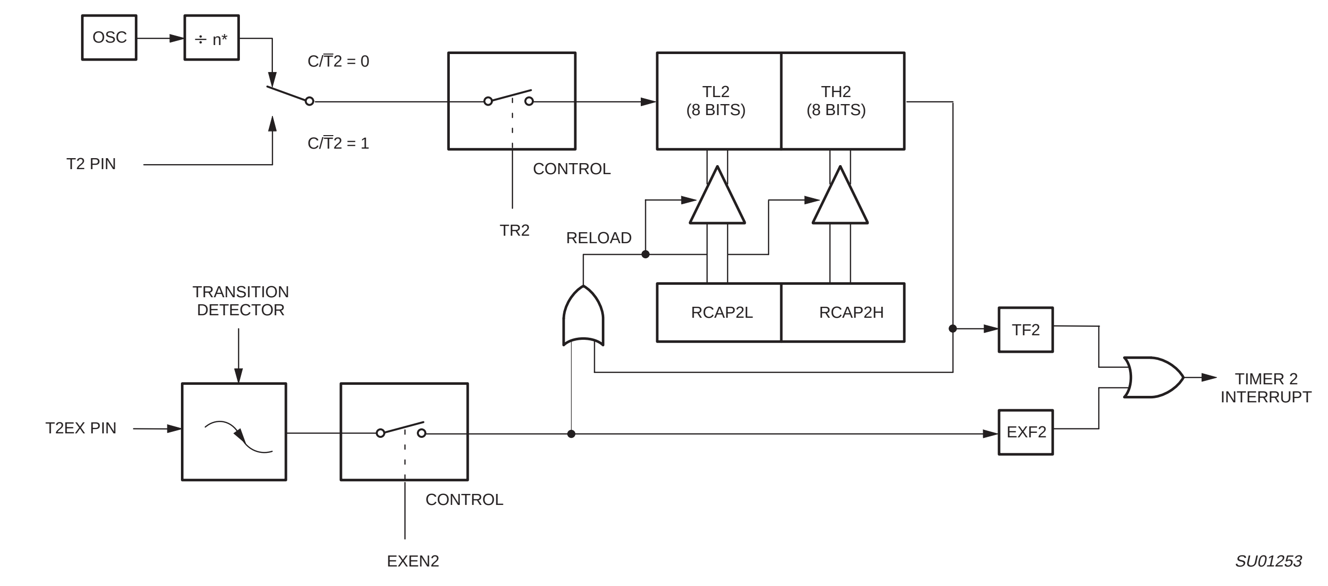
Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	T2OE	DCEN

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.

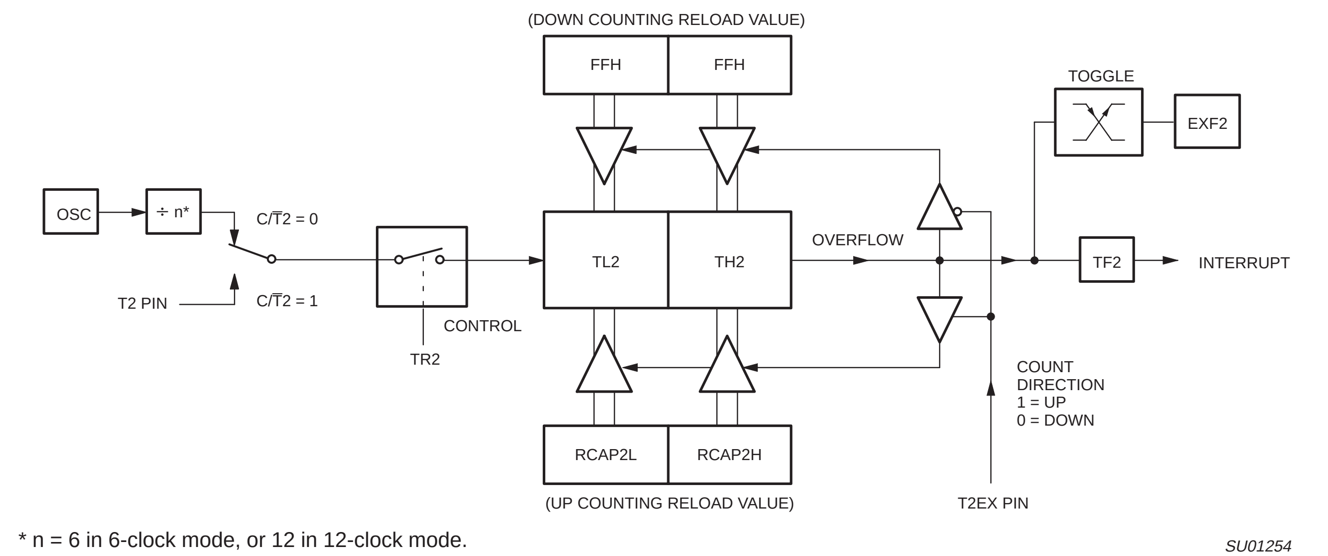
SU00729

Figure 8. Timer 2 Mode (T2MOD) Control Register



* 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)

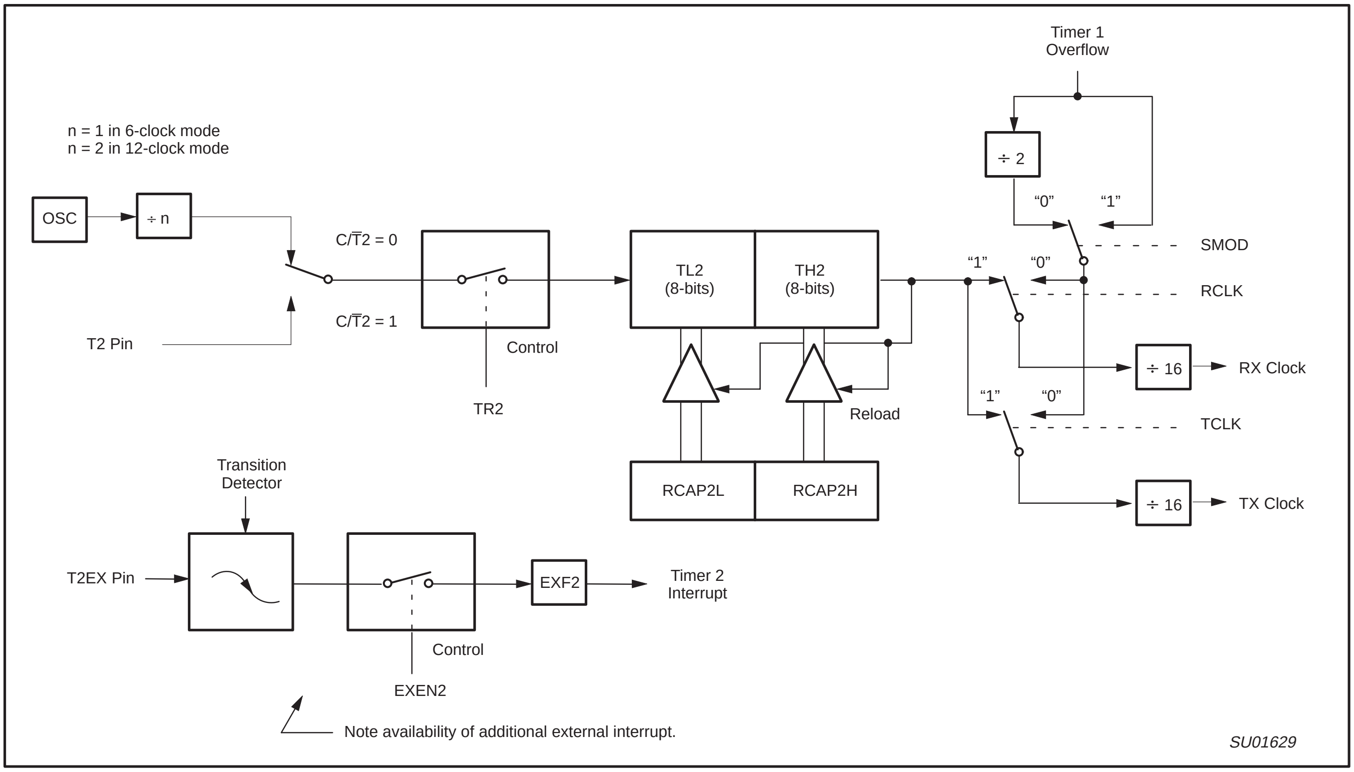


Figure 11. Timer 2 in Baud Rate Generator Mode

Table 4. Timer 2 Generated Commonly Used Baud Rates

Baud Rate		Osc Freq	Timer 2	
12-clock mode	6-clock mode		RCAP2H	RCAP2L
375 k	750 k	12 MHz	FF	FF
9.6 k	19.2 k	12 MHz	FF	D9
4.8 k	9.6 k	12 MHz	FF	B2
2.4 k	4.8 k	12 MHz	FF	64
1.2 k	2.4 k	12 MHz	FE	C8
300	600	12 MHz	FB	1E
110	220	12 MHz	F2	AF
300	600	6 MHz	FD	8F
110	220	6 MHz	F9	57

Baud Rate Generator Mode

Bits TCLK and/or RCLK in T2CON (Table 4) allow the serial port transmit and receive baud rates to be derived from either Timer 1 or Timer 2. When TCLK= 0, Timer 1 is used as the serial port transmit baud rate generator. When TCLK= 1, Timer 2 is used as the serial port transmit baud rate generator. RCLK has the same effect for the serial port receive baud rate. With these two bits, the serial port can have different receive and transmit baud rates – one generated by Timer 1, the other by Timer 2.

Figure 11 shows the Timer 2 in baud rate generation mode. The baud rate generation mode is like the auto-reload mode,in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in modes 1 and 3 are determined by Timer 2's overflow rate given below:

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The timer can be configured for either "timer" or "counter" operation. In many applications, it is configured for "timer" operation ($C/\overline{T}2=0$). Timer operation is different for Timer 2 when it is being used as a baud rate generator.

Usually, as a timer it would increment every machine cycle (i.e., $^1/_6$ the oscillator frequency in 6-clock mode, $^1/_{12}$ the oscillator frequency in 12-clock mode). As a baud rate generator, it increments at the oscillator frequency in 6-clock mode ($^{OSC}/_2$ in 12-clock mode). Thus the baud rate formula is as follows:

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Oscillator Frequency}}{[\,n^* \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]]}$$

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

Where: (RCAP2H, RCAP2L)= The content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

The Timer 2 as a baud rate generator mode shown in Figure 11, is valid only if RCLK and/or TCLK = 1 in T2CON register. Note that a rollover in TH2 does not set TF2, and will not generate an interrupt. Thus, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. Also if the EXEN2 (T2 external enable flag) is set, a 1-to-0 transition in T2EX (Timer/counter 2 trigger input) will set EXF2 (T2 external flag) but will not cause a reload from (RCAP2H, RCAP2L) to (TH2,TL2). Therefore when Timer 2 is in use as a baud rate generator, T2EX can be used as an additional external interrupt, if needed.

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.

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 =

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

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.

SCON Address = 98H Reset Value = 00H

Bit Addressable

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Where SM0, SM1 specify the serial port mode, as follows:

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

SM2 Enables the multiprocessor communication feature in Modes 2 and 3. In Mode 2 or 3, if SM2 is set to 1, then Rl will not be activated if the received 9th data bit (RB8) is 0. In Mode 1, if SM2=1 then RI will not be activated if a valid stop bit was not received. 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, is the 9th data bit that was received. In Mode 1, it 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.

SU01626

Figure 12. Serial Port Control (SCON) Register

Baud Rate			f_{OSC}	SMOD	Timer 1		
Mode	12-clock mode	6-clock mode			$C/\overline{T}$	Mode	Reload Value
Mode 0 Max	1.67 MHz	3.34 MHz	20 MHz	X	X	X	X
Mode 2 Max	625 k	1250 k	20 MHz	1	X	X	X
Mode 1, 3 Max	104.2 k	208.4 k	20 MHz	1	0	2	FFH
Mode 1, 3	19.2 k	38.4 k	11.059 MHz	1	0	2	FDH
	9.6 k	19.2 k	11.059 MHz	0	0	2	FDH
	4.8 k	9.6 k	11.059 MHz	0	0	2	FAH
	2.4 k	4.8 k	11.059 MHz	0	0	2	F4H
	1.2 k	2.4 k	11.059 MHz	0	0	2	E8H
	137.5	275	11.986 MHz	0	0	2	1DH
	110	220	6 MHz	0	0	2	72H
	110	220	12 MHz	0	0	1	FEEBH

Figure 13. Timer 1 Generated Commonly Used Baud Rates

More About Mode 0

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

Figure 14 shows a simplified functional diagram of the serial port in Mode 0, and associated timing.

Transmission is initiated by any instruction that uses SBUF as a destination register. The “write to SBUF” signal at S6P2 also loads a 1 into the 9th position of the transmit shift register and tells the TX Control block to commence a transmission. The internal timing is such that one full machine cycle will elapse between “write to SBUF” and activation of SEND.

SEND enables the output of the shift register to the alternate output function line of P3.0 and also enable SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK is low during S3, S4, and S5 of every machine cycle, and high during S6, S1, and S2. At S6P2 of every machine cycle in which SEND is active, the contents of the transmit shift are shifted to the right one position.

As data bits shift out to the right, zeros come in from the left. When the MSB of the data byte is at the output position of the shift register, then the 1 that was initially loaded into the 9th position, is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control block to do one last shift and then deactivate SEND and set T1. Both of these actions occur at S1P1 of the 10th machine cycle after “write to SBUF.”

Reception is initiated by the condition REN = 1 and R1 = 0. At S6P2 of the next machine cycle, the RX Control unit writes the bits 11111110 to the receive shift register, and in the next clock phase activates RECEIVE.

RECEIVE enable SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK makes transitions at S3P1 and S6P1 of every machine cycle. At S6P2 of every machine cycle in which RECEIVE is active, the contents of the receive shift register are shifted to the left one position. The value that comes in from the right is the value that was sampled at the P3.0 pin at S5P2 of the same machine cycle.

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

More About Mode 1

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

Figure 15 shows a simplified functional diagram of the serial port in Mode 1, and associated timings for transmit receive.

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

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

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

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

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

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

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

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

More About Modes 2 and 3

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

Figures 16 and 17 show a functional diagram of the serial port in Modes 2 and 3. The receive portion is exactly the same as in Mode 1. The transmit portion differs from Mode 1 only in the 9th bit of the transmit shift register.

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

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

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

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

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

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

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

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

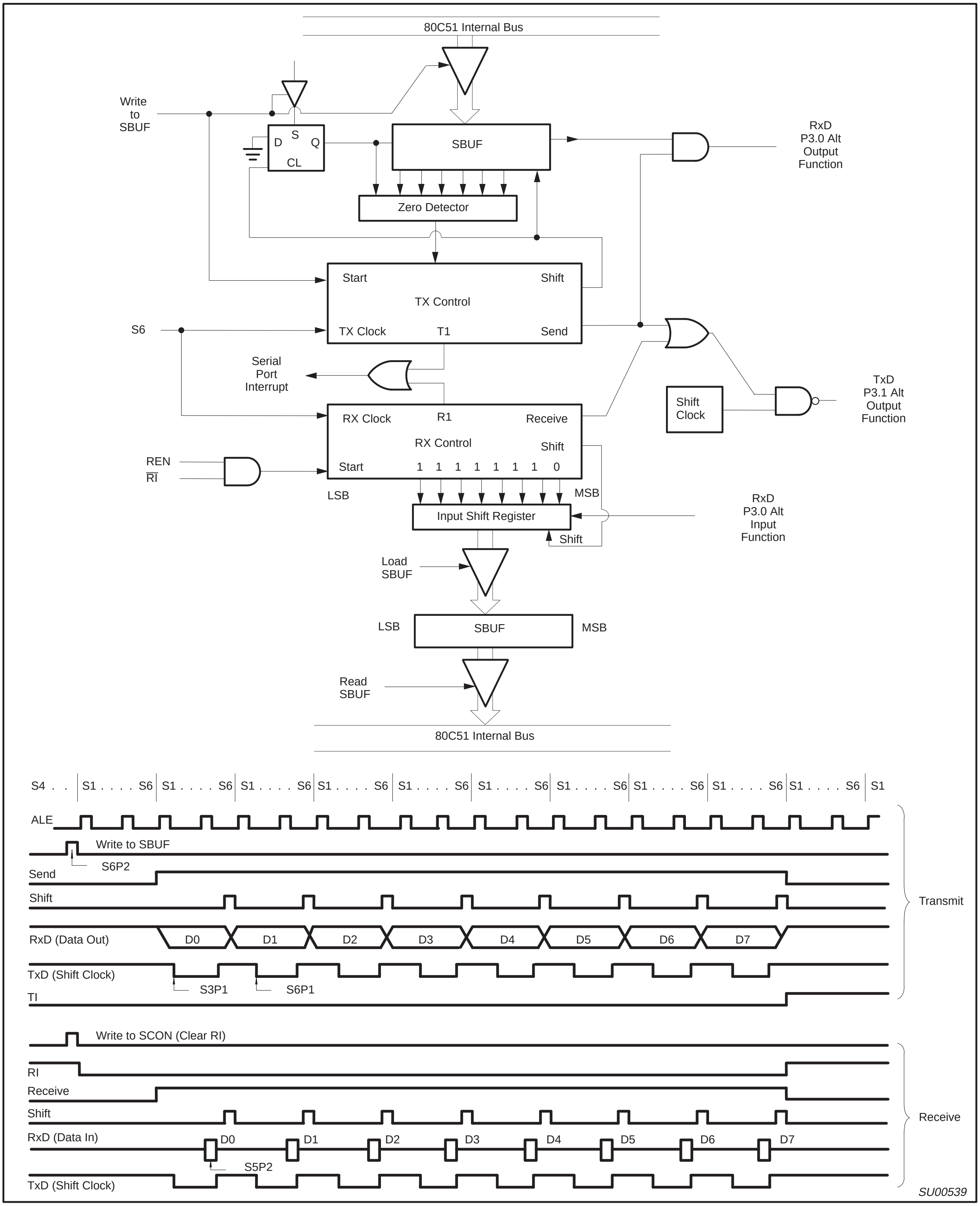


Figure 14. Serial Port Mode 0

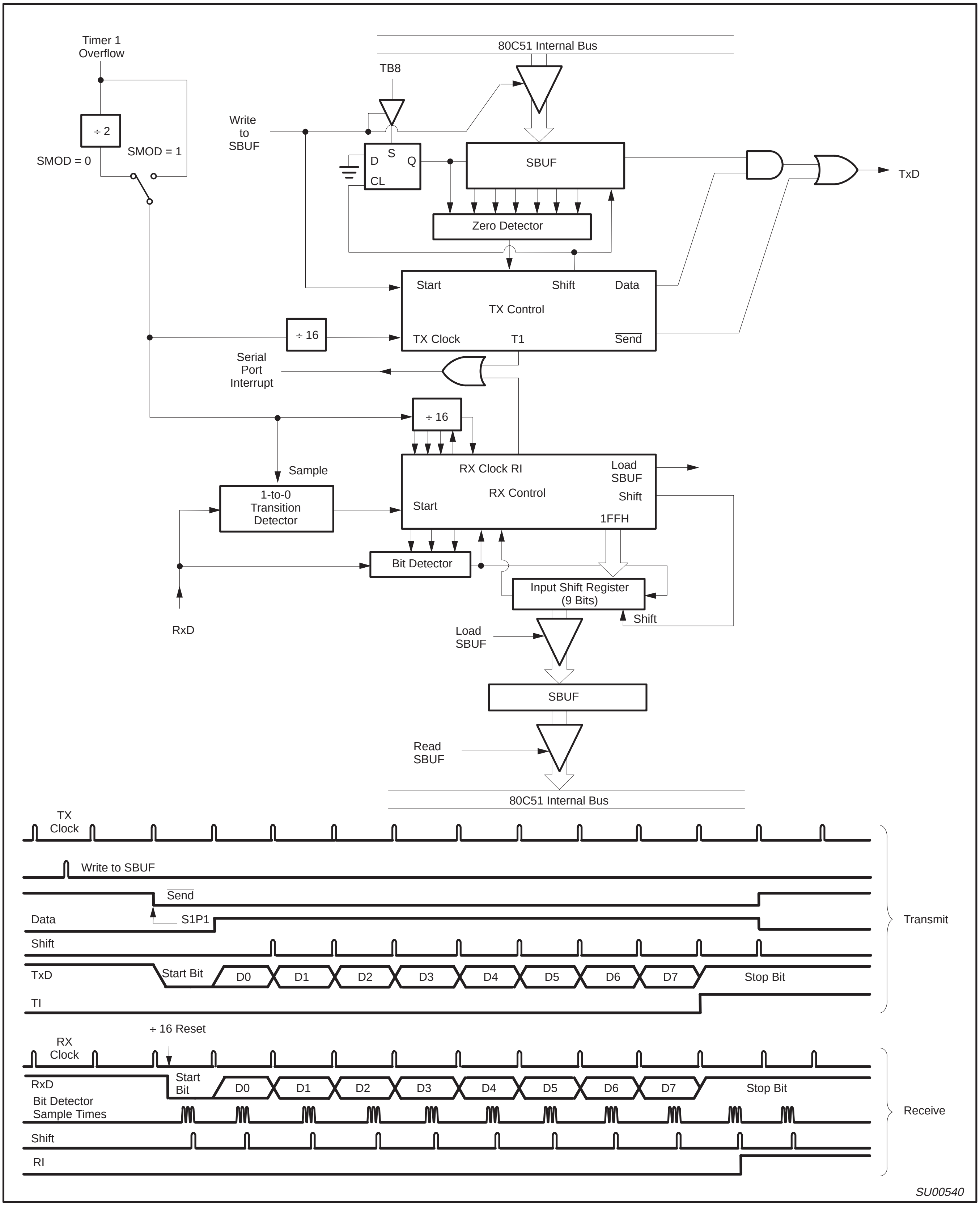


Figure 15. Serial Port Mode 1

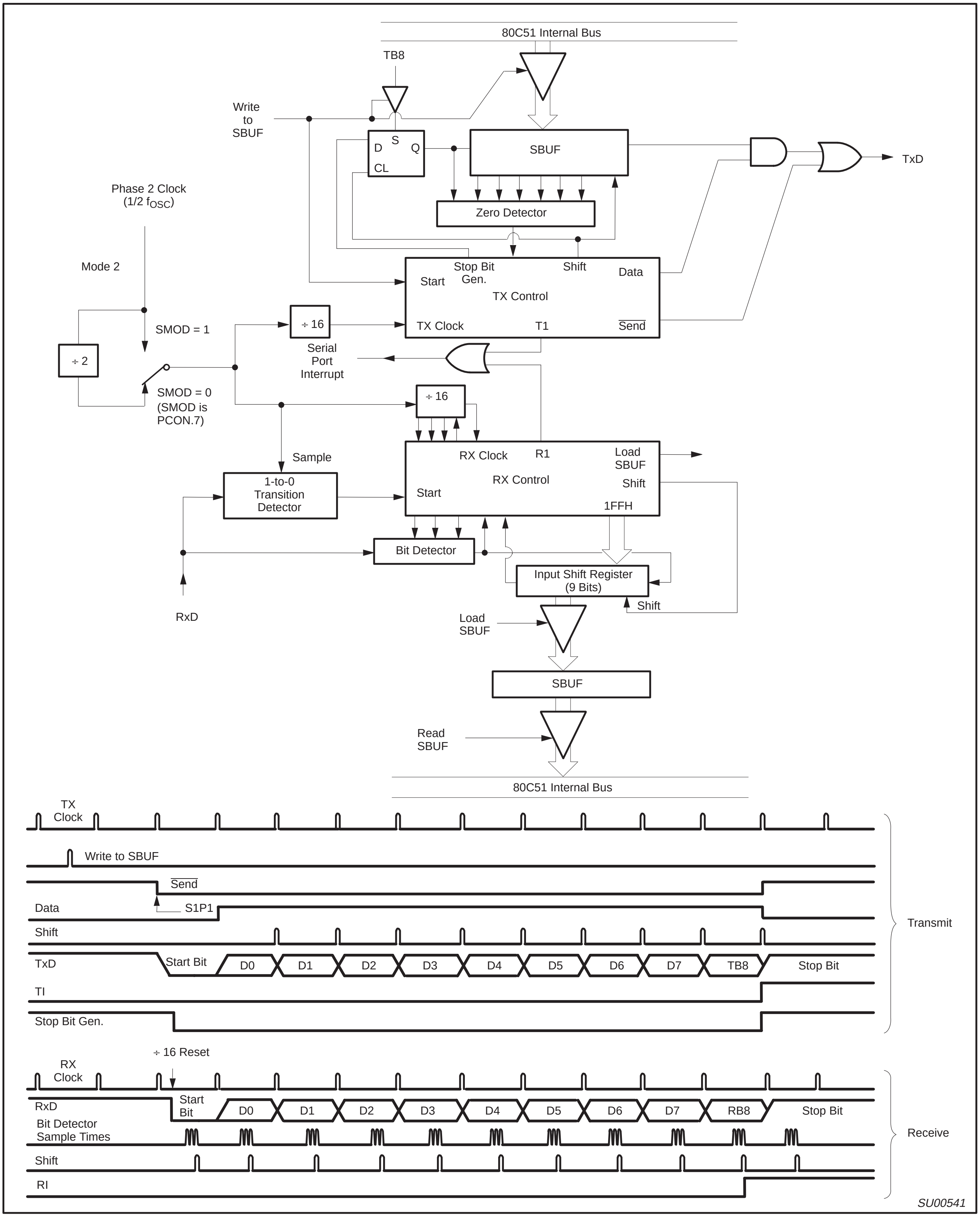


Figure 16. Serial Port Mode 2

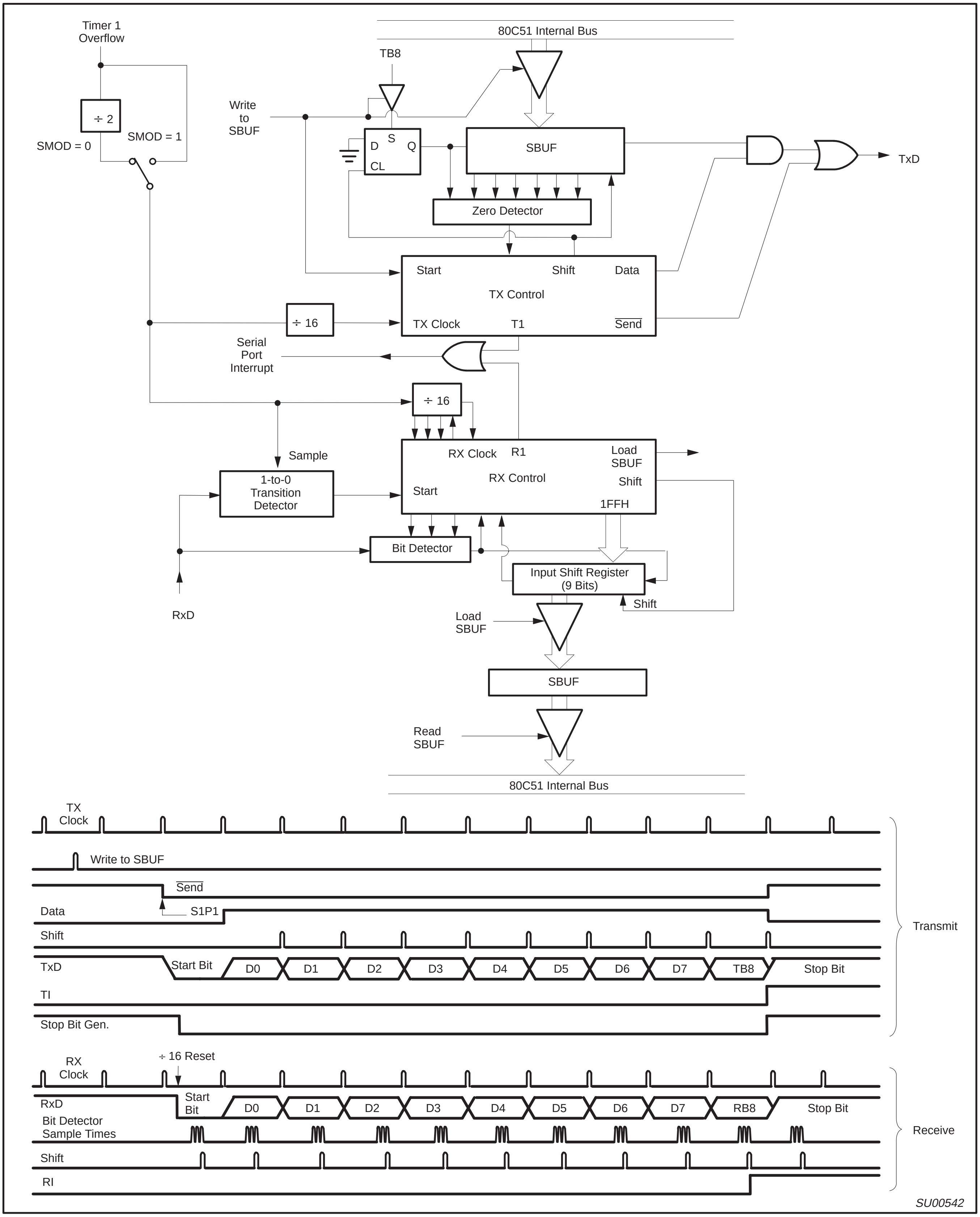


Figure 17. Serial Port Mode 3

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 SCON register. The FE bit shares the SCON.7 bit with SM0 and the function of SCON.7 is determined by PCON.6 (SMOD0) (see Figure 18). If SMOD0 is set then SCON.7 functions as FE. SCON.7 functions as SM0 when SMOD0 is cleared. When used as FE SCON.7 can only be cleared by software. Refer to Figure 19.

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 SCON. 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 20.

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 b 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.

SCON Address = 98H Reset Value = 0000 0000B

Bit Addressable

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

(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

Symbol	Function
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

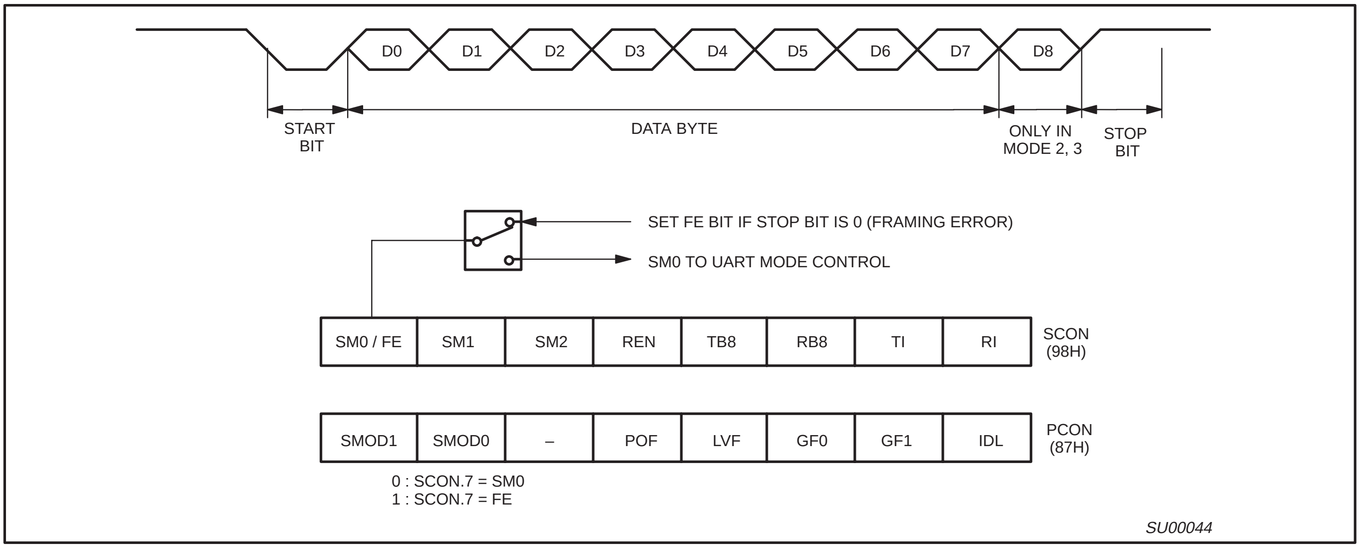


Figure 19. UART Framing Error Detection

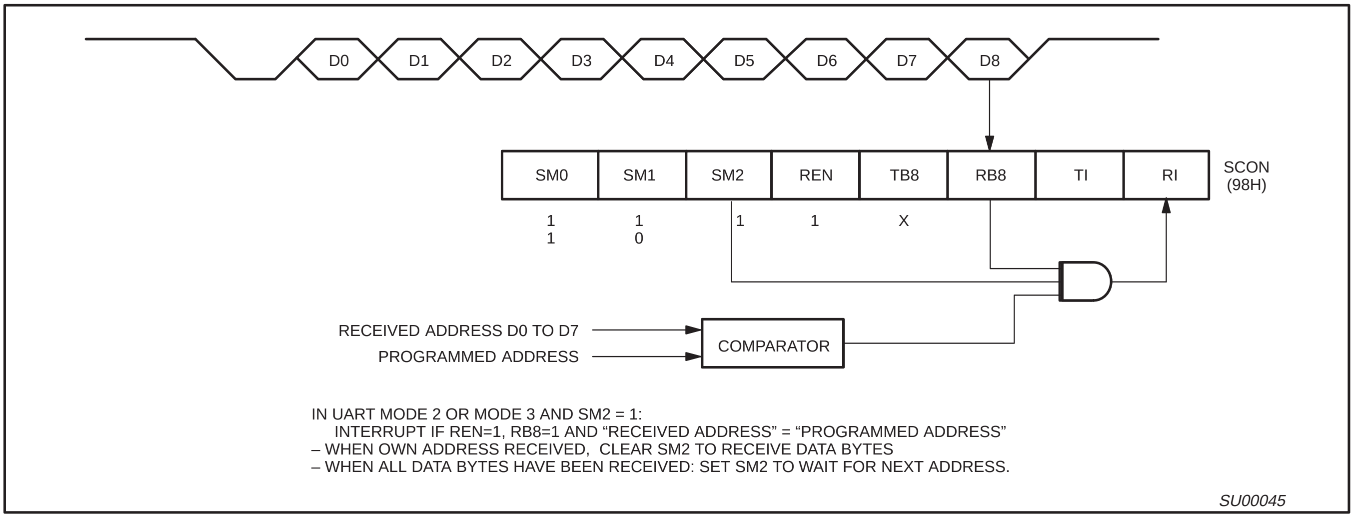


Figure 20. UART Multiprocessor Communication, Automatic Address Recognition

Interrupt Priority Structure

The P89C51RA2/RB2/RC2/RD2xx has a 7 source four-level interrupt structure (see Table 7).

There are 3 SFRs associated with the four-level interrupt. They are the IE, IP, and IPH. (See Figures 21, 22, and 23.) 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 23.

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 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 7. Interrupt Table

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

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

	7	6	5	4	3	2	1	0
IE (0A8H)	EA	EC	ET2	ES	ET1	EX1	ET0	EX0

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

BIT	SYMBOL	FUNCTION
IE.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.
IE.6	EC	PCA interrupt enable bit
IE.5	ET2	Timer 2 interrupt enable bit.
IE.4	ES	Serial Port interrupt enable bit.
IE.3	ET1	Timer 1 interrupt enable bit.
IE.2	EX1	External interrupt 1 enable bit.
IE.1	ET0	Timer 0 interrupt enable bit.
IE.0	EX0	External interrupt 0 enable bit.

SU01290

Figure 21. IE Registers

	7	6	5	4	3	2	1	0
IP (0B8H)	–	PPC	PT2	PS	PT1	PX1	PT0	PX0

Priority Bit = 1 assigns high priority
Priority Bit = 0 assigns low priority

BIT	SYMBOL	FUNCTION
IP.7	–	–
IP.6	PPC	PCA interrupt priority bit
IP.5	PT2	Timer 2 interrupt priority bit.
IP.4	PS	Serial Port interrupt priority bit.
IP.3	PT1	Timer 1 interrupt priority bit.
IP.2	PX1	External interrupt 1 priority bit.
IP.1	PT0	Timer 0 interrupt priority bit.
IP.0	PX0	External interrupt 0 priority bit.

SU01291

Figure 22. IP Registers

	7	6	5	4	3	2	1	0
IPH (B7H)	–	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H

Priority Bit = 1 assigns higher priority
Priority Bit = 0 assigns lower priority

BIT	SYMBOL	FUNCTION
IPH.7	–	–
IPH.6	PPCH	PCA interrupt priority bit
IPH.5	PT2H	Timer 2 interrupt priority bit high.
IPH.4	PSH	Serial Port interrupt priority bit high.
IPH.3	PT1H	Timer 1 interrupt priority bit high.
IPH.2	PX1H	External interrupt 1 priority bit high.
IPH.1	PT0H	Timer 0 interrupt priority bit high.
IPH.0	PX0H	External interrupt 0 priority bit high.

SU01292

Figure 23. IPH Registers

Reduced EMI Mode

The AO bit (AUXR.0) in the AUXR register when set disables the ALE output unless the CPU needs to perform an off-chip memory access.

Reduced EMI Mode

AUXR (8EH)

7	6	5	4	3	2	1	0
–	–	–	–	–	–	EXTRAM	AO

AUXR.1 EXTRAM
AUXR.0 AO

See more detailed description in Figure 38.

Dual DPTR

The dual DPTR structure (see Figure 24) is a way by which the chip will specify the address of an external data memory location. There are two 16-bit DPTR registers that address the external memory, and a single bit called DPS = AUXR1/bit0 that allows the program code to switch between them.

- New Register Name: AUXR1#
- SFR Address: A2H
- Reset Value: xxxxxxx0B

AUXR1 (A2H)

7	6	5	4	3	2	1	0
–	–	ENBOOT	–	GF2	0	–	DPS

Where:
DPS = AUXR1/bit0 = Switches between DPTR0 and DPTR1.

Select Reg	DPS
DPTR0	0
DPTR1	1

The DPS bit status should be saved by software when switching between DPTR0 and DPTR1.

The GF2 bit is a general purpose user-defined flag. Note that bit 2 is not writable and is always read as a zero. This allows the DPS bit to be quickly toggled simply by executing an INC AUXR1 instruction without affecting the GF2 bit.

The ENBOOT bit determines whether the BOOTROM is enabled or disabled. This bit will automatically be set if the status byte is non zero during reset or $\overline{PSEN}$ is pulled low, ALE floats high, and EA > V_{IH} on the falling edge of reset. Otherwise, this bit will be cleared during reset.

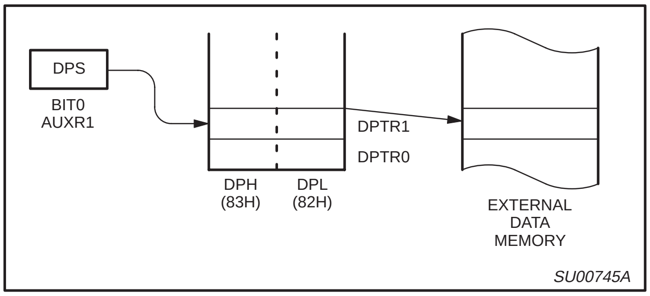


Figure 24.

DPTR Instructions

The instructions that refer to DPTR refer to the data pointer that is currently selected using the AUXR1/bit 0 register. The six instructions that use the DPTR are as follows:

INC DPTR	Increments the data pointer by 1
MOV DPTR, #data16	Loads the DPTR with a 16-bit constant
MOV A, @ A+DPTR	Move code byte relative to DPTR to ACC
MOVX A, @ DPTR	Move external RAM (16-bit address) to ACC
MOVX @ DPTR , A	Move ACC to external RAM (16-bit address)
JMP @ A + DPTR	Jump indirect relative to DPTR

The data pointer can be accessed on a byte-by-byte basis by specifying the low or high byte in an instruction which accesses the SFRs. See *Application Note AN458* for more details.

Programmable Counter Array (PCA)

The Programmable Counter Array available on the P89C51RA2/RB2/RC2/RD2xx 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 25.

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 28):

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 26.

The watchdog timer function is implemented in module 4 (see Figure 35).

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 29). 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 27.

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 30). 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 31 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.

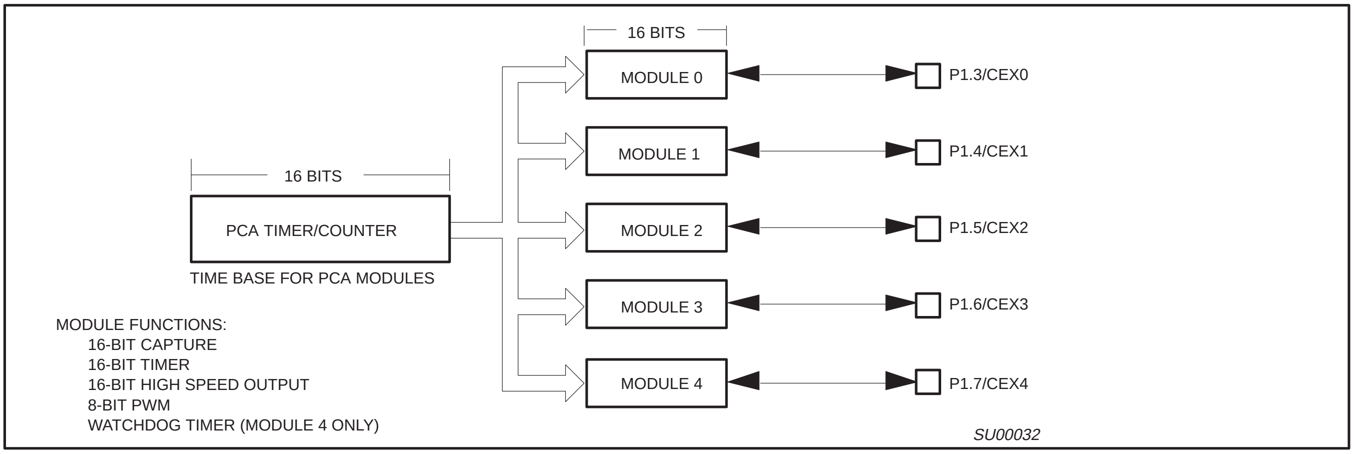


Figure 25. Programmable Counter Array (PCA)

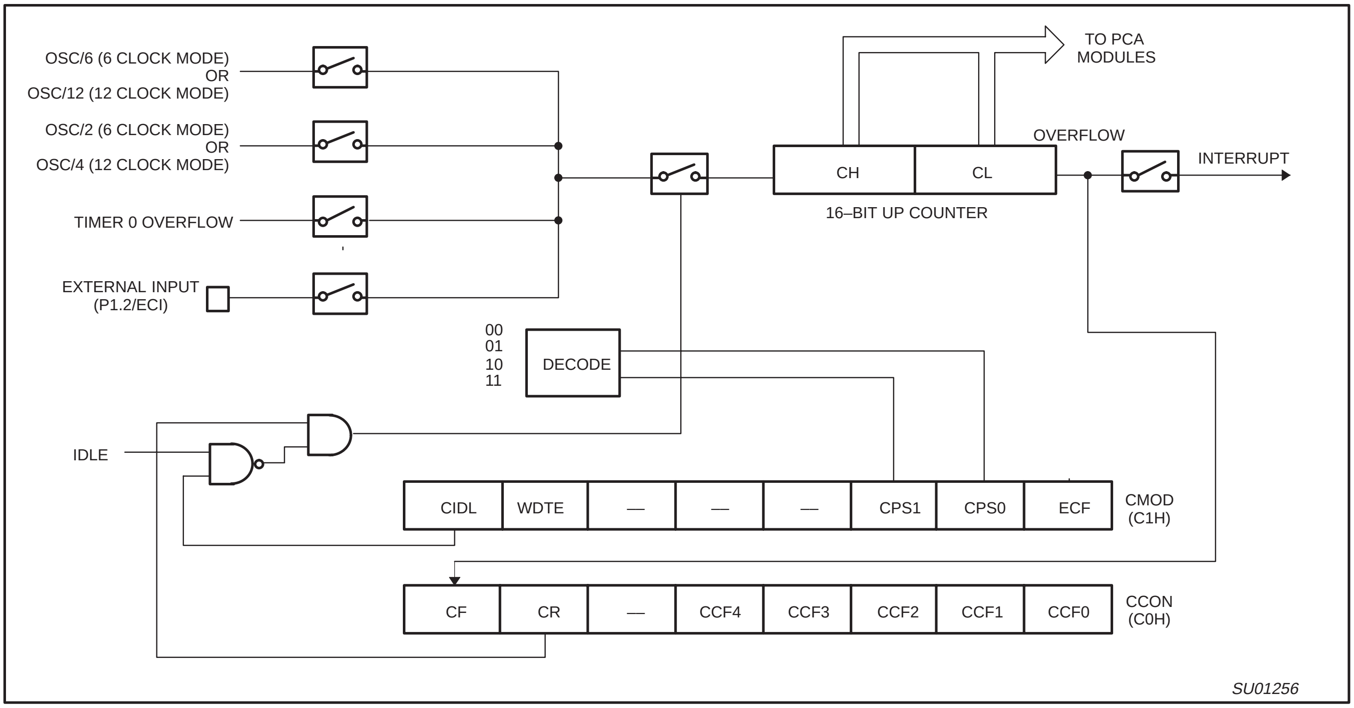


Figure 26. PCA Timer/Counter

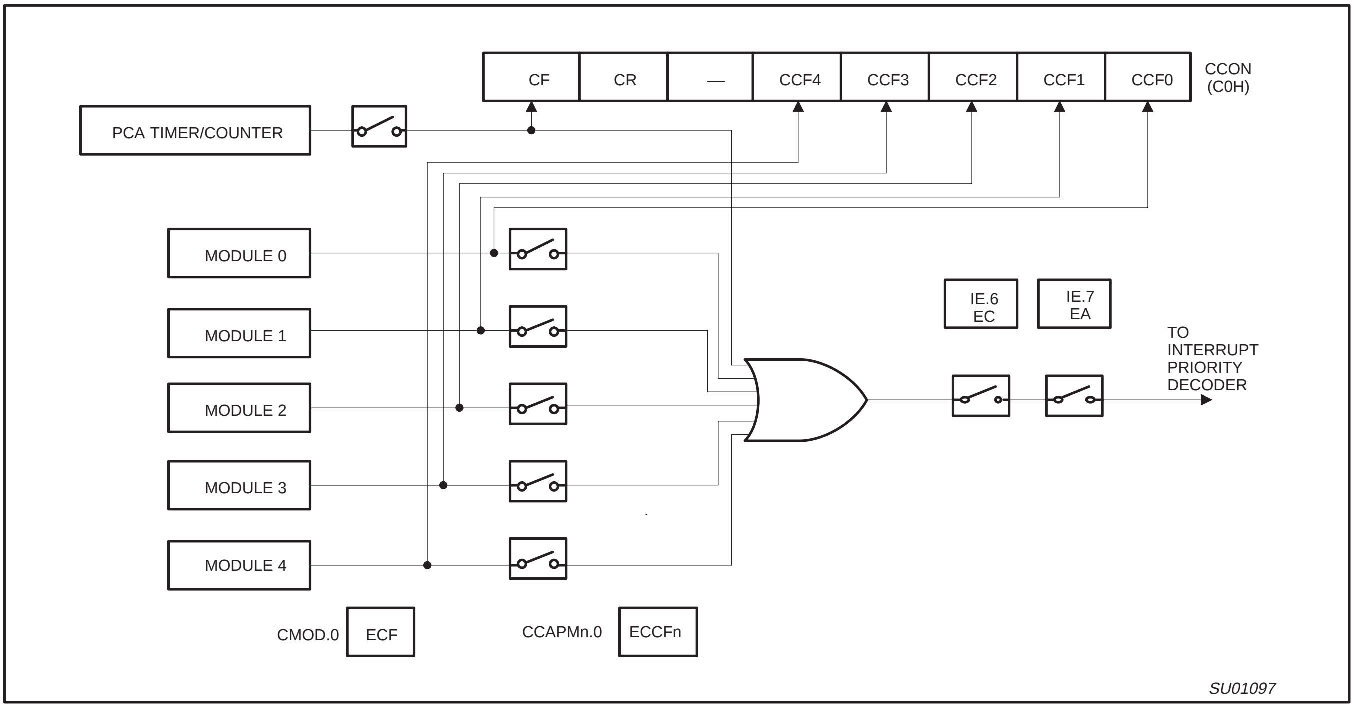


Figure 27. PCA Interrupt System

CMOD Address = D9H Reset Value = 00XX X000B

Bit:	7	6	5	4	3	2	1	0
	CIDL	**WDTE**	**–**	**–**	**–**	**CPS1**	**CPS0**	**ECF**

Symbol	**Function**
CIDL	Counter Idle control: CIDL = 0 programs the PCA Counter to continue functioning during idle Mode. CIDL = 1 programs it to be gated off during idle.
WDTE	Watchdog Timer Enable: WDTE = 0 disables Watchdog Timer function on PCA Module 4. WDTE = 1 enables it.
–	Not implemented, reserved for future use.*
CPS1	PCA Count Pulse Select bit 1.
CPS0	PCA Count Pulse Select bit 0.

CPS1	**CPS0**	**Selected PCA Input****	
0	0	0	Internal clock, $f_{OSC}/6$ in 6-clock mode ($f_{OSC}/12$ in 12-clock mode)
0	1	1	Internal clock, $f_{OSC}/2$ in 6-clock mode ($f_{OSC}/4$ in 12-clock mode)
1	0	2	Timer 0 overflow
1	1	3	External clock at ECI/P1.2 pin (max. rate = $f_{OSC}/4$ in 6-clock mode, $f_{OCS}/8$ in 12-clock mode)

Symbol	**Function**
ECF	PCA Enable Counter Overflow interrupt: ECF = 1 enables CF bit in CCON to generate an interrupt. ECF = 0 disables that function of CF.

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

SU01318

Figure 28. CMOD: PCA Counter Mode Register

CCON Address = D8H Reset Value = 00X0 0000B

Bit Addressable

Bit:	7	6	5	4	3	2	1	0
	CF	**CR**	**–**	**CCF4**	**CCF3**	**CCF2**	**CCF1**	**CCF0**

Symbol	**Function**
CF	PCA Counter Overflow flag. Set by hardware when the counter rolls over. CF flags an interrupt if bit ECF in CMOD is set. CF may be set by either hardware or software but can only be cleared by software.
CR	PCA Counter Run control bit. Set by software to turn the PCA counter on. Must be cleared by software to turn the PCA counter off.
–	Not implemented, reserved for future use*.
CCF4	PCA Module 4 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.
CCF3	PCA Module 3 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.
CCF2	PCA Module 2 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.
CCF1	PCA Module 1 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.
CCF0	PCA Module 0 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.

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

SU01319

Figure 29. CCON: PCA Counter Control Register

CCAPMn Address CCAPM0 0DAH
CCAPM1 0DBH
CCAPM2 0DCH
CCAPM3 0DDH
CCAPM4 0DEH

Reset Value = X000 0000B

Not Bit Addressable

Bit:	7	6	5	4	3	2	1	0
	–	**ECOMn**	**CAPPn**	**CAPNn**	**MATn**	**TOGn**	**PWMn**	**ECCFn**

Symbol	Function
–	Not implemented, reserved for future use*.
ECOMn	Enable Comparator. ECOMn = 1 enables the comparator function.
CAPPn	Capture Positive, CAPPn = 1 enables positive edge capture.
CAPNn	Capture Negative, CAPNn = 1 enables negative edge capture.
MATn	Match. When MATn = 1, a match of the PCA counter with this module's compare/capture register causes the CCFn bit in CCON to be set, flagging an interrupt.
TOGn	Toggle. When TOGn = 1, a match of the PCA counter with this module's compare/capture register causes the CEXn pin to toggle.
PWMn	Pulse Width Modulation Mode. PWMn = 1 enables the CEXn pin to be used as a pulse width modulated output.
ECCFn	Enable CCF interrupt. Enables compare/capture flag CCFn in the CCON register to generate an interrupt.

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

SU01320

Figure 30. CCAPMn: PCA Modules Compare/Capture Registers

–	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	MODULE FUNCTION
X	0	0	0	0	0	0	0	No operation
X	X	1	0	0	0	0	X	16-bit capture by a positive-edge trigger on CEXn
X	X	0	1	0	0	0	X	16-bit capture by a negative trigger on CEXn
X	X	1	1	0	0	0	X	16-bit capture by a transition on CEXn
X	1	0	0	1	0	0	X	16-bit Software Timer
X	1	0	0	1	1	0	X	16-bit High Speed Output
X	1	0	0	0	0	1	0	8-bit PWM
X	1	0	0	1	X	0	X	Watchdog Timer

Figure 31. PCA Module Modes (CCAPMn Register)

PCA Capture Mode

To use one of the PCA modules in the capture mode either one or both of the CCAPM bits CAPN and CAPP for that module must be set. The external CEX input for the module (on port 1) is sampled for a transition. When a valid transition occurs the PCA hardware loads the value of the PCA counter registers (CH and CL) into the module's capture registers (CCAPnL and CCAPnH). If the CCFn bit for the module in the CCON SFR and the ECCFn bit in the CCAPMn SFR are set then an interrupt will be generated. Refer to Figure 32.

16-bit Software Timer Mode

The PCA modules can be used as software timers by setting both the ECOM and MAT bits in the modules CCAPMn register. The PCA timer will be compared to the module's capture registers and when a match occurs an interrupt will occur if the CCFn (CCON SFR) and the ECCFn (CCAPMn SFR) bits for the module are both set (see Figure 33).

High Speed Output Mode

In this mode the CEX output (on port 1) associated with the PCA module will toggle each time a match occurs between the PCA counter and the module's capture registers. To activate this mode the TOG, MAT, and ECOM bits in the module's CCAPMn SFR must be set (see Figure 34).

Pulse Width Modulator Mode

All of the PCA modules can be used as PWM outputs. Figure 35 shows the PWM function. The frequency of the output depends on the source for the PCA timer. All of the modules will have the same frequency of output because they all share the PCA timer. The duty cycle of each module is independently variable using the module's capture register CCAPLn. When the value of the PCA CL SFR is less than the value in the module's CCAPLn SFR the output will be low, when it is equal to or greater than the output will be high. When CL overflows from FF to 00, CCAPLn is reloaded with the value in CCAPHn. the allows updating the PWM without glitches. The PWM and ECOM bits in the module's CCAPMn register must be set to enable the PWM mode.

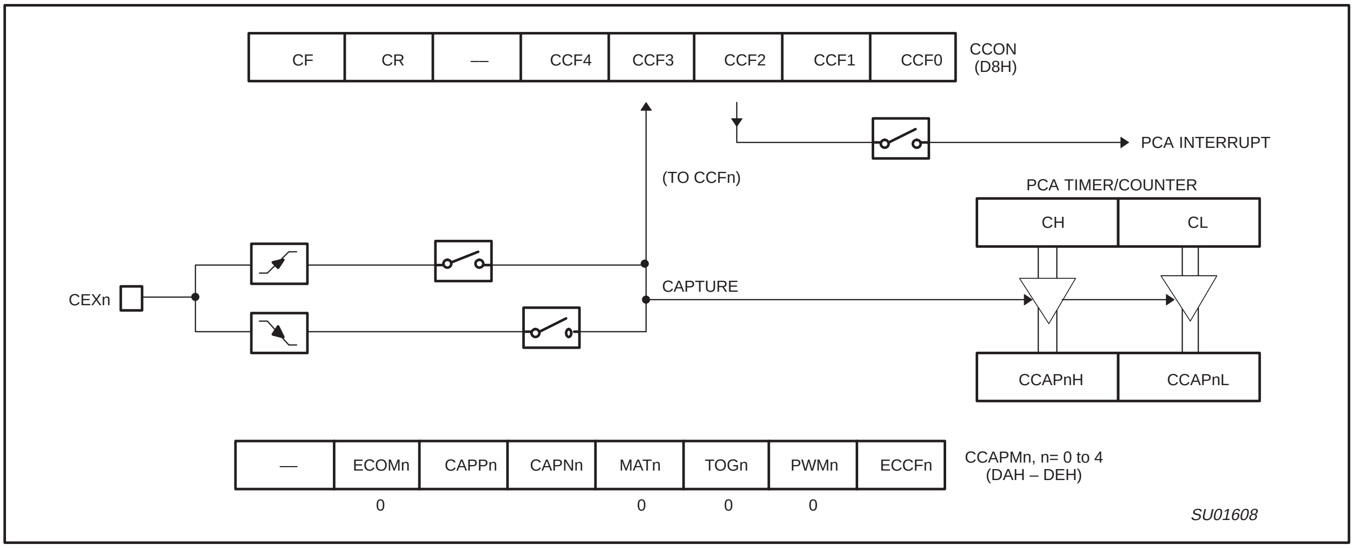


Figure 32. PCA Capture Mode

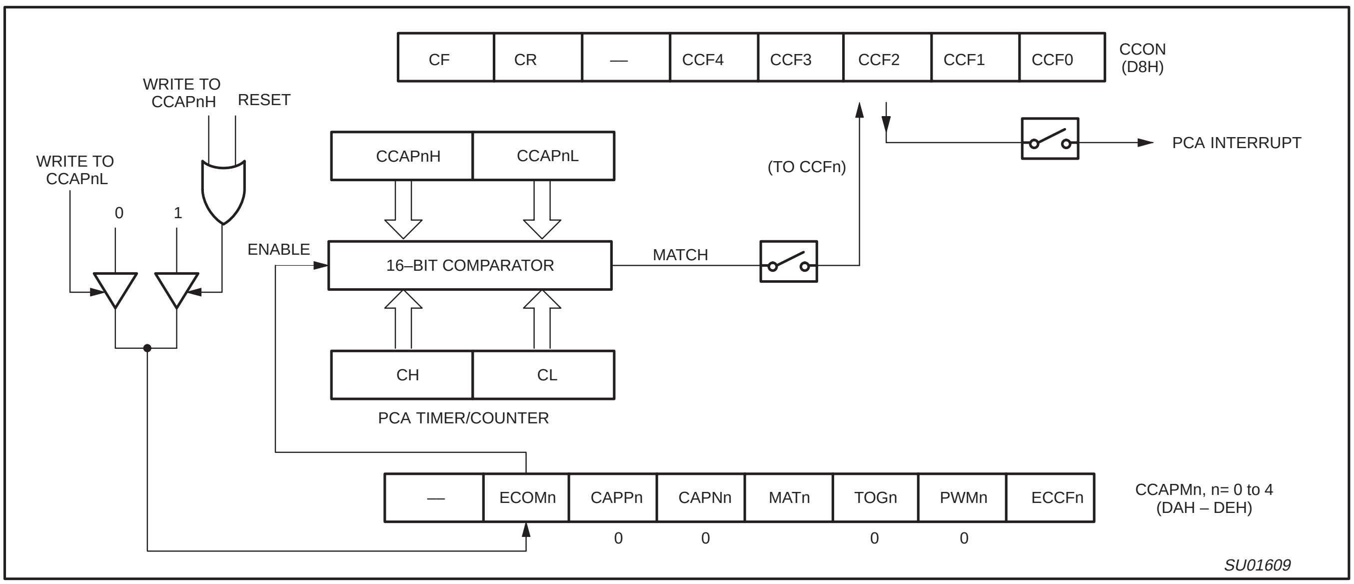


Figure 33. PCA Compare Mode

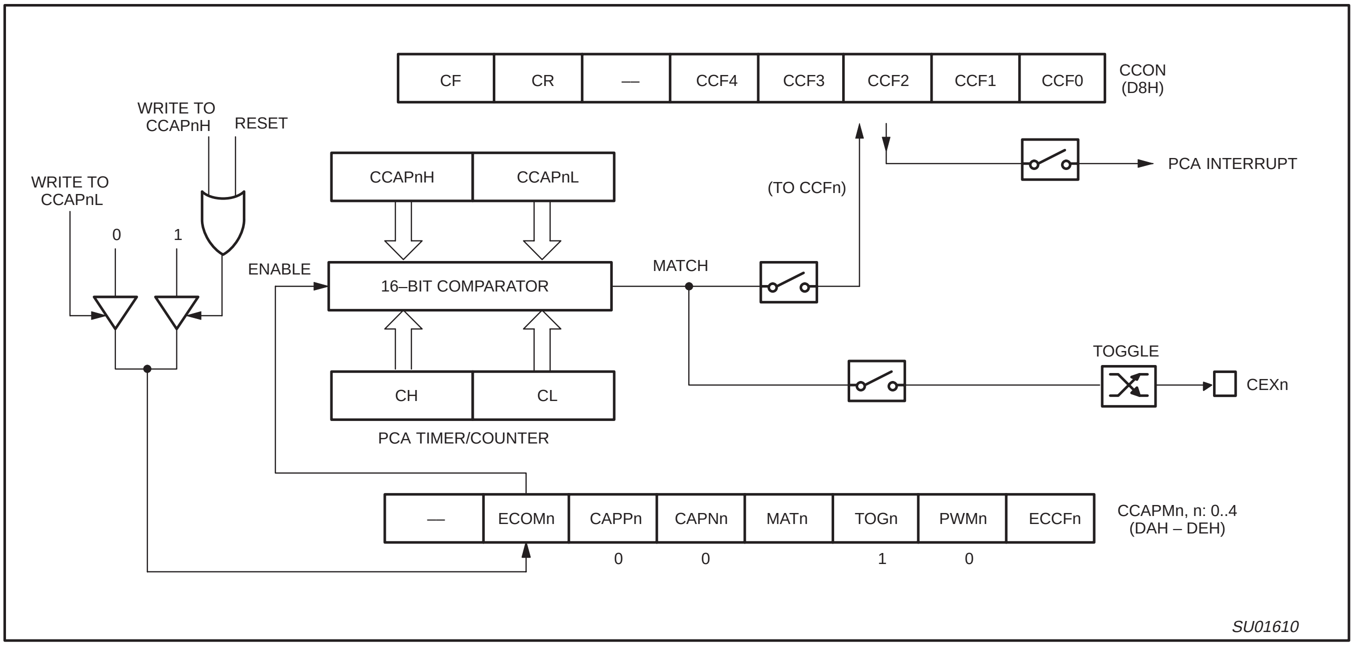


Figure 34. PCA High Speed Output Mode

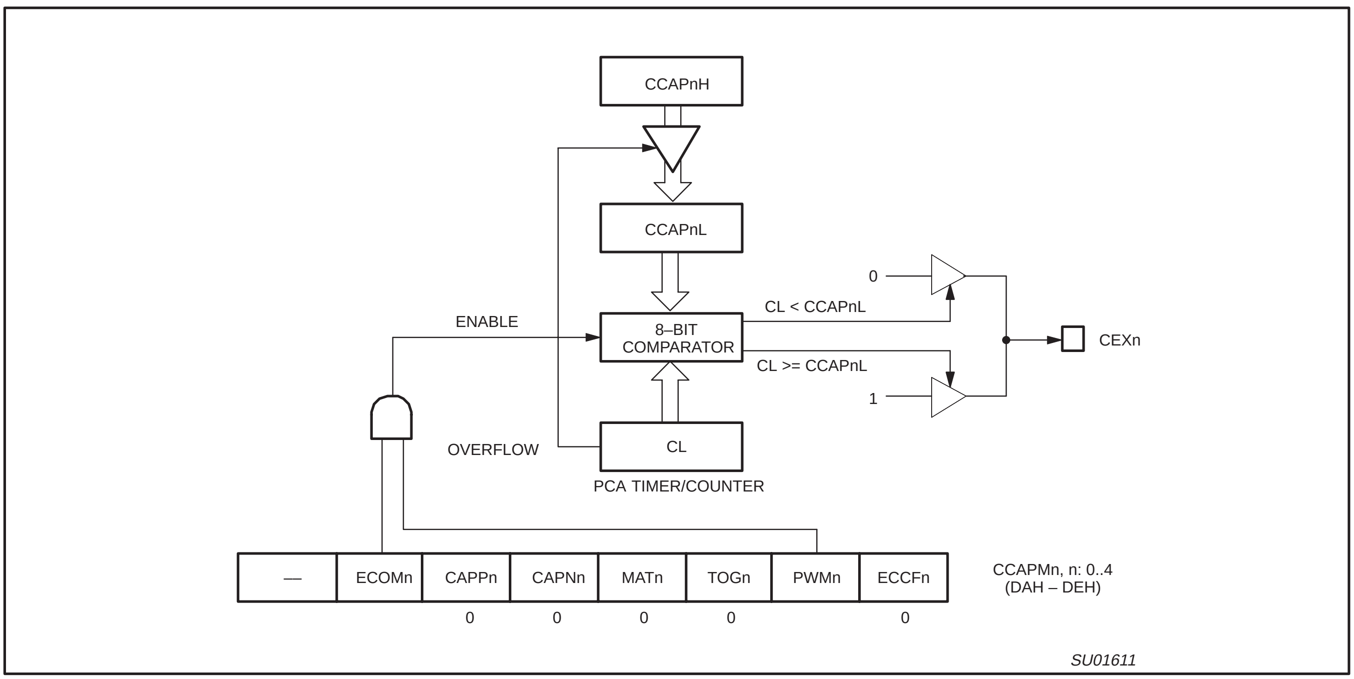


Figure 35. PCA PWM Mode

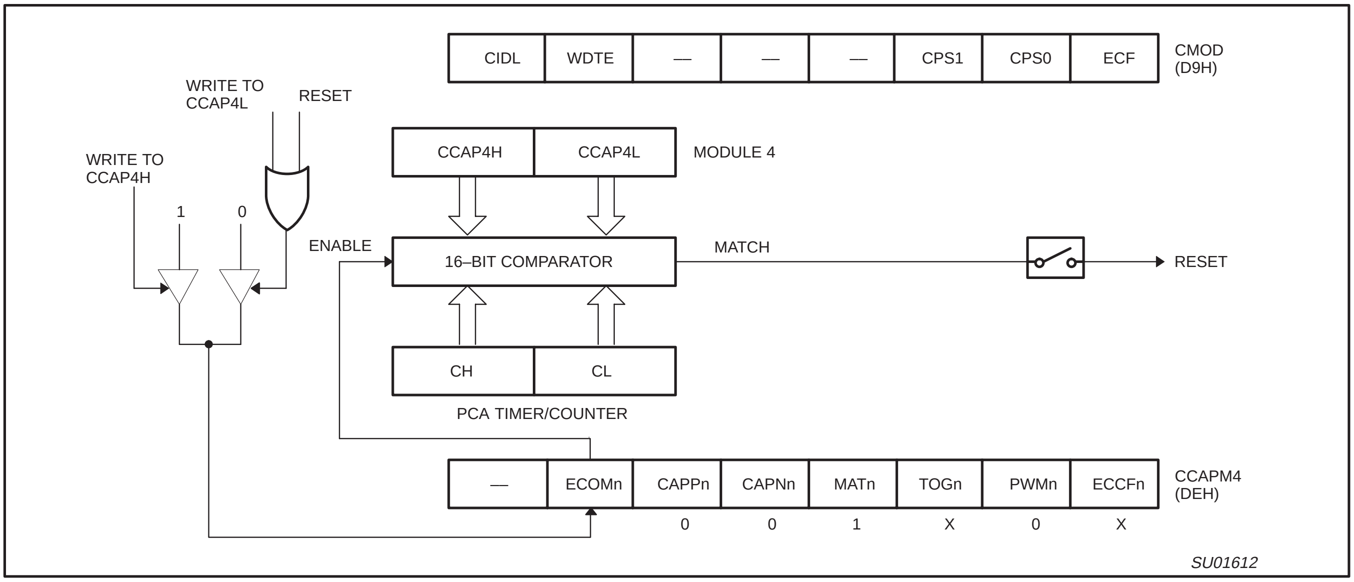


Figure 36. PCA Watchdog Timer mode (Module 4 only)

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 36 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 37 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 then 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 37.

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.

```
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

Expanded Data RAM Addressing

The P89C51RA2/RB2/RC2/RD2xx has internal data memory that is mapped into four separate segments: the lower 128 bytes of RAM, upper 128 bytes of RAM, 128 bytes Special Function Register (SFR), and 256 bytes expanded RAM (ERAM) (768 bytes for the RD2xx).

The four segments are:

1. The Lower 128 bytes of RAM (addresses 00H to 7FH) are directly and indirectly addressable.
2. The Upper 128 bytes of RAM (addresses 80H to FFH) are indirectly addressable only.
3. The Special Function Registers, SFRs, (addresses 80H to FFH) are directly addressable only.
4. The 256/768-bytes expanded RAM (ERAM, 00H – 1FFH/2FFH) are indirectly accessed by move external instruction, MOVX, and with the EXTRAM bit cleared, see Figure 38.

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

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

MOV 0A0H,#data

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

For example:

MOV @R0,acc

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

The ERAM can be accessed by indirect addressing, with EXTRAM bit cleared and MOVX instructions. This part of memory is physically located on-chip, logically occupies the first 256/768 bytes of external data memory in the P89C51RA2/RB2/RC2/89C51RD2.

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

MOVX @R0,acc

where R0 contains 0A0H, accesses the ERAM at address 0A0H rather than external memory. An access to external data memory locations higher than the ERAM will be performed with the MOVX DPTR instructions in the same way as in the standard 80C51, so with P0 and P2 as data/address bus, and P3.6 and P3.7 as write and read timing signals. Refer to Figure 39.

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

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

AUXR Address = 8EH Reset Value = xxxx xx00B

Not Bit Addressable

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

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

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

SU01613

Figure 38. AUXR: Auxiliary Register

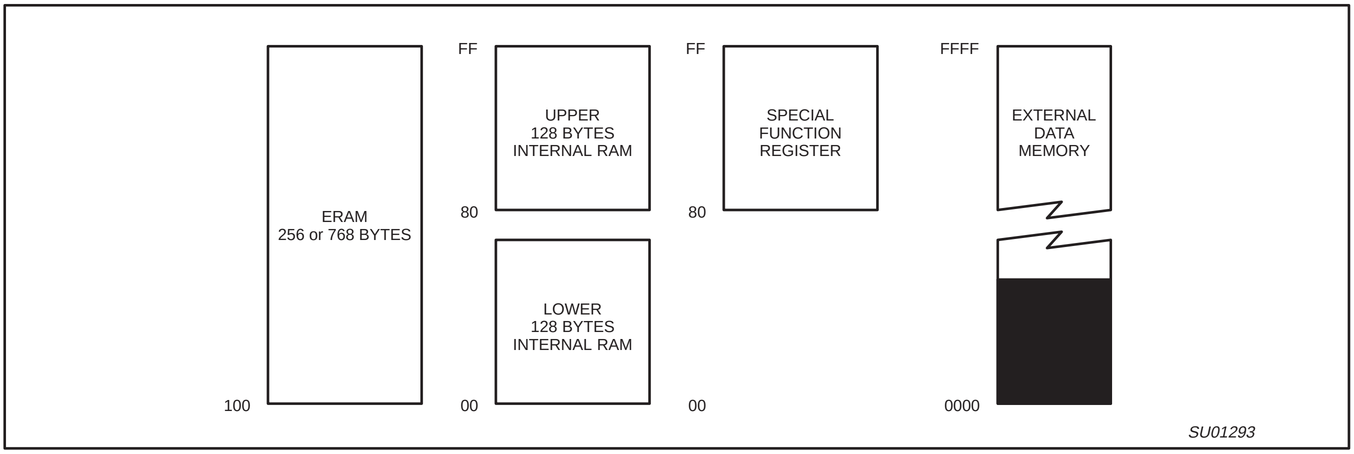


Figure 39. Internal and External Data Memory Address Space with EXTRAM = 0

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.

FLASH EPROM MEMORY

GENERAL DESCRIPTION

The P89C51RA2/RB2/RC2/RD2xx Flash memory augments EPROM functionality with in-circuit electrical erasure and programming. The Flash can be read and written as bytes. The Chip Erase operation will erase the entire program memory. The Block Erase function can erase any Flash block. In-system programming and standard parallel programming are both available. On-chip erase and write timing generation contribute to a user friendly programming interface.

The P89C51RA2/RB2/RC2/RD2xx Flash reliably stores memory contents even after 10,000 erase and program cycles. The cell is designed to optimize the erase and programming mechanisms. In addition, the combination of advanced tunnel oxide processing and low internal electric fields for erase and programming operations produces reliable cycling. The P89C51RA2/RB2/RC2/RD2xx uses a +5 V V_{PP} supply to perform the Program/Erase algorithms.

FEATURES – IN-SYSTEM PROGRAMMING (ISP) AND IN-APPLICATION PROGRAMMING (IAP)

- Flash EPROM internal program memory with Block Erase.
- Internal 1-kbyte fixed BootROM, containing low-level in-system programming routines and a default serial loader. User program can call these routines to perform In-Application Programming (IAP). The BootROM can be turned off to provide access to the full 64-kbyte Flash memory.
- Boot Vector allows user provided Flash loader code to reside anywhere in the Flash memory space. This configuration provides flexibility to the user.
- Default loader in BootROM allows programming via the serial port without the need for a user provided loader.
- Up to 64-kbyte external program memory if the internal program memory is disabled ($\overline{EA}$ = 0).
- Programming and erase voltage +5 V (+12 V tolerant).
- Read/Programming/Erase using ISP/IAP:
  - Byte Programming (8 μs).
  - Typical quick erase times:
    Block Erase (4 kbyte) in 3 seconds.
    Full Chip Erase:
    – RD2xx (64K) in 11 seconds
    – RC2 (32K) in 7 seconds
    – RB2 (16K) in 5 seconds
    – RA2 (4K) in 4 seconds
- Parallel programming with 87C51 compatible hardware interface to programmer.
- In-system programming (ISP).
- In-application programming (IAP).
- Programmable security for the code in the Flash.
- 10,000 minimum erase/program cycles for each byte.
- 10-year minimum data retention.

FLASH PROGRAMMING AND ERASURE

In general, there are three methods of erasing or programming of the Flash memory that may be used. First, the Flash may be programmed or erased in the end-user application by calling low-level routines through entry point in the BootROM. The end-user application, though, must be executing code from a different block than the block that is being erased or programmed. Second, the on-chip ISP boot loader may be invoked. This ISP boot loader will, in turn, call low-level routines through the common entry point in the BootROM that can be used by end-user applications. Third, the Flash may be programmed or erased using parallel method by using a commercially available EPROM programmer. The parallel programming method used by these devices is similar to that used by EPROM 87C51, but it is not identical, and the commercially available programmer will need to have support for these devices.

FLASH MEMORY SPACES

Flash User Code Memory Organization

The P89C51RA2/RB2/RC2/RD2xx contains 8KB/16KB/32KB/64KB Flash user code program memory organized into 4-kbyte blocks. ISP and IAP BootROM routines will support the new 4-kbyte block sizes through additional block number assignments while maintaining compatibility with previous 8-kbyte and 16-kbyte block assignments. This memory space is programmable via IAP, ISP, and parallel modes.

Status Byte/Boot Vector Block

This device includes a 4-kbyte block which contains the Status Byte and Boot Vector (Status Byte Block) . The Status Byte and Boot Vector are programmable via IAP, ISP, and parallel modes. Note that erasing of either the Status Byte and Boot Vector will erase the entire contents of this block. Thus the Status Byte and Boot Vector are erased together but are programmable separately.

Security & User Configuration Block

This device includes a 4-kbyte block (Security Block) which contains the Security Bits, the 6-clock/12-clock Flash-based clock mode bit FX2, and 4095 user programmable bytes. This block is programmable via IAP, ISP, and parallel modes. Security bits will prevent, as required, parallel programmers from reading or writing, however, IAP or ISP inhibitions will be software controlled. This block may only be erased using full-chip erase functions in ISP, IAP, or parallel mode. This security feature protects against software piracy and prevents the contents of the Flash from being read. The Security bits are located in the Flash. There are three programmable security bits that will provide different levels of protection for the on-chip code and data (See Table 11). The 4095 user programmable bytes are not part of user code memory are intended to be programmed or read through IAP, ISP, or parallel programmer functions.

The 6-clock/12-clock Flash-based clock mode bit FX2 will be latched at power-on. This allows the bit to be changed via IAP or ISP and delay taking effect until the next reset. This avoids changing baud rates during ISP operations.

Boot ROM

When the microcontroller programs its Flash memory, all of the low level details are handled by code that is contained in a 1-kbyte

BootROM that is shadowed over a portion of the user code memory space. A user program simply calls the common entry point with appropriate parameters in the BootROM to accomplish the desired operation. BootROM operations include: erase block, program byte, verify byte, program security bit, etc. The BootROM overlays the program memory space at the top of the address space from FC00 to FFFF hex, when it is enabled. The BootROM may be turned off so that the upper 1 kbyte of user program memory is accessible for execution.

Clock Mode

The clock mode feature sets operating frequency to be 1/12 or 1/6 of the oscillator frequency. The clock mode configuration bit, FX2, is located in the Security Block (See Table 8). FX2, when programmed, will override the SFR clock mode bit (X2) in the CKCON register. If FX2 is erased, then the SFR bit (X2) may be used to select between 6-clock and 12-clock mode.

Table 8.

CLOCK MODE CONFIG BIT (FX2)	X2 bit in CKCON	DESCRIPTION
erased	0	12-clock mode (default)
erased	1	6-clock mode
programmed	x	6-clock mode

NOTE:
1. Default clock mode after ChipErase is set to SFR selection.

FLASH MEMORY SPACES

Flash User Code Memory Organization

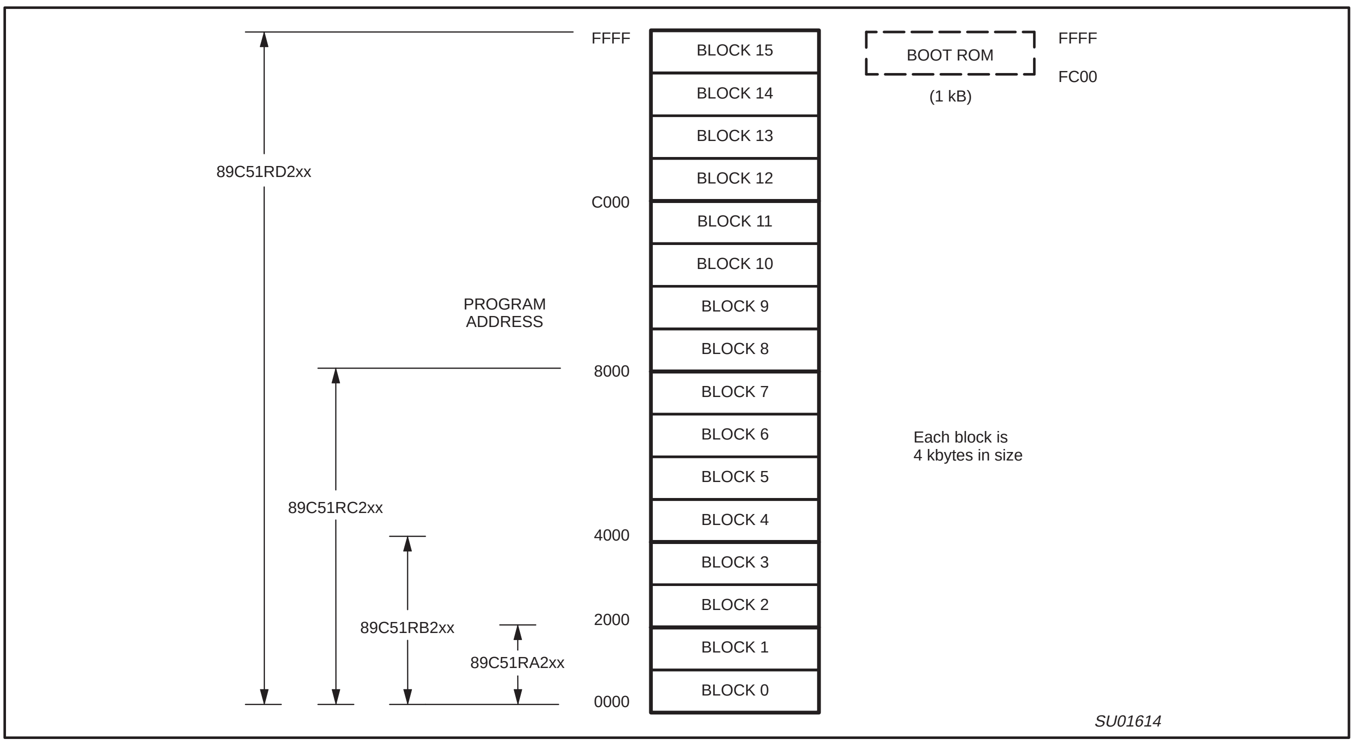


Figure 40. Flash Memory Configurations

Power-On Reset Code Execution

The P89C51RA2/RB2/RC2/RD2xx contains two special Flash registers: the BOOT VECTOR and the STATUS BYTE. At the falling edge of reset, the P89C51RA2/RB2/RC2/RD2xx examines the contents of the Status Byte. If the Status Byte is set to zero, power-up execution starts at location 0000H, which is the normal start address of the user's application code. When the Status Byte is set to a value other than zero, the contents of the Boot Vector is used as the high byte of the execution address and the low byte is set to 00H. The factory default setting is 0FCH, corresponds to the address 0FC00H for the factory masked-ROM ISP boot loader. A custom boot loader can be written with the Boot Vector set to the custom boot loader.

NOTE: When erasing the Status Byte or Boot Vector, both bytes are erased at the same time. It is necessary to reprogram the Boot Vector after erasing and updating the Status Byte.

Hardware Activation of the Boot Loader

The boot loader can also be executed by holding $\overline{PSEN}$ LOW, $\overline{EA}$ 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.

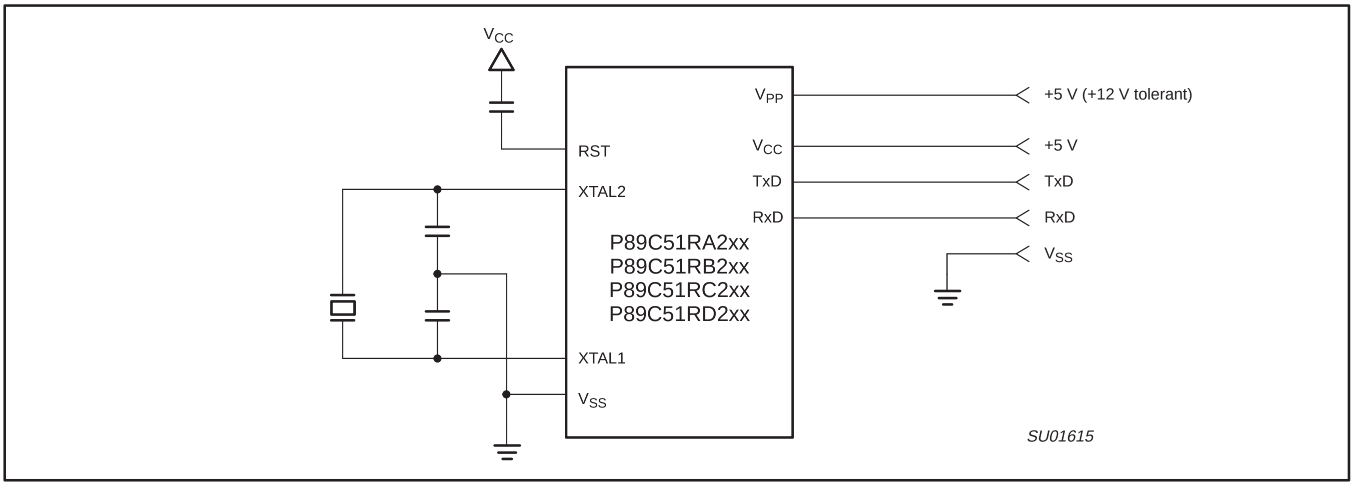


Figure 41. In-System Programming with a Minimum of Pins

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.

Table 9. Intel-Hex Records Used by In-System Programming

RECORD TYPE	COMMAND/DATA FUNCTION

00	Program Data
	`:nnaaaa00dd....ddcc`
	Where:
	`nn = number of bytes (hex) in record`
	`aaaa = memory address of first byte in record`
	`dd....dd = data bytes`
	`cc = checksum`
	Example:
	`:10008000AF5F67F0602703E0322CFA92007780C3FD`

01	End of File (EOF), no operation
	`:xxxxxx01cc`
	Where:
	`xxxxxx = required field, but value is a “don’t care”`
	`cc = checksum`
	Example:
	`:00000001FF`

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 8K/16K Code Blocks)
	`ff = 01`
	`ss = block code as shown below:`
	`block 0, 0k to 8k, 00H`
	`block 1, 8k to 16k, 20H (RB2, RC2, RD2)`
	`block 2, 16k to 32k, 40H (RC2, RD2)`
	`block 3, 32k to 48k, 80H (RD2 only)`
	`block 4, 48k to 64k, C0H (RD2 only)`
	Example:
	`:0200000301C03A 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`
	`02 program FX2 bit (dd = 80)`
	`dd = data`
	Example 1:
	`:030000030601FCF7 program boot vector with 0FCH`
	Example 2:
	`:0300000306028072 program FX2 bit (select 12-clock mode)`

RECORD TYPE	COMMAND/DATA FUNCTION
03 (Cont.)	Subfunction Code = 07 (Full Chip Erase) Erases all blocks, security bits, and sets status byte and boot vector to default values ff = 07 ss = don’t care dd = don’t care Example: :0100000307F5 full chip erase Subfunction Code = 0C (Erase 4K Blocks) ff = 0C ss = block code as shown below: Block 0 , 0k~4k , 00H Block 1 , 4k~8k , 10H Block 2 , 8k~12k , 20H (only available on RD2 / RC2 / RB2) Block 3 , 12k~16k , 30H (only available on RD2 / RC2 / RB2) Block 4 , 16k~20k , 40H (only available on RD2 / RC2) Block 5 , 20k~24k , 50H (only available on RD2 / RC2) Block 6 , 24k~28k , 60H (only available on RD2 / RC2) Block 7 , 28k~32k , 70H (only available on RD2 / RC2) Block 8 , 32k~36k , 80H (only available on RD2) Block 9 , 36k~40k , 90H (only available on RD2) Block 10, 40k~44k , A0H (only available on RD2) Block 11, 44k~48k , B0H (only available on RD2) Block 12, 48k~52k , C0H (only available on RD2) Block 13, 52k~56k , D0H (only available on RD2) Block 14, 56k~60k , E0H (only available on RD2) Block 15, 60k~62k , F0H (only available on RD2) Example: :020000030C20CF (Erase 4k block #2)
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 02 = display data in data block (valid addresses: 0001~0FFFH) cc = checksum Example 1: :0500000440004FFF0069 display 4000–4FFF Example 2: :0500000400000FFF02E7 display data in data block (the data at address 0000 is invalid)

<table>
<tr><th>RECORD TYPE</th><th>COMMAND/DATA FUNCTION</th></tr>
<tr><td>05</td><td>Miscellaneous Read Functions (Selection)

General Format of Function 05
:02xxxx05ffsscc
Where:
02 = number of bytes (hex) in record
xxxx = required field, but value is a “don’t care”
05 = “Miscellaneous Read” function code
ffss = subfunction and selection code
0000 = read signature byte – manufacturer id (15H)
0001 = read signature byte – device id # 1 (C2H)
0002 = read signature byte – device id # 2
0003 = read FX2 bit
0080 = read ROM Code Revision
0700 = read security bits
0701 = read status byte
0702 = read boot vector
cc = checksum
Example 1:
:020000050001F8 read signature byte – device id # 1
Example 2:
:020000050003F6 read FX2 bit
(bit7=0 represent 12–clock mode, bit7=1 represent 6–clock mode)
Example 3:
:02000005008079 read ROM Code Revision (0A: Rev. A, 0B:Rev. B)</td></tr>
<tr><td>06</td><td>Direct Load of Baud Rate

General Format of Function 06
:02xxxx06hhllcc
Where:
02 = number of bytes (hex) in record
xxxx = required field, but value is a “don’t care”
06 = ”Direct Load of Baud Rate” function code
hh = high byte of Timer 2
ll = low byte of Timer 2
cc = checksum
Example:
:02000006F500F3</td></tr>
<tr><td>07</td><td>Program Data in Data Block
:nnaaaa07dd....ddcc
Where:
nn = number of bytes (hex) in record
aaaa = memory address of first byte in record (the valid address:0001~0FFFH)
dd....dd = data bytes
cc = checksum
Example:
:10008007AF5F67F0602703E0322CFA92007780C3F6</td></tr>
</table>

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`

IAP CALL	PARAMETER

PROGRAM SECURITY BITS	Input Parameter:
	`R0 = osc freq (integer)`
	`R1 = 05h or R1 = 85h (WDT feed)`
	`DPH = 00h`
	`DPL = 00h , security bit #1`
	`DPL = 01h , security bit #2`
	`DPL = 02h , security bit #3`
	Return Parameter:
	`ACC = 00 if pass , !=0 if fail`

PROGRAM STATUS BYTE	Input Parameter:
	`R0 = osc freq (integer)`
	`R1 = 06h or R1 = 86h (WDT feed)`
	`DPH = 00h`
	`DPL = 00H - program status byte`
	`ACC = status byte`
	Return Parameter:
	`ACC = 00 if pass , !=0 if fail`

PROGRAM BOOT VECTOR	Input Parameter:
	`R0 = osc freq (integer)`
	`R1 = 06h or R1 = 86h (WDT feed)`
	`DPH = 00h`
	`DPL = 01H - program boot vector`
	`ACC = boot vector`
	Return Parameter:
	`ACC = 00 if pass , !=0 if fail`

PROGRAM 6–CLK/12–CLK CONFIGURATION BIT
(New function)
	Input Parameter:
	`R0 = osc freq (integer)`
	`R1 = 06h or R1 = 86h (WDT feed)`
	`DPH = 00h`
	`DPL = 02H - program config bit`
	`ACC = 80H (MSB = 6clk/12clk bit)`
	Return Parameter:
	`ACC = 00 if pass , !=0 if fail`

PROGRAM DATA BLOCK
(New function)
	Input Parameter:
	`R0 = osc freq (integer)`
	`R1 = 0Dh or R1 = 8Dh (WDT feed)`
	`DPTR = address of byte to program`
	`(valid addresses = 0001h~0FFFh)`
	`ACC = data`
	Return Parameter:
	`ACC = 00 if pass , !=0 if fail`

READ DEVICE DATA	Input Parameter:
	`R0 = osc freq (integer)`
	`R1 = 03h or R1 = 83h (WDT feed)`
	`DPTR = address of byte to read`
	Return Parameter:
	`ACC = value of byte read`

READ DATA BLOCK
(New function)
	Input Parameter:
	`R0 = osc freq (integer)`
	`R1 = 0Eh or R1 = 8Eh (WDT feed)`
	`DPTR = address of byte to read`
	`(valid addresses = 0001h~0FFFh)`
	Return Parameter:
	`ACC = value of byte read`

READ MANUFACTURER ID	Input Parameter:
	`R0 = osc freq (integer)`
	`R1 = 00h or R1 = 80h (WDT feed)`
	`DPH = 00h`
	`DPL = 00h - read manufacturer ID`
	Return Parameter:
	`ACC = value of byte read`

IAP CALL	PARAMETER

READ DEVICE ID #1	Input Parameter:
	`R0 = osc freq (integer)`
	`R1 = 00h or R1 = 80h (WDT feed)`
	`DPH = 00h`
	`DPL = 01h - read device ID #1`
	Return Parameter:
	`ACC = value of byte read`

READ DEVICE ID #2	Input Parameter:
	`R0 = osc freq (integer)`
	`R1 = 00h or R1 = 80h (WDT feed)`
	`DPH = 00h`
	`DPL = 02h - read device ID #2`
	Return Parameter:
	`ACC = value of byte read`

READ SECURITY BITS	Input Parameter:
	`R0 = osc freq (integer)`
	`R1 = 07h or R1 = 87h (WDT feed)`
	`DPH = 00h`
	`DPL = 00h - read lock byte`
	Return Parameter:
	`ACC = value of byte read`

READ STATUS BYTE	Input Parameter:
	`R0 = osc freq (integer)`
	`R1 = 07h or R1 = 87h (WDT feed)`
	`DPH = 00h`
	`DPL = 01h - read status byte`
	Return Parameter:
	`ACC = value of byte read`

READ BOOT VECTOR	Input Parameter:
	`R0 = osc freq (integer)`
	`R1 = 07h or R1 = 87h (WDT feed)`
	`DPH = 00h`
	`DPL = 02h - read boot vector`
	Return Parameter:
	`ACC = value of byte read`

READ CONFIG
(New function)
	Input Parameter:
	`R0 = osc freq (integer)`
	`R1 = 00h or R1 = 80h (WDT feed)`
	`DPH = 00h`
	`DPL = 03h - read config byte`
	Return Parameter:
	`ACC = value of byte read`

READ REVISION
(New function)
	Input Parameter:
	`R0 = osc freq (integer)`
	`R1 = 00h or R1 = 80h (WDT feed)`
	`DPH = 00h`
	`DPL = 80h - read revision of ROM Code`
	Return Parameter:
	`ACC = value of byte read`

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.

ABSOLUTE MAXIMUM RATINGS[1, 2, 3]

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

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

DC ELECTRICAL CHARACTERISTICS

T_{amb} = 0 °C to +70 °C or –40 °C to +85 °C; V_{CC} = 5 V ± 10%; V_{SS} = 0 V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
V_{IL}	Input low voltage	4.5 V < V_{CC} < 5.5 V	–0.5		$0.2V_{CC}$–0.1	V
V_{IH}	Input high voltage (ports 0, 1, 2, 3, $\overline{EA}$)		$0.2V_{CC}$+0.9		V_{CC}+0.5	V
V_{IH1}	Input high voltage, XTAL1, RST		$0.7V_{CC}$		V_{CC}+0.5	V
V_{OL}	Output low voltage, ports 1, 2, 3[8]	V_{CC} = 4.5 V I_{OL} = 1.6 mA[2]			0.4	V
V_{OL1}	Output low voltage, port 0, ALE, $\overline{PSEN}$ [7, 8]	V_{CC} = 4.5 V I_{OL} = 3.2 mA[2]			0.45	V
V_{OH}	Output high voltage, ports 1, 2, 3 [3]	V_{CC} = 4.5 V I_{OH} = –30 μA	V_{CC} – 0.7			V
V_{OH1}	Output high voltage (port 0 in external bus mode), ALE[9], $\overline{PSEN}$[3]	V_{CC} = 4.5 V I_{OH} = –3.2 mA	V_{CC} – 0.7			V
I_{IL}	Logical 0 input current, ports 1, 2, 3	V_{IN} = 0.4 V	–1		–75	μA
I_{TL}	Logical 1-to-0 transition current, ports 1, 2, 3[6]	V_{IN} = 2.0 V See Note 4			–650	μA
I_{LI}	Input leakage current, port 0	0.45 < V_{IN} < V_{CC} – 0.3			±10	μA
I_{CC}	Power supply current (see Figure 49): Active mode (see Note 5) Idle mode (see Note 5) Power-down mode or clock stopped (see Figure 55 for conditions) Programming and erase mode	See Note 5 T_{amb} = 0 °C to 70 °C T_{amb} = –40 °C to +85 °C f_{osc} = 20 MHz		 < 30 < 40 60	 100 125	 μA μA mA
R_{RST}	Internal reset pull-down resistor		40		225	kΩ
C_{IO}	Pin capacitance[10] (except $\overline{EA}$)				15	pF

NOTES:

1. Typical ratings are not guaranteed. The values listed are at room temperature, 5 V.
2. Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the V_{OL}s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE pin may exceed 0.8 V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input. I_{OL} can exceed these conditions provided that no single output sinks more than 5 mA and no more than two outputs exceed the test conditions.
3. Capacitive loading on ports 0 and 2 may cause the V_{OH} on ALE and $\overline{PSEN}$ to momentarily fall below the V_{CC}–0.7 specification when the address bits are stabilizing.
4. Pins of ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when V_{IN} is approximately 2 V.
5. See Figures 52 through 55 for I_{CC} test conditions and Figure 49 for I_{CC} vs Freq.
   Active mode: $I_{CC(MAX)}$ = (10.5 + 0.9 × FREQ.[MHz])mA in 12-clock mode
   Idle mode: $I_{CC(MAX)}$ = (2.5 + 0.33 × FREQ.[MHz])mA in 12-clock mode
6. This value applies to T_{amb} = 0 °C to +70 °C.
7. Load capacitance for port 0, ALE, and $\overline{PSEN}$ = 100 pF, load capacitance for all other outputs = 80 pF.
8. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 15 mA (*NOTE: This is 85 °C specification.)
   Maximum I_{OL} per 8-bit port: 26 mA
   Maximum total I_{OL} for all outputs: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
9. ALE is tested to V_{OH1}, except when ALE is off then V_{OH} is the voltage specification.
10. Pin capacitance is characterized but not tested. Pin capacitance is less than 25 pF. Pin capacitance of ceramic package is less than 15 pF (except $\overline{EA}$ is 25 pF).

AC ELECTRICAL CHARACTERISTICS (12-CLOCK MODE)

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]

SYMBOL	FIGURE	PARAMETER	VARIABLE CLOCK[4]		33 MHz CLOCK[4]		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	42	Oscillator frequency	0	33			MHz
t_{LHLL}	42	ALE pulse width	$2t_{CLCL}$–40		21		ns
t_{AVLL}	42	Address valid to ALE low	t_{CLCL}–25		5		ns
t_{LLAX}	42	Address hold after ALE low	t_{CLCL}–25		5		ns
t_{LLIV}	42	ALE low to valid instruction in		$4t_{CLCL}$–65		55	ns
t_{LLPL}	42	ALE low to $\overline{PSEN}$ low	t_{CLCL}–25		5		ns
t_{PLPH}	42	$\overline{PSEN}$ pulse width	$3t_{CLCL}$–45		45		ns
t_{PLIV}	42	$\overline{PSEN}$ low to valid instruction in		$3t_{CLCL}$–60		30	ns
t_{PXIX}	42	Input instruction hold after $\overline{PSEN}$	0		0		ns
t_{PXIZ}	42	Input instruction float after $\overline{PSEN}$		t_{CLCL}–25		5	ns
t_{AVIV}	42	Address to valid instruction in		$5t_{CLCL}$–80		70	ns
t_{PLAZ}	42	$\overline{PSEN}$ low to address float		10		10	ns
Data Memory							
t_{RLRH}	43, 44	$\overline{RD}$ pulse width	$6t_{CLCL}$–100		82		ns
t_{WLWH}	43, 44	$\overline{WR}$ pulse width	$6t_{CLCL}$–100		82		ns
t_{RLDV}	43, 44	$\overline{RD}$ low to valid data in		$5t_{CLCL}$–90		60	ns
t_{RHDX}	43, 44	Data hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	43, 44	Data float after $\overline{RD}$		$2t_{CLCL}$–28		32	ns
t_{LLDV}	43, 44	ALE low to valid data in		$8t_{CLCL}$–150		90	ns
t_{AVDV}	43, 44	Address to valid data in		$9t_{CLCL}$–165		105	ns
t_{LLWL}	43, 44	ALE low to $\overline{RD}$ or $\overline{WR}$ low	$3t_{CLCL}$–50	$3t_{CLCL}$+50	40	140	ns
t_{AVWL}	43, 44	Address valid to $\overline{WR}$ low or $\overline{RD}$ low	$4t_{CLCL}$–75		45		ns
t_{QVWX}	43, 44	Data valid to $\overline{WR}$ transition	t_{CLCL}–30		0		ns
t_{WHQX}	43, 44	Data hold after $\overline{WR}$	t_{CLCL}–25		5		ns
t_{QVWH}	44	Data valid to $\overline{WR}$ high	$7t_{CLCL}$–130		80		ns
t_{RLAZ}	43, 44	$\overline{RD}$ low to address float		0		0	ns
t_{WHLH}	43, 44	$\overline{RD}$ or $\overline{WR}$ high to ALE high	t_{CLCL}–25	t_{CLCL}+25	5	55	ns
External Clock							
t_{CHCX}	46	High time	17	t_{CLCL}–t_{CLCX}			ns
t_{CLCX}	46	Low time	17	t_{CLCL}–t_{CHCX}			ns
t_{CLCH}	46	Rise time		5			ns
t_{CHCL}	46	Fall time		5			ns
Shift Register							
t_{XLXL}	45	Serial port clock cycle time	$12t_{CLCL}$		360		ns
t_{QVXH}	45	Output data setup to clock rising edge	$10t_{CLCL}$–133		167		ns
t_{XHQX}	45	Output data hold after clock rising edge	$2t_{CLCL}$–80		50		ns
t_{XHDX}	45	Input data hold after clock rising edge	0		0		ns
t_{XHDV}	45	Clock rising edge to input data valid		$10t_{CLCL}$–133		167	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 45 ns is permitted. This limited bus contention will not cause damage to Port 0 drivers.
4. Parts are tested to 3.5 MHz, but guaranteed to operate down to 0 Hz.

AC ELECTRICAL CHARACTERISTICS (6-CLOCK MODE)

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]

SYMBOL	FIGURE	PARAMETER	VARIABLE CLOCK[4]		20 MHz CLOCK[4]		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	42	Oscillator frequency	0	20			MHz
t_{LHLL}	42	ALE pulse width	t_{CLCL}–40		10		ns
t_{AVLL}	42	Address valid to ALE low	0.5t_{CLCL}–20		5		ns
t_{LLAX}	42	Address hold after ALE low	0.5t_{CLCL}–20		5		ns
t_{LLIV}	42	ALE low to valid instruction in		2t_{CLCL}–65		35	ns
t_{LLPL}	42	ALE low to $\overline{PSEN}$ low	0.5t_{CLCL}–20		5		ns
t_{PLPH}	42	$\overline{PSEN}$ pulse width	1.5t_{CLCL}–45		30		ns
t_{PLIV}	42	$\overline{PSEN}$ low to valid instruction in		1.5t_{CLCL}–60		15	ns
t_{PXIX}	42	Input instruction hold after $\overline{PSEN}$	0		0		ns
t_{PXIZ}	42	Input instruction float after $\overline{PSEN}$		0.5t_{CLCL}–20		5	ns
t_{AVIV}	42	Address to valid instruction in		2.5t_{CLCL}–80		45	ns
t_{PLAZ}	42	$\overline{PSEN}$ low to address float		10		10	ns
Data Memory							
t_{RLRH}	43, 44	$\overline{RD}$ pulse width	3t_{CLCL}–100		50		ns
t_{WLWH}	43, 44	$\overline{WR}$ pulse width	3t_{CLCL}–100		50		ns
t_{RLDV}	43, 44	$\overline{RD}$ low to valid data in		2.5t_{CLCL}–90		35	ns
t_{RHDX}	43, 44	Data hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	43, 44	Data float after $\overline{RD}$		t_{CLCL}–20		5	ns
t_{LLDV}	43, 44	ALE low to valid data in		4t_{CLCL}–150		50	ns
t_{AVDV}	43, 44	Address to valid data in		4.5t_{CLCL}–165		60	ns
t_{LLWL}	43, 44	ALE low to $\overline{RD}$ or $\overline{WR}$ low	1.5t_{CLCL}–50	1.5t_{CLCL}+50	25	125	ns
t_{AVWL}	43, 44	Address valid to $\overline{WR}$ low or $\overline{RD}$ low	2t_{CLCL}–75		25		ns
t_{QVWX}	43, 44	Data valid to $\overline{WR}$ transition	0.5t_{CLCL}–25		0		ns
t_{WHQX}	43, 44	Data hold after $\overline{WR}$	0.5t_{CLCL}–20		5		ns
t_{QVWH}	44	Data valid to $\overline{WR}$ high	3.5t_{CLCL}–130		45		ns
t_{RLAZ}	43, 44	$\overline{RD}$ low to address float		0		0	ns
t_{WHLH}	43, 44	$\overline{RD}$ or $\overline{WR}$ high to ALE high	0.5t_{CLCL}–20	0.5t_{CLCL}+20	5	45	ns
External Clock							
t_{CHCX}	46	High time	20	t_{CLCL}–t_{CLCX}			ns
t_{CLCX}	46	Low time	20	t_{CLCL}–t_{CHCX}			ns
t_{CLCH}	46	Rise time		5			ns
t_{CHCL}	46	Fall time		5			ns
Shift Register							
t_{XLXL}	45	Serial port clock cycle time	6t_{CLCL}		300		ns
t_{QVXH}	45	Output data setup to clock rising edge	5t_{CLCL}–133		117		ns
t_{XHQX}	45	Output data hold after clock rising edge	t_{CLCL}–30		20		ns
t_{XHDX}	45	Input data hold after clock rising edge	0		0		ns
t_{XHDV}	45	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 45 ns 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.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always ‘t’ (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A – Address
C – Clock
D – Input data
H – Logic level high
I – Instruction (program memory contents)
L – Logic level low, or ALE
P – $\overline{\text{PSEN}}$
Q – Output data
R – $\overline{\text{RD}}$ signal
t – Time
V – Valid
W – $\overline{\text{WR}}$ signal
X – No longer a valid logic level
Z – Float

Examples: t_{AVLL} = Time for address valid to ALE low.
t_{LLPL} = Time for ALE low to $\overline{\text{PSEN}}$ low.

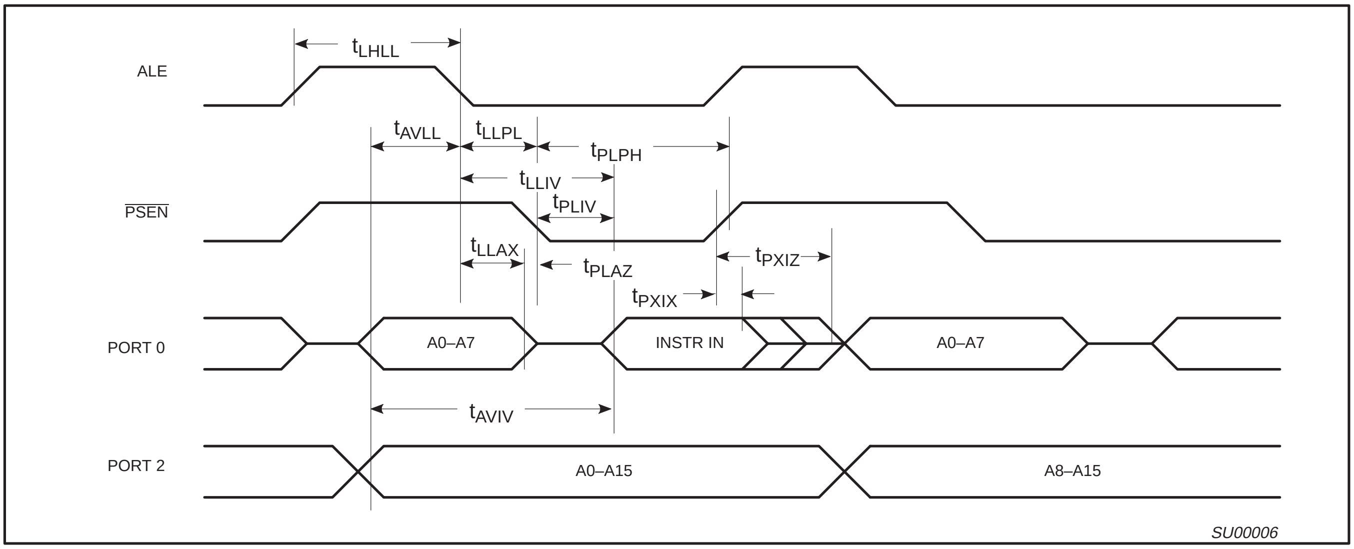


Figure 42. External Program Memory Read Cycle

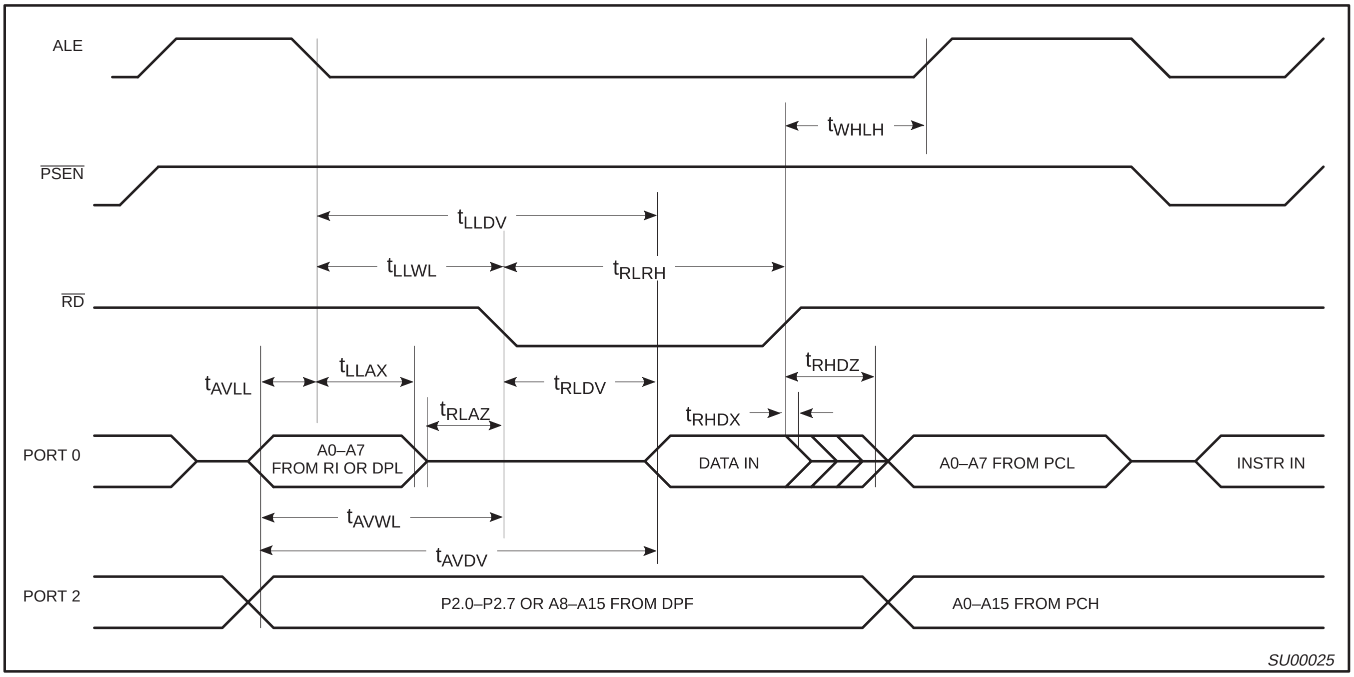


Figure 43. External Data Memory Read Cycle

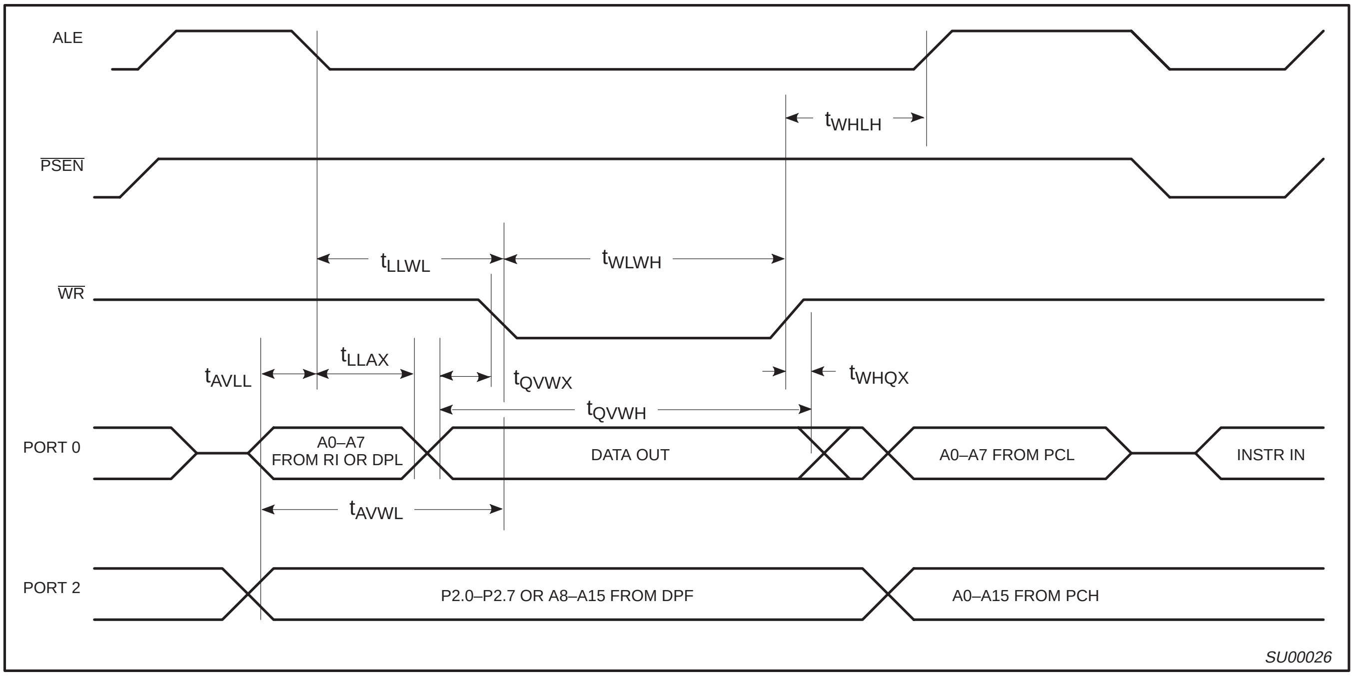


Figure 44. External Data Memory Write Cycle

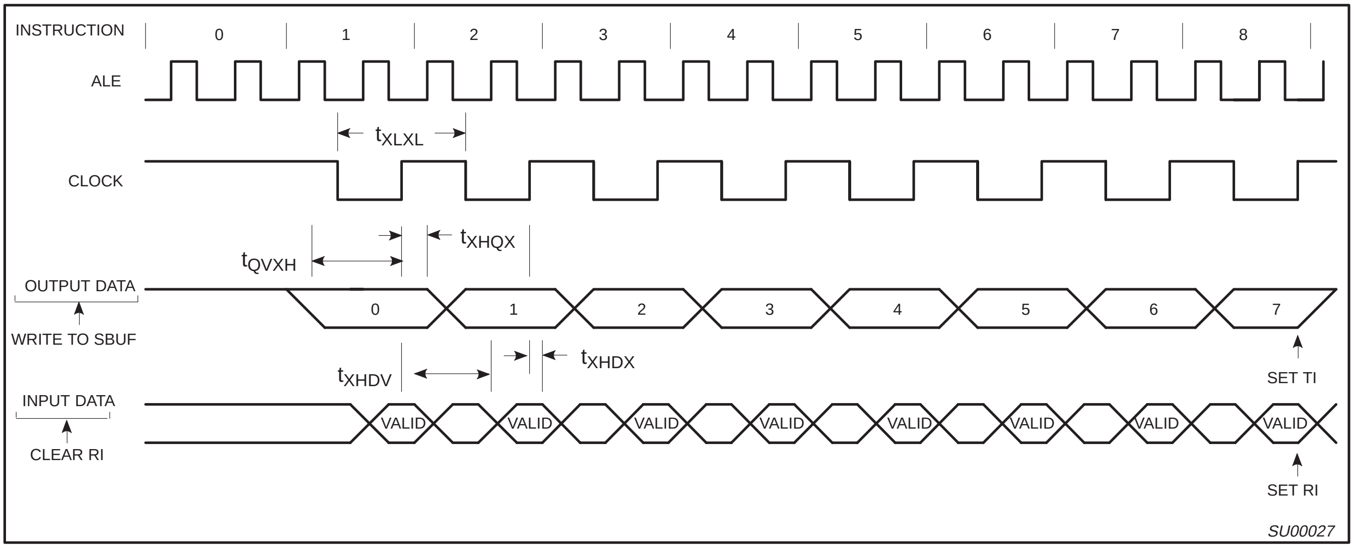


Figure 45. Shift Register Mode Timing

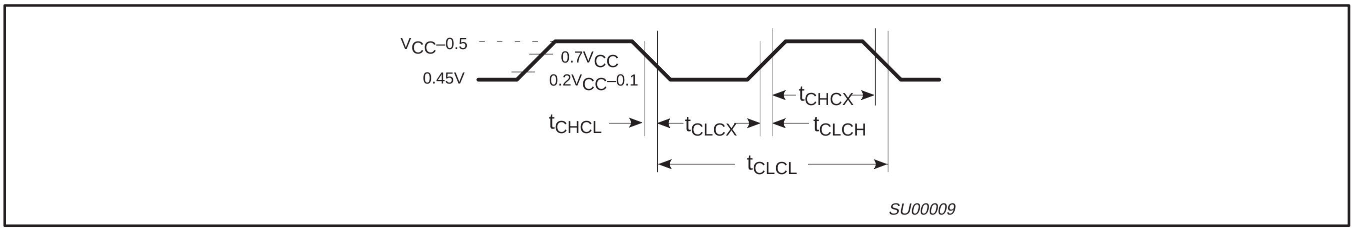


Figure 46. External Clock Drive

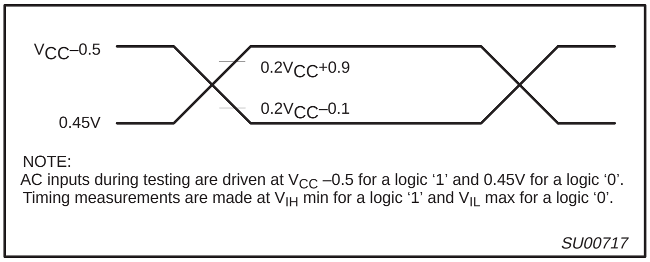


Figure 47. AC Testing Input/Output

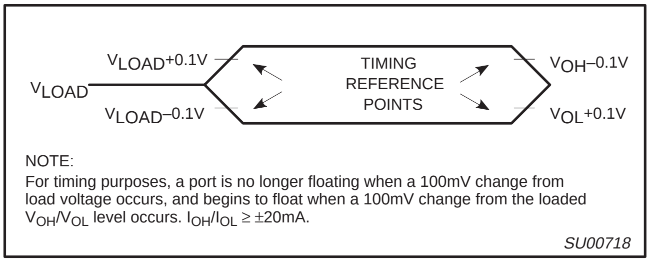


Figure 48. Float Waveform

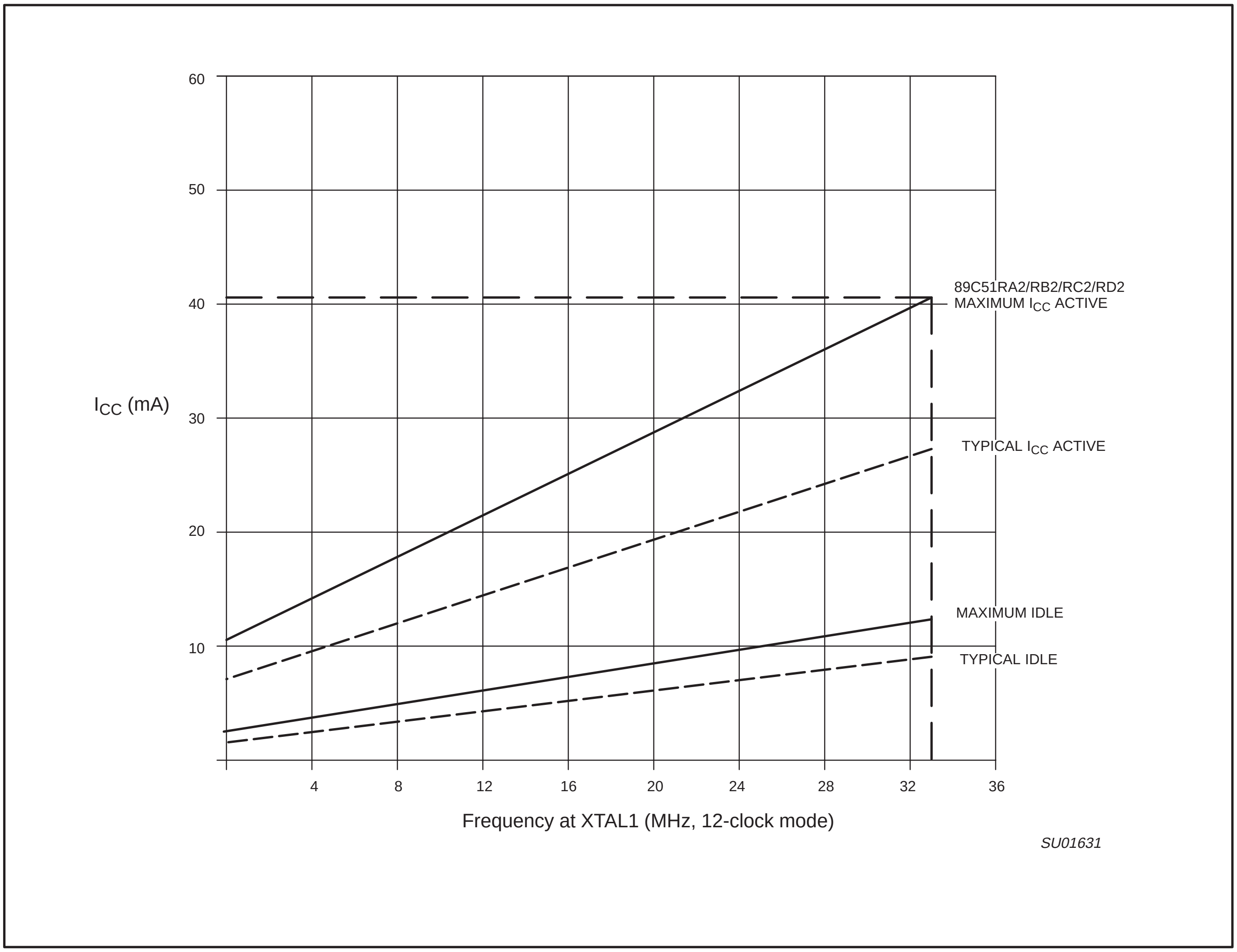
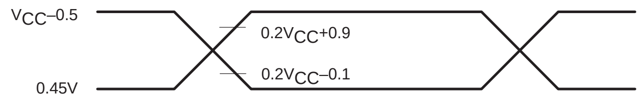


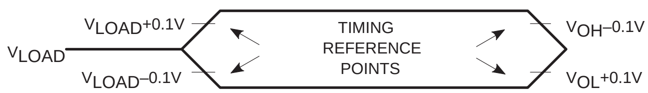
Figure 49. I_{CC} vs. FREQ
Valid only within frequency specifications of the device under test



NOTE:
AC inputs during testing are driven at V_{CC} –0.5 for a logic '1' and 0.45V for a logic '0'.
Timing measurements are made at V_{IH} min for a logic '1' and V_{IL} max for a logic '0'.

SU00010

Figure 50. AC Testing Input/Output



NOTE:
For timing purposes, a port is no longer floating when a 100mV change from load voltage occurs, and begins to float when a 100mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OH}/I_{OL} \geq \pm 20mA$.

SU00011

Figure 51. Float Waveform

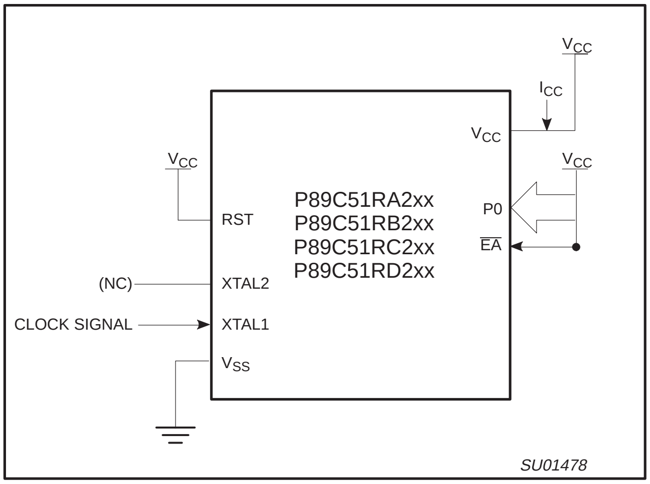


Figure 52. I_{CC} Test Condition, Active Mode, T_{amb} = 25 °C. All other pins are disconnected

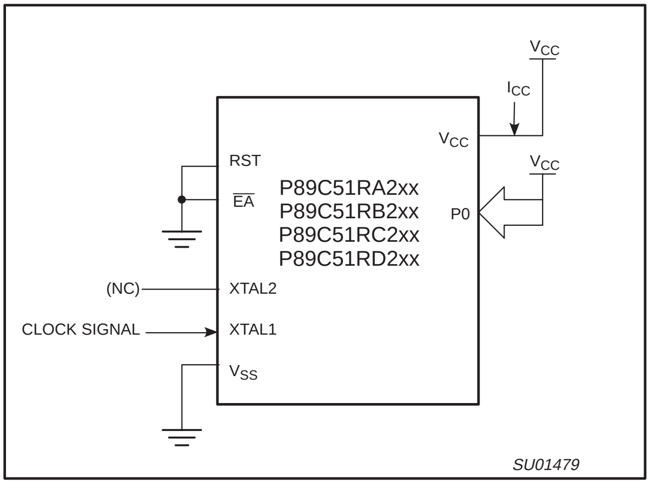


Figure 53. I_{CC} Test Condition, Idle Mode, T_{amb} = 25 °C. All other pins are disconnected

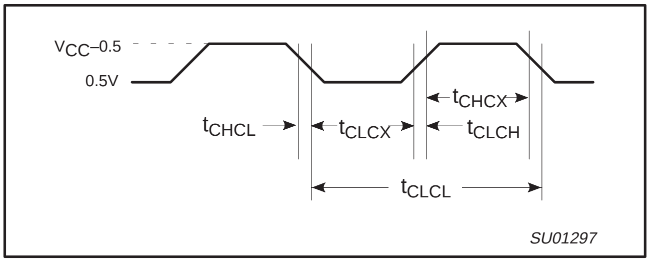


Figure 54. Clock Signal Waveform for I_{CC} Tests in Active and Idle Modes.
$t_{CLCL} = t_{CHCL}$ = 10 ns

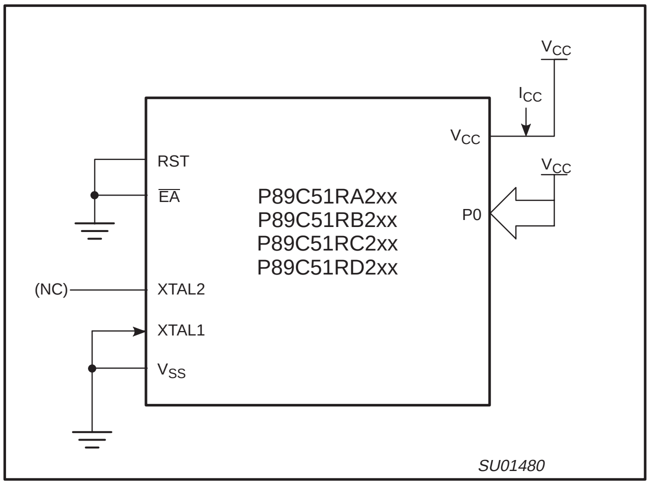
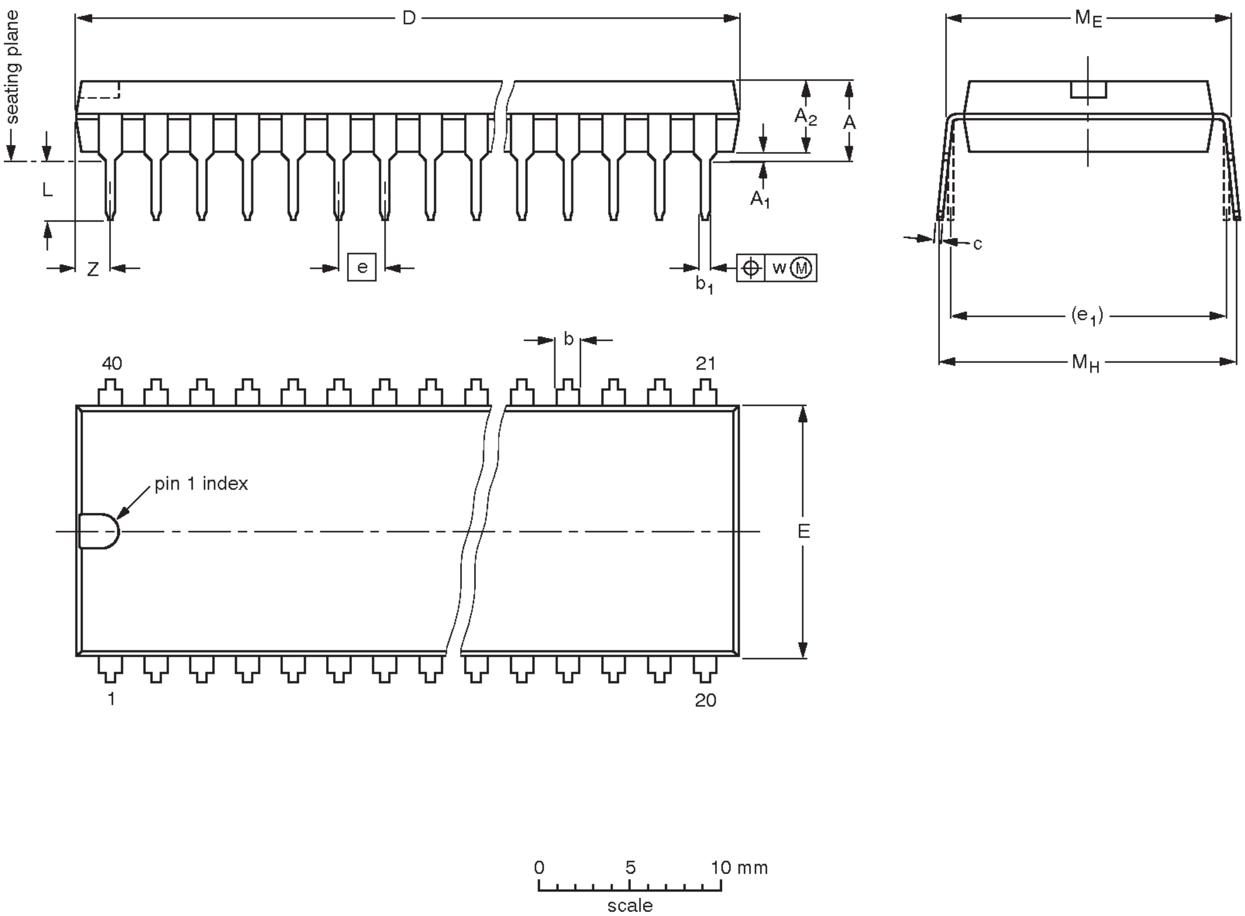


Figure 55. I_{CC} Test Condition, Power Down Mode. All other pins are disconnected; V_{CC} = 2 V to 5.5 V

DIP40: plastic dual in-line package; 40 leads (600 mil) **SOT129-1**

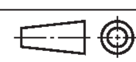


DIMENSIONS (inch dimensions are derived from the original mm dimensions)

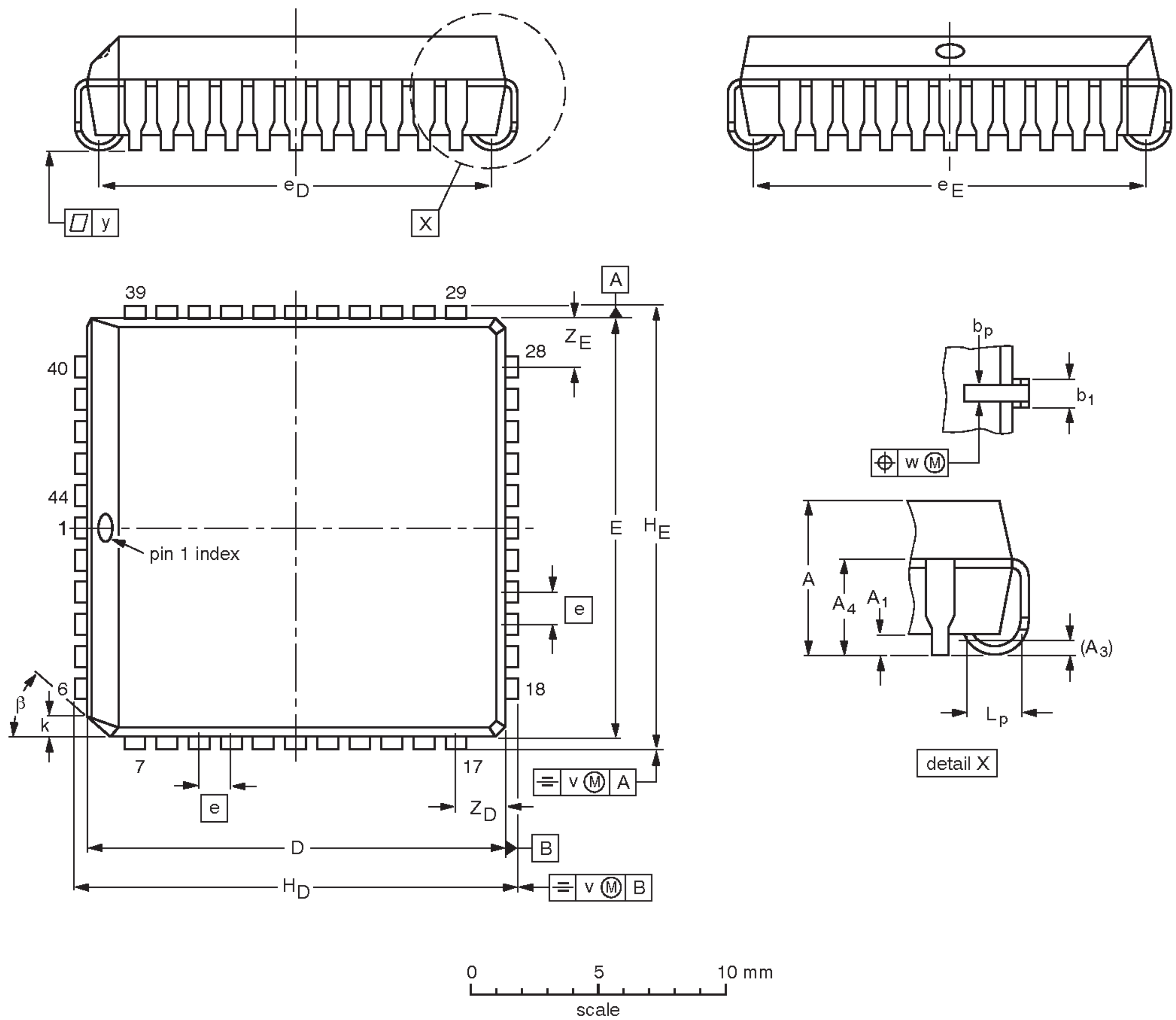
UNIT	A max.	A_1 min.	A_2 max.	b	b_1	c	D (1)	E (1)	e	e_1	L	M_E	M_H	w	Z (1) max.
mm	4.7	0.51	4.0	1.70 1.14	0.53 0.38	0.36 0.23	52.50 51.50	14.1 13.7	2.54	15.24	3.60 3.05	15.80 15.24	17.42 15.90	0.254	2.25
inches	0.19	0.020	0.16	0.067 0.045	0.021 0.015	0.014 0.009	2.067 2.028	0.56 0.54	0.10	0.60	0.14 0.12	0.62 0.60	0.69 0.63	0.01	0.089

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	EIAJ			
SOT129-1	051G08	MO-015	SC-511-40			~~95-01-14~~ 99-12-27

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



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

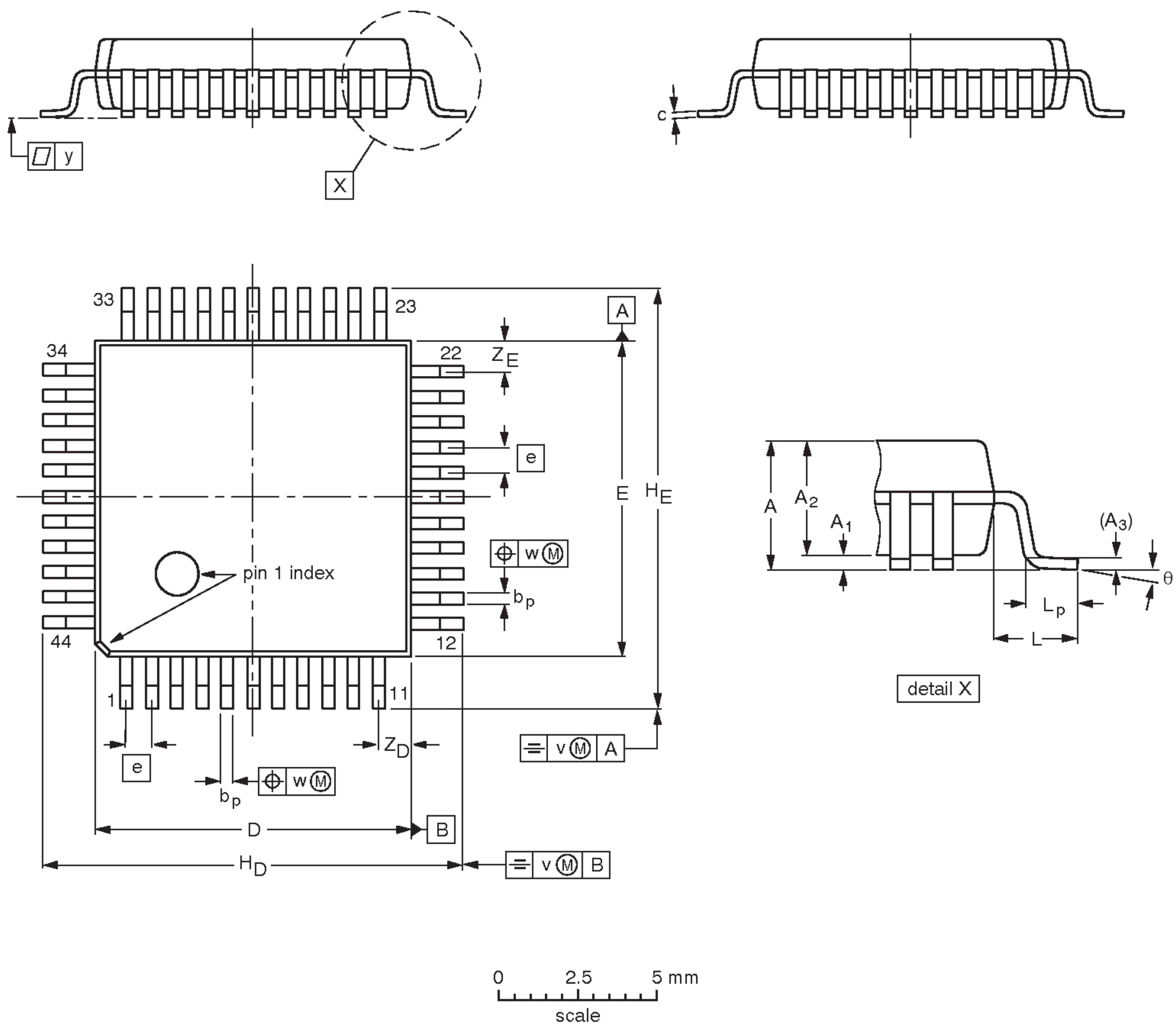
UNIT	A	A_1 min.	A_3	A_4 max.	b_p	b_1	$D^{(1)}$	$E^{(1)}$	e	e_D	e_E	H_D	H_E	k	L_p	v	w	y	$Z_D^{(1)}$ max.	$Z_E^{(1)}$ max.	β
mm	4.57 4.19	0.51	0.25	3.05	0.53 0.33	0.81 0.66	16.66 16.51	16.66 16.51	1.27	16.00 14.99	16.00 14.99	17.65 17.40	17.65 17.40	1.22 1.07	1.44 1.02	0.18	0.18	0.1	2.16	2.16	45°
inches	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

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

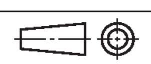


DIMENSIONS (mm are the original dimensions)

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	1.60	0.15 0.05	1.45 1.35	0.25	0.45 0.30	0.20 0.12	10.10 9.90	10.10 9.90	0.80	12.15 11.85	12.15 11.85	1.0	0.75 0.45	0.20	0.20	0.10	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	EIAJ			
SOT389-1	136E08	MS-026				~~99-12-17~~ 00-01-19

REVISION HISTORY

Date	CPCN	Description
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: 05-02

Document order number: 9397 750 09843

Let's make things better.

Philips
Semiconductors

PHILIPS