DATA SHEET

P89C51RC+/P89C51RD+

80C51 8-bit Flash microcontroller family
32K/64K ISP FLASH with 512–1K RAM

Product specification
Replaces 89C51RC+/RD+ of 1999 Apr 01
(see Notes 1 and 2 on page 2)
Supersedes data of 1999 Apr 01
IC28 Data Handbook

1999 Oct 27

Philips
Semiconductors

PHILIPS

80C51 8-bit Flash microcontroller family 32K/64K ISP FLASH with 512–1K RAM

P89C51RC+/P89C51RD+

DESCRIPTION

The 89C51RX+ devices contain a non-volatile FLASH program memory (up to 64 k bytes in the 89C51RD+) that is both parallel programmable and Serial In-System Programmable. In-System Programming allows devices to alter their own program memory, in the actual end product, under software control. This opens up a range of applications that can include the ability to field update the application firmware.

A default serial loader (boot loader) program in ROM allows In-System serial programming of the FLASH memory without the need for a loader in the FLASH code. User programs may erase and reprogram the FLASH memory at will through the use of standard routines contained in ROM.

These devices are Single-Chip 8-Bit Microcontrollers manufactured in advanced CMOS process and are derivatives of the 80C51 microcontroller family. All the devices have the same instruction set as the 80C51.

FLASH/EPROM Memory Size (X by 8)	RAM Size (X by 8)	Programmable Timer Counter (PCA)
89C51RC+		
32 k	512	Yes
89C51RD+		
64 k	1024	Yes

See P89C51RX2 data sheet for devices which do not require a 12 V programming voltage.

The devices also have 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. For systems that require extra memory capability up to 64 k bytes, each can be expanded using standard TTL-compatible memories and logic.

The added features of the P89C51RX+ Family makes them even more powerful microcontrollers 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) capability
- Boot ROM contains low level FLASH programming routines and a default serial loader
- Speed up to 33 MHz
- Full static operation
- RAM expandable externally to 64 k bytes
- 4 level priority interrupt
- 7 interrupt sources, depending on device
- Four 8-bit I/O ports
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Power control modes
  - Clock can be stopped and resumed
  - Idle mode
  - Power down mode
- Programmable clock out
- Second DPTR register
- Asynchronous port reset
- Low EMI (inhibit ALE)
- Watchdog timer

ORDERING INFORMATION

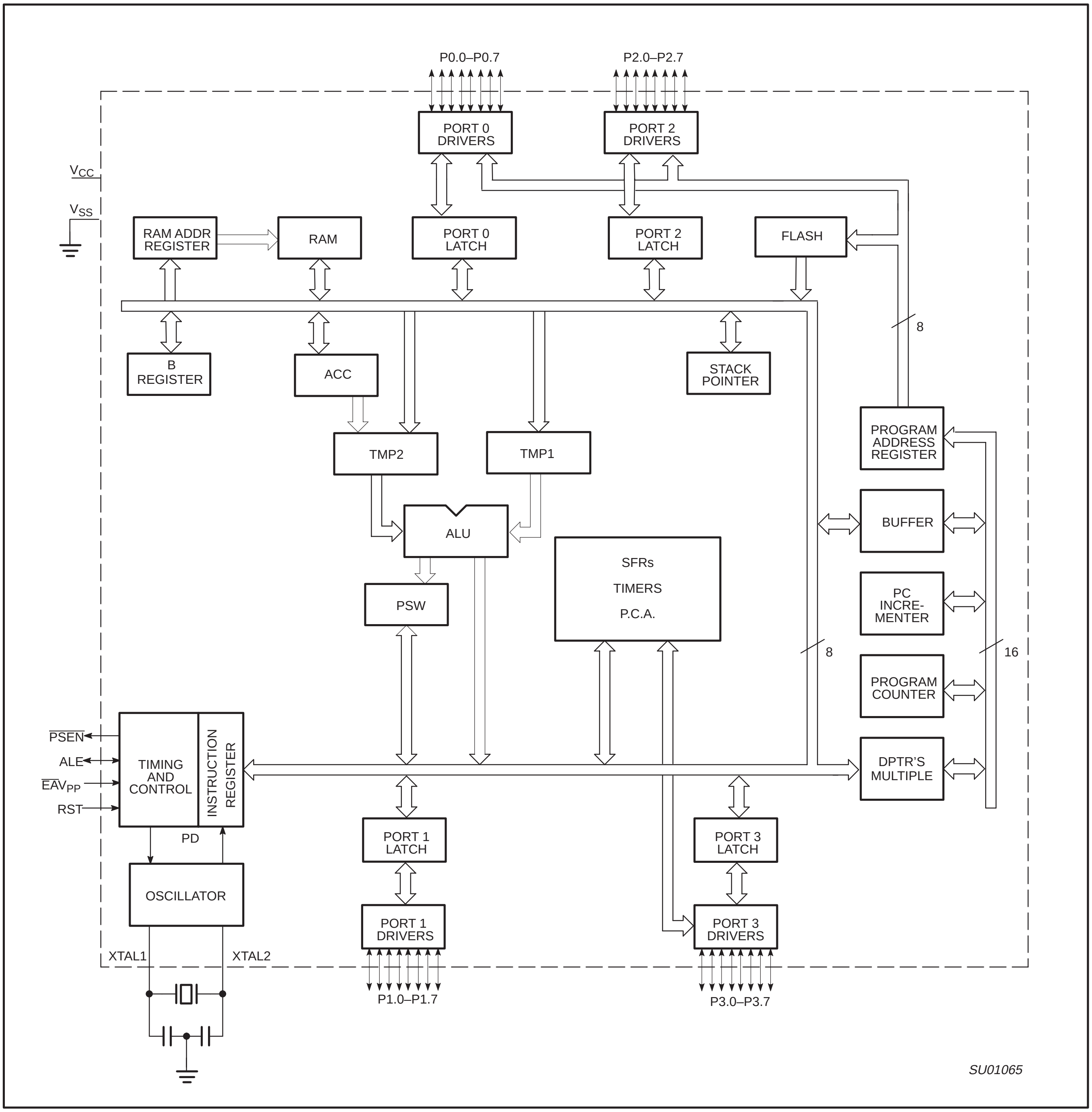
MEMORY SIZE 32 k × 8	MEMORY SIZE 64 k × 8	TEMPERATURE RANGE °C AND PACKAGE	VOLTAGE RANGE	FREQ. (MHz)	DWG. #
P89C51RC+IN	P89C51RD+IN	0 to +70, 40-Pin Plastic Dual In-line Pkg.	5 V	0 to 33	SOT129-1
P89C51RC+IA	P89C51RD+IA	0 to +70, 44-Pin Plastic Leaded Chip Carrier	5 V	0 to 33	SOT187-2
P89C51RC+IB	P89C51RD+IB	0 to +70, 44-Pin Plastic Quad Flat Pack	5 V	0 to 33	QFP44[1]
P89C51RC+JN	P89C51RD+JN	–40 to +85, 40-Pin Plastic Dual In-line Pkg.	5 V	0 to 33	SOT129-1
P89C51RC+JA	P89C51RD+JA	–40 to +85, 44-Pin Plastic Leaded Chip Carrier	5 V	0 to 33	SOT187-2
P89C51RC+JB	P89C51RD+JB	–40 to +85, 44-Pin Plastic Quad Flat Pack	5 V	0 to 33	QFP44[1]

NOTE:
1. SOT not assigned for this package outline.

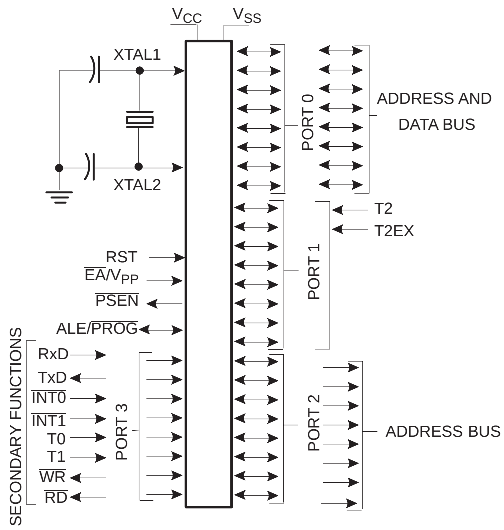
ORDERING INFORMATION

DEVICE NUMBER (P89C51RC+)	TEMPERATURE RANGE/OPERATING FREQUENCY, MAX (I)	PACKAGE (A)
P89C51RC+ (FLASH)	I = 33 MHz, 0°C to 70°C	A = PLCC
P89C51RD+ (FLASH)	J = 33 MHz, –40°C to +85°C	B = PQFP
		N = PDIP

BLOCK DIAGRAM

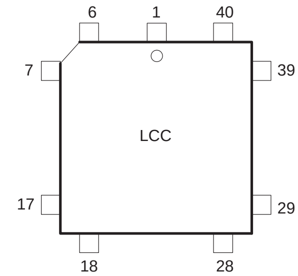


LOGIC SYMBOL



SU00830

PLASTIC LEADED CHIP CARRIER PIN FUNCTIONS



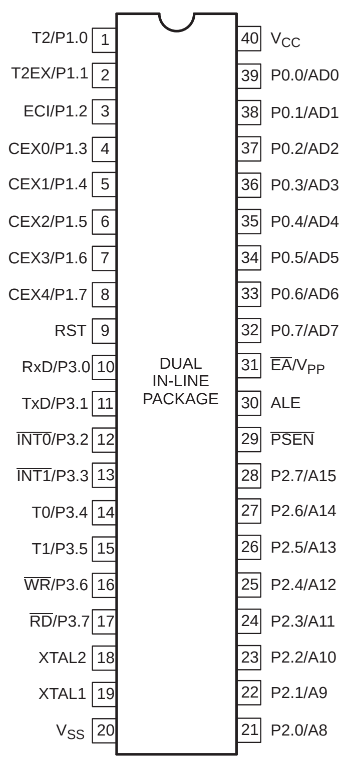
Pin	Function	Pin	Function	Pin	Function
1	NIC*	16	P3.4/T0	31	P2.7/A15
2	P1.0/T2	17	P3.5/T1	32	$\overline{PSEN}$
3	P1.1/T2EX	18	P3.6/$\overline{WR}$	33	ALE
4	P1.2/ECI	19	P3.7/$\overline{RD}$	34	NIC*
5	P1.3/CEX0	20	XTAL2	35	$\overline{EA}$/V_{PP}
6	P1.4/CEX1	21	XTAL1	36	P0.7/AD7
7	P1.5/CEX2	22	V_{SS}	37	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

SU00890

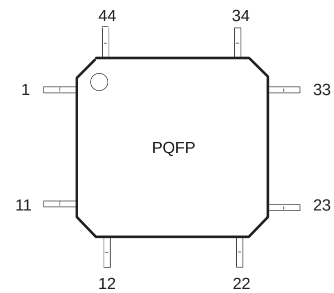
PIN CONFIGURATIONS

DUAL IN-LINE PACKAGE PIN FUNCTIONS



SU00888

PLASTIC QUAD FLAT PACK PIN FUNCTIONS



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	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

SU00891

PIN DESCRIPTIONS

MNEMONIC	PIN NUMBER DIP	LCC	QFP	TYPE	NAME AND FUNCTION
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. 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 89C51RX+ 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 89C51RX+ family, 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 at a constant rate of 1/6 the oscillator frequency, 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.

PIN DESCRIPTIONS (Continued)

MNEMONIC	PIN NUMBER			TYPE	NAME AND FUNCTION
	DIP	LCC	QFP		
$\overline{PSEN}$	29	32	26	O	**Program Store Enable:** The read strobe to external program memory. When executing code from the external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory. $\overline{PSEN}$ is not activated during fetches from internal program memory.
$\overline{EA}/V_{PP}$	31	35	29	I	**External Access Enable/Programming Supply Voltage:** $\overline{EA}$ must be externally held low to enable the device to fetch code from external program memory locations 0000H and 7FFFH. If $\overline{EA}$ is held high, the device executes from internal program memory unless the program counter contains an address greater than 7FFFH for 32 k devices. The value on the $\overline{EA}$ pin is latched when RST is released and any subsequent changes have no effect. Since the RD+ has 64 k internal memory, the RD+ will execute only from internal memory when $\overline{EA}$ is held high. This pin also receives the 12.00 V 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}) at any time must not be higher than V_{CC} + 0.5 V or V_{SS} – 0.5 V, respectively.

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#[2]	Auxiliary 1	A2H	–	–	ENBOOT	–	GF2	0	–	DPS	xx0x00x0B
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
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	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
			B7	B6	B5	B4	B3	B2	B1	B0	
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.
2. The state of the ENBOOT bit depends on the status byte and $\overline{PSEN}$ when reset is exited. See the AUXR1 description on page 20.

Table 1. 89C51RC+/RD+ 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	–	P	000000x0B
RACAP2H#	Timer 2 Capture High	CBH									00H
RACAP2L#	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/$\overline{T2}$	CP/$\overline{RL2}$	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. There are no requirements on the duty cycle of the external clock signal, because the input to the internal clock circuitry is through a divide-by-two flip-flop. However, minimum and maximum high and low times specified in the data sheet must be observed.

RESET

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. To insure 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 RESET.

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 2.0V 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 10ms).

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 OFF FLAG

The Power Off Flag (POF) is set by on-chip circuitry when the V_{CC} level on the 89C51RX+ rises from 0 to 5V. 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 3V 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.

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:

$$\frac{\text{Oscillator Frequency}}{4 \times (65536 - \text{RCAP2H, RCAP2L})}$$

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 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{\text{T}}$2* in the special function register T2CON (see Figure 1). 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{\text{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 2 (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/12 pulses.).

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{\text{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 3). 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 4 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 5, 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/$\overline{\text{T2}}$	CP/$\overline{\text{RL2}}$

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/$\overline{\text{T2}}$	T2CON.1	Timer or counter select. (Timer 2) 0 = Internal timer (OSC/12) 1 = External event counter (falling edge triggered).
CP/$\overline{\text{RL2}}$	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.

SU00728

Figure 1. 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)

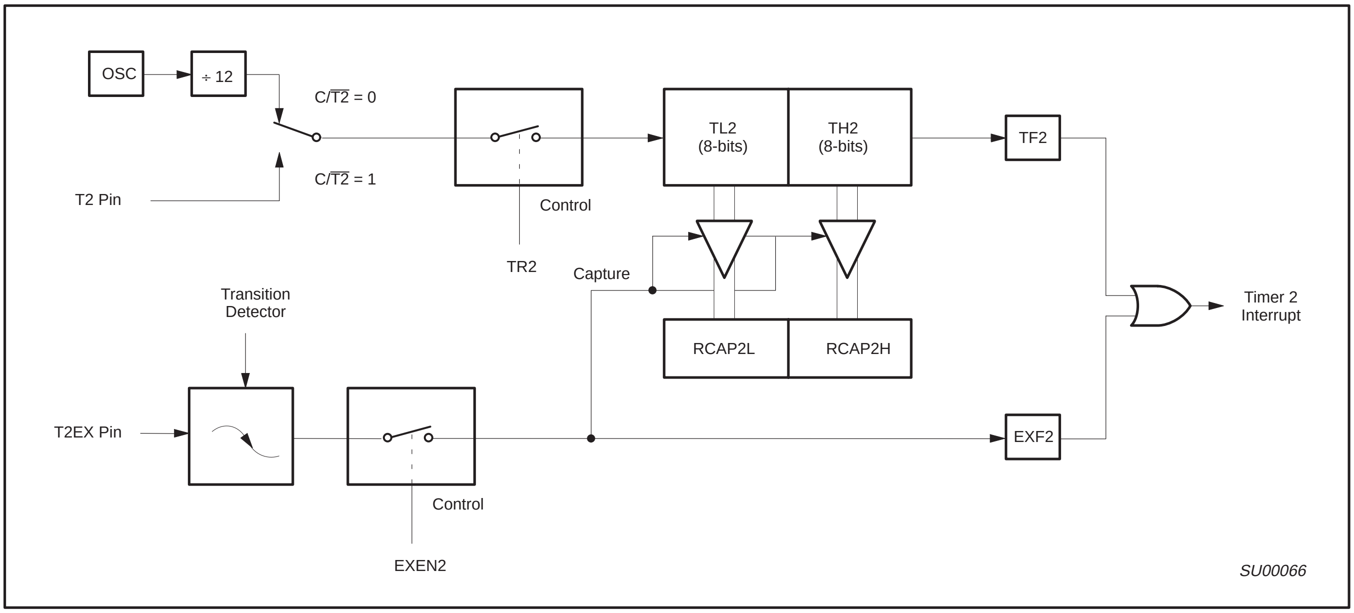


Figure 2. Timer 2 in Capture Mode

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 3. Timer 2 Mode (T2MOD) Control Register

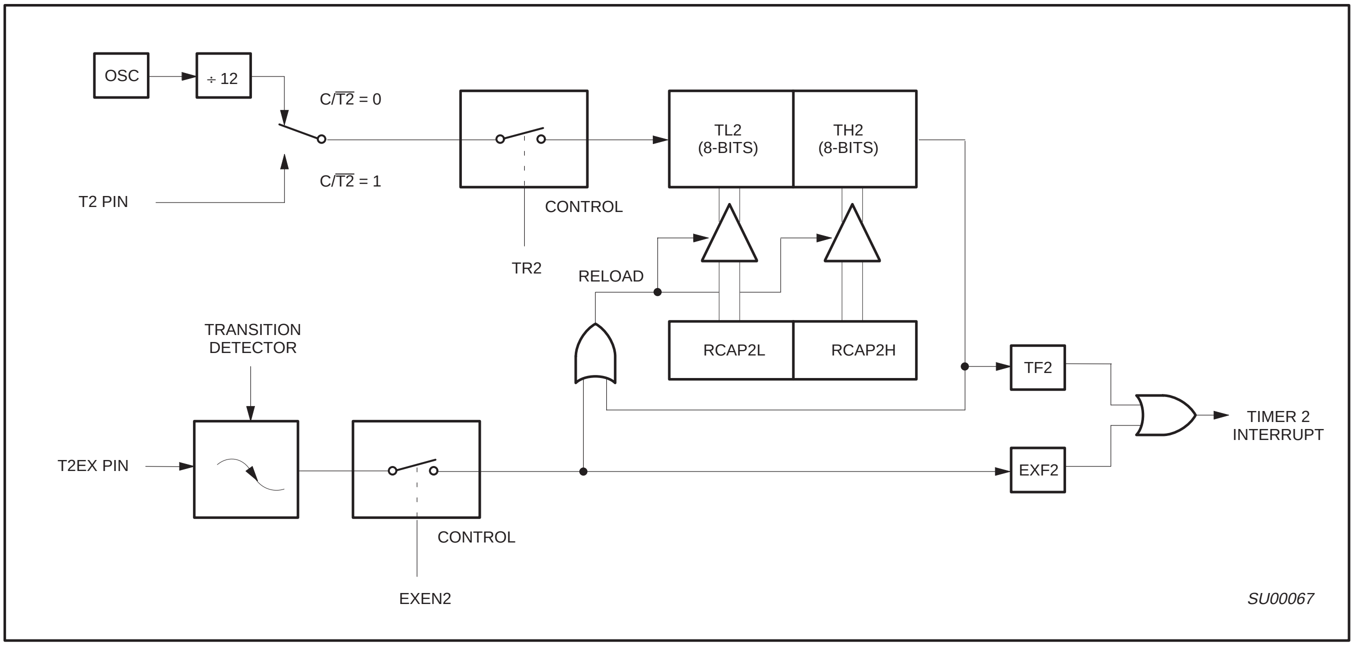


Figure 4. Timer 2 in Auto-Reload Mode (DCEN = 0)

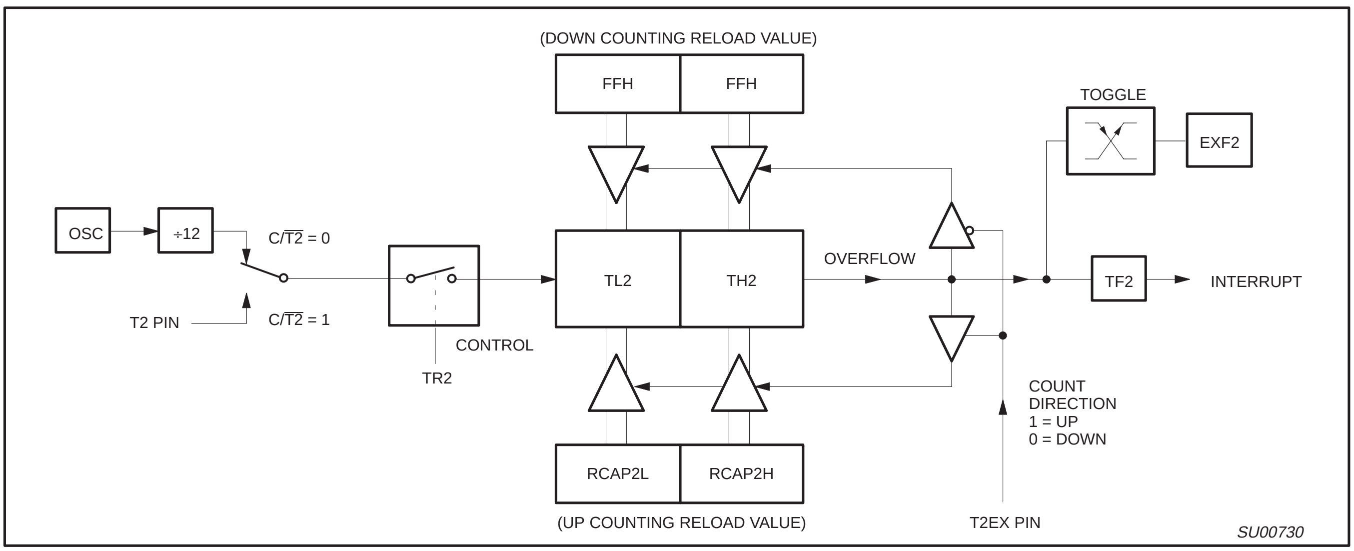


Figure 5. Timer 2 Auto Reload Mode (DCEN = 1)

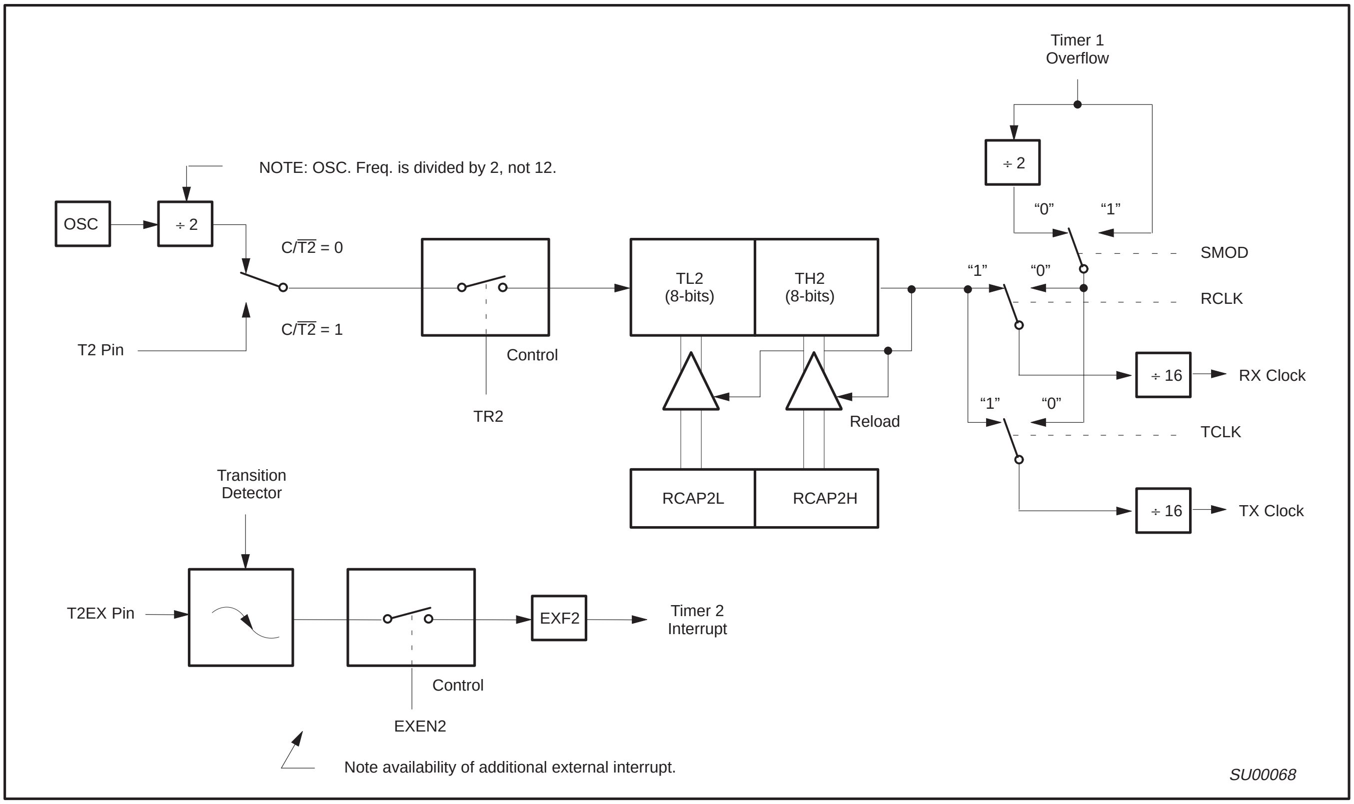


Figure 6. Timer 2 in Baud Rate Generator Mode

Table 4. Timer 2 Generated Commonly Used Baud Rates

Baud Rate	Osc Freq	Timer 2	
		RCAP2H	RCAP2L
375 K	12 MHz	FF	FF
9.6 K	12 MHz	FF	D9
2.8 K	12 MHz	FF	B2
2.4 K	12 MHz	FF	64
1.2 K	12 MHz	FE	C8
300	12 MHz	FB	1E
110	12 MHz	F2	AF
300	6 MHz	FD	8F
110	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 6 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/$\overline{T2}$*=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., 1/12 the oscillator frequency). As a baud rate generator, it increments every state time (i.e., 1/2 the oscillator frequency). Thus the baud rate formula is as follows:

Modes 1 and 3 Baud Rates =

$$\frac{\text{Oscillator Frequency}}{[32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]]}$$

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 6, 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:

$$\text{Baud Rate} = \frac{f_{OSC}}{[32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]]}$$

Where f_{OSC}= Oscillator Frequency

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

$$\text{RCAP2H}, \text{RCAP2L} = 65536 - \left(\frac{f_{OSC}}{32 \times \text{Baud Rate}}\right)$$

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.

Enhanced UART

The UART operates in all of the usual modes that are described in the first section of *Data Handbook IC20, 80C51-Based 8-Bit Microcontrollers*. In addition 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 7). 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 8.

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 9.

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

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

(SMOD0 = 0/1)*

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

SM0	SM1	Mode	Description	Baud Rate**
0	0	0	shift register	$f_{OSC}/12$
0	1	1	8-bit UART	variable
1	0	2	9-bit UART	$f_{OSC}/64$ or $f_{OSC}/32$
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

SU00043

Figure 7. SCON: Serial Port Control Register

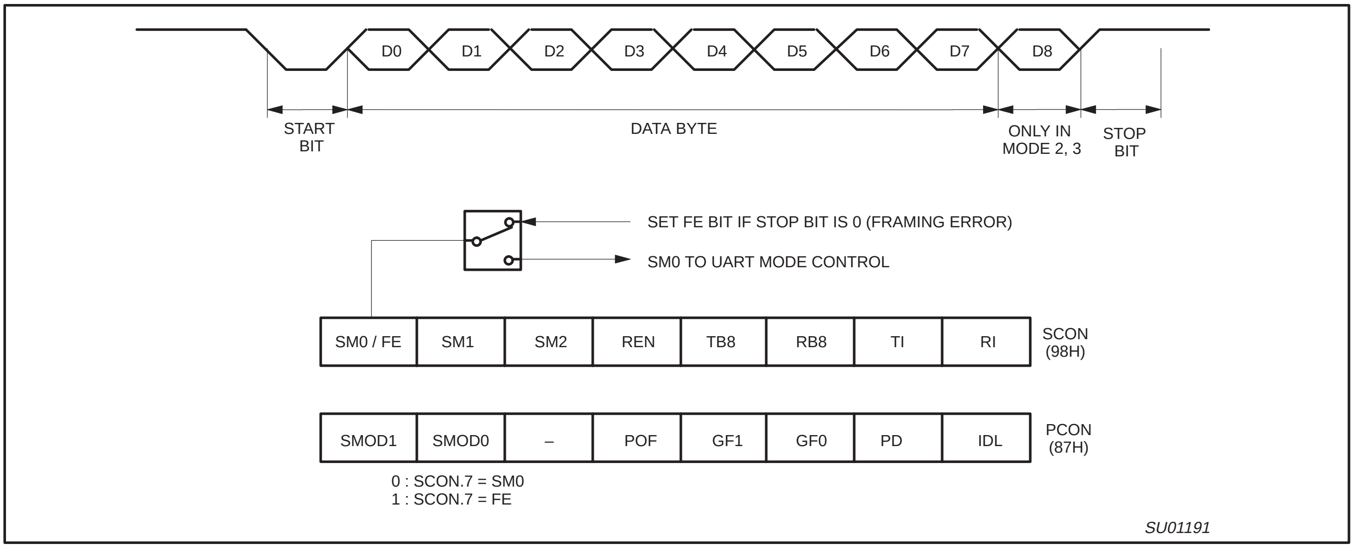


Figure 8. UART Framing Error Detection

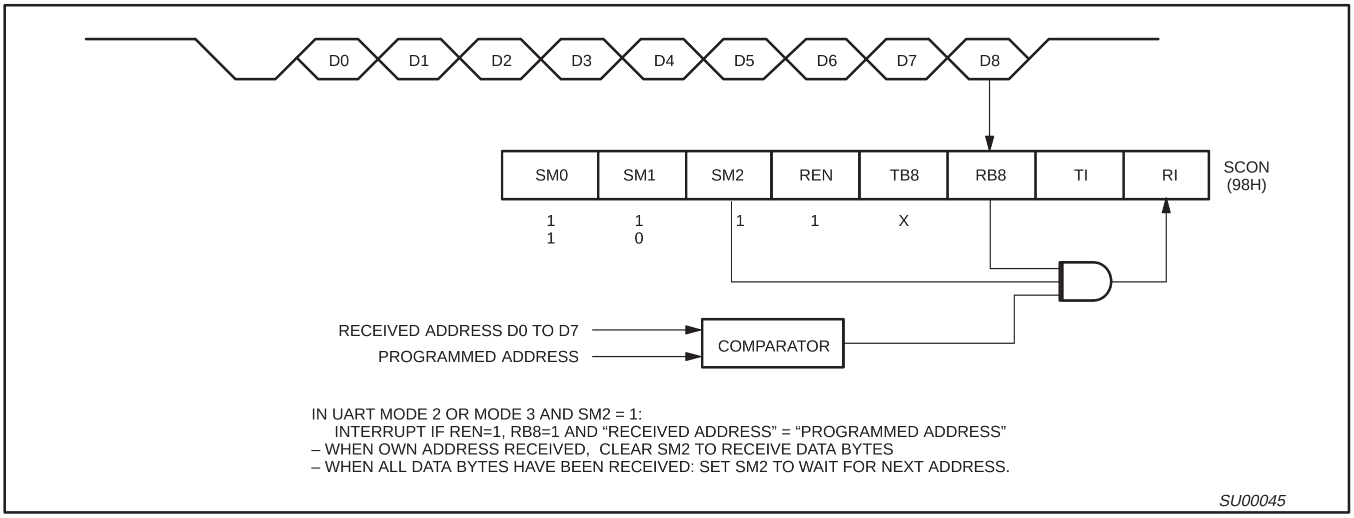


Figure 9. UART Multiprocessor Communication, Automatic Address Recognition

Interrupt Priority Structure

The 89C51RC+/RD+ have 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 10, 11, and 12.) 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 12.

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. There are four interrupt levels. 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.

SU01037

Figure 10. 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	—	Not implemented, reserved for future use.
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.

SU01038

Figure 11. 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	—	Not implemented, reserved for future use.
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.

SU01039

Figure 12. IPH Registers

Reduced EMI Mode

The AO bit (AUXR.0) in the AUXR register when set disables the ALE output.

Reduced EMI Mode

AUXR (8EH)

7	6	5	4	3	2	1	0
–	–	–	–	–	–	EXTRAM	AO

AUXR.1	EXTRAM	
AUXR.0	AO	Turns off ALE output.

Dual DPTR

The dual DPTR structure (see Figure 13) 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: xx0x00x0B

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.

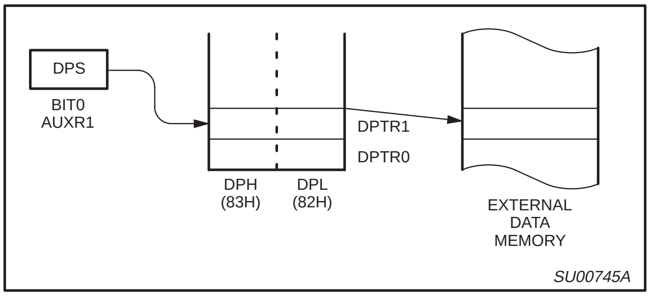


Figure 13.

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 89C51RX+, 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 14.

The PCA timer is a common time base for all five modules and can be programmed to run at: 1/12 the oscillator frequency, 1/4 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 17):

CPS1	CPS0	PCA Timer Count Source
0	0	1/12 oscillator frequency
0	1	1/4 oscillator frequency
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 15.

The watchdog timer function is implemented in module 4 (see Figure 24).

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 18). 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 16.

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 19). 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 20 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.

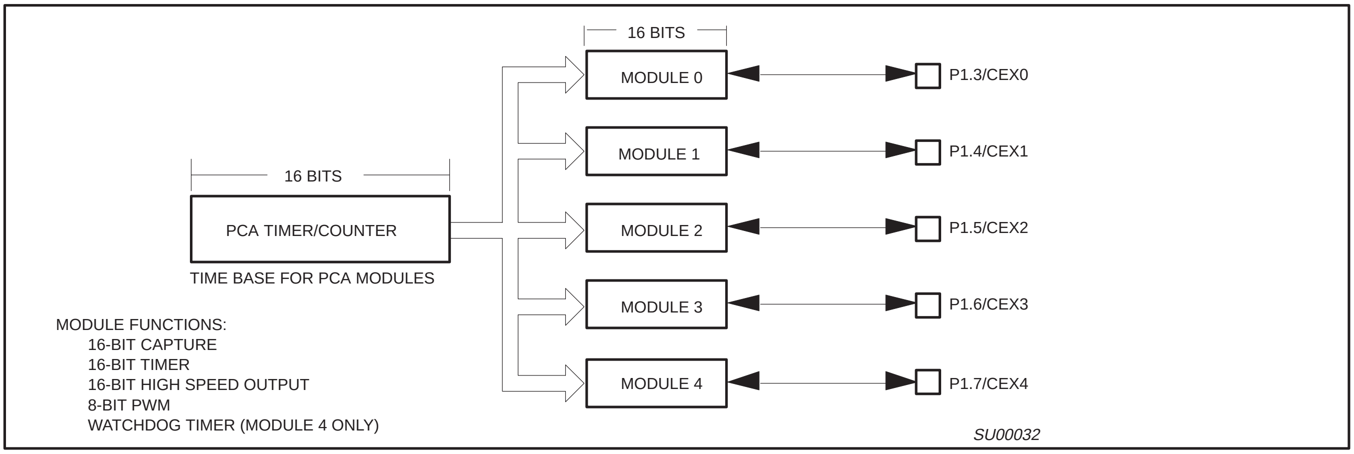


Figure 14. Programmable Counter Array (PCA)

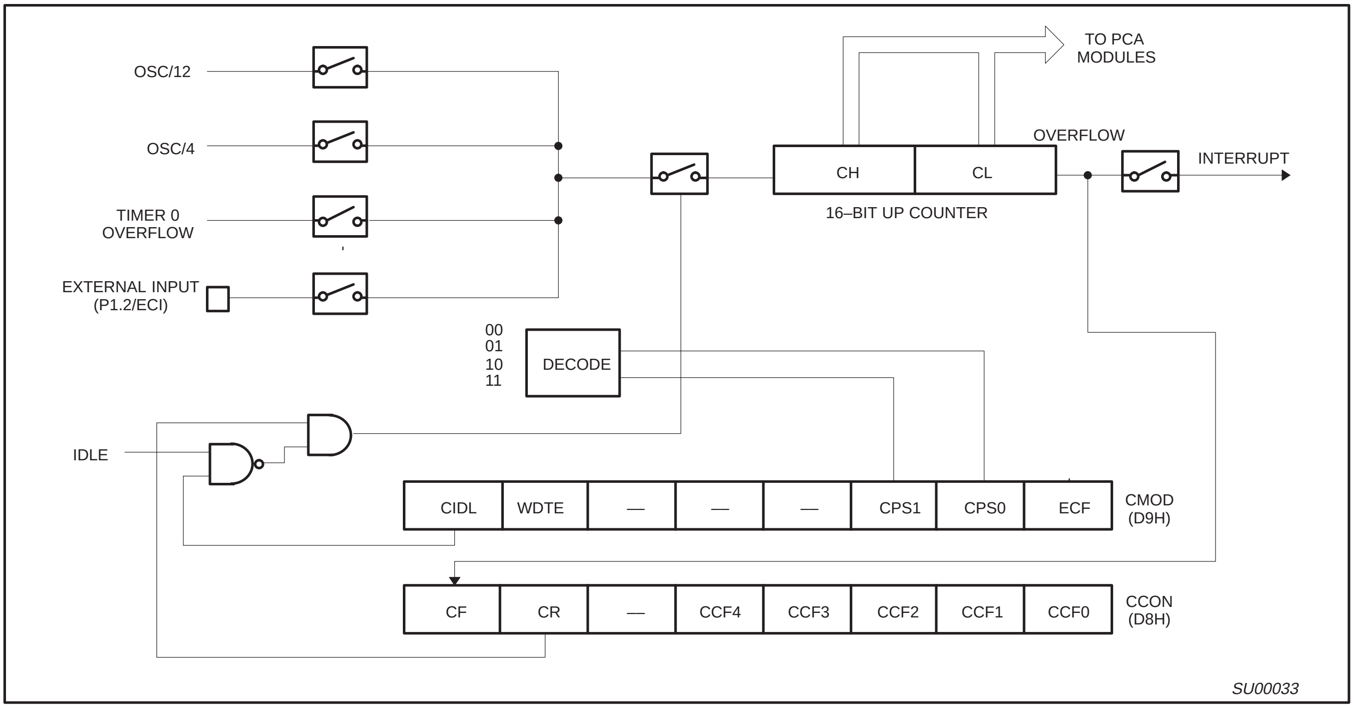


Figure 15. PCA Timer/Counter

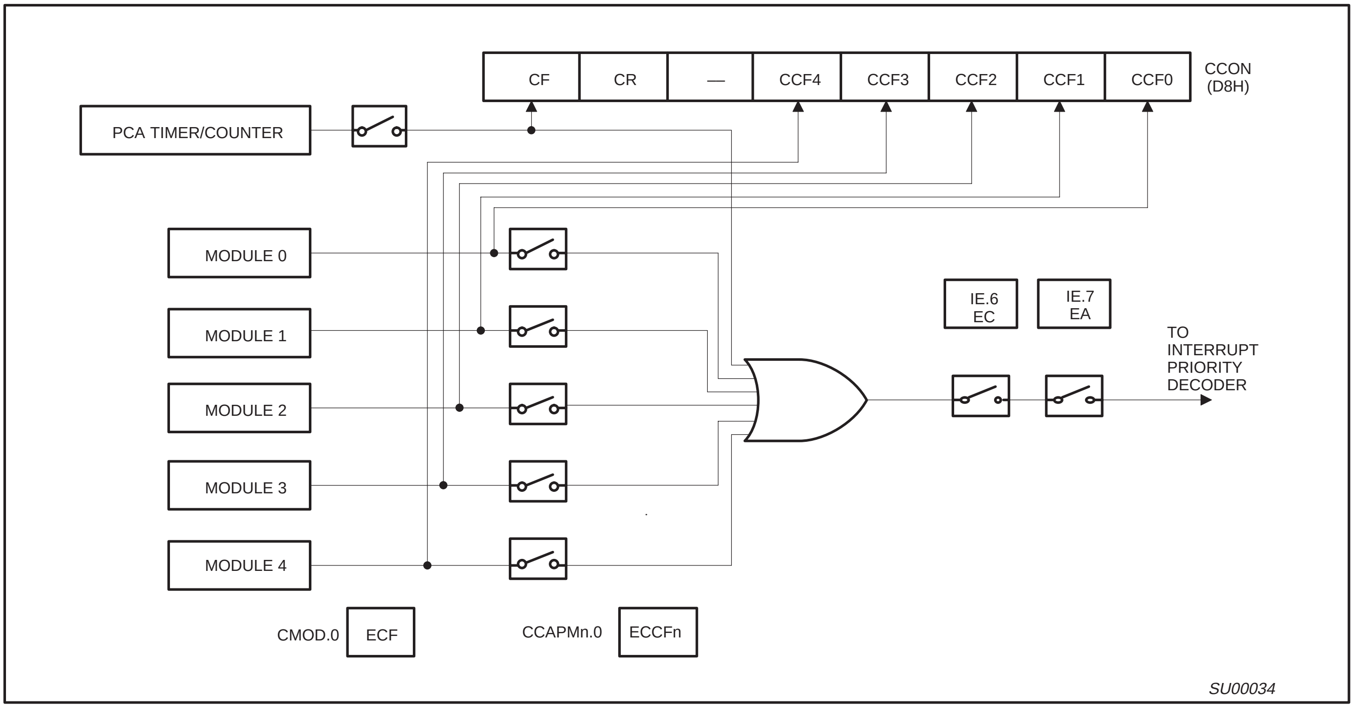


Figure 16. PCA Interrupt System

CMOD Address = OD9H 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, $f_{OSC} \div 12$
0	1	1	Internal clock, $f_{OSC} \div 4$
1	0	2	Timer 0 overflow
1	1	3	External clock at ECI/P1.2 pin (max. rate = $f_{OSC} \div 8$)

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

SU00035

Figure 17. CMOD: PCA Counter Mode Register

CCON Address = OD8H 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.

SU00036

Figure 18. 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.

SU00037

Figure 19. 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 20. 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 21.

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 22).

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 23).

Pulse Width Modulator Mode

All of the PCA modules can be used as PWM outputs. Figure 24 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.

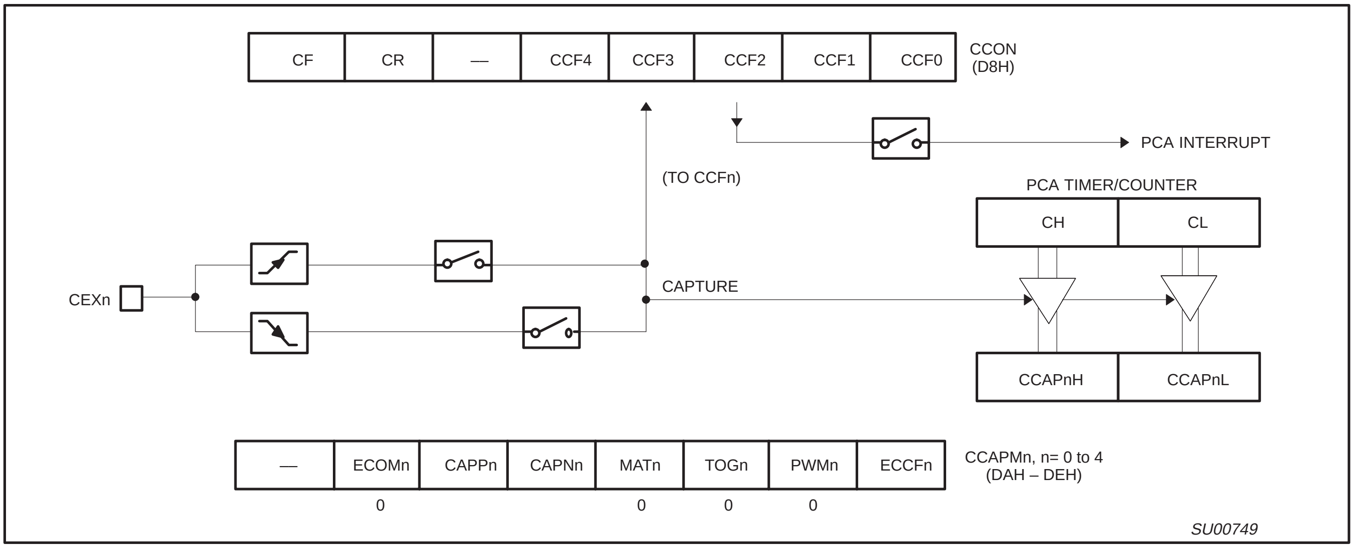


Figure 21. PCA Capture Mode

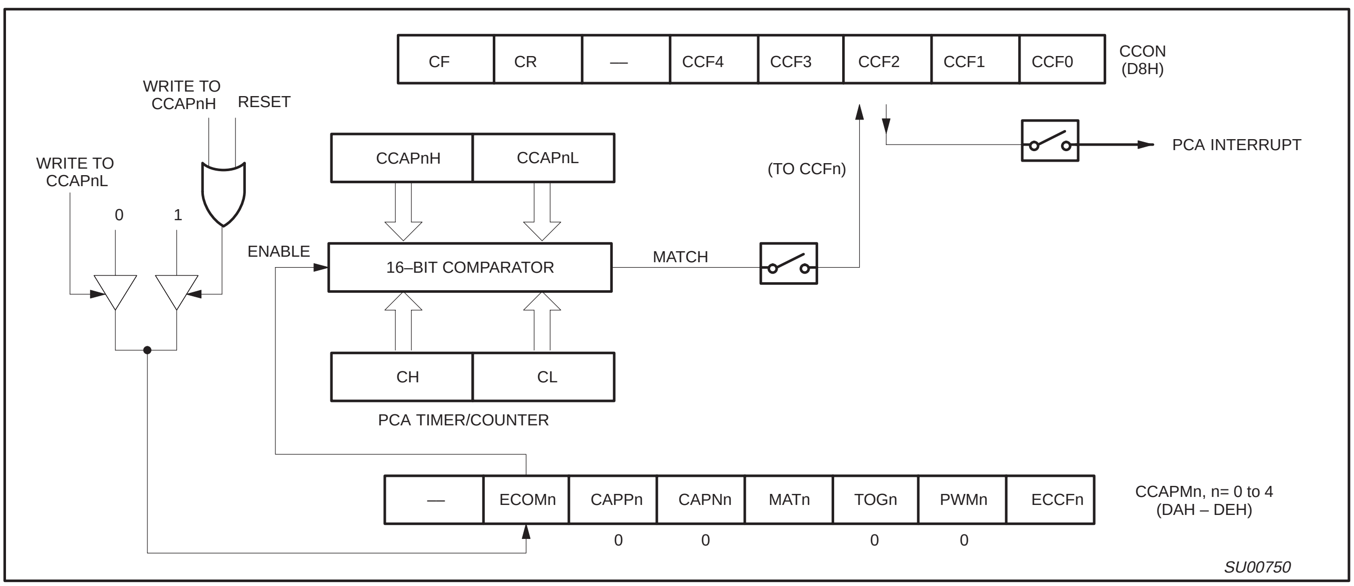


Figure 22. PCA Compare Mode

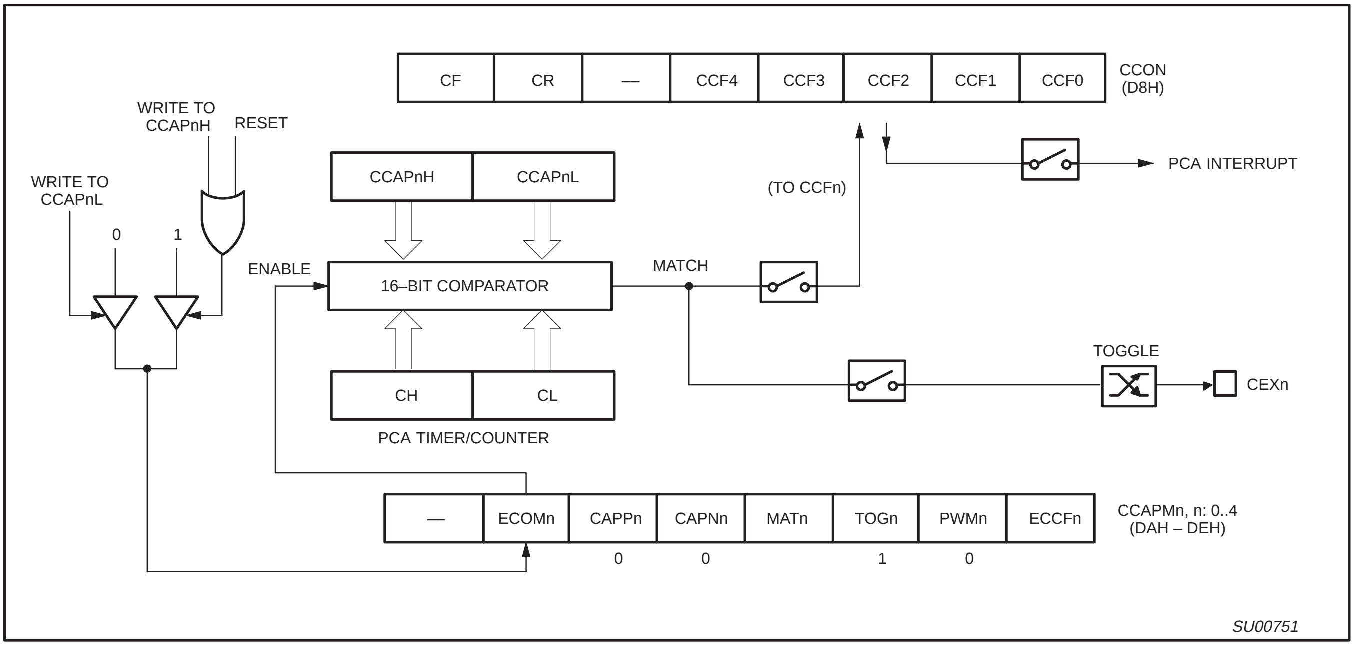


Figure 23. PCA High Speed Output Mode

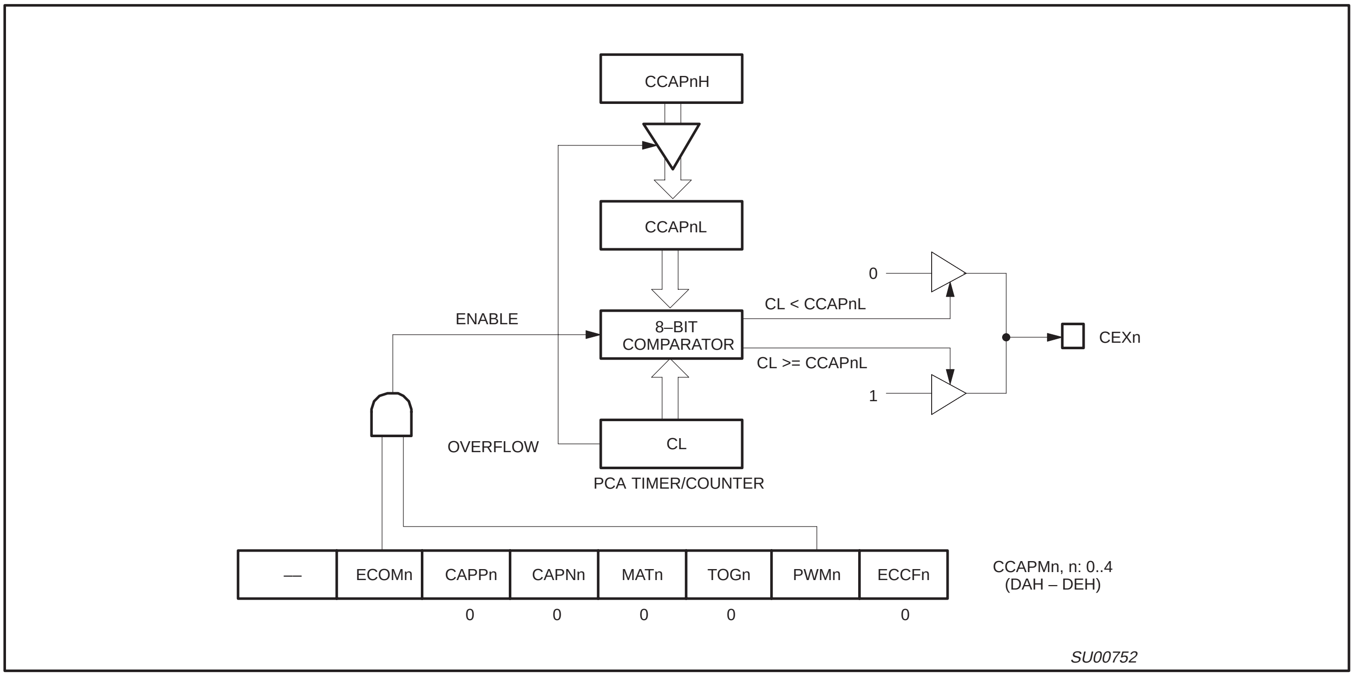


Figure 24. PCA PWM Mode

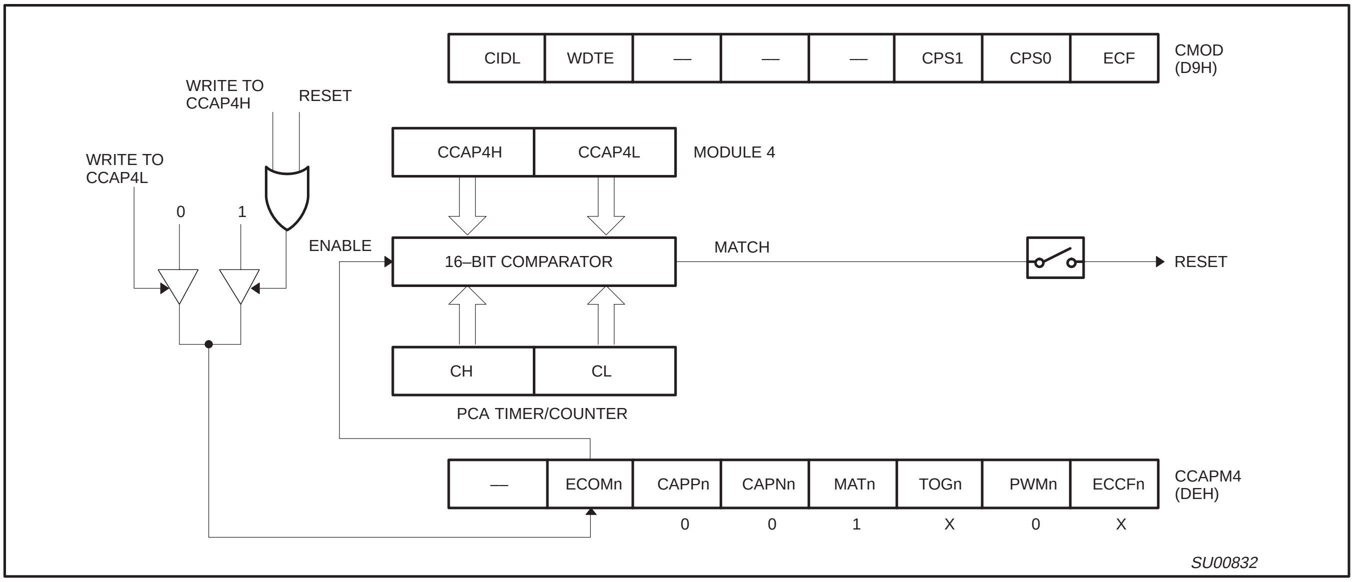


Figure 25. PCA Watchdog Timer m(Module 4 only)

PCA Watchdog Timer

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

Figure 25 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 26 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 26.

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 26. PCA Watchdog Timer Initialization Code

Expanded Data RAM Addressing

The 89C51RX+ 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 (768 for RD+) expanded RAM (ERAM).

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-bytes (768 for RD+) expanded RAM (ERAM, 00H – FFH) are indirectly accessed by move external instruction, MOVX, and with the EXTRAM bit cleared, see Figure 27.

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,#data

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

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

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

MOVX @R0,#data

where R0 contains 0A0H, access the ERAM at address 0A0H rather than external memory. An access to external data memory locations higher than FFH (2FF for RD+) (i.e., 0100H to FFFFH) 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 28.

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.
	1 — ALE is active only during a MOVX or MOVC instruction.
EXTRAM	Internal/External RAM (00H – FFH) access using MOVX @Ri/@DPTR
	EXTRAM — **Operating Mode**
	0 — Internal ERAM (00H–FFH) (00H–2FFH for RD+) 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.

SU00833A

Figure 27. AUXR: Auxiliary Register (RX+ only)

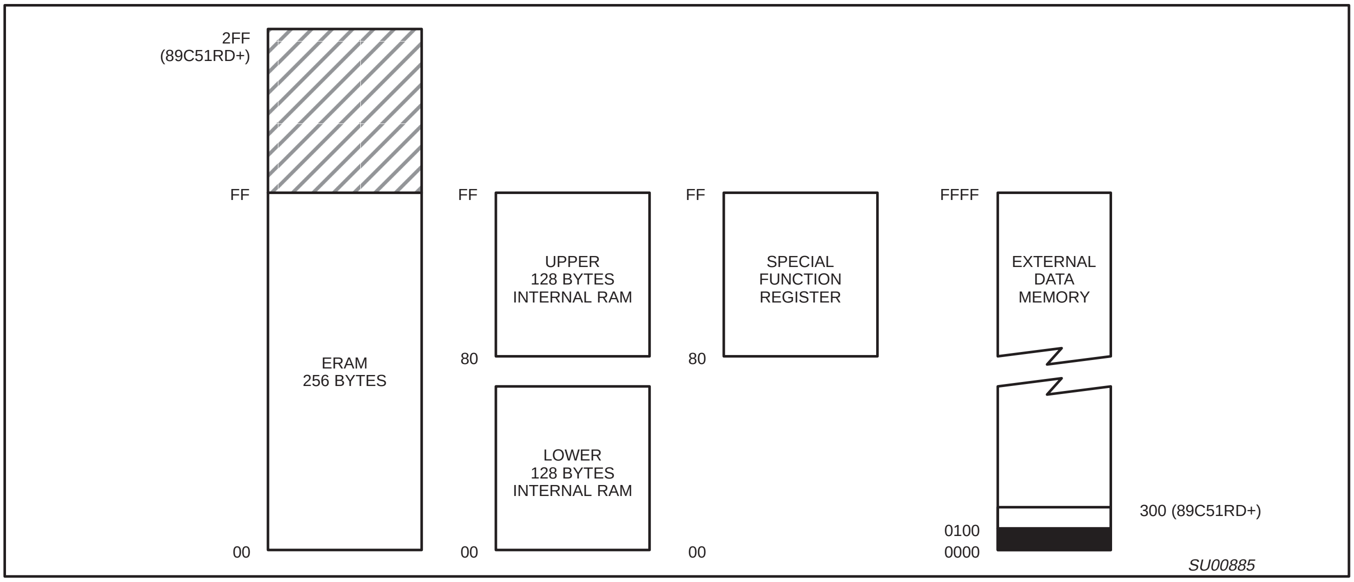


Figure 28. Internal and External Data Memory Address Space with EXTRAM = 0

HARDWARE WATCHDOG TIMER (ONE-TIME ENABLED WITH RESET-OUT FOR 89C51RX+)

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, user must write 01EH and 0E1H in sequence to the WDTRST, SFR location 0A6H. When 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 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, user must write 01EH and 0E1H in sequence to the WDTRST, SFR location 0A6H. When WDT is enabled, the user needs to service it by writing to 01EH and 0E1H to WDTRST to avoid WDT overflow. The 14-bit counter overflows when it reaches 16383 (3FFFH) and this will reset the device. When 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 WDT overflows, it will generate an output RESET pulse at the reset pin (see note below). The RESET pulse duration is $98 \times T_{OSC}$, 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.

NOTE:
Applications which use the Watchdog Timer will need to include a series resistor (1 k, ±20%) between the reset pin and ANY external components. Without this series resistor the Watchdog Timer will not function.

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; 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, $\overline{PSEN}$ [8, 7]	V_{CC} = 4.5 V I_{OL} = 3.2 mA[2]			0.4	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 36): Active mode (see Note 5) Idle mode (see Note 5) Power-down mode or clock stopped (see Figure 40 for conditions)	See Note 5 T_{amb} = 0°C to 70°C T_{amb} = –40°C to +85°C		 20	 100 125	 µA µA
I_{PP}	Programming current	11.5 V ≤ V_{PP} ≤ 12.5 V			50	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 37 through 40 for I_{CC} test conditions and Figure 36 for I_{CC} vs Freq.
   Active mode: $I_{CC(MAX)}$ = (0.9 × FREQ. + 20)mA for all devices.
   Idle mode: $I_{CC(MAX)}$ = (0.37 × FREQ. +1.0)mA
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

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]		33MHz CLOCK		UNIT
			MIN	MAX	MIN	MAX	
$1/t_{CLCL}$	29	Oscillator frequency Speed versions: I;J;U (33 MHz)	3.5	33	3.5	33	MHz
t_{LHLL}	29	ALE pulse width	$2t_{CLCL}–40$		21		ns
t_{AVLL}	29	Address valid to ALE low	$t_{CLCL}–25$		5		ns
t_{LLAX}	29	Address hold after ALE low	$t_{CLCL}–25$		5		ns
t_{LLIV}	29	ALE low to valid instruction in		$4t_{CLCL}–65$		55	ns
t_{LLPL}	29	ALE low to $\overline{PSEN}$ low	$t_{CLCL}–25$		5		ns
t_{PLPH}	29	$\overline{PSEN}$ pulse width	$3t_{CLCL}–45$		45		ns
t_{PLIV}	29	$\overline{PSEN}$ low to valid instruction in		$3t_{CLCL}–60$		30	ns
t_{PXIX}	29	Input instruction hold after $\overline{PSEN}$	0		0		ns
t_{PXIZ}	29	Input instruction float after $\overline{PSEN}$		$t_{CLCL}–25$		5	ns
t_{AVIV}	29	Address to valid instruction in		$5t_{CLCL}–80$		70	ns
t_{PLAZ}	29	$\overline{PSEN}$ low to address float		10		10	ns
Data Memory							
t_{RLRH}	30, 31	$\overline{RD}$ pulse width	$6t_{CLCL}–100$		82		ns
t_{WLWH}	30, 31	$\overline{WR}$ pulse width	$6t_{CLCL}–100$		82		ns
t_{RLDV}	30, 31	$\overline{RD}$ low to valid data in		$5t_{CLCL}–90$		60	ns
t_{RHDX}	30, 31	Data hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	30, 31	Data float after $\overline{RD}$		$2t_{CLCL}–28$		32	ns
t_{LLDV}	30, 31	ALE low to valid data in		$8t_{CLCL}–150$		90	ns
t_{AVDV}	30, 31	Address to valid data in		$9t_{CLCL}–165$		105	ns
t_{LLWL}	30, 31	ALE low to $\overline{RD}$ or $\overline{WR}$ low	$3t_{CLCL}–50$	$3t_{CLCL}+50$	40	140	ns
t_{AVWL}	30, 31	Address valid to $\overline{WR}$ low or $\overline{RD}$ low	$4t_{CLCL}–75$		45		ns
t_{QVWX}	30, 31	Data valid to $\overline{WR}$ transition	$t_{CLCL}–30$		0		ns
t_{WHQX}	30, 31	Data hold after $\overline{WR}$	$t_{CLCL}–25$		5		ns
t_{QVWH}	31	Data valid to $\overline{WR}$ high	$7t_{CLCL}–130$		80		ns
t_{RLAZ}	30, 31	$\overline{RD}$ low to address float		0		0	ns
t_{WHLH}	30, 31	$\overline{RD}$ or $\overline{WR}$ high to ALE high	$t_{CLCL}–25$	$t_{CLCL}+25$	5	55	ns
External Clock							
t_{CHCX}	33	High time	17	$t_{CLCL}–t_{CLCX}$			ns
t_{CLCX}	33	Low time	17	$t_{CLCL}–t_{CHCX}$			ns
t_{CLCH}	33	Rise time		5			ns
t_{CHCL}	33	Fall time		5			ns
Shift Register							
t_{XLXL}	32	Serial port clock cycle time	$12t_{CLCL}$		360		ns
t_{QVXH}	32	Output data setup to clock rising edge	$10t_{CLCL}–133$		167		ns
t_{XHQX}	32	Output data hold after clock rising edge	$2t_{CLCL}–80$		50		ns
t_{XHDX}	32	Input data hold after clock rising edge	0		0		ns
t_{XHDV}	32	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 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 – $\overline{PSEN}$
Q – Output data
R – $\overline{RD}$ signal
t – Time
V – Valid
W – $\overline{WR}$ 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 $\overline{PSEN}$ low.

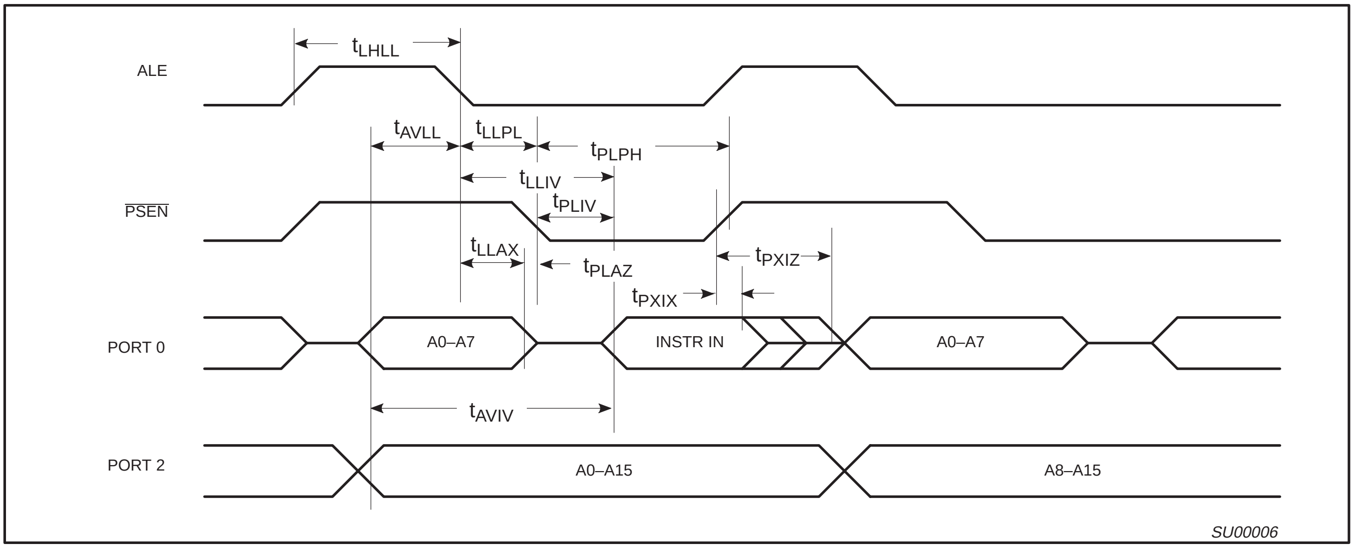


Figure 29. External Program Memory Read Cycle

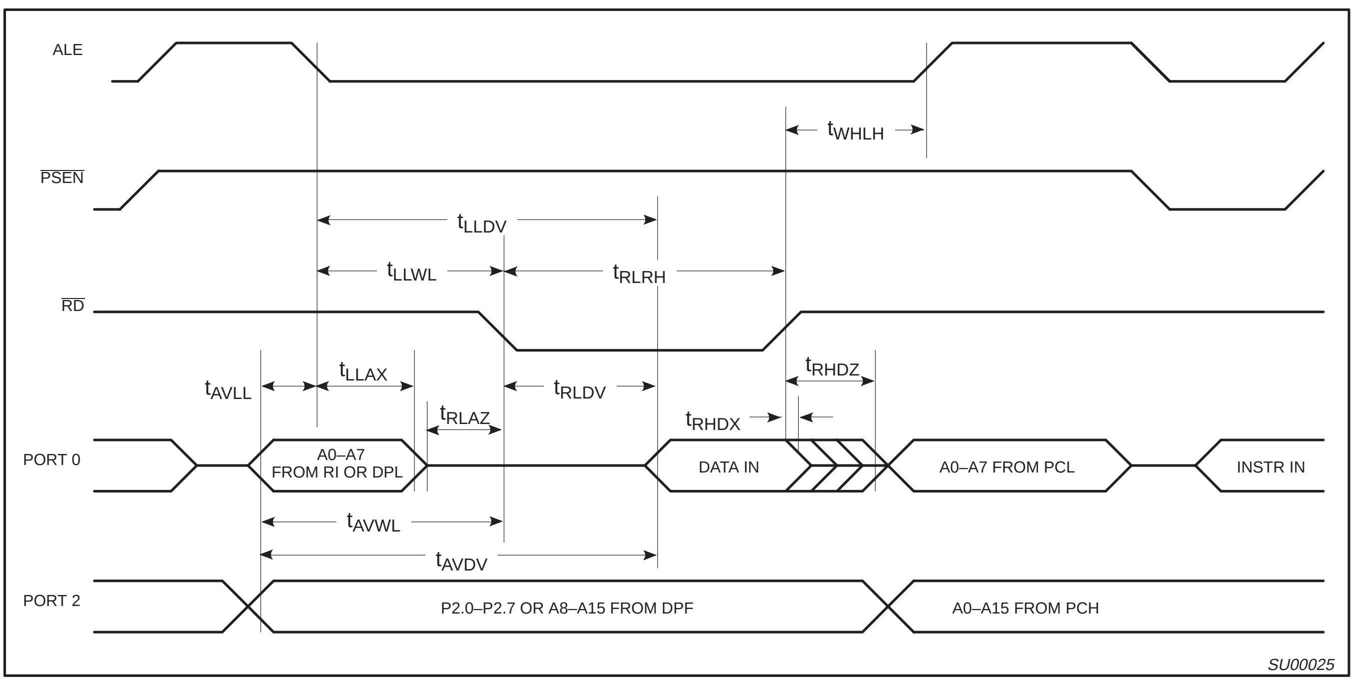


Figure 30. External Data Memory Read Cycle

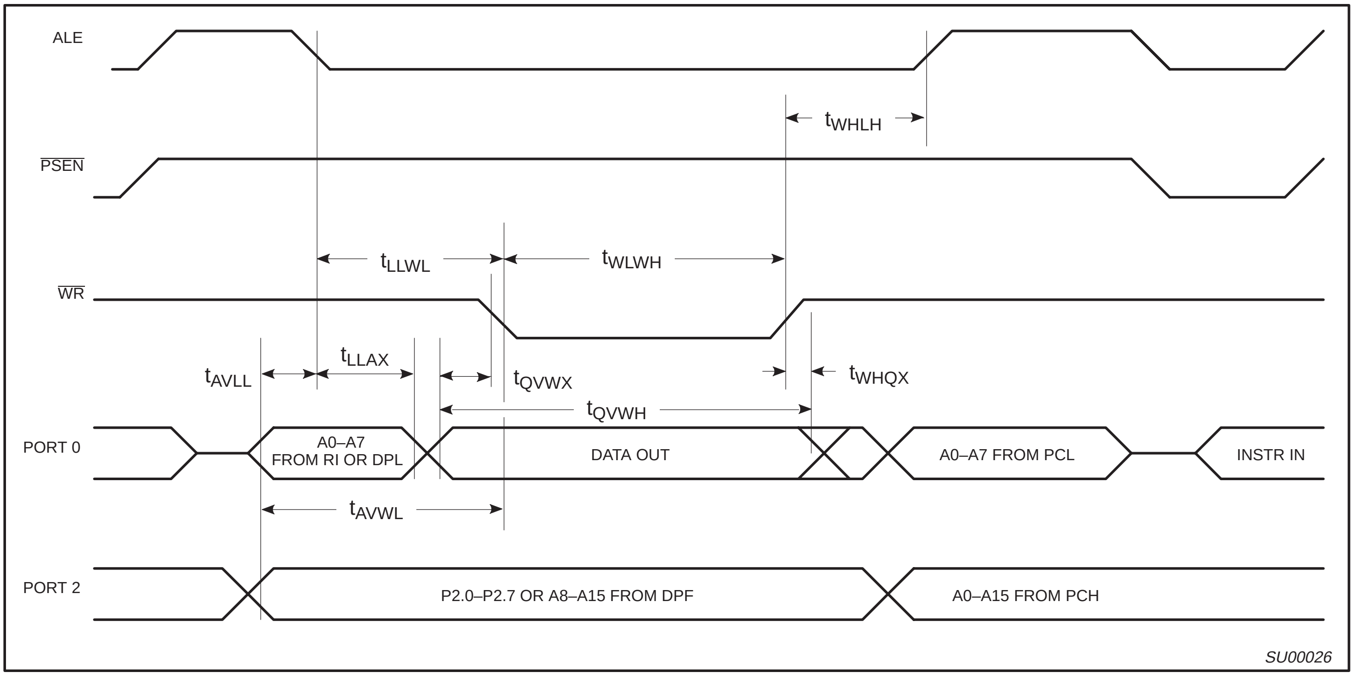


Figure 31. External Data Memory Write Cycle

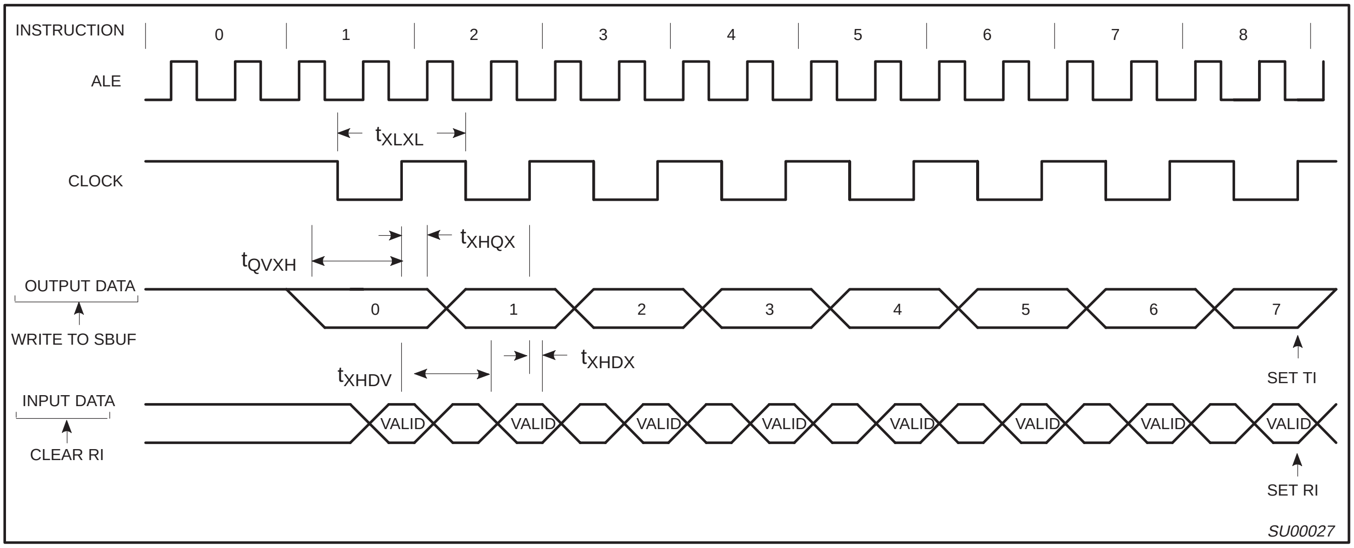


Figure 32. Shift Register Mode Timing

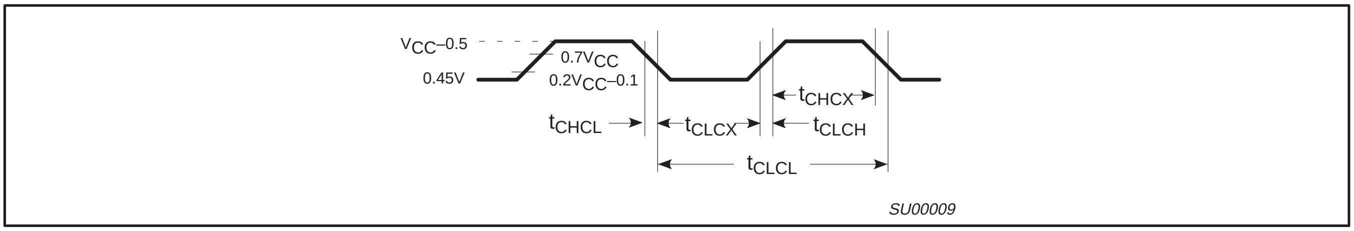


Figure 33. External Clock Drive

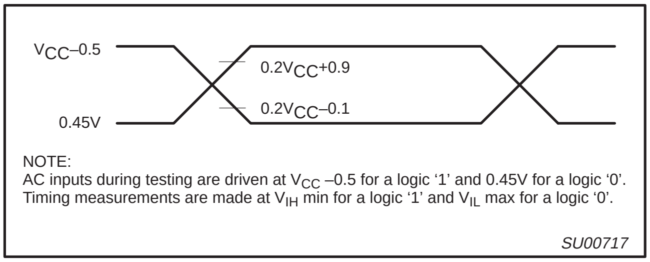


Figure 34. AC Testing Input/Output

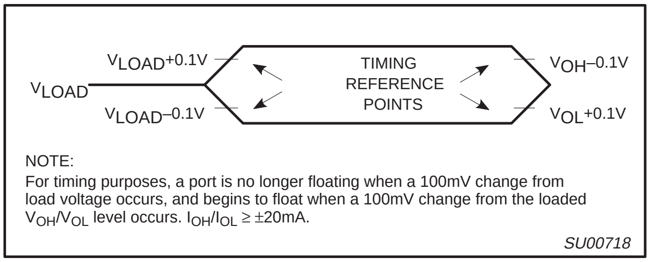


Figure 35. Float Waveform

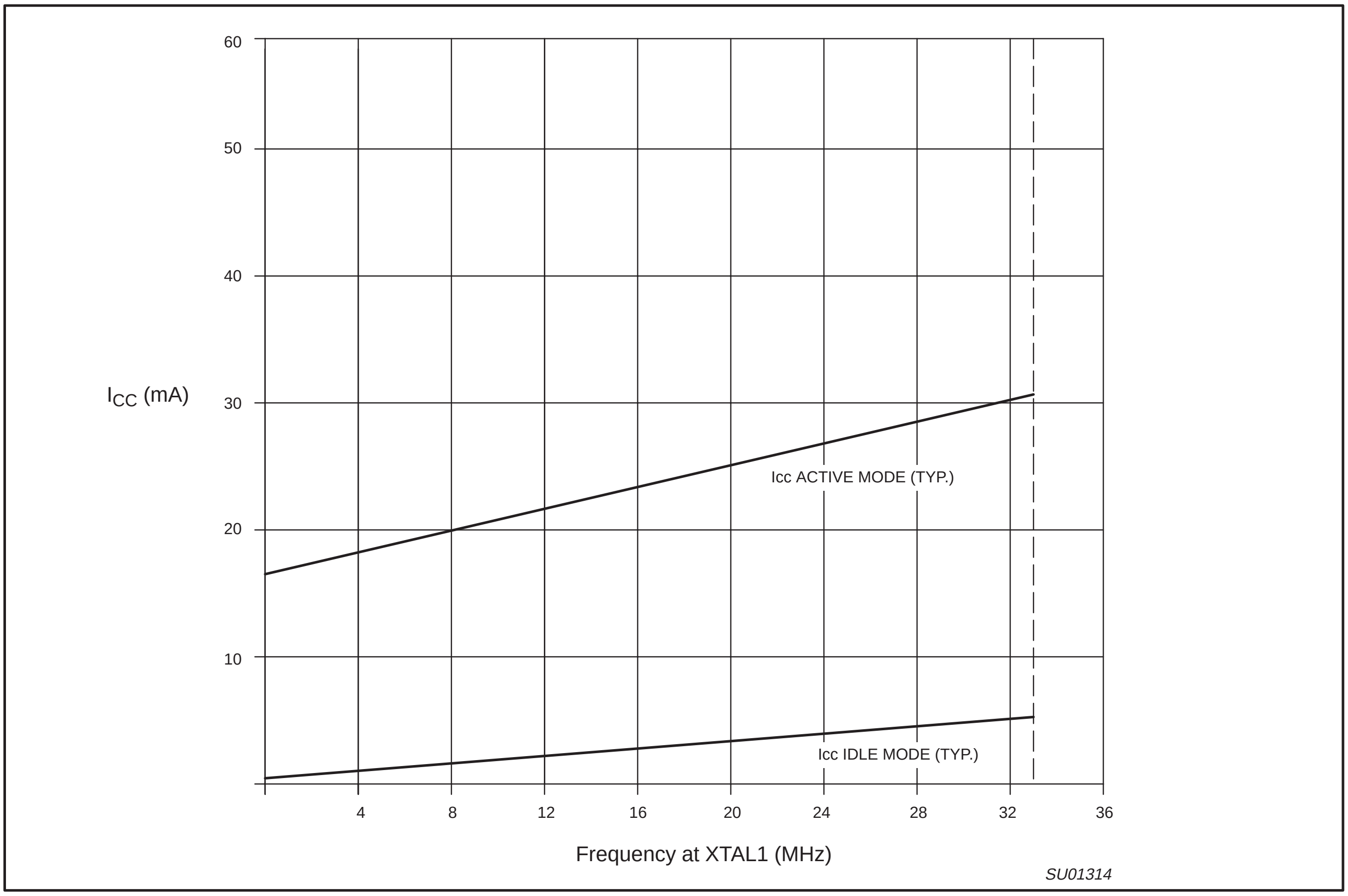


Figure 36. I_{CC} vs. FREQ
Valid only within frequency specifications of the device under test

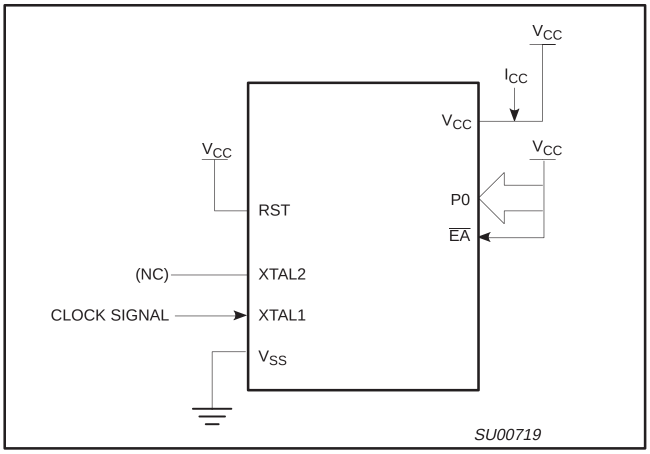


Figure 37. I_{CC} Test Condition, Active Mode
All other pins are disconnected

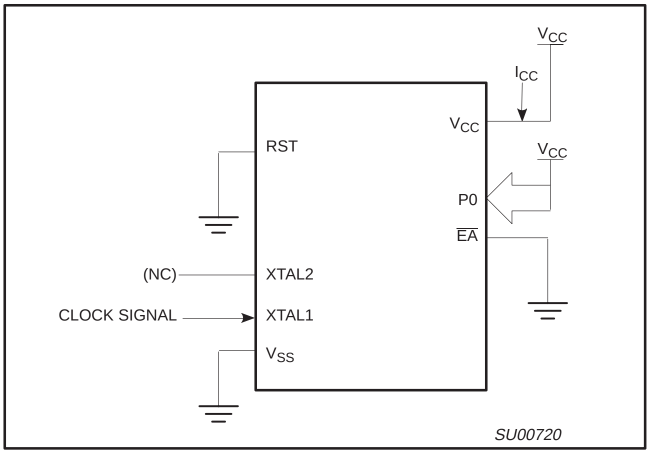


Figure 38. I_{CC} Test Condition, Idle Mode
All other pins are disconnected

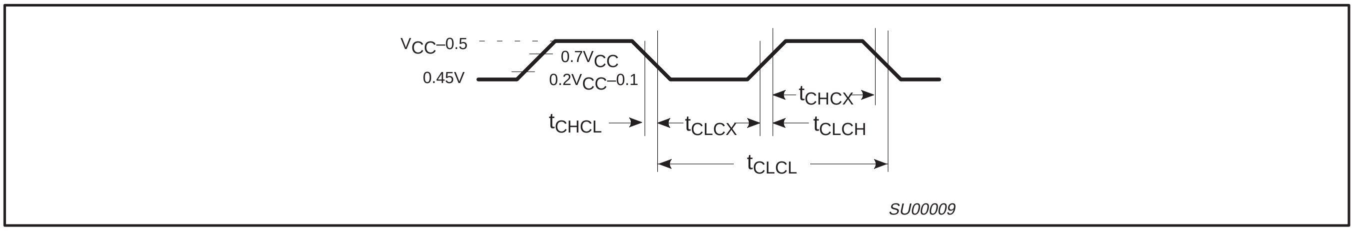


Figure 39. Clock Signal Waveform for I_{CC} Tests in Active and Idle Modes
$t_{CLCH} = t_{CHCL} = 5$ ns

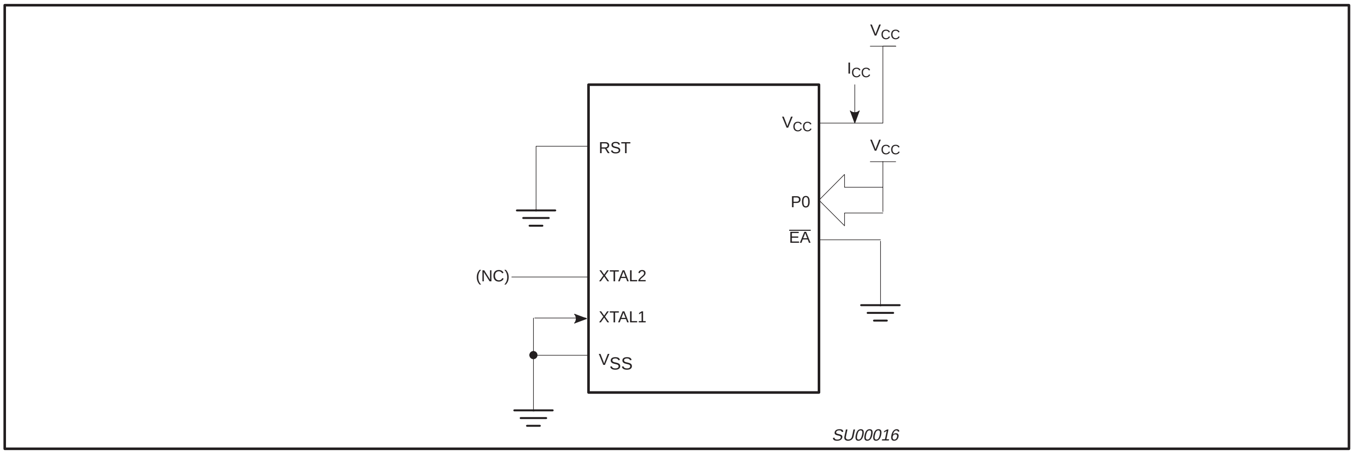


Figure 40. I_{CC} Test Condition, Power Down Mode
All other pins are disconnected. V_{CC} = 2 V to 5.5 V

FLASH EPROM MEMORY

GENERAL DESCRIPTION

The 89C51RX+ 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 byte 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 89C51RX+ FLASH reliably stores memory contents even after 1000 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 89C51RX+ uses a 12.0 V ±0.5 V V_{PP} supply to perform the Program/Erase algorithms.

FEATURES

- FLASH EPROM internal program memory with Block Erase.
- Internal 1 k byte fixed boot ROM, containing low-level in-system programming routines and a default serial loader. The Boot ROM can be turned off to provide access to the full 64 k byte 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 Boot ROM allows programming via the serial port without the need for a user provided loader.
- Up to 64 k byte external program memory if the internal program memory is disabled ($\overline{EA}$ = 0).
- Programming and erase voltage 12 V ±0.5 V.
- Read/Programming/Erase:
  - Byte-wise read (100 ns access time).
  - Byte Programming (20 μs).
  - Typical erase times:
    Block Erase (8 k bytes or 16 k bytes) in 3 seconds.
    Full Erase (64 k bytes) in 3 seconds.
- Parallel programming with 87C51 compatible hardware interface to programmer.
- In-circuit programming.
- Programmable security for the code in the FLASH.
- 1000 minimum erase/program cycles for each byte.
- 10 year minimum data retention.

CAPABILITIES OF THE PHILIPS 89C51 FLASH-BASED MICROCONTROLLERS

FLASH organization (89C51RC+ and 89C51RD+)

The 89C51RD+ contains 64 k bytes of FLASH program memory. This memory is organized as 5 separate blocks. The first two blocks are 8 k bytes in size, filling the program memory space from address 0 through 3FFF hex. The final three blocks are 16 k bytes in size and occupy addresses from 4000 through FFFF hex.

The 89C51RC+ contains 32 k bytes of FLASH program memory, which is organized as two blocks of 8k bytes, followed by one block of 16 k bytes.

Figure 41 depicts the FLASH memory configurations.

FLASH Programming and Erasure

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 a common entry point in the Boot ROM. 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 same common entry point in the Boot ROM that can be used by the end-user application. Third, the FLASH may be programmed or erased using the 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.

Boot ROM

When the microcontroller programs its own FLASH memory, all of the low level details are handled by code that is permanently contained in a 1 k byte “Boot ROM” that is separate from the FLASH memory. A user program simply calls the common entry point with appropriate parameters in the Boot ROM to accomplish the desired operation. Boot ROM operations include things like: erase block, program byte, verify byte, program security lock bit, etc. The Boot ROM overlays the program memory space at the top of the address space from FC00 to FFFF hex, when it is enabled. The Boot ROM may be turned off so that the upper 1k bytes of FLASH program memory are accessible for execution.

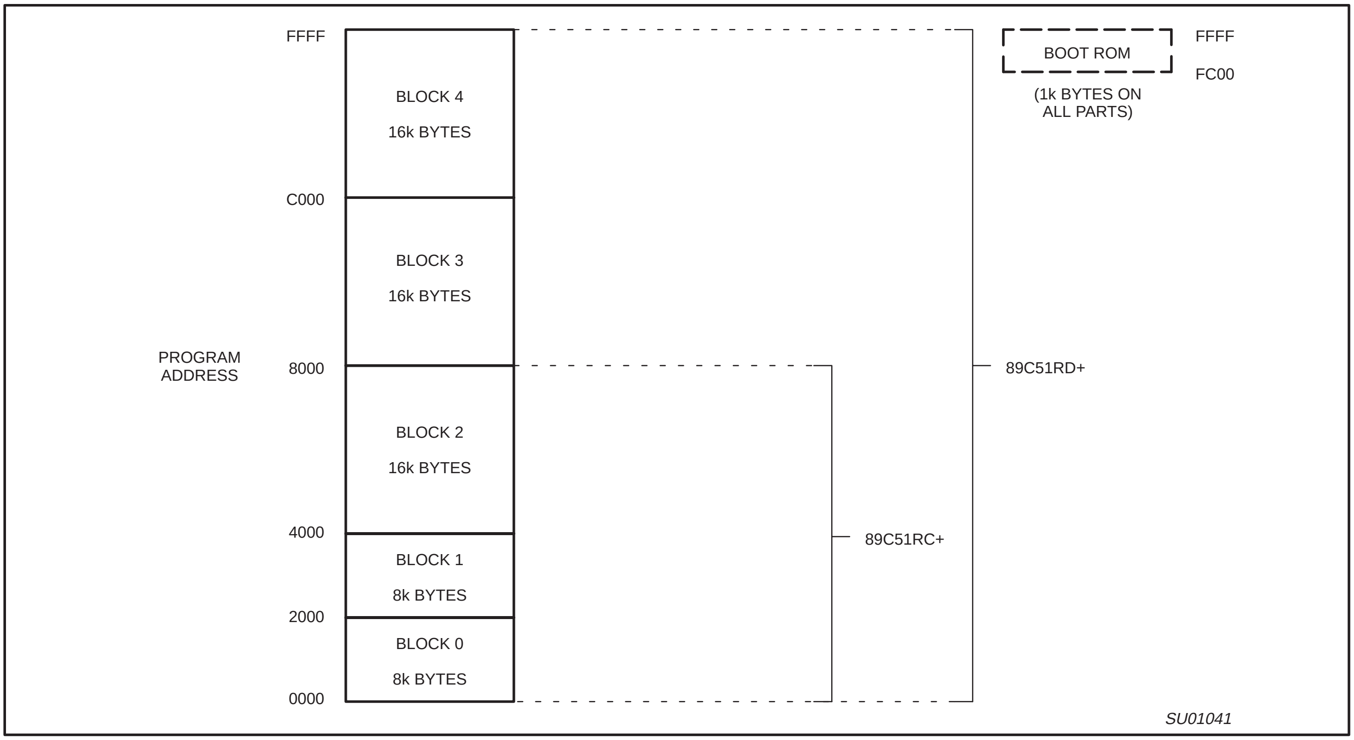


Figure 41. FLASH Memory Configurations

Power-On Reset Code Execution

The 89C51RX+ contains two special FLASH registers: the BOOT VECTOR and the STATUS BYTE. At the falling edge of reset, the 89C51RX+ 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 PSEN LOW, $\overline{EA}$ greater than V_{IH} (such as +12 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.

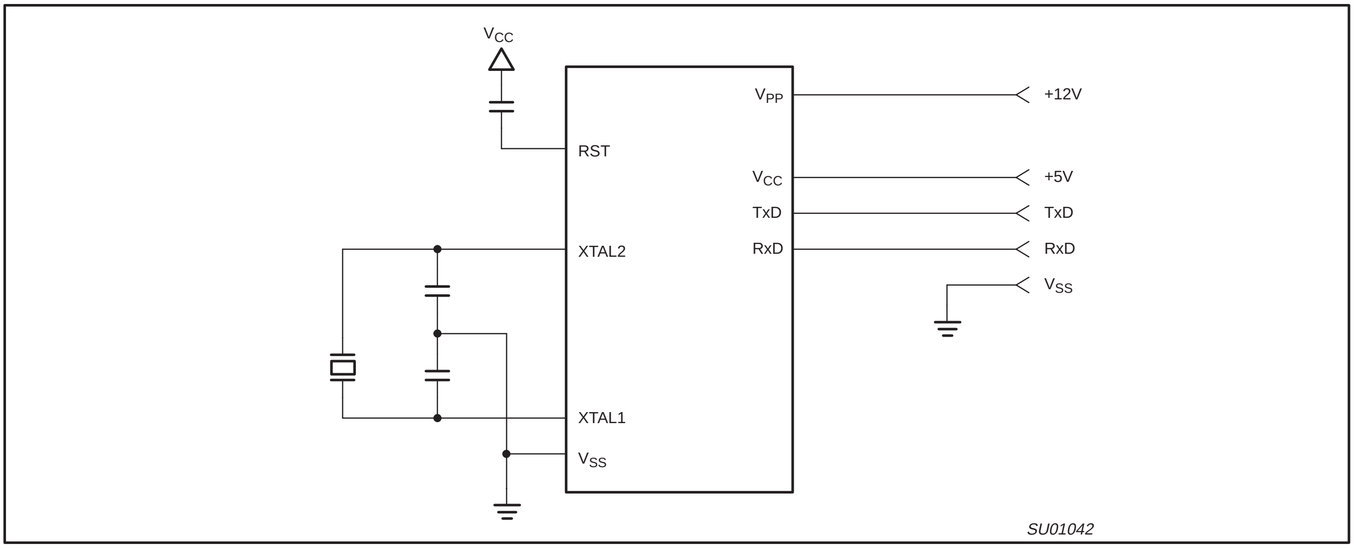


Figure 42. In-System Programming with a Minimum of Pins

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 89C51RX+ through the serial port. This firmware is provided by Philips and embedded within each 89C51RX+ 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 42). 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 provide a minimum 50mA. V_{PP} is not allowed to exceed datasheet limits.

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 89C51RX+ 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 89C51RX+ 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 8.

As a record is received by the 89C51RX+, 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 89C51RX+ 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 89C51RX+ 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 89C51RX+ with information required to generate the proper timing. Record type 02 is provided for this purpose.

WinISP, a software utility to implement ISP programming with a PC, is available from Philips.

Table 8. 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
02	Specify Oscillator Frequency :01xxxx02ddcc Where: xxxx = required field, but value is a “don’t care” dd = integer oscillator frequency rounded down to nearest MHz cc = checksum Example: :0100000210ED (dd = 10h = 16, used for 16.0–16.9 MHz)
03	Miscellaneous Write Functions :nnxxxx03ffssddcc Where: nn = number of bytes (hex) in record xxxx = required field, but value is a “don’t care” 03 = Write Function ff = subfunction code ss = selection code dd = data input (as needed) cc = checksum Subfunction Code = 01 (Erase Blocks) ff = 01 ss = block code as shown below: block 0, 0k to 8k, 00H block 1, 8k to 16k, 20H block 2, 16k to 32k, 40H block 3, 32k to 48k, 80H block 4, 48k to 64k, C0H Example: :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 Example: :030000030601FCF7 program boot vector with 0FCH

<table>
<tr><th>RECORD TYPE</th><th>COMMAND/DATA FUNCTION</th></tr>
<tr><td>04</td><td>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
<code>:05xxxx04sssseeeeffcc</code>
Where:
<code>05 = number of bytes (hex) in record</code>
<code>xxxx = required field, but value is a “don’t care”</code>
<code>04 = “Display Device Data or Blank Check” function code</code>
<code>ssss = starting address</code>
<code>eeee = ending address</code>
<code>ff = subfunction</code>
<code>00 = display data</code>
<code>01 = blank check</code>
<code>cc = checksum</code>
Example:
<code>:0500000440004FFF0069 display 4000–4FFF</code></td></tr>
<tr><td>05</td><td>Miscellaneous Read Functions

General Format of Function 05
<code>:02xxxx05ffsscc</code>
Where:
<code>02 = number of bytes (hex) in record</code>
<code>xxxx = required field, but value is a “don’t care”</code>
<code>05 = “Miscellaneous Read” function code</code>
<code>ffss = subfunction and selection code</code>
<code>0000 = read signature byte – manufacturer id (15H)</code>
<code>0001 = read signature byte – device id # 1 (C2H)</code>
<code>0002 = read signature byte – device id # 2</code>

<code>0700 = read security bits</code>
<code>0701 = read status byte</code>
<code>0702 = read boot vector</code>
<code>cc = checksum</code>
Example:
<code>:020000050001F8 read signature byte – device id # 1</code></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 9.

Table 9. IAP calls

IAP CALL	PARAMETER

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

ERASE BLOCK	Input Parameters:
	`R0 = osc freq (integer)`
	`R1 = 01h`
	`DPH = block code as shown below:`
	`block 0,  0k to  8k, 00H`
	`block 1,  8k to 16k, 20H`
	`block 2, 16k to 32k, 40H`
	`block 3, 32k to 48k, 80H`
	`block 4, 48k to 64k, C0H`
	
	`DPL = 00h`
	Return Parameter
	`none`

ERASE BOOT VECTOR	Input Parameters:
	`R0 = osc freq (integer)`
	`R1 = 04h`
	`DPH = 00h`
	`DPL = don't care`
	Return Parameter
	`none`

PROGRAM SECURITY BIT	Input Parameters:
	`R0 = osc freq (integer)`
	`R1 = 05h`
	`DPH = 00h`
	`DPL = 00h – security bit # 1 (inhibit writing to Flash)`
	`01h – security bit # 2 (inhibit Flash verify)`
	`02h – security bit # 3 (disable external memory)`
	Return Parameter
	`none`

PROGRAM STATUS BYTE	Input Parameters:
	`R0 = osc freq (integer)`
	`R1 = 06h`
	`DPH = 00h`
	`DPL = 00h – program status byte`
	`ACC = status byte`
	Return Parameter
	`ACC = status byte`

PROGRAM BOOT VECTOR	Input Parameters:
	`R0 = osc freq (integer)`
	`R1 = 06h`
	`DPH = 00h`
	`DPL = 01h – program boot vector`
	`ACC = boot vector`
	Return Parameter
	`ACC = boot vector`

READ DEVICE DATA	Input Parameters:
	`R1 = 03h`
	`DPTR = address of byte to read`
	Return Parameter
	`ACC = value of byte read`

IAP CALL	PARAMETER

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

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

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

READ SECURITY BITS	Input Parameters:
	`R0 = osc freq (integer)`
	`R1 = 07h`
	`DPH = 00h`
	`DPL = 00h (security bits)`
	Return Parameter
	`ACC = value of byte read`

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

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

Security

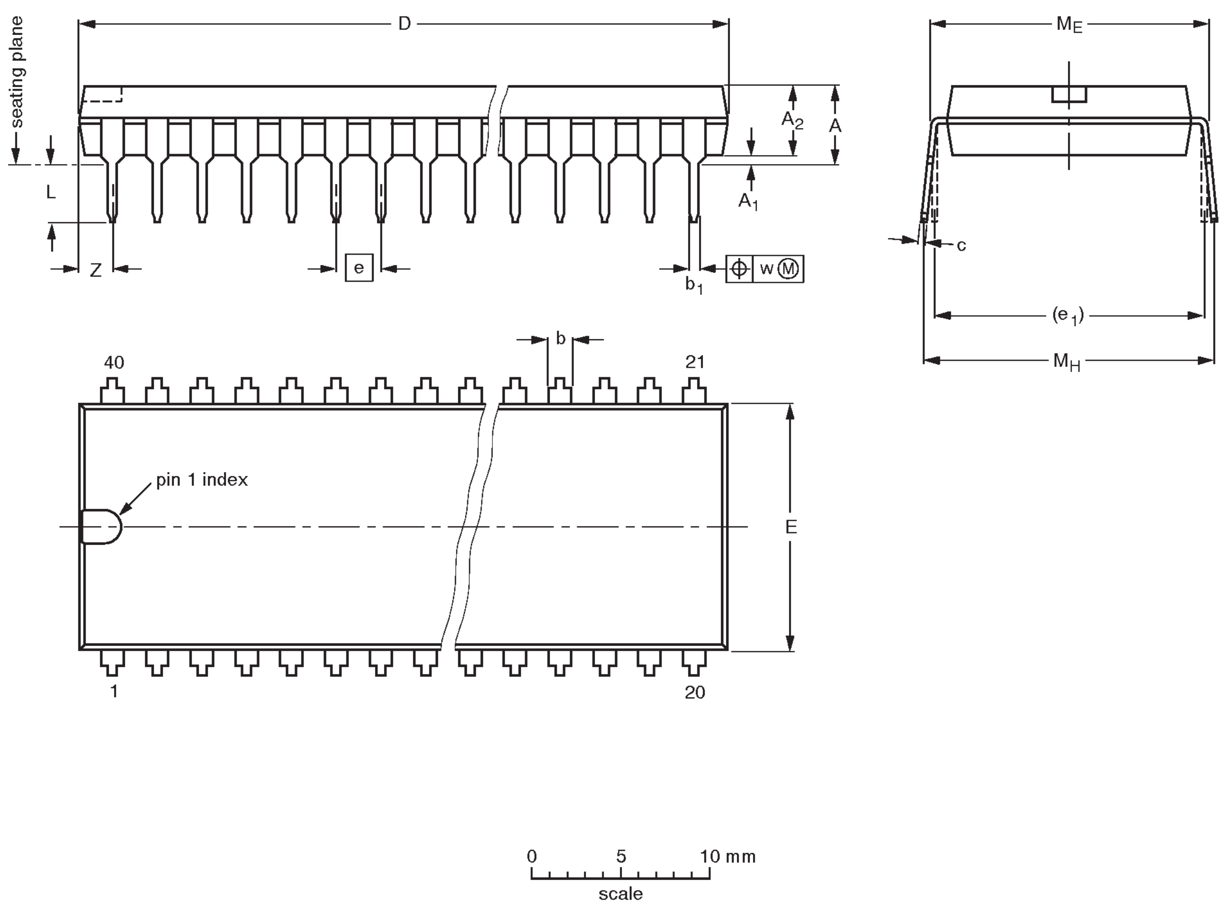
The security feature protects against software piracy and prevents the contents of the Flash from being read. The Security Lock bits are located in Flash. The P89C51RC+/P89C51RD+ has three programmable security lock bits that will provide different levels of protection for the on-chip code and data (see Table 10).

Table 10.

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

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



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

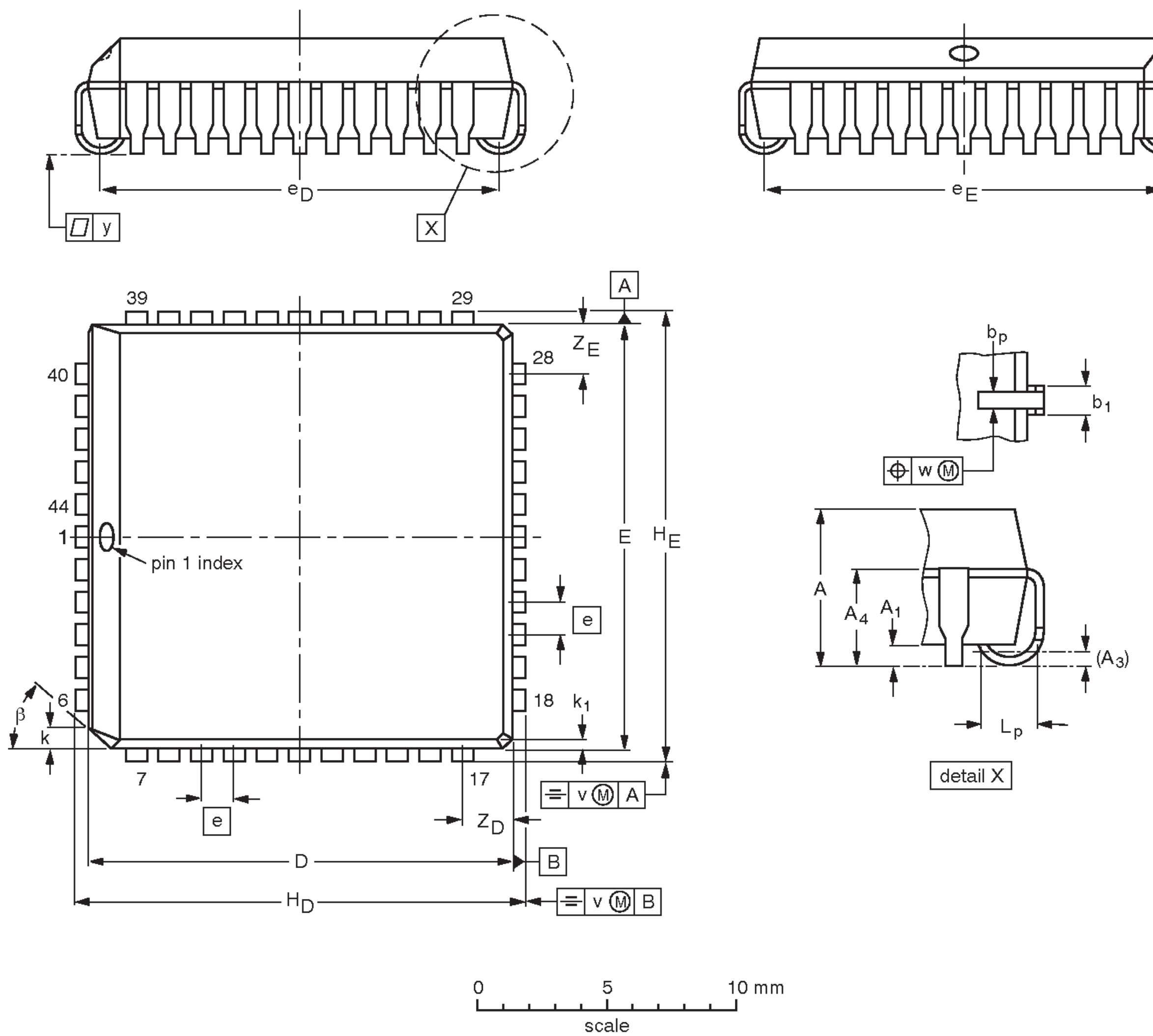
UNIT	A max.	A_1 min.	A_2 max.	b	b_1	c	D (1)	E (1)	e	e_1	L	M_E	M_H	w	Z (1) max.
mm	4.7	0.51	4.0	1.70 1.14	0.53 0.38	0.36 0.23	52.50 51.50	14.1 13.7	2.54	15.24	3.60 3.05	15.80 15.24	17.42 15.90	0.254	2.25
inches	0.19	0.020	0.16	0.067 0.045	0.021 0.015	0.014 0.009	2.067 2.028	0.56 0.54	0.10	0.60	0.14 0.12	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-015AJ				~~92-11-17~~ 95-01-14

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

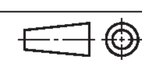


DIMENSIONS (millimetre dimensions are derived from the original inch dimensions)

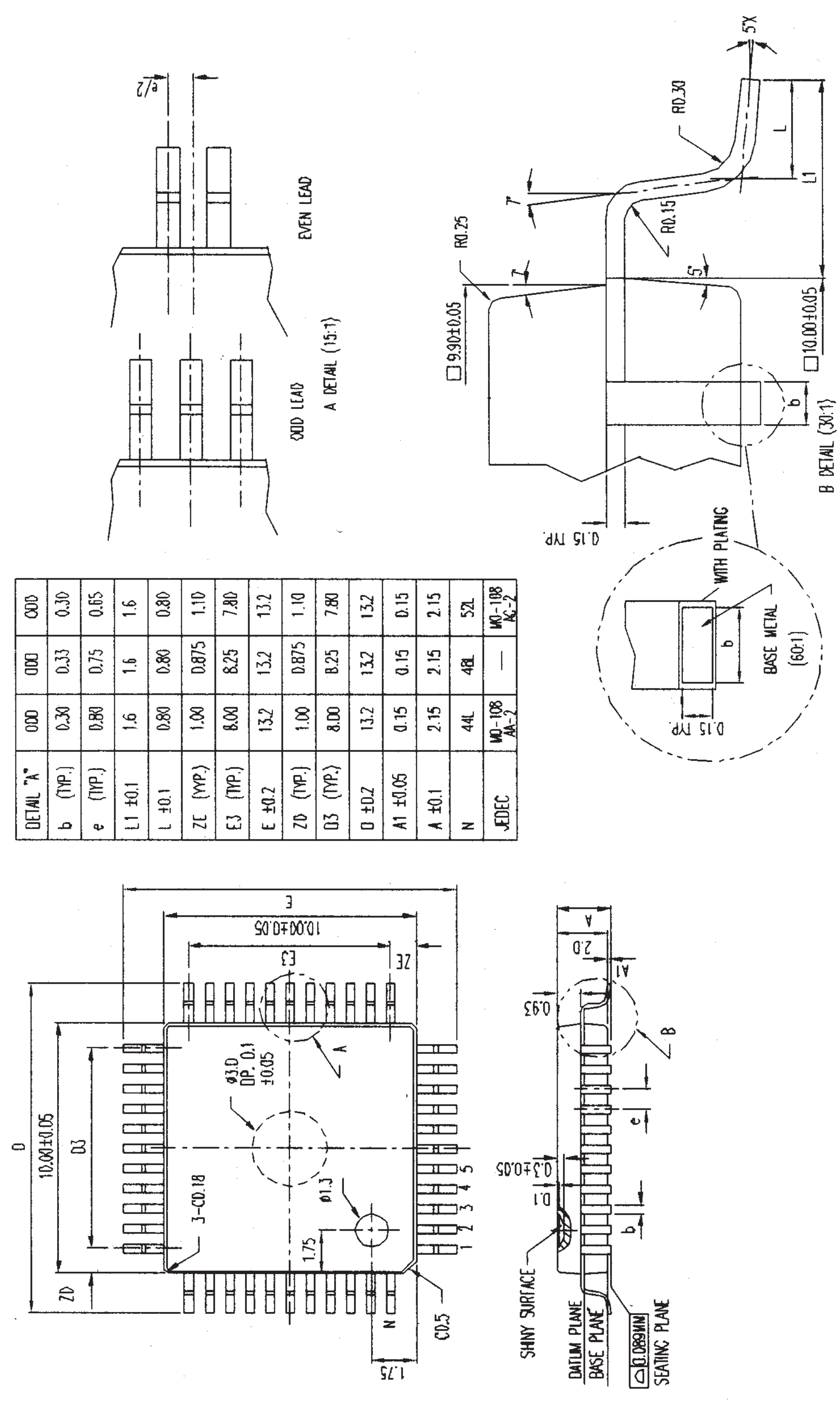
UNIT	A	A_1 min.	A_3	A_4 max.	b_p	b_1	$D^{(1)}$	$E^{(1)}$	e	e_D	e_E	H_D	H_E	k	k_1 max.	L_p	v	w	y	$Z_D^{(1)}$ max.	$Z_E^{(1)}$ max.	β
mm	4.57 4.19	0.51	0.25	3.05	0.53 0.33	0.81 0.66	16.66 16.51	16.66 16.51	1.27	16.00 14.99	16.00 14.99	17.65 17.40	17.65 17.40	1.22 1.07	0.51	1.44 1.02	0.18	0.18	0.10	2.16	2.16	45°
inches	0.180 0.165	0.020	0.01	0.12	0.021 0.013	0.032 0.026	0.656 0.650	0.656 0.650	0.05	0.630 0.590	0.630 0.590	0.695 0.685	0.695 0.685	0.048 0.042	0.020	0.057 0.040	0.007	0.007	0.004	0.085	0.085	

Note

1. Plastic or metal protrusions of 0.01 inches maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT187-2	112E10	MO-047AC				~~95-02-25~~ 97-12-16

QFP44: plastic quad flat package; 44 leads



DETAIL "A"	ODD	ODD	ODD
b (TYP.)	0.30	0.33	0.30
e (TYP.)	0.80	0.75	0.65
L1 ±0.1	1.6	1.6	1.6
L ±0.1	0.80	0.80	0.80
ZE (YYP.)	1.00	0.875	1.10
E3 (TYP.)	8.00	8.25	7.80
E ±0.2	13.2	13.2	13.2
ZD (TYP.)	1.00	0.875	1.10
D3 (TYP.)	8.00	8.25	7.80
D ±0.2	13.2	13.2	13.2
A1 ±0.05	0.15	0.15	0.15
A ±0.1	2.15	2.15	2.15
N	44L	48L	52L
JEDEC	MO-108 AA-2	—	MO-108 AC-2

NOTES

Data sheet status

Data sheet status	Product status	Definition [1]
Objective specification	Development	This data sheet contains the design target or goal specifications for product development. Specification may change in any manner without notice.
Preliminary specification	Qualification	This data sheet contains preliminary data, and supplementary data will be published at a later date. Philips Semiconductors reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Product specification	Production	This data sheet contains final specifications. Philips Semiconductors reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

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

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 134). 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.

Philips Semiconductors
811 East Arques Avenue
P.O. Box 3409
Sunnyvale, California 94088–3409
Telephone 800-234-7381

Date of release: 10-99

Document order number: 9397-750-06605