

# P89LV51RB2/RC2/RD2

8-bit 80C51 3 V low power 16/32/64 kB Flash microcontroller with 1 kB RAM

Rev. 04 — 02 December 2004

Product data

## 1. General description

---

The P89LV51RB2/RC2/RD2 are 80C51 microcontrollers with 16/32/64 kB Flash and 1024 bytes of data RAM.

A key feature of the P89LV51RB2/RC2/RD2 is its X2 mode option. The design engineer can choose to run the application with the conventional 80C51 clock rate (12 clocks per machine cycle) or select the X2 mode (6 clocks per machine cycle) to achieve twice the throughput at the same clock frequency. Another way to benefit from this feature is to keep the same performance by reducing the clock frequency by half, thus dramatically reducing the EMI.

The Flash program memory supports both parallel programming and in serial In-System Programming (ISP). Parallel programming mode offers gang-programming at high speed, reducing programming costs and time to market. ISP allows a device to be reprogrammed in the end product under software control. The capability to field/update the application firmware makes a wide range of applications possible.

The P89LV51RB2/RC2/RD2 is also In-Application Programmable (IAP), allowing the Flash program memory to be reconfigured even while the application is running.

## 2. Features

---

- 80C51 Central Processing Unit
- 3 V Operating voltage from 0 MHz to 33 MHz
- 16/32/64 kB of on-chip Flash program memory with ISP (In-System Programming) and IAP (In-Application Programming)
- Supports 12-clock (default) or 6-clock mode selection via software or ISP
- SPI (Serial Peripheral Interface) and enhanced UART
- PCA (Programmable Counter Array) with PWM and Capture/Compare functions
- Four 8-bit I/O ports with three high-current Port 1 pins (16 mA each)
- Three 16-bit timers/counters
- Programmable Watchdog timer (WDT)
- Eight interrupt sources with four priority levels
- Second DPTR register
- Low EMI mode (ALE inhibit)
- TTL- and CMOS-compatible logic levels



**PHILIPS**

- Brown-out detection
- Low power modes
  - ◆ Power-down mode with external interrupt wake-up
  - ◆ Idle mode
- DIP40, PLCC44 and TQFP44 packages

### 3. Ordering information

Table 1: Ordering information

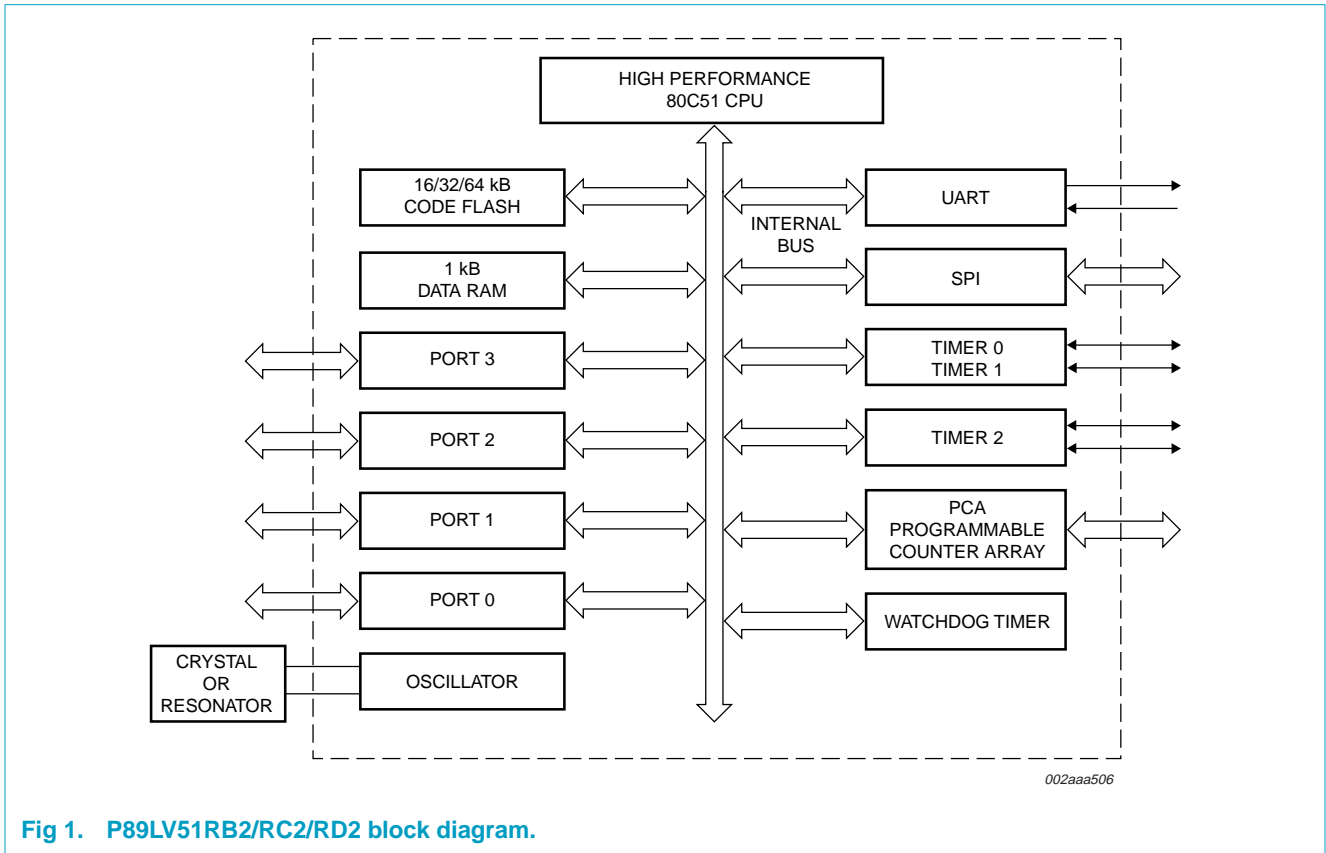
Type number	Package		Version
	Name	Description	
P89LV51RB2BA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2
P89LV51RB2BBC	TQFP44	plastic thin quad flat package; 44 leads; body 10 × 10 × 1.0 mm	SOT376-1
P89LV51RC2FA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2
P89LV51RC2FBC	TQFP44	plastic thin quad flat package; 44 leads; body 10 × 10 × 1.0 mm	SOT376-1
P89LV51RC2BN	DIP40	plastic dual in-line package; 40 leads (600 mil)	SOT129-1
P89LV51RD2BA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2
P89LV51RD2FA			
P89LV51RD2BBC	TQFP44	plastic thin quad flat package; 44 leads; body 10 × 10 × 1.0 mm	SOT376-1
P89LV51RD2BN	DIP40	plastic dual in-line package; 40 leads (600 mil)	SOT129-1

#### 3.1 Ordering options

Table 2: Ordering options

Type number	Flash memory	Temperature range	Frequency
P89LV51RB2BA	16 kB	0 °C to +70 °C	0 MHz to 40 MHz
P89LV51RB2BBC	16 kB	0 °C to +70 °C	
P89LV51RC2FA	32 kB	−40 °C to +85 °C	
P89LV51RC2FBC	32 kB	−40 °C to +85 °C	
P89LV51RC2BN	32 kB	0 °C to +70 °C	
P89LV51RD2BA	64 kB	0 °C to +70 °C	
P89LV51RD2FA	64 kB	−40 °C to +85 °C	
P89LV51RD2BBC	64 kB	0 °C to +70 °C	
P89LV51RD2BN	64 kB	0 °C to +70 °C	

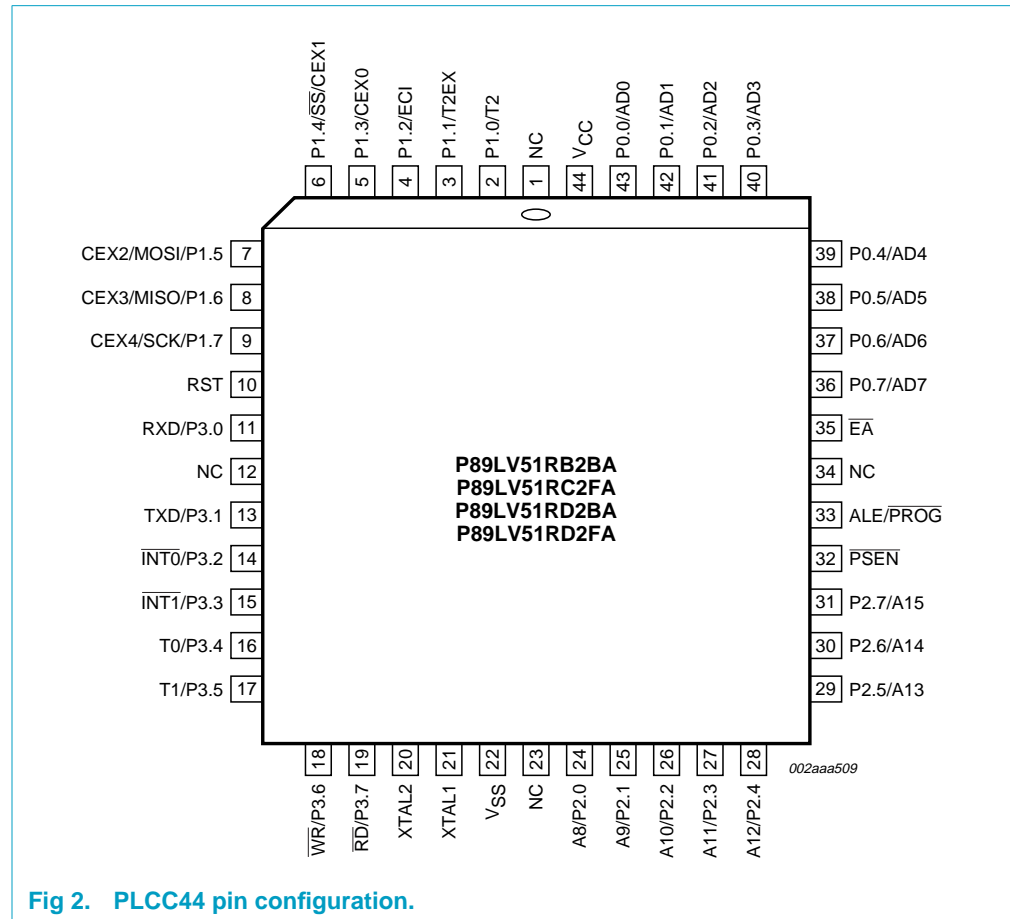
**4. Block diagram**



**Fig 1. P89LV51RB2/RC2/RD2 block diagram.**

## 5. Pinning information

### 5.1 Pinning



**Fig 2. PLCC44 pin configuration.**

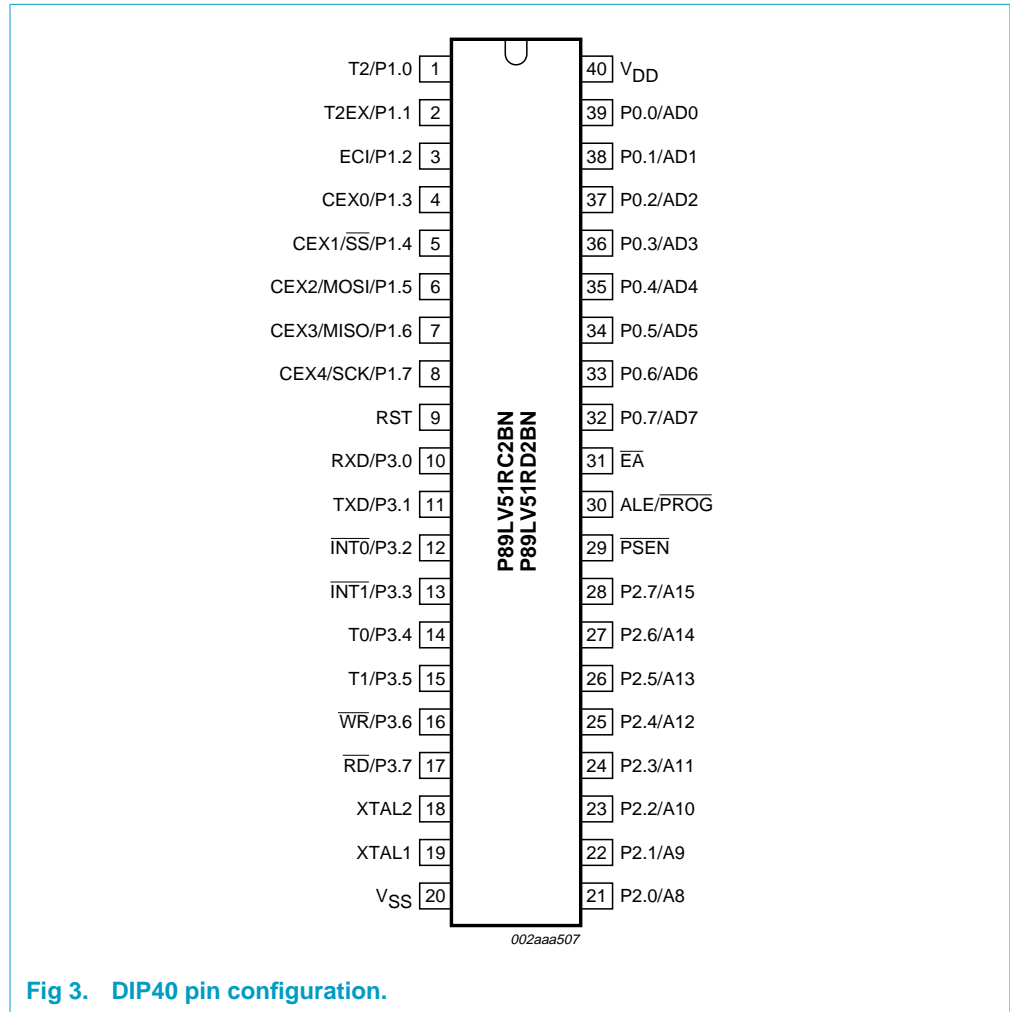


Fig 3. DIP40 pin configuration.

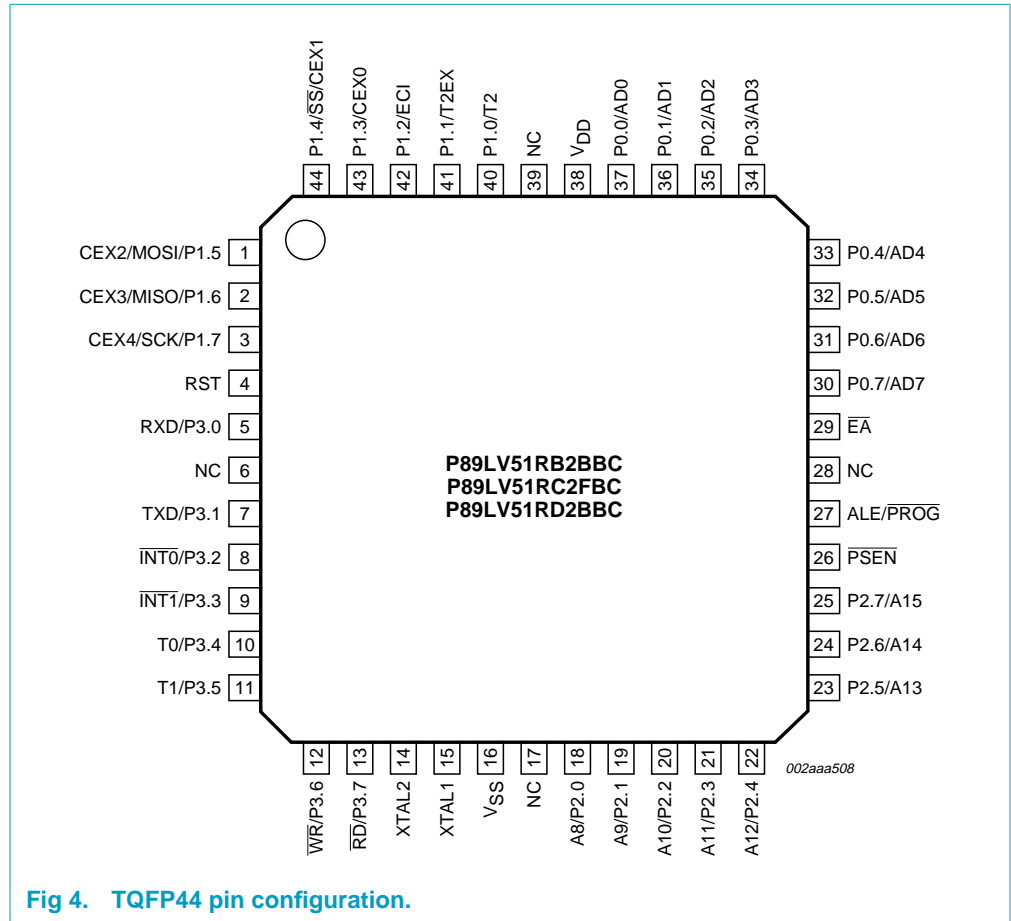


Fig 4. TQFP44 pin configuration.

## 5.2 Pin description

Table 3: P89LV51RB2/RC2/RD2 pin description

Symbol	Pin			Type	Description
	DIP40	TQFP44	PLCC44		
P0.0 to P0.7	39-32	37-30	43-36	I/O	<b>Port 0:</b> Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have '1's written to them float, and in this state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external code and data memory. In this application, it uses strong internal pull-ups when transitioning to '1's. Port 0 also receives the code bytes during the external host mode programming, and outputs the code bytes during the external host mode verification. External pull-ups are required during program verification or as a general purpose I/O port.
P1.0 to P1.7	1-8	40-44, 1-3	2-9	I/O with internal pull-up	<b>Port 1:</b> Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 1 pins are pulled high by the internal pull-ups when '1's are written to them and can be used as inputs in this state. As inputs, Port 1 pins that are externally pulled LOW will source current ( $I_{IL}$ ) because of the internal pull-ups. P1.5, P1.6, P1.7 have high current drive of 16 mA. Port 1 also receives the low-order address bytes during the external host mode programming and verification.
P1.0	1	40	2	I/O	<b>T2:</b> External count input to Timer/Counter 2 or Clock-out from Timer/Counter 2
P1.1	2	41	3	I	<b>T2EX:</b> Timer/Counter 2 capture/reload trigger and direction control
P1.2	3	42	4	I	<b>ECI:</b> External clock input. This signal is the external clock input for the PCA.
P1.3	4	43	5	I/O	<b>CEX0:</b> Capture/compare external I/O for PCA Module 0. Each capture/compare module connects to a Port 1 pin for external I/O. When not used by the PCA, this pin can handle standard I/O.
P1.4	5	44	6	I/O	<b>SS:</b> Slave port select input for SPI <b>CEX1:</b> Capture/compare external I/O for PCA Module 1
P1.5	6	1	7	I/O	<b>MOSI:</b> Master Output Slave Input for SPI <b>CEX2:</b> Capture/compare external I/O for PCA Module 2
P1.6	7	2	8	I/O	<b>MISO:</b> Master Input Slave Output for SPI <b>CEX3:</b> Capture/compare external I/O for PCA Module 3
P1.7	8	3	9	I/O	<b>SCK:</b> Master Output Slave Input for SPI <b>CEX4:</b> Capture/compare external I/O for PCA Module 4

Table 3: P89LV51RB2/RC2/RD2 pin description...continued

Symbol	Pin			Type	Description
	DIP40	TQFP44	PLCC44		
P2.0 to P2.7	21-28	18-25	24-31	I/O with internal pull-up	<b>Port 2:</b> Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. Port 2 pins are pulled HIGH by the internal pull-ups when '1's are written to them and can be used as inputs in this state. As inputs, Port 2 pins that are externally pulled LOW will source current ( $I_{IL}$ ) because of the internal pull-ups. Port 2 sends the high-order address byte during fetches from external program memory and during accesses to external Data Memory that use 16-bit address (MOVX@DPTR). In this application, it uses strong internal pull-ups when transitioning to '1's. Port 2 also receives some control signals and a partial of high-order address bits during the external host mode programming and verification.
P3.0 to P3.7	10-17	5, 7-13	11, 13-19	I/O with internal pull-up	<b>Port 3:</b> Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins are pulled HIGH by the internal pull-ups when '1's are written to them and can be used as inputs in this state. As inputs, Port 3 pins that are externally pulled LOW will source current ( $I_{IL}$ ) because of the internal pull-ups. Port 3 also receives some control signals and a partial of high-order address bits during the external host mode programming and verification.
P3.0	10	5	11	I	<b>RXD:</b> serial input port
P3.1	11	7	13	O	<b>TXD:</b> serial output port
P3.2	12	8	14	I	<b>INT0:</b> external interrupt 0 input
P3.3	13	9	15	I	<b>INT1:</b> external interrupt 1 input
P3.4	14	10	16	I	<b>T0:</b> external count input to Timer/Counter 0
P3.5	15	11	17	I	<b>T1:</b> external count input to Timer/Counter 1
P3.6	16	12	18	O	<b>WR:</b> external data memory write strobe
P3.7	17	13	19	O	<b>RD:</b> external data memory read strobe
PSEN	29	26	32	I/O	<b>Program Store Enable:</b> PSEN is the read strobe for external program memory. When the device is executing from internal program memory, PSEN is inactive (HIGH). When the device is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory. A forced HIGH-to-LOW input transition on the PSEN pin while the RST input is continually held HIGH for more than 10 machine cycles will cause the device to enter external host mode programming.
RST	9	4	10	I	<b>Reset:</b> While the oscillator is running, a HIGH logic state on this pin for two machine cycles will reset the device. If the PSEN pin is driven by a HIGH-to-LOW input transition while the RST input pin is held HIGH, the device will enter the external host mode, otherwise the device will enter the normal operation mode.



Table 3: P89LV51RB2/RC2/RD2 pin description...continued

Symbol	Pin			Type	Description
	DIP40	TQFP44	PLCC44		
$\overline{EA}$	31	29	35	I	<b>External Access Enable:</b> $\overline{EA}$ must be connected to $V_{SS}$ in order to enable the device to fetch code from the external program memory. $\overline{EA}$ must be strapped to $V_{DD}$ for internal program execution. However, Security lock level 4 will disable $\overline{EA}$ , and program execution is only possible from internal program memory. The $\overline{EA}$ pin can tolerate a high voltage of 12 V.
ALE/ PROG	30	27	33	I/O	<b>Address Latch Enable:</b> ALE is the output signal for latching the low byte of the address during an access to external memory. This pin is also the programming pulse input ( $\overline{PROG}$ ) for flash programming. Normally the ALE <sup>[1]</sup> is emitted at a constant rate of $\frac{1}{6}$ the crystal frequency <sup>[2]</sup> and can be used for external timing and clocking. One ALE pulse is skipped during each access to external data memory. However, if AO is set to '1', ALE is disabled.
NC	-	6, 17, 28, 39	1, 12, 23, 34	I/O	<b>No Connect</b>
XTAL1	19	15	21	I	<b>Crystal 1:</b> Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	14	20	O	<b>Crystal 2:</b> Output from the inverting oscillator amplifier.
$V_{DD}$	40	38	44	I	<b>Power supply</b>
$V_{SS}$	20	16	22	I	<b>Ground</b>

[1] ALE loading issue: When ALE pin experiences higher loading (>30 pF) during the reset, the microcontroller may accidentally enter into modes other than normal working mode. The solution is to add a pull-up resistor of 3 k $\Omega$  to 50 k $\Omega$  to  $V_{DD}$ , e.g., for ALE pin.

[2] For 6-clock mode, ALE is emitted at  $\frac{1}{3}$  of crystal frequency.

## 6. Special function registers

---

**Remark:** Special Function Registers (SFRs) accesses are restricted in the following ways:

- User must **not** attempt to access any SFR locations not defined.
- Accesses to any defined SFR locations must be strictly for the functions for the SFRs.
- SFR bits labeled '-', '0' or '1' can **only** be written and read as follows:
  - '-' Unless otherwise specified, **must** be written with '0', but can return any value when read (even if it was written with '0'). It is a reserved bit and may be used in future derivatives.
  - '0' **must** be written with '0', and will return a '0' when read.
  - '1' **must** be written with '1', and will return a '1' when read.

**Table 4: Special function registers**  
 \* indicates SFRs that are bit addressable.

Name	Description	SFR addr.	Bit functions and addresses							
			MSB				LSB			
		<b>Bit address</b>	<b>E7</b>	<b>E6</b>	<b>E5</b>	<b>E4</b>	<b>E3</b>	<b>E2</b>	<b>E1</b>	<b>E0</b>
ACC*	Accumulator	E0H								
AUXR	Auxiliary function register	8EH	-	-	-	-	-	-	EXTRAM	AO
AUXR1	Auxiliary function register 1	A2H	-	-	-		GF2	0	-	DPS
		<b>Bit address</b>	<b>F7</b>	<b>F6</b>	<b>F5</b>	<b>F4</b>	<b>F3</b>	<b>F2</b>	<b>F1</b>	<b>F0</b>
B*	B register	F0H								
CCAP0H	Module 0 Capture HIGH	FAH								
CCAP1H	Module 1 Capture HIGH	FBH								
CCAP2H	Module 2 Capture HIGH	FCH								
CCAP3H	Module 3 Capture HIGH	FDH								
CCAP4H	Module 4 Capture HIGH	FEH								
CCAP0L	Module 0 Capture LOW	EAH								
CCAP1L	Module 1 Capture LOW	EBH								
CCAP2L	Module 2 Capture LOW	ECH								
CCAP3L	Module 3 Capture LOW	EDH								
CCAP4L	Module 4 Capture LOW	EEH								
CCAPM0	Module 0 Mode	DAH	-	ECOM_0	CAPP_0	CAPN_0	MAT_0	TOG_0	PWM_0	ECCF_0
CCAPM1	Module 1 Mode	DBH	-	ECOM_1	CAPP_1	CAPN_1	MAT_1	TOG_1	PWM_1	ECCF_1
CCAPM2	Module 2 Mode	DCH	-	ECOM_2	CAPP_2	CAPN_2	MAT_2	TOG_2	PWM_2	ECCF_2
CCAPM3	Module 3 Mode	DDH	-	ECOM_3	CAPP_3	CAPN_3	MAT_3	TOG_3	PWM_3	ECCF_3
CCAPM4	Module 4 Mode	DEH	-	ECOM_4	CAPP_4	CAPN_4	MAT_4	TOG_4	PWM_4	ECCF_4
		<b>Bit address</b>	<b>DF</b>	<b>DE</b>	<b>DD</b>	<b>DC</b>	<b>DB</b>	<b>DA</b>	<b>D9</b>	<b>D8</b>
CCON*	PCA Counter Control	D8H	CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0
CH	PCA Counter HIGH	F9H								
CL	PCA Counter LOW	E9H								
CMOD	PCA Counter Mode	D9H	CIDL	WDTE	-	-	-	CPS1	CPS0	ECF
DPTR	Data Pointer (2 bytes)									
DPH	Data Pointer HIGH	83H								
DPL	Data Pointer LOW	82H								

**Table 4: Special function registers...continued**

\* indicates SFRs that are bit addressable.

Name	Description	SFR addr.	Bit functions and addresses							
			MSB				LSB			
FST	Flash Status Register	B6	-	SB	-	-	EDC	-	-	-
		<b>Bit address</b>	<b>AF</b>	<b>AE</b>	<b>AD</b>	<b>AC</b>	<b>AB</b>	<b>AA</b>	<b>A9</b>	<b>A8</b>
IEN0*	Interrupt Enable 0	A8H	EA	EC	ET2	ES0	ET1	EX1	ET0	EX0
		<b>Bit address</b>	<b>EF</b>	<b>EE</b>	<b>ED</b>	<b>EC</b>	<b>EB</b>	<b>EA</b>	<b>E9</b>	<b>E8</b>
IEN1*	Interrupt Enable 1	E8H	-	-	-	-	EBO			
		<b>Bit address</b>	<b>BF</b>	<b>BE</b>	<b>BD</b>	<b>BC</b>	<b>BB</b>	<b>BA</b>	<b>B9</b>	<b>B8</b>
IPO*	Interrupt Priority	B8H	-	PPC	PT2	PS	PT1	PX1	PT0	PX0
IPOH	Interrupt Priority 0 HIGH	B7H	-	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
		<b>Bit address</b>	<b>FF</b>	<b>FE</b>	<b>FD</b>	<b>FC</b>	<b>FB</b>	<b>FA</b>	<b>F9</b>	<b>F8</b>
IP1*	Interrupt Priority 1	F8H	-	-	-	-	PBO			
IP1H	Interrupt Priority 1 HIGH	F7H	-	-	-	-	PBOH			
FCF		B1H	-	-	-	-	-	-	SWR	BSEL
		<b>Bit address</b>	<b>87</b>	<b>86</b>	<b>85</b>	<b>84</b>	<b>83</b>	<b>82</b>	<b>81</b>	<b>80</b>
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
		<b>Bit address</b>	<b>97</b>	<b>96</b>	<b>95</b>	<b>94</b>	<b>93</b>	<b>92</b>	<b>91</b>	<b>90</b>
P1*	Port 1	90H	CEX4/ SPICLK	CEX3/ MISO	CEX2/ MOSI	CEX1/SS	CEX0	ECI	T2EX	T2
		<b>Bit address</b>	<b>A7</b>	<b>A6</b>	<b>A5</b>	<b>A4</b>	<b>A3</b>	<b>A2</b>	<b>A1</b>	<b>A0</b>
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8
		<b>Bit address</b>	<b>B7</b>	<b>B6</b>	<b>B5</b>	<b>B4</b>	<b>B3</b>	<b>B2</b>	<b>B1</b>	<b>B0</b>
P3*	Port 3	B0H	RD	WR	T1	T0	INT1	INT0	TxD	RxD
PCON	Power Control Register	87H	SMOD1	SMOD0	BOF	POF	GF1	GF0	PD	IDL
		<b>Bit address</b>	<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>
PSW*	Program Status Word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P
RCAP2H	Timer2 Capture HIGH	CBH								
RCAP2L	Timer2 Capture LOW	CAH								
		<b>Bit address</b>	<b>9F</b>	<b>9E</b>	<b>9D</b>	<b>9C</b>	<b>9B</b>	<b>9A</b>	<b>99</b>	<b>98</b>
SCON*	Serial Port Control	98H	SM0/FE_	SM1	SM2	REN	TB8	RB8	TI	RI
SBUF	Serial Port Data Buffer Register	99H								

**Table 4: Special function registers...continued**

\* indicates SFRs that are bit addressable.

Name	Description	SFR addr.	Bit functions and addresses							
			MSB				LSB			
SADDR	Serial Port Address Register	A9H								
SADEN	Serial Port Address Enable	B9H								
		<b>Bit address</b>	<b>87<sup>[1]</sup></b>	<b>86<sup>[1]</sup></b>	<b>85<sup>[1]</sup></b>	<b>84<sup>[1]</sup></b>	<b>83<sup>[1]</sup></b>	<b>82<sup>[1]</sup></b>	<b>81<sup>[1]</sup></b>	<b>80<sup>[1]</sup></b>
SPCTL	SPI Control Register	D5H	SPIE	SPEN	DORD	MSTR	CPOL	CPHA	PSC1	PSC0
SPCFG	SPI Configuration Register	AAH	SPIF	SPWCOL	-	-	-	-	-	-
SPDAT	SPI Data	86H								
SP	Stack Pointer	81H								
		<b>Bit address</b>	<b>8F</b>	<b>8E</b>	<b>8D</b>	<b>8C</b>	<b>8B</b>	<b>8A</b>	<b>89</b>	<b>88</b>
TCON*	Timer Control Register	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
		<b>Bit address</b>	<b>CF</b>	<b>CE</b>	<b>CD</b>	<b>CC</b>	<b>CB</b>	<b>CA</b>	<b>C9</b>	<b>C8</b>
T2CON*	Timer2 Control Register	C8H	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
T2MOD	Timer2 Mode Control	C9H	-	-	ENT2				T2OE	DCEN
TH0	Timer 0 HIGH	8CH								
TH1	Timer 1 HIGH	8DH								
TH2	Timer 2 HIGH	CDH								
TL0	Timer 0 LOW	8AH								
TL1	Timer 1 LOW	8BH								
TL2	Timer 2 LOW	CCH								
TMOD	Timer 0 and 1 Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0
WDTC	Watchdog Timer Control	C0H	-	-	-	WDOUT	WDRE	WDTS	WDT	SWDT
WDTD	Watchdog Timer Data/Reload	85H								

[1] Unimplemented bits in SFRs (labeled '-') are 'X's (unknown) at all times. Unless otherwise specified, '1's should not be written to these bits since they may be used for other purposes in future derivatives. The reset values shown for these bits are '0's although they are unknown when read.

## 7. Functional description

### 7.1 Memory organization

The device has separate address spaces for program and data memory.

#### 7.1.1 Flash program memory bank selection

There are two internal flash memory blocks in the device. Block 0 has 16/32/64 kB and is organized as 128/256/512 sectors, each sector consists of 128 Bytes. Block 1 contains the IAP/ISP routines and may be enabled such that it overlays the first 8 kB of the user code memory. The overlay function is controlled by the combination of the Software Reset Bit (SWR) at FCF.1 and the Bank Select Bit (BSEL) at FCF.0. The combination of these bits and the memory source used for instructions is shown in [Table 5](#).

**Table 5:** Code memory bank selection

SWR (FCF.1)	BSEL (FCF.0)	addresses from 0000h to 1FFFh	addresses above 1FFFh
0	0	Bootcode (in Block 1)	User code (in Block 0)
0	1	User code (in Block 0)	
1	0		
1	1		

Access to the IAP routines in Block 1 may be enabled by clearing the BSEL bit (FCF.0), provided that the SWR bit (FCF.1) is cleared. Following a power-on sequence, the bootcode is automatically executed and attempts to autobaud to a host. If no autobaud occurs within approximately 400 ms and the SoftICE flag is not set, control will be passed to the user code. A software reset is used to accomplish this control transfer and as a result the SWR bit will remain set. **Therefore the user's code will need to clear the SWR bit in order to access the IAP routines in Block 1.** However, caution must be taken when dynamically changing the BSEL bit. Since this will cause different physical memory to be mapped to the logical program address space, the user must avoid clearing the BSEL bit when executing user code within the address range 0000H to 1FFFH.

#### 7.1.2 Power-on reset code execution

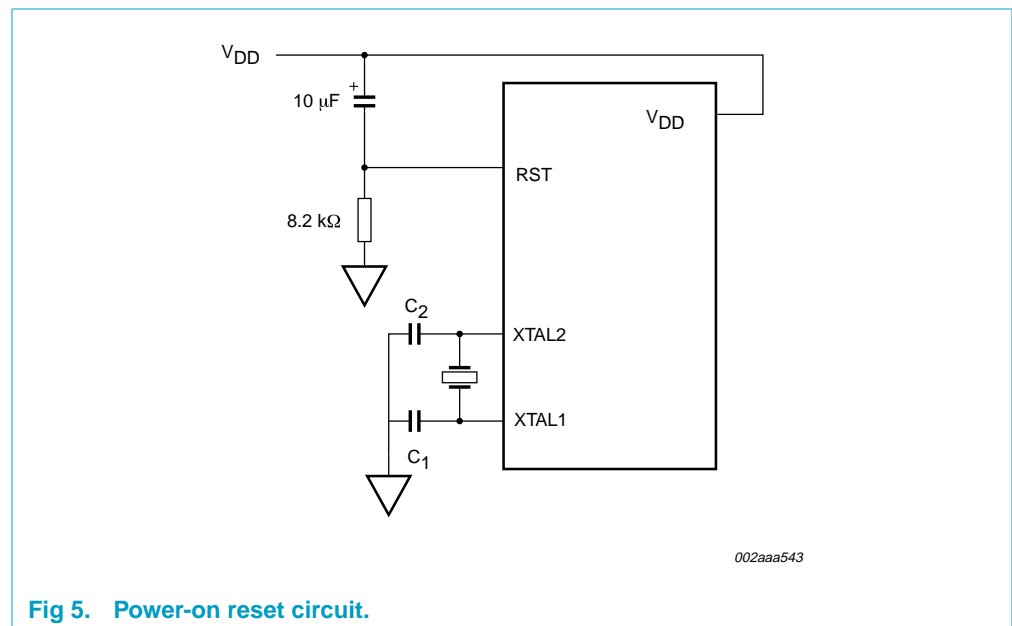
At initial power up, the port pins will be in a random state until the oscillator has started and the internal reset algorithm has weakly pulled all pins high. Powering up the device without a valid reset could cause the MCU to start executing instructions from an indeterminate location. Such undefined states may inadvertently corrupt the code in the flash. A system reset will not affect the 1 kB of on-chip RAM while the device is running, however, the contents of the on-chip RAM during power up are indeterminate.

When power is applied to the device, the RST pin must be held high long enough for the oscillator to start up (usually several milliseconds for a low frequency crystal), in addition to two machine cycles for a valid power-on reset. An example of a method to extend the RST signal is to implement a RC circuit by connecting the RST pin to  $V_{DD}$  through a 10 F capacitor and to VSS through an 8.2KW resistor as shown in

**Figure 5.** Note that if an RC circuit is being used, provisions should be made to ensure the  $V_{DD}$  rise time does not exceed 1 millisecond and the oscillator start-up time does not exceed 10 milliseconds.

For a low frequency oscillator with slow start-up time the reset signal must be extended in order to account for the slow start-up time. This method maintains the necessary relationship between  $V_{DD}$  and RST to avoid programming at an indeterminate location, which may cause corruption in the code of the flash. The power-on detection is designed to work during initial power up, before the voltage reaches the brown-out detection level. The POF flag in the PCON register is set to indicate an initial power up condition. The POF flag will remain active until cleared by software.

Following a power-on or external reset the P89LV51RB2/RC2/RD2 will force the SWR and BSEL bits ( $FCF[1:0] = 00$ ). This causes the bootblock to be mapped into the lower 8 kB of code memory and the device will execute the ISP code in the boot block and attempt to autobaud to the host. If the autobaud is successful the device will remain in ISP mode. If, after approximately 400 ms, the autobaud is unsuccessful the boot block code will check to see if the SoftICE flag is set (from a previous programming operation). If the SoftICE flag is set the device will enter SoftICE mode. If the SoftICE flag is cleared, the bootcode will execute a software reset causing the device to execute the user code from block 0 starting at address 0000h. Note that an external reset applied to the RST pin has the same effect as a power-on reset.



**Fig 5. Power-on reset circuit.**

### 7.1.3 Software reset

A software reset is executed by changing the SWR bit ( $FCF.1$ ) from '0' to '1'. A software reset will reset the program counter to address 0000H and force both the SWR and BSEL bits ( $FCF[1:0] = 10$ ). This will result in the lower 8 kB of the user code memory being mapped into the user code memory space. Thus the user's code will be executed starting at address 0000h. A software reset will not change WDTC.2 or RAM data. Other SFRs will be set to their reset values.

#### 7.1.4 Brown-out detect reset

The device includes a brown-out detection circuit to protect the system from severe supply voltage fluctuations. The P89LV51RB2/RC2/RD2's brown-out detection threshold is 2.35 V. When  $V_{DD}$  drops below this voltage threshold, the brown-out detect triggers the circuit to generate a brown-out interrupt but the CPU still runs until the supplied voltage returns to the brown-out detection voltage  $V_{BOD}$ . The default operation for a brown-out detection is to cause a processor reset.

$V_{DD}$  must stay below  $V_{BOD}$  at least four oscillator clock periods before the brown-out detection circuit will respond.

Brown-out interrupt can be enabled by setting the EBO bit (IEA.3). If EBO bit is set and a brown-out condition occurs, a brown-out interrupt will be generated to execute the program at location 004BH. It is required that the EBO bit be cleared by software after the brown-out interrupt is serviced. Clearing EBO bit when the brown-out condition is active will properly reset the device. If brown-out interrupt is not enabled, a brown-out condition will reset the program to resume execution at location 0000H. A brown-out detect reset will clear the BSEL bit (FCF.0) but will not change the SWR bit (FCF.1) and therefore will not change the banking of the lower 8 kB of user code memory space.

#### 7.1.5 Watchdog reset

Like a brown-out detect reset, the watchdog timer reset will clear the BSEL bit (FCF.0) but will not change the SWR bit (FCF.1) and therefore will not change the banking of the lower 8 kB of user code memory space.

The state of the SWR and BSEL bits after different types of resets is shown in [Table 6](#). This results in the code memory bank selections as shown.

**Table 6: Effects of reset sources on bank selection**

Reset source	SWR bit result (FCF.1)	BSEL bit result (FCF.0)	addresses from 0000h to 1FFFh	addresses above 1FFFh
External reset	0	0	Bootcode (in Block 1)	User code (in Block 0)
Power-on reset				
Watchdog reset	x	0	Retains state of SWR bit. If SWR, BSEL =00 then uses Bootcode. If SWR, BSEL = 10 then uses user code.	
Brown-out detect reset				
Software reset	1	0	User code (in Block 0)	

#### 7.1.6 Data RAM memory

The data RAM has 1024 bytes of internal memory. The device can also address up to 64 kB for external data memory.

#### 7.1.7 Expanded data RAM addressing

The P89LV51RB2/RC2/RD2 has 1 kB of RAM. See [Figure 6 "Internal and external data memory structure."](#) on page 19.

The device has four sections of internal data memory:

1. The lower 128 bytes of RAM (00H to 7FH) are directly and indirectly addressable.



2. The higher 128 bytes of RAM (80H to FFH) are indirectly addressable.
3. The special function registers (80H to FFH) are directly addressable only.
4. The expanded RAM of 768 bytes (00H to 2FFH) is indirectly addressable by the move external instruction (MOVX) and clearing the EXTRAM bit. (See 'Auxiliary Register (AUXR) in [Section 6 "Special function registers" on page 10](#))

Since the upper 128 bytes occupy the same addresses as the SFRs, the RAM must be accessed indirectly. The RAM and SFRs space are physically separate even though they have the same addresses.

**Table 7: AUXR - Auxiliary register (address 8EH) bit allocation**

Not bit addressable; Reset value 00H

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	EXTRAM	AO

**Table 8: AUXR - Auxiliary register (address 8EH) bit description**

Bit	Symbol	Description
7 to 2	-	Reserved for future use. Should be set to '0' by user programs.
1	EXTRAM	Internal/External RAM access using MOVX @Ri/@DPTR. When '0', core attempts to access internal XRAM with address specified in MOVX instruction. If address supplied with this instruction exceeds on-chip available XRAM, off-chip XRAM is going to be selected and accessed. When '1', every MOVX @Ri/@DPTR instruction targets external data memory by default.
0	AO	ALE off: disables/enables ALE. AO = 0 results in ALE emitted at a constant rate of $\frac{1}{2}$ the oscillator frequency. In case of AO = 1, ALE is active only during a MOVX or MOVC.

When instructions access addresses in the upper 128 bytes (above 7FH), the MCU determines whether to access the SFRs or RAM by the type of instruction given. If it is indirect, then RAM is accessed. If it is direct, then an SFR is accessed. See the examples below.

Indirect Access:

```
MOV @R0, #data; R0 contains 90H
```

Register R0 points to 90H which is located in the upper address range. Data in '#data' is written to RAM location 90H rather than port 1.

Direct Access:

```
MOV 90H, #data; write data to P1
```

Data in '#data' is written to port 1. Instructions that write directly to the address write to the SFRs.

To access the expanded RAM, the EXTRAM bit must be cleared and MOVX instructions must be used. The extra 768 bytes of memory is physically located on the chip and logically occupies the first 768 bytes of external memory (addresses 000H to 2FFH).

When EXTRAM = 0, the expanded RAM is indirectly addressed using the MOVX instruction in combination with any of the registers R0, R1 of the selected bank or DPTR. Accessing the expanded RAM does not affect ports P0, P3.6 ( $\overline{WR}$ ), P3.7 ( $\overline{RD}$ ), or P2. With EXTRAM = 0, the expanded RAM can be accessed as in the following example.

Expanded RAM Access (Indirect Addressing only):

```
MOVX@DPTR, A DPTR contains 0A0H
```

DPTR points to 0A0H and data in 'A' is written to address 0A0H of the expanded RAM rather than external memory. Access to external memory higher than 2FFH using the MOVX instruction will access external memory (0300H to FFFFH) and will perform in the same way as the standard 8051, with P0 and P2 as data/address bus, and P3.6 and P3.7 as write and read timing signals.

When EXTRAM = 1, MOVX @Ri and MOVX @DPTR will be similar to the standard 8051. Using MOVX @Ri provides an 8-bit address with multiplexed data on Port 0. Other output port pins can be used to output higher order address bits. This provides external paging capabilities. Using MOVX @DPTR generates a 16-bit address. This allows external addressing up to the 64 kB. Port 2 provides the high-order eight address bits (DPH), and Port 0 multiplexes the low order eight address bits (DPL) with data. Both MOVX @Ri and MOVX @DPTR generates the necessary read and write signals (P3.6 -  $\overline{WR}$  and P3.7 -  $\overline{RD}$ ) for external memory use. Table 9 shows external data memory  $\overline{RD}$ ,  $\overline{WR}$  operation with EXTRAM bit.

The stack pointer (SP) can be located anywhere within the 256 bytes of internal RAM (lower 128 bytes and upper 128 bytes). The stack pointer may not be located in any part of the expanded RAM.

**Table 9: External data memory  $\overline{RD}$ ,  $\overline{WR}$  with EXTRAM bit**

AUXR	MOVX @DPTR, A or MOVX A, @DPTR		MOVX @Ri, A or MOVX A, @Ri
	ADDR < 0300H	ADDR ≥ 0300H	ADDR = any
EXTRAM = 0	$\overline{RD}/\overline{WR}$ not asserted	$\overline{RD}/\overline{WR}$ asserted	$\overline{RD}/\overline{WR}$ not asserted
EXTRAM = 1	$\overline{RD}/\overline{WR}$ asserted	$\overline{RD}/\overline{WR}$ asserted	$\overline{RD}/\overline{WR}$ asserted

[1] Access limited to ERAM address within 0 to 0FFH; cannot access 100H to 02FFH.

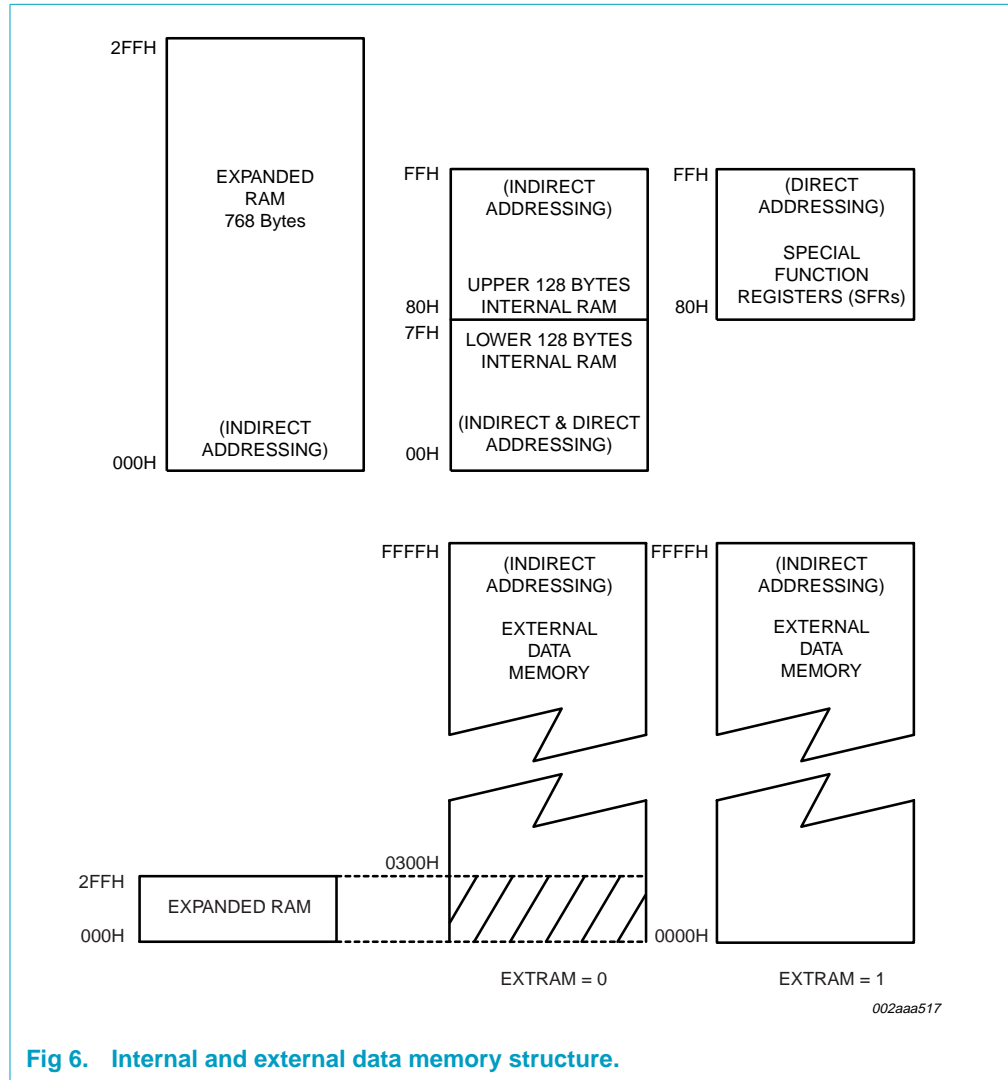


Fig 6. Internal and external data memory structure.

### 7.1.8 Dual data pointers

The device has two 16-bit data pointers. The DPTR Select (DPS) bit in AUXR1 determines which of the two data pointers is accessed. When DPS = 0, DPTR0 is selected; when DPS = 1, DPTR1 is selected. Quickly switching between the two data pointers can be accomplished by a single INC instruction on AUXR1 (see Figure 7).

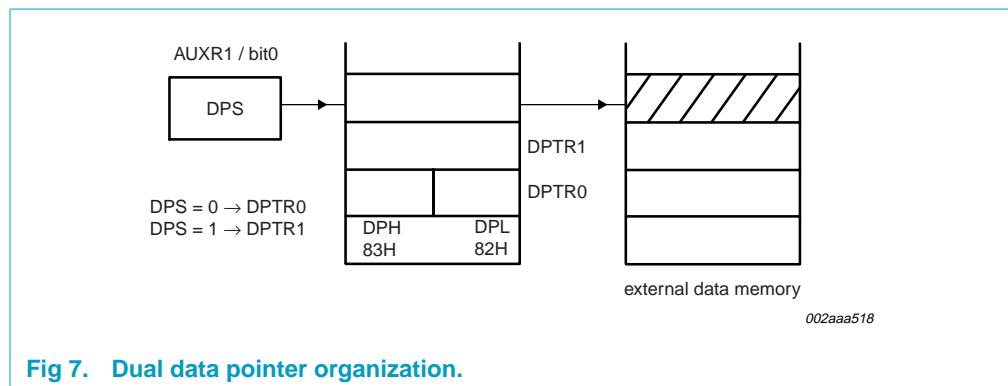


Fig 7. Dual data pointer organization.

**Table 10: AUXR1 - Auxiliary register 1 (address A2H) bit allocation***Not bit addressable; Reset value 00H*

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	GF2	0	-	DPS

**Table 11: AUXR1 - Auxiliary register 1 (address A2H) bit description**

Bit	Symbol	Description
7 to 4	-	Reserved for future use. Should be set to '0' by user programs.
3	GF2	General purpose user-defined flag.
2	0	This bit contains a hard-wired '0'. Allows toggling of the DPS bit by incrementing AUXR1, without interfering with other bits in the register.
1	-	Reserved for future use. Should be set to '0' by user programs.
0	DPS	Data pointer select. Chooses one of two Data Pointers for use by the program. See text for details.

## 7.2 Flash memory In-Application Programming

### 7.2.1 Flash organization

The P89LV51RB2/RC2/RD2 program memory consists of a 16/32/64 kB block. An In-System Programming (ISP) capability, in a second 8 kB block, is provided to allow the user code to be programmed in-circuit through the serial port. 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 (IAP). 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 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 which supports this device.

### 7.2.2 Boot block (Block 1)

When the microcontroller programs its own Flash memory, all of the low level details are handled by code that is contained in Block 1. A user program calls the common entry point in the Block 1 with appropriate parameters to accomplish the desired operation. Boot block operations include erase user code, program user code, program security bits, etc.

A Chip-Erase operation can be performed using a commercially available parallel programmer. This operation will erase the contents of this Boot Block and it will be necessary for the user to reprogram this Boot Block (Block 1) with the Philips-provided ISP/IAP code in order to use the ISP or IAP capabilities of this device. Contact <http://www.semiconductors.philips.com> to obtain the hex file for this device. Questions may be directed to [micro.support@philips.com](mailto:micro.support@philips.com).

### 7.2.3 In-System Programming (ISP)

In-System Programming is performed without removing the microcontroller from the system. The In-System Programming facility consists of a series of internal hardware resources coupled with internal firmware to facilitate remote programming of the

P89LV51RB2/RC2/RD2 through the serial port. This firmware is provided by Philips and embedded within each P89LV51RB2/RC2/RD2 device. The Philips In-System Programming 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 ( $V_{DD}$ ,  $V_{SS}$ , TxD, RxD, and RST). Only a small connector needs to be available to interface your application to an external circuit in order to use this feature.

#### 7.2.4 Using the In-System Programming

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 P89LV51RB2/RC2/RD2 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 P89LV51RB2/RC2/RD2 will accept up to 32 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 32 (decimal). ISP commands are summarized in [Table 12](#). As a record is received by the P89LV51RB2/RC2/RD2, 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 P89LV51RB2/RC2/RD2 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.

Table 12: In-System Programming (ISP) hex record formats

Record type	Command/data function
00	<p>Program User Code Memory</p> <p>:nnaaaa00dd..ddcc</p> <p>Where:</p> <p>nn = number of bytes to program</p> <p>aaaa = address</p> <p>dd..dd = data bytes</p> <p>cc = checksum</p> <p>Example:</p> <p>:100000000102030405006070809cc</p>
01	<p>End of File (EOF), no operation</p> <p>:xxxxxx01cc</p> <p>Where:</p> <p>xxxxxx = required field but value is a 'don't care'</p> <p>cc = checksum</p> <p>Example:</p> <p>:00000001FF</p>
02	<p>Set SoftICE mode</p> <p>Following the next reset the device will enter the SoftICE mode. Will erase user code memory, erase device serial number.</p> <p>:00000002cc</p> <p>Where:</p> <p>xxxxxx = required field but value is a 'don't care'</p> <p>cc = checksum</p> <p>Example:</p> <p>:00000002FE</p>

Table 12: In-System Programming (ISP) hex record formats...continued

Record type	Command/data function
03	<p>Miscellaneous Write Functions</p> <p>:nnxxxx03ffssddcc</p> <p>Where:</p> <p>nn = number of bytes in the record</p> <p>xxxx = required field but value is a 'don't care'</p> <p>ff = subfunction code</p> <p>ss = selection code</p> <p>dd = data (if needed)</p> <p>cc = checksum</p> <p>Subfunction code = 01 (Erase Block 0)</p> <p>ff = 01</p> <p>Subfunction code = 05 (Program security bit, Double Clock)</p> <p>ff = 05</p> <p>ss = 01 program security bit</p> <p>ss = 05 program double clock bit</p> <p>Subfunction code = 08 (Erase sector, 128 bytes)</p> <p>ff = 08</p> <p>ss = high byte of sector address (A15:8)</p> <p>dd = low byte of sector address (A7, A6:0 = 0)</p> <p>Example:</p> <p>:0300000308E000F2 (erase sector at E000h)</p>
04	<p>Display Device Data or Blank Check</p> <p>:05xxxx04sssseeeeffcc</p> <p>Where</p> <p>05 = number of bytes in the record</p> <p>xxxx = required field but value is a 'don't care'</p> <p>04 = function code for display or blank check</p> <p>ssss = starting address, MSB first</p> <p>eeee = ending address, MSB first</p> <p>ff = subfunction</p> <p>00 = display data</p> <p>01 = blank check</p> <p>cc = checksum</p> <p>Subfunction codes:</p> <p>Example:</p> <p>:0500000400001FFF00D9 (display from 0000h to 1FFFh)</p>

Table 12: In-System Programming (ISP) hex record formats...continued

Record type	Command/data function
05	<p>Miscellaneous Read Functions</p> <p>:02xxxx05ffsscc</p> <p>Where:</p> <p>02 = number of bytes in the record</p> <p>xxxx = required field but value is a 'don't care'</p> <p>05 = function code for misc read</p> <p>ffss = subfunction and selection code</p> <p>0000 = read manufacturer id</p> <p>0001 = read device id 1</p> <p>0002 = read bootcode version</p> <p>0700 = read security bit (00 SoftICE serial number match 0 SB 0 Double Clock)</p> <p>cc = checksum</p> <p>Example:</p> <p>:020000050000F9 (display manufacturer id)</p>
06	<p>Direct Load of Baud Rate</p> <p>:02xxxx06HLLcc</p> <p>Where:</p> <p>02 = number of bytes in the record</p> <p>xxxx = required field but value is a 'don't care'</p> <p>HH = high byte of timer</p> <p>LL = low byte of timer</p> <p>cc = checksum</p> <p>Example:</p> <p>:02000006FFFFcc (load T2 = FFFF)</p>
07	<p>Reset serial number, erase user code, clear SoftICE mode</p> <p>:xxxxxx07cc</p> <p>Where:</p> <p>xxxxxx = required field but value is a 'don't care'</p> <p>07 = reset serial number function</p> <p>cc = checksum</p> <p>Example:</p> <p>:00000007F9</p>
08	<p>Verify serial number</p> <p>:nnxxxx08ss..sscc</p> <p>Where:</p> <p>xxxxxx = required field but value is a 'don't care'</p> <p>08 = verify serial number function</p> <p>ss..ss = serial number contents</p> <p>cc = checksum</p> <p>Example:</p> <p>:03000008010203EF (verify s/n = 010203)</p>



**Table 12: In-System Programming (ISP) hex record formats...continued**

Record type	Command/data function
09	<p>Write serial number</p> <p>:nnxxxx09ss..sscc</p> <p>Where:</p> <p>xxxxxx = required field but value is a 'don't care'</p> <p>09 = write serial number function</p> <p>ss..ss = serial number contents</p> <p>cc = checksum</p> <p>Example:</p> <p>:03000009010203EE (write s/n = 010203)</p>
0A	<p>Display serial number</p> <p>:xxxxxx0Acc</p> <p>Where:</p> <p>xxxxxx = required field but value is a 'don't care'</p> <p>0A = display serial number function</p> <p>cc = checksum</p> <p>Example:</p> <p>:0000000AF6</p>
0B	<p>Reset and run user code</p> <p>:xxxxxx0Bcc</p> <p>Where:</p> <p>xxxxxx = required field but value is a 'don't care'</p> <p>0B = Reset and run user code</p> <p>cc = checksum</p> <p>Example:</p> <p>:0000000BF5</p>

### 7.2.5 Using the serial number

This device has the option of storing a 31-byte serial number along with the length of the serial number (for a total of 32 bytes) in a non-volatile memory space. When ISP mode is entered, the serial number length is evaluated to determine if the serial number is in use. If the length of the serial number is programmed to either 00H or FFH, the serial number is considered not in use. If the serial number is in use, reading, programming, or erasing of the user code memory or the serial number is blocked until the user transmits a 'verify serial number' record containing a serial number and length that matches the serial number and length previously stored in the device. The user can reset the serial number to all zeros and set the length to zero by sending the 'reset serial number' record. In addition, the 'reset serial number' record will also erase all user code.

### 7.2.6 In-Application Programming method

Several In-Application Programming (IAP) calls are available for use by an application program to permit selective erasing, reading and programming of Flash sectors, security bit, configuration bytes, and device id. 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 1FF0H. The IAP calls are shown in [Table 13](#)

**Table 13: IAP function calls**

IAP function	IAP call parameters
Read Id	<p><b>Input parameters:</b></p> <p>R1 = 00h            DPH = 00H            DPL = 00H = mfgr id            DPL = 01H = device id 1            DPL = 02H = bootcode version number</p> <p><b>Return parameter(s):</b></p> <p>ACC = requested parameter</p>
Erase Block 0	<p><b>Input parameters:</b></p> <p>R1 = 01h</p> <p><b>Return parameter(s):</b></p> <p>ACC = 00 = pass            ACC = !00 = fail</p>
Program User Code	<p><b>Input parameters:</b></p> <p>R1 = 02h            DPH = memory address MSB            DPL = memory address LSB            ACC = byte to program</p> <p><b>Return parameter(s):</b></p> <p>ACC = 00 = pass            ACC = !00 = fail</p>
Read User Code	<p><b>Input parameters:</b></p> <p>R1 = 03h            DPH = memory address MSB            DPL = memory address LSB</p> <p><b>Return parameter(s):</b></p> <p>ACC = device data</p>
Program Security Bit, Double Clock	<p><b>Input parameters:</b></p> <p>R1 = 05h            DPL = 01H = security bit            DPL = 05H = Double Clock</p> <p><b>Return parameter(s):</b></p> <p>ACC = 00 = pass            ACC = !00 = fail</p>

Table 13: IAP function calls...continued

IAP function	IAP call parameters
Read Security Bit, Double Clock, SoftICE	<b>Input parameters:</b> ACC = 07h <b>Return parameter(s):</b> ACC = 000 S/N-match 0 SB 0 DBL_CLK
Read Security Bit, Double Clock, SoftICE	<b>Input parameters:</b> ACC = 07h <b>Return parameter(s):</b> ACC = 00 SoftICE S/N-match 0 SB 0 DBL_CLK
Erase sector	<b>Input parameters:</b> R1 = 08h DPH = sector address high byte DPL = sector address low byte <b>Return parameter(s):</b> ACC = 00 = pass ACC = !00 = fail

### 7.3 Timers/counters 0 and 1

The two 16-bit Timer/Counter registers: Timer 0 and Timer 1 can be configured to operate either as timers or event counters (see [Table 14](#) and [Table 15](#)).

In the 'Timer' function, the register is incremented every machine cycle. Thus, one can think of it as counting machine cycles. Since a machine cycle consists of six oscillator periods, the count rate is  $\frac{1}{6}$  of the oscillator frequency.

In the 'Counter' function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0 or T1. In this function, the external input is sampled once every machine cycle.

When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register in the machine cycle following the one in which the transition was detected. Since it takes two machine cycles (12 oscillator periods) for 1-to-0 transition to be recognized, the maximum count rate is  $\frac{1}{12}$  of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full machine cycle. In addition to the 'Timer' or 'Counter' selection, Timer 0 and Timer 1 have four operating modes from which to select.

The 'Timer' or 'Counter' function is selected by control bits C/T in the Special Function Register TMOD. These two Timer/Counters have four operating modes, which are selected by bit-pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both Timers/Counters. Mode 3 is different. The four operating modes are described in the following text.

**Table 14: TMOD - Timer/Counter mode control register (address 89H) bit allocation***Not bit addressable; Reset value: 0000000B; Reset source(s): any source*

Bit	7	6	5	4	3	2	1	0
Symbol	T1GATE	T1C/ $\bar{T}$	T1M1	T1M0	T0GATE	T0C/ $\bar{T}$	T0M1	T0M0

**Table 15: TMOD - Timer/Counter mode control register (address 89H) bit description**

Bit	Symbol	Description
	T1/T0	Bits controlling Timer1/Timer0
	GATE	Gating control when set. Timer/Counter 'x' is enabled only while 'INTx' pin is HIGH and 'TRx' control pin is set. When cleared, Timer 'x' is enabled whenever 'TRx' control bit is set.
	C/ $\bar{T}$	Gating Timer or Counter Selector cleared for Timer operation (input from internal system clock.) Set for Counter operation (input from 'Tx' input pin).

**Table 16: TMOD - Timer/Counter mode control register (address 89H) M1/M0 operating mode**

M1	M0	Operating mode	
0	0	0	8048 timer 'TLx' serves as 5-bit prescaler
0	1	1	16-bit Timer/Counter 'THx' and 'TLx' are cascaded; there is no prescaler.
1	0	2	8-bit auto-reload Timer/Counter 'THx' holds a value which is to be reloaded into 'TLx' each time it overflows.
1	1	3	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

**Table 17: TCON - Timer/Counter control register (address 88H) bit allocation***Bit addressable; Reset value: 0000000B; Reset source(s): any reset*

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

**Table 18: TCON - Timer/Counter control register (address 88H) bit description**

Bit	Symbol	Description
7	TF1	Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when the processor vectors to Timer 1 Interrupt routine, or by software.
6	TR1	Timer 1 Run control bit. Set/cleared by software to turn Timer/Counter 1 on/off.
5	TF0	Timer 0 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when the processor vectors to Timer 0 Interrupt routine, or by software.
4	TR0	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter 0 on/off.

Table 18: TCON - Timer/Counter control register (address 88H) bit description

Bit	Symbol	Description
3	IE1	Interrupt 1 Edge flag. Set by hardware when external interrupt 1 edge/low level is detected. Cleared by hardware when the interrupt is processed, or by software.
2	IT1	Interrupt 1 Type control bit. Set/cleared by software to specify falling edge/low level that triggers external interrupt 1.
1	IE0	Interrupt 0 Edge flag. Set by hardware when external interrupt 0 edge/low level is detected. Cleared by hardware when the interrupt is processed, or by software.
0	IT0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level that triggers external interrupt 0.

### 7.3.1 Mode 0

Putting either Timer into Mode 0 makes it look like an 8048 Timer, which is an 8-bit Counter with a fixed divide-by-32 prescaler. Figure 8 shows Mode 0 operation.

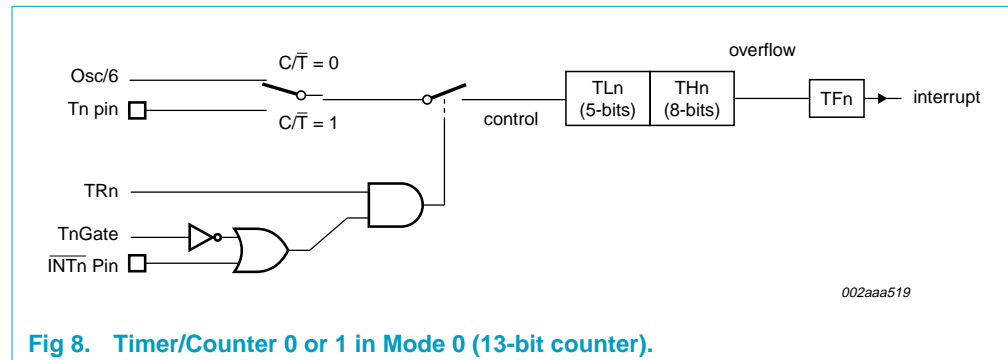


Fig 8. Timer/Counter 0 or 1 in Mode 0 (13-bit counter).

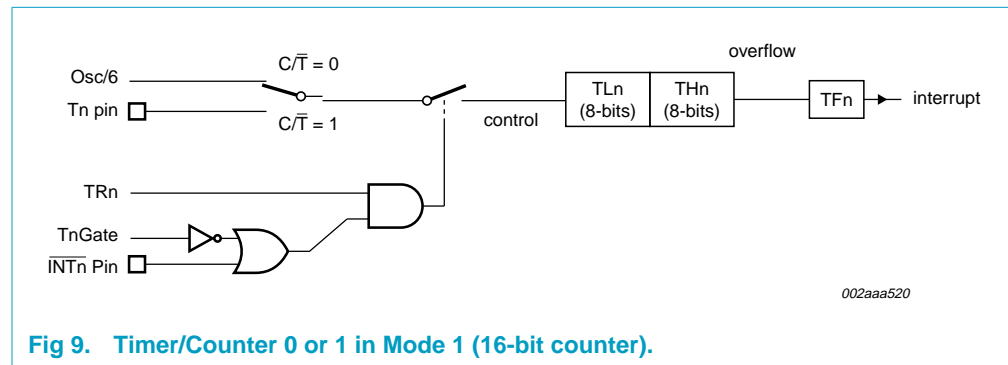
In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the Timer interrupt flag TF<sub>n</sub>. The count input is enabled to the Timer when TR<sub>n</sub> = 1 and either GATE = 0 or  $\overline{\text{INTn}}$  = 1. (Setting GATE = 1 allows the Timer to be controlled by external input  $\overline{\text{INTn}}$ , to facilitate pulse width measurements). TR<sub>n</sub> is a control bit in the Special Function Register TCON (Figure 7). The GATE bit is in the TMOD register.

The 13-bit register consists of all 8 bits of TH<sub>n</sub> and the lower 5 bits of TL<sub>n</sub>. The upper 3 bits of TL<sub>n</sub> are indeterminate and should be ignored. Setting the run flag (TR<sub>n</sub>) does not clear the registers.

Mode 0 operation is the same for Timer 0 and Timer 1 (see Figure 8). There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

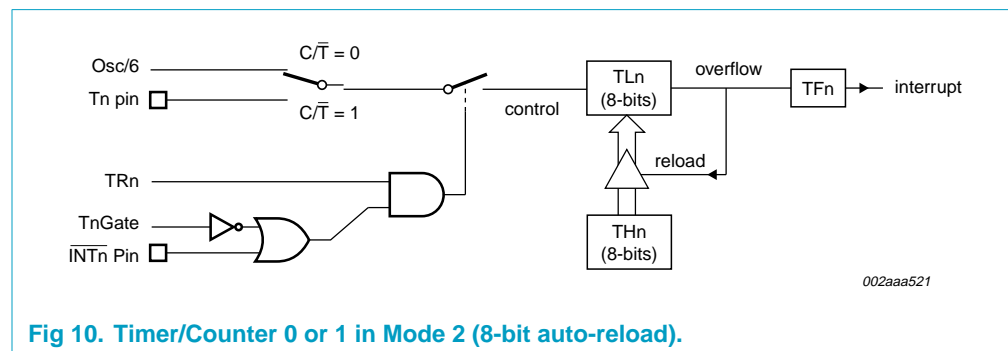
### 7.3.2 Mode 1

Mode 1 is the same as Mode 0, except that all 16 bits of the timer register (TH<sub>n</sub> and TL<sub>n</sub>) are used. See Figure 9.



### 7.3.3 Mode 2

Mode 2 configures the Timer register as an 8-bit Counter (TLn) with automatic reload, as shown in [Figure 10](#). Overflow from TLn not only sets TFn, but also reloads TLn with the contents of THn, which must be preset by software. The reload leaves THn unchanged. Mode 2 operation is the same for Timer 0 and Timer 1.



### 7.3.4 Mode 3

When timer 1 is in Mode 3 it is stopped (holds its count). The effect is the same as setting TR1 = 0.

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate 8-bit counters. The logic for Mode 3 and Timer 0 is shown in [Figure 11](#). TL0 uses the Timer 0 control bits: T0C/T̄, T0GATE, TR0, INT̄0, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the 'Timer 1' interrupt.

Mode 3 is provided for applications that require an extra 8-bit timer. With Timer 0 in Mode 3, the P89LV51RB2/RC2/RD2 can look like it has an additional Timer.

**Note:** When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it into and out of its own Mode 3. It can still be used by the serial port as a baud rate generator, or in any application not requiring an interrupt.

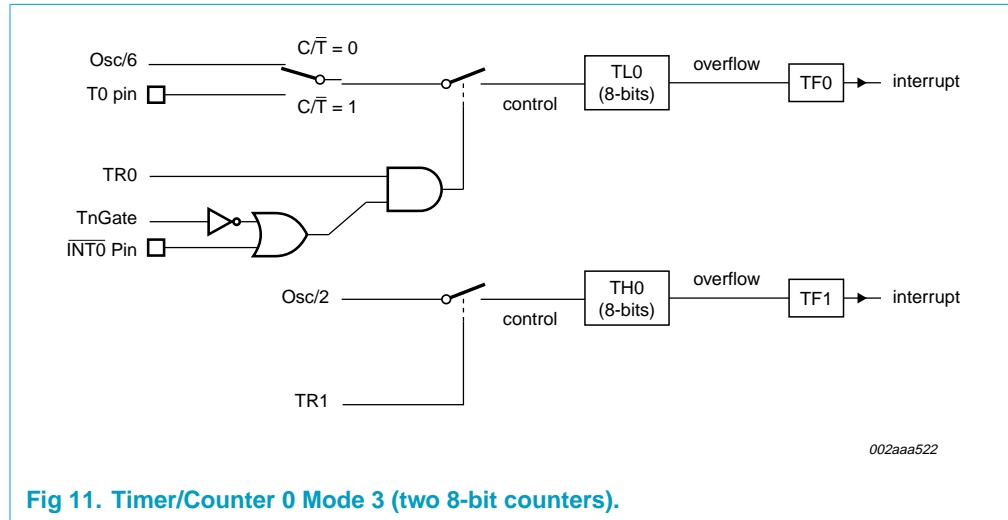


Fig 11. Timer/Counter 0 Mode 3 (two 8-bit counters).

### 7.4 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/T2 in the special function register T2CON. Timer 2 has four operating modes: Capture, Auto-reload (up or down counting), Clock-out, and Baud Rate Generator which are selected according to Table 19 using T2CON (Table 20 and Table 21) and T2MOD (Table 22 and Table 23).

Table 19: Timer 2 operating mode

RCLK+TCLK	CP/RL2	TR2	T2OE	Mode
0	0	1	0	16-BIT auto reload
0	1	1	0	16-bit capture
0	0	1	1	Programmable Clock-Out
1	X	1	0	Baud rate generator
X	X	0	X	off

Table 20: T2CON - Timer/Counter 2 control register (address C8H) bit allocation

Bit addressable; Reset value: 00H

Bit	7	6	5	4	3	2	1	0
Symbol	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

**Table 21: T2CON - Timer/Counter 2 control register (address C8H) bit description**

Bit	Symbol	Description
7	TF2	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 or when Timer 2 is in Clock-out Mode.
6	EXF2	Timer 2 external flag is set when Timer 2 is in capture, reload or baud-rate mode, EXEN2 = 1 and a negative transition on T2EX occurs. If Timer 2 interrupt is enabled EXF2 = 1 causes the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.
5	RCLK	Receive clock flag. When set, causes the UART 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.
4	TCLK	Transmit clock flag. When set, causes the UART 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.
3	EXEN2	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.
2	TR2	Start/stop control for Timer 2. A logic '1' enables the timer to run.
1	C/T2	Timer or counter select. (Timer 2) 0 = internal timer ( $f_{osc}/6$ ) 1 = External event counter (falling edge triggered; external clock's maximum rate = $f_{osc}/12$ )
0	CP/RL2	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.

**Table 22: T2MOD - Timer 2 mode control register (address C9H) bit allocation**

Not bit addressable; Reset value: XX00000B

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	T2OE	DCEN

**Table 23: T2MOD - Timer 2 mode control register (address C9H) bit description**

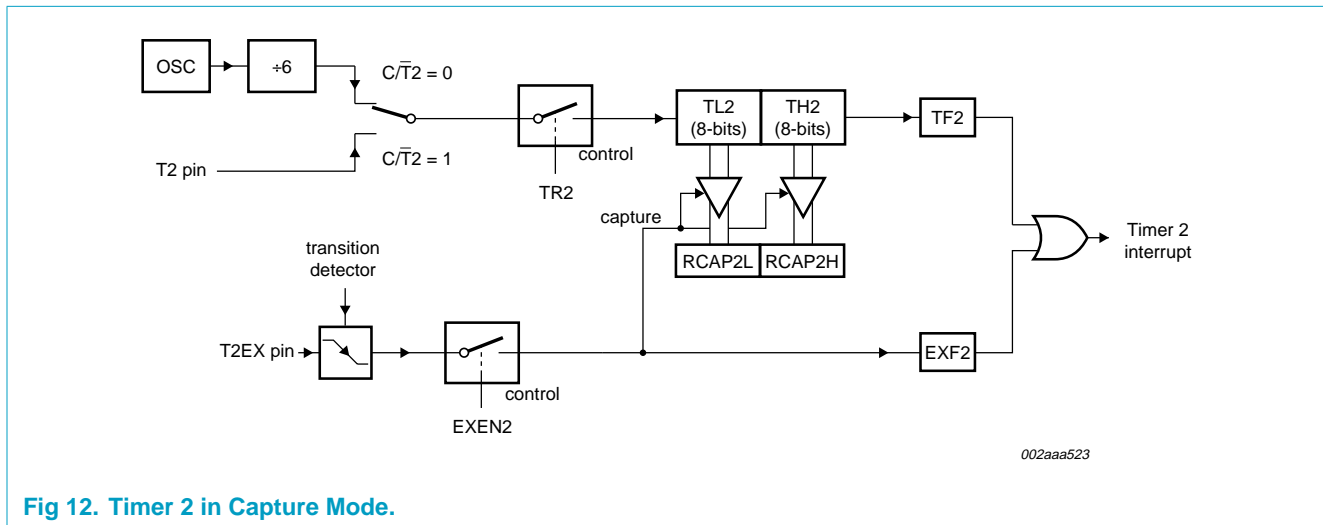
Bit	Symbol	Description
7 to 2	-	Reserved for future use. Should be set to '0' by user programs.
1	T2OE	Timer 2 Output Enable bit. Used in programmable clock-out mode only.
0	DCEN	Down Count Enable bit. When set, this allows Timer 2 to be configured as an up/down counter.

#### 7.4.1 Capture mode

In the Capture Mode there are two options which are selected by bit EXEN2 in T2CON. If EXEN2 = 0 Timer 2 is a 16-bit timer or counter (as selected by C/T2 in T2CON) which upon overflowing sets bit TF2, the Timer 2 overflow bit.



The capture mode is illustrated in [Figure 12](#).



**Fig 12. Timer 2 in Capture Mode.**

This bit can be used to generate an interrupt (by enabling the Timer 2 interrupt bit in the IEN0 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.

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 T2 pin transitions or  $f_{osc}/6$  pulses. Since once loaded contents of RCAP2L and RCAP2H registers are not protected, once Timer2 interrupt is signalled it has to be serviced before new capture event on T2EX pin occurs. Otherwise, the next falling edge on T2EX pin will initiate reload of the current value from TL2 and TH2 to RCAP2L and RCAP2H and consequently corrupt their content related to previously reported interrupt.

#### 7.4.2 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 (via  $C/\bar{T}2$  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 [Table 22](#) and [Table 23](#)). When reset is applied, DCEN = 0 and Timer 2 will default to counting up. If the DCEN bit is set, Timer 2 can count up or down depending on the value of the T2EX pin.

[Figure 13](#) shows Timer 2 counting up automatically (DCEN = 0).

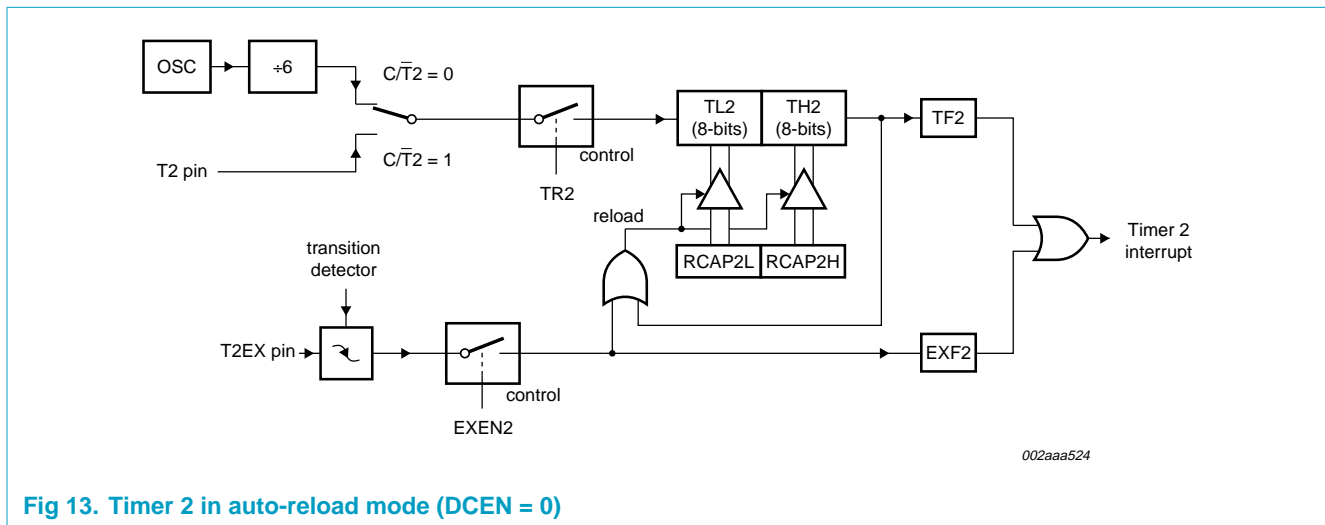


Fig 13. Timer 2 in auto-reload mode (DCEN = 0)

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.

Auto reload frequency when Timer 2 is counting up can be determined from this formula:

$$\frac{\text{SupplyFrequency}}{(65536 \angle (RCAP2H, RCAP2L))} \quad (1)$$

Where SupplyFrequency is either  $f_{\text{osc}}$  ( $C/\overline{T2} = 0$ ) or frequency of signal on T2 pin ( $C/\overline{T2} = 1$ ).

If EXEN2 = 1, 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 is '1'.

Microcontroller's hardware will need three consecutive machine cycles in order to recognize falling edge on T2EX and set EXF2 = 1: in the first machine cycle pin T2EX has to be sampled as '1'; in the second machine cycle it has to be sampled as '0', and in the third machine cycle EXF2 will be set to '1'.

In **Figure 14**, DCEN = 1 and Timer 2 is enabled 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.

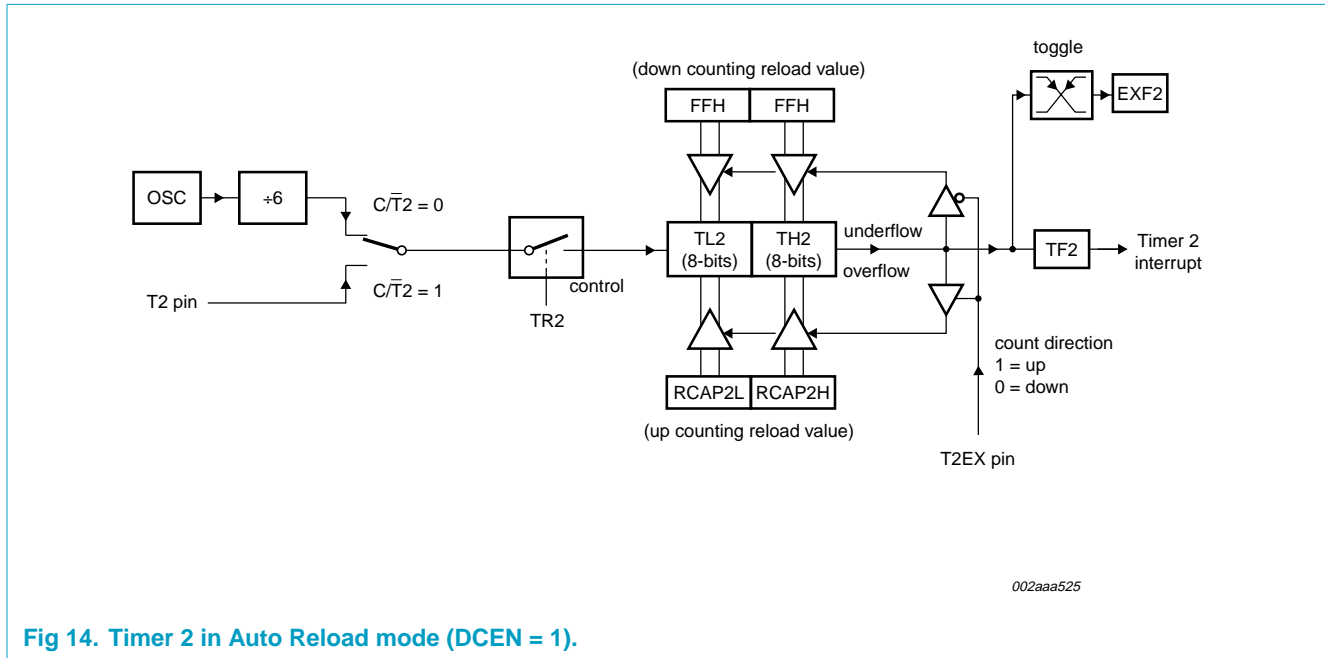


Fig 14. Timer 2 in Auto Reload mode (DCEN = 1).

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.

#### 7.4.3 Programmable clock-out

A 50 % duty cycle clock can be programmed to come out on pin T2 (P1.0). This pin, besides being a regular I/O pin, has two additional 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 122 Hz to 8 MHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit  $C/\bar{T}2$  (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 Equation 2:

$$\frac{\text{Oscillator Frequency}}{2 \times (65536 \angle (\text{RCAP2H}, \text{RCAP2L}))} \quad (2)$$

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.

7.4.4 Baud rate generator mode

Bits TCLK and/or RCLK in T2CON allow the UART) transmit and receive baud rates to be derived from either Timer 1 or Timer 2 (See Section 7.5 “UARTs” on page 37 for details). When TCLK = 0, Timer 1 is used as the UART transmit baud rate generator. When TCLK = 1, Timer 2 is used as the UART transmit baud rate generator. RCLK has the same effect for the UART receive baud rate. With these two bits, the serial port can have different receive and transmit baud rates – Timer 1 or Timer 2.

Figure 15 shows Timer 2 in baud rate generator mode:

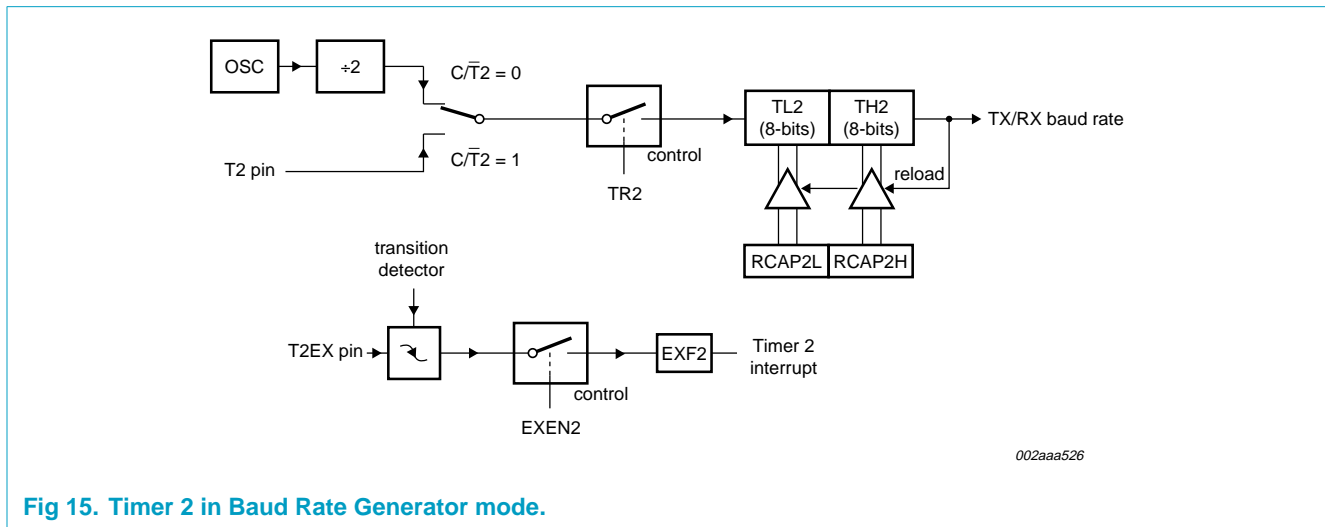


Fig 15. Timer 2 in Baud Rate Generator mode.

The baud rate generation mode is like the auto-reload mode, when 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:

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

The timer can be configured for either ‘timer’ or ‘counter’ operation. In many applications, it is configured for ‘timer’ operation ( $C/\bar{T}2 = 0$ ). Timer operation is different for Timer 2 when it is being used as a baud rate generator.

Usually, as a timer it would increment every machine cycle (i.e.,  $1/6$  the oscillator frequency). As a baud rate generator, it increments at the oscillator frequency. Thus the baud rate formula is as follows:

$$\text{Modes 1 and 3 Baud Rates} =$$

$$\frac{\text{OscillatorFrequency}}{(16 \times (65536 - (RCAP2H, RCAP2L)))} \tag{3}$$

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

#### 7.4.5 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:

$$\text{Baud rate} = \text{Timer 2 overflow rate} / 16$$

If Timer 2 is being clocked internally, the baud rate is:

$$\text{Baud rate} = f_{\text{osc}} / (16 \times (65536 - (\text{RCAP2H}, \text{RCAP2L})))$$

Where  $f_{\text{osc}}$  = oscillator frequency

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

$$\text{RCAP2H}, \text{RCAP2L} = 65536 - f_{\text{osc}} / (16 \times \text{baud rate})$$

**Table 24: Timer 2 generated commonly used baud rates**

Baud rate	Osc freq	Timer 2	
		RCAP2H	RCAP2L
750K	12 MHz	FF	FF
19.2K	12 MHz	FF	D9
9.6K	12 MHz	FF	B2
4.8K	12 MHz	FF	64
2.4K	12 MHz	FE	C8
600	12 MHz	FB	1E
220	12 MHz	F2	AF
600	6 MHz	FD	8F
220	6 MHz	F9	57

## 7.5 UARTs

The UART operates in all standard modes. Enhancements over the standard 80C51 UART include Framing Error detection, and automatic address recognition.

**7.5.1 Mode 0**

Serial data enters and exits through RxD and TxD outputs the shift clock. Only 8 bits are transmitted or received, LSB first. The baud rate is fixed at  $\frac{1}{6}$  of the CPU clock frequency. UART configured to operate in this mode outputs serial clock on TxD line no matter whether it sends or receives data on RxD line.

**7.5.2 Mode 1**

10 bits are transmitted (through TxD) or received (through RxD): a start bit (logical 0), 8 data bits (LSB first), and a stop bit (logical 1). When data is received, the stop bit is stored in RB8 in Special Function Register SCON. The baud rate is variable and is determined by the Timer  $\frac{1}{2}$  overflow rate.

**7.5.3 Mode 2**

11 bits are transmitted (through TxD) or received (through RxD): start bit (logical 0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (logical 1). When data is transmitted, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or (e.g. the parity bit (P, in the PSW) could be moved into TB8). When data is received, the 9th data bit goes into RB8 in Special Function Register SCON, while the stop bit is ignored. The baud rate is programmable to either  $\frac{1}{16}$  or  $\frac{1}{32}$  of the CPU clock frequency, as determined by the SMOD1 bit in PCON.

**7.5.4 Mode 3**

11 bits are transmitted (through TxD) or received (through RxD): a start bit (logical 0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (logical 1). In fact, Mode 3 is the same as Mode 2 in all respects except baud rate. The baud rate in Mode 3 is variable and is determined by the Timer  $\frac{1}{2}$  overflow rate.

**Table 25: SCON - Serial port control register (address 98H) bit allocation**

Bit addressable; Reset value: 00H

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

**Table 26: SCON - Serial port control register (address 98H) bit description**

Bit	Symbol	Description
7	SM0/FE	The usage of this bit is determined by SMOD0 in the PCON register. If SMOD0 = 0, this bit is SM0, which with SM1, defines the serial port mode. If SMOD0 = 1, this bit is FE (Framing Error). FE is set by the receiver when an invalid stop bit is detected. Once set, this bit cannot be cleared by valid frames but can only be cleared by software. (Note: It is recommended to set up UART mode bits SM0 and SM1 before setting SMOD0 to '1'.)
6	SM1	With SM0, defines the serial port mode (see <a href="#">Table 27</a> below).
5	SM2	Enables the multiprocessor communication feature in Modes 2 and 3. In Mode 2 or 3, if SM2 is set to '1', then RI will not be activated if the received 9th data bit (RB8) is '0'. In Mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In Mode 0, SM2 should be '0'.
4	REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.

**Table 26: SCON - Serial port control register (address 98H) bit description...continued**

Bit	Symbol	Description
3	TB8	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.
2	RB8	In Modes 2 and 3, is 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 undefined.
1	TI	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the stop bit in the other modes, in any serial transmission. Must be cleared by software.
0	RI	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or approximately halfway through the stop bit time in all other modes. (See SM2 for exceptions). Must be cleared by software.

**Table 27: SCON - Serial port control register (address 98H) SM0/SM1 mode definition**

SM0, SM1	UART mode	Baud rate
0 0	0: shift register	CPU clock/6
0 1	1: 8-bit UART	variable
1 0	2: 9-bit UART	CPU clock/32 or CPU clock/16
1 1	3: 9-bit UART	variable

### 7.5.5 Framing error

Framing error (FE) is reported in the SCON.7 bit if SMOD0 (PCON.6) = 1. If SMOD0 = 0, SCON.7 is the SM0 bit for the UART, it is recommended that SM0 is set up before SMOD0 is set to '1'.

### 7.5.6 More about UART mode 1

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

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

The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated: (a) RI = 0, and (b) Either SM2 = 0, or the received stop bit = 1.

If either of these two conditions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated.

### 7.5.7 More about UART modes 2 and 3

Reception is performed in the same manner as in mode 1.

The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated: (a) RI = 0, and (b) Either SM2 = 0, or the received 9th data bit = 1.

If either of these conditions is not met, the received frame is irretrievably lost, and RI is not set. If both conditions are met, the received 9th data bit goes into RB8, and the first 8 data bits go into SBUF.

### 7.5.8 Multiprocessor communications

UART modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received or transmitted. When data is received, the 9th bit is stored in RB8. The UART can be programmed so that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON. One way to use this feature in multiprocessor systems is as follows:

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

SM2 has no effect in Mode 0, and in Mode 1 can be used to check the validity of the stop bit, although this is better done with the Framing Error flag. When UART receives data in mode 1 and SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

### 7.5.9 Automatic address recognition

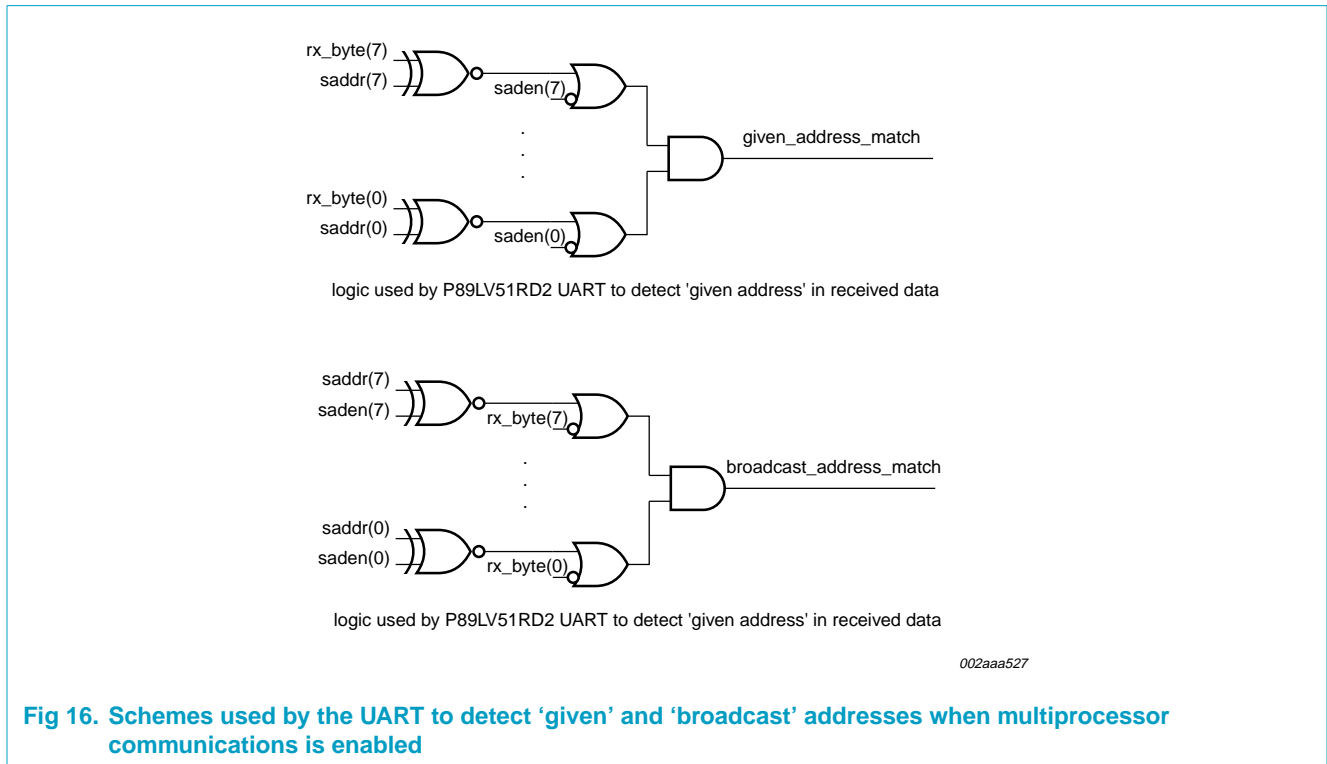
Automatic Address Recognition is a feature which allows the UART to recognize certain addresses in the serial bit stream by using hardware to make the comparisons. This feature saves a great deal of software overhead by eliminating the need for the software to examine every serial address which passes by the serial port. This feature is enabled for the UART 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.

Using the Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the Given slave address or addresses. All of the slaves may be contacted by using the Broadcast address. Two Special Function Registers are used to define the slave's address, SADDR, and the address mask, SADEN. SADEN is used to define which bits in the SADDR are to be used and which bits are 'don't care'. The SADEN mask can be logically ANDed with



the SADDR to create the 'Given' address which the master will use for addressing each of the slaves. Use of the Given address allows multiple slaves to be recognized while excluding others.

This device uses the methods presented in Figure 16 to determine if a 'Given' or 'Broadcast' address has been received or not.



**Fig 16. Schemes used by the UART to detect 'given' and 'broadcast' addresses when multiprocessor communications is enabled**

The following examples will help to show the versatility of this scheme.

**Table 28: Slaves 0 and 1 scheme examples**

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:

**Table 29: Slaves 0, 1 and 2 scheme examples**

Slave 0	SADDR = 1110 0000
	SADEN = 1111 1001
	Given = 1110 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 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 treated as don't-cares. In most cases, interpreting the don't-cares as ones, the broadcast address will be FF hexadecimal. Upon reset SADDR and SADEN are loaded 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 UART drivers which do not make use of this feature.

## 7.6 Serial peripheral interface

### 7.6.1 SPI features

- Master or slave operation
- 10 MHz bit frequency (max)
- LSB first or MSB first data transfer
- Four programmable bit rates
- End of transmission (SPIF)
- Write collision flag protection (WCOL)
- Wake-up from idle mode (slave mode only)

### 7.6.2 SPI description

The serial peripheral interface (SPI) allows high-speed synchronous data transfer between the P89LV51RB2/RC2/RD2 and peripheral devices or between several P89LV51RB2/RC2/RD2 devices. **Figure 17** shows the correspondence between master and slave SPI devices. The SCK pin is the clock output and input for the master and slave modes, respectively. The SPI clock generator will start following a write to the master devices SPI data register. The written data is then shifted out of the MOSI pin on the master device into the MOSI pin of the slave device. Following a complete transmission of one byte of data, the SPI clock generator is stopped and the SPIF flag is set. An SPI interrupt request will be generated if the SPI Interrupt Enable bit (SPIE) and the Serial Port Interrupt Enable bit (ES) are both set.

An external master drives the Slave Select input pin,  $\overline{SS}/P1[4]$ , low to select the SPI module as a slave. If  $\overline{SS}/P1[4]$  has not been driven low, then the slave SPI unit is not active and the MOSI/P1[5] port can also be used as an input port pin.

CPHA and CPOL control the phase and polarity of the SPI clock. Figure 18 and Figure 19 show the four possible combinations of these two bits.

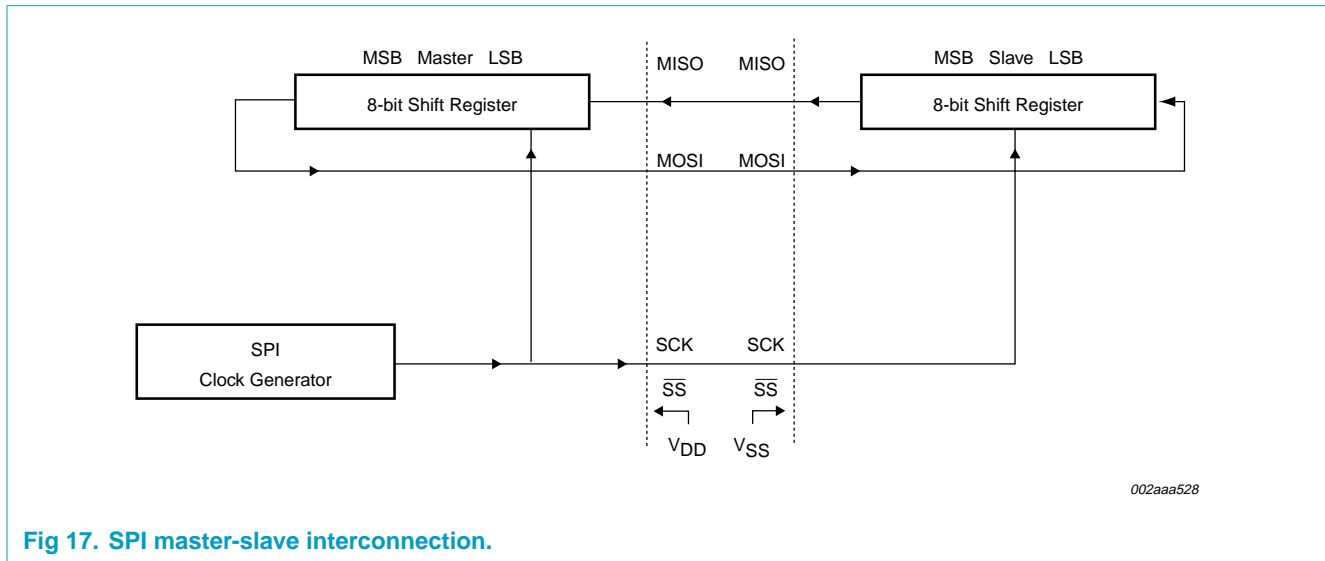


Fig 17. SPI master-slave interconnection.

Table 30: SPCR - SPI control register (address D5H) bit allocation

Bit addressable; Reset source(s): any reset; Reset value: 0000000B

Bit	7	6	5	4	3	2	1	0
Symbol	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0

Table 31: SPCR - SPI control register (address D5H) bit description

Bit	Symbol	Description
7	SPIE	If both SPIE and ES are set to one, SPI interrupts are enabled.
6	SPE	SPI enable bit. When set enables SPI.
5	DORD	Data transmission order. 0 = MSB first; 1 = LSB first in data transmission.
4	MSTR	Master/slave select. 1 = master mode, 0 = slave mode.
3	CPOL	Clock polarity. 1 = SCK is high when idle (active LOW), 0 = SCK is low when idle (active HIGH).
2	CPHA	Clock Phase control bit. 1 = shift triggered on the trailing edge of the clock; 0 = shift triggered on the leading edge of the clock.
1	SPR1	SPI Clock Rate Select bit 1. Along with SPR0 controls the SCK rate of the device when a master. SPR1 and SPR0 have no effect on the slave. See Table 32 below.
0	SPR0	SPI Clock Rate Select bit 0. Along with SPR1 controls the SCK rate of the device when a master. SPR1 and SPR0 have no effect on the slave. See Table 32 below.

**Table 32: SPCR - SPI control register (address D5H) clock rate selection**

SPR1	SPR0	SCK = f <sub>osc</sub> divided by
0	0	4
0	1	16
1	0	64
1	1	128

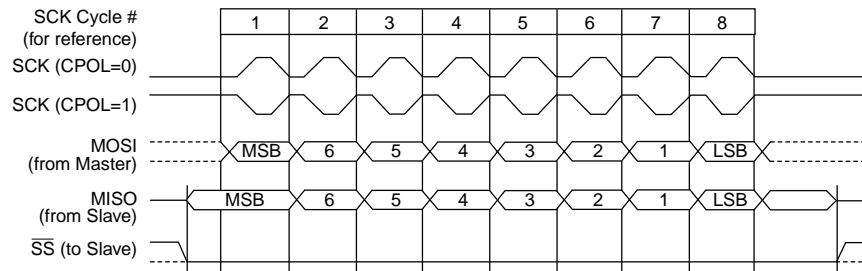
**Table 33: SPSR - SPI status register (address AAH) bit allocation**

Bit addressable; Reset source(s): any reset; Reset value: 0000000B

Bit	7	6	5	4	3	2	1	0
Symbol	SPIF	WCOL	-	-	-	-	-	-

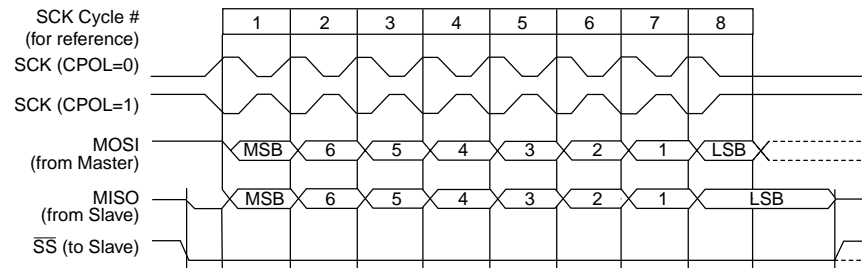
**Table 34: SPSR - SPI status register (address AAH) bit description**

Bit	Symbol	Description
7	SPIF	SPI interrupt flag. Upon completion of data transfer, this bit is set to '1'. If SPIE = 1 and ES = 1, an interrupt is then generated. This bit is cleared by software.
6	WCOL	Write Collision Flag. Set if the SPI data register is written to during data transfer. This bit is cleared by software.
5 to 0	-	Reserved for future use. Should be set to '0' by user programs.



002aaa529

**Fig 18. SPI transfer format with CPHA = 0.**



002aaa530

**Fig 19. SPI transfer format with CPHA = 1.**

### 7.7 Watchdog timer

The device offers a programmable Watchdog Timer (WDT) for fail safe protection against software deadlock and automatic recovery.

To protect the system against software deadlock, the user software must refresh the WDT within a user-defined time period. If the software fails to do this periodical refresh, an internal hardware reset will be initiated if enabled (WDRE = 1). The software can be designed such that the WDT times out if the program does not work properly.

The WDT in the device uses the system clock (XTAL1) as its time base. So strictly speaking, it is a Watchdog counter rather than a Watchdog timer. The WDT register will increment every 344,064 crystal clocks. The upper 8-bits of the time base register (WDTD) are used as the reload register of the WDT.

The WDTS flag bit is set by WDT overflow and is not changed by WDT reset. User software can clear WDTS by writing '1' to it.

Figure 20 provides a block diagram of the WDT. Two SFRs (WDTC and WDTD) control Watchdog timer operation. During idle mode, WDT operation is temporarily suspended, and resumes upon an interrupt exit from idle.

The time-out period of the WDT is calculated as follows:

$$\text{Period} = (255 - \text{WDTD}) \times 344064 \times 1/f_{\text{CLK (XTAL1)}}$$

where WDTD is the value loaded into the WDTD register and  $f_{\text{osc}}$  is the oscillator frequency.

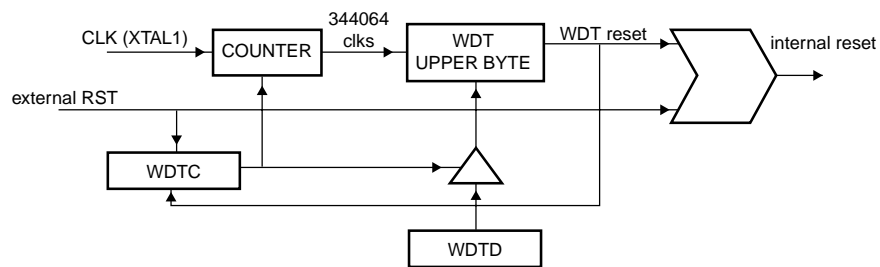


Fig 20. Block diagram of programmable Watchdog timer

Table 35: WDTC - Watchdog control register (address COH) bit allocation

Bit addressable; Reset value: 00H

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	-	WDOUT	WDRE	WDTS	WDT	SWDT

Table 36: WDTC - Watchdog control register (address COH) bit description

Bit	Symbol	Description
7 to 5	-	Reserved for future use. Should be set to '0' by user programs.
4	WDOUT	Watchdog output enable. When this bit and WDRE are both set, a Watchdog reset will drive the reset pin active for 32 clocks.
3	WDRE	Watchdog timer reset enable. When set enables a Watchdog timer reset.
2	WDTS	Watchdog timer reset flag, when set indicates that a WDT reset occurred. Reset in software.
1	WDT	Watchdog timer refresh. Set by software to force a WDT reset.
0	SWDT	Start Watchdog timer, when set starts the WDT. When cleared, stops the WDT.

## 7.8 Programmable Counter Array (PCA)

The PCA includes 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. Registers CH and CL contain current value of the free running up counting 16-bit PCA timer. The PCA timer is a common time base for all five modules and can be programmed to run at:  $\frac{1}{6}$  the oscillator frequency,  $\frac{1}{2}$  the oscillator frequency, the Timer 0 overflow, or the input on the ECI pin (P1.2). The timer count source is determined from the CPS1 and CPS0 bits in the CMOD SFR (see Table 37 and Table 38).

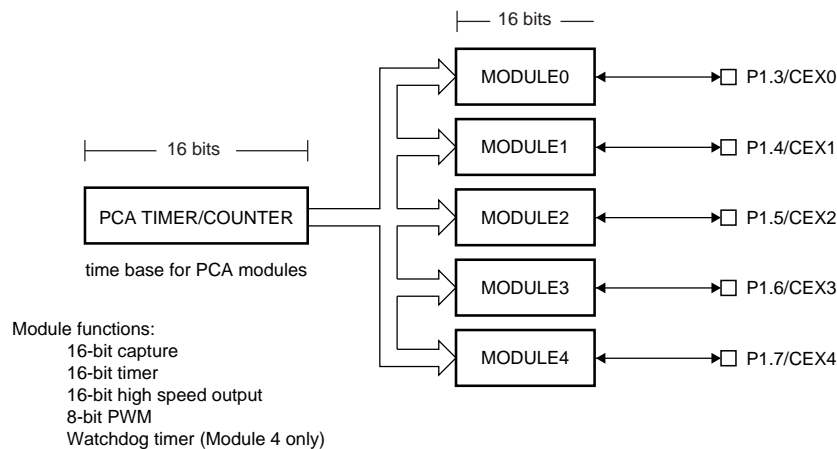


Fig 21.

In the CMOD SFR there 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.

The Watchdog timer function is implemented in module 4 of PCA.

The CCON SFR contains the run control bit for the PCA (CR) and the flags for the PCA timer (CF) and each module (CCF4:0). 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 can only be cleared by software. All the modules share one interrupt vector. The PCA interrupt system is shown in [Figure 22](#).

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. The registers contain the bits that control the mode that each module will operate in.

The ECCF bit (from CCAPMn.0 where n = 0, 1, 2, 3, or 4 depending on the module) enables the CCFn flag in the CCON SFR to generate an interrupt when a match or compare occurs in the associated module (see [Figure 22](#)).

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.

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.

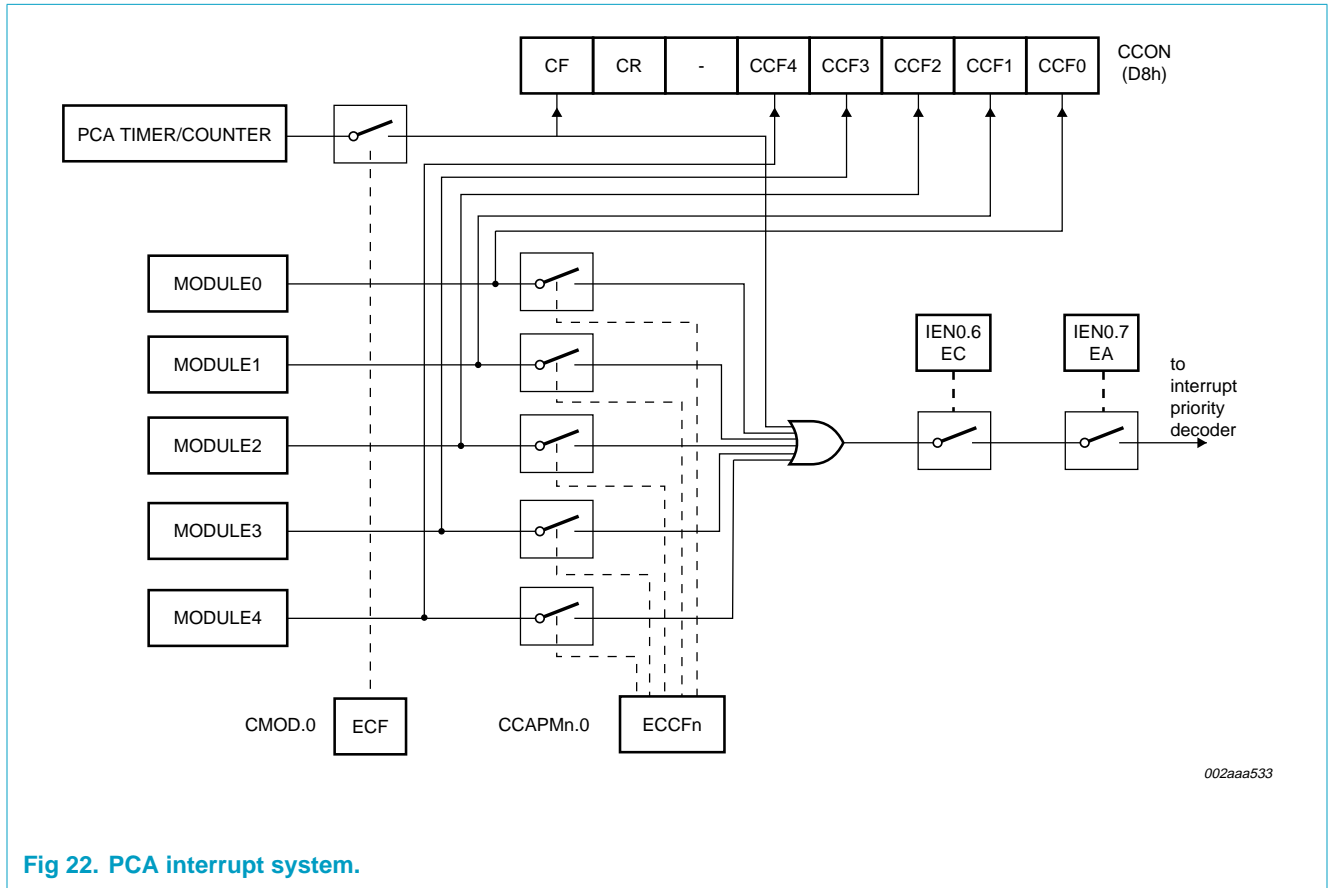


Fig 22. PCA interrupt system.

Table 37: CMOD - PCA counter mode register (address D9H) bit allocation  
 Not bit addressable; Reset value: 00H

Bit	7	6	5	4	3	2	1	0
Symbol	CIDL	WDTE	-	-	-	CPS1	CPS0	ECF

Table 38: CMOD - PCA counter mode register (address D9H) bit description

Bit	Symbol	Description
7	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.
6	WDTE	Watchdog Timer Enable: WDTE = 0 disables Watchdog timer function on module 4. WDTE = 1 enables it.
5 to 3	-	Reserved for future use. Should be set to '0' by user programs.
2 to 1	CPS1, CPS0	PCA Count Pulse Select (see Table 39 below).
0	ECF	PCA Enable Counter Overflow Interrupt: ECF = 1 enables CF bit in CCON to generate an interrupt. ECF = 0 disables that function.



**Table 39: CMOD - PCA counter mode register (address D9H) count pulse select**

CPS1	CPS0	Select PCA input
0	0	0 Internal clock, $f_{osc} / 6$
0	1	1 Internal clock, $f_{osc} / 6$
1	0	2 Timer 0 overflow
1	1	3 External clock at ECI/P1.2 pin (max rate = $f_{osc} / 4$ )

**Table 40: CCON - PCA counter control register (address 0D8H) bit allocation**

Bit addressable; Reset value: 00H

Bit	7	6	5	4	3	2	1	0
Symbol	CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0

**Table 41: CCON - PCA counter control register (address 0D8H) bit description**

Bit	Symbol	Description
7	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.
6	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.
5	-	Reserved for future use. Should be set to '0' by user programs.
4	CCF4	PCA Module 4 Interrupt Flag. Set by hardware when a match or capture occurs. Must be cleared by software.
3	CCF3	PCA Module 3 Interrupt Flag. Set by hardware when a match or capture occurs. Must be cleared by software.
2	CCF2	PCA Module 2 Interrupt Flag. Set by hardware when a match or capture occurs. Must be cleared by software.
1	CCF1	PCA Module 1 Interrupt Flag. Set by hardware when a match or capture occurs. Must be cleared by software.
0	CCF0	PCA Module 0 Interrupt Flag. Set by hardware when a match or capture occurs. Must be cleared by software.

**Table 42: CCAPMn - PCA modules compare/capture register (address CCAPM0 0DAH, CCAPM1 0DBH, CCAPM2 0DCH, CCAPM3 0DDH, CCAPM4 0DEH) bit alloc.**

Not bit addressable; Reset value: 00H

Bit	7	6	5	4	3	2	1	0
Symbol	-	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn

**Table 43: CCAPMn - PCA modules compare/capture register (address CCAPM0 0DAH, CCAPM1 0DBH, CCAPM2 0DCH, CCAPM3 0DDH, CCAPM4 0DEH) bit desc.**

Bit	Symbol	Description
7	-	Reserved for future use. Should be set to '0' by user programs.
6	ECOMn	Enable Comparator. ECOMn = 1 enables the comparator function.
5	CAPPn	Capture Positive, CAPPn = 1 enables positive edge capture.
4	CAPNn	Capture Negative, CAPNn = 1 enables negative edge capture.

**Table 43: CCAPMn - PCA modules compare/capture register (address CCAPM0 0DAH, CCAPM1 0DBH, CCAPM2 0DCH, CCAPM3 0DDH, CCAPM4 0DEH) bit desc.**

Bit	Symbol	Description
3	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.
2	TOGn	Toggle. When TOGn = 1, a match of the PCA counter with this module's compare/capture register causes the CEXn pin to toggle.
1	PWMn	Pulse Width Modulation Mode. PWMn = 1 enables the CEXn pin to be used as a pulse width modulated output.
0	ECCFn	Enable CCF Interrupt. Enables compare/capture flag CCFn in the CCON register to generate an interrupt.

**Table 44: PCA module modes (CCAPMn register)**

ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	Module function
0	0	0	0	0	0	0	no operation
x	1	0	0	0	0	x	16-bit capture by a positive-edge trigger on CEXn
x	0	1	0	0	0	x	16-bit capture by a negative-edge trigger on CEXn
x	1	1	0	0	0	x	16-bit capture by any transition on CEXn
1	0	0	1	0	0	x	16-bit software timer
1	0	0	1	1	0	x	16-bit high speed output
1	0	0	0	0	1	0	8-bit PWM
1	0	0	1	x	0	x	Watchdog timer

### 7.8.1 PCA capture mode

To use one of the PCA modules in the capture mode ([Figure 23](#)) 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).

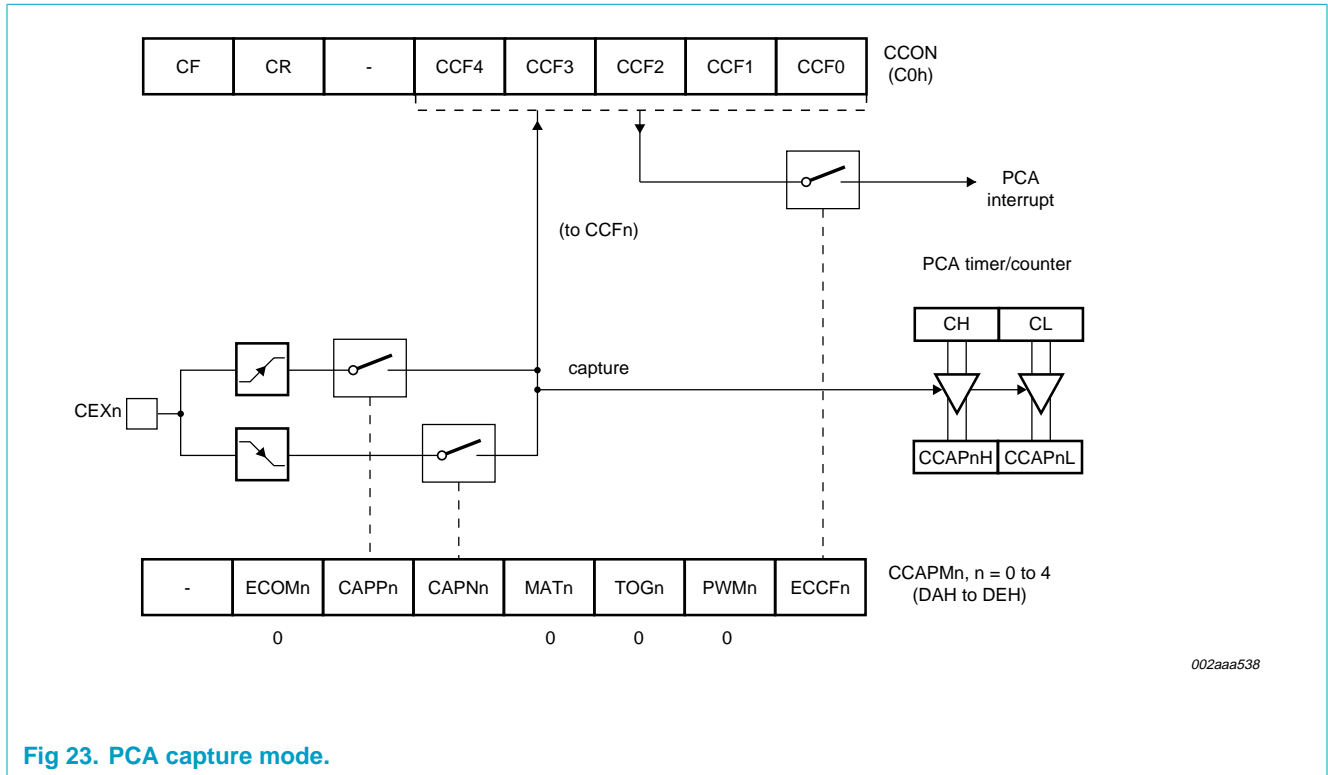


Fig 23. PCA capture mode.

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.

7.8.2 16-bit software timer mode

The PCA modules can be used as software timers (Figure 24) 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.

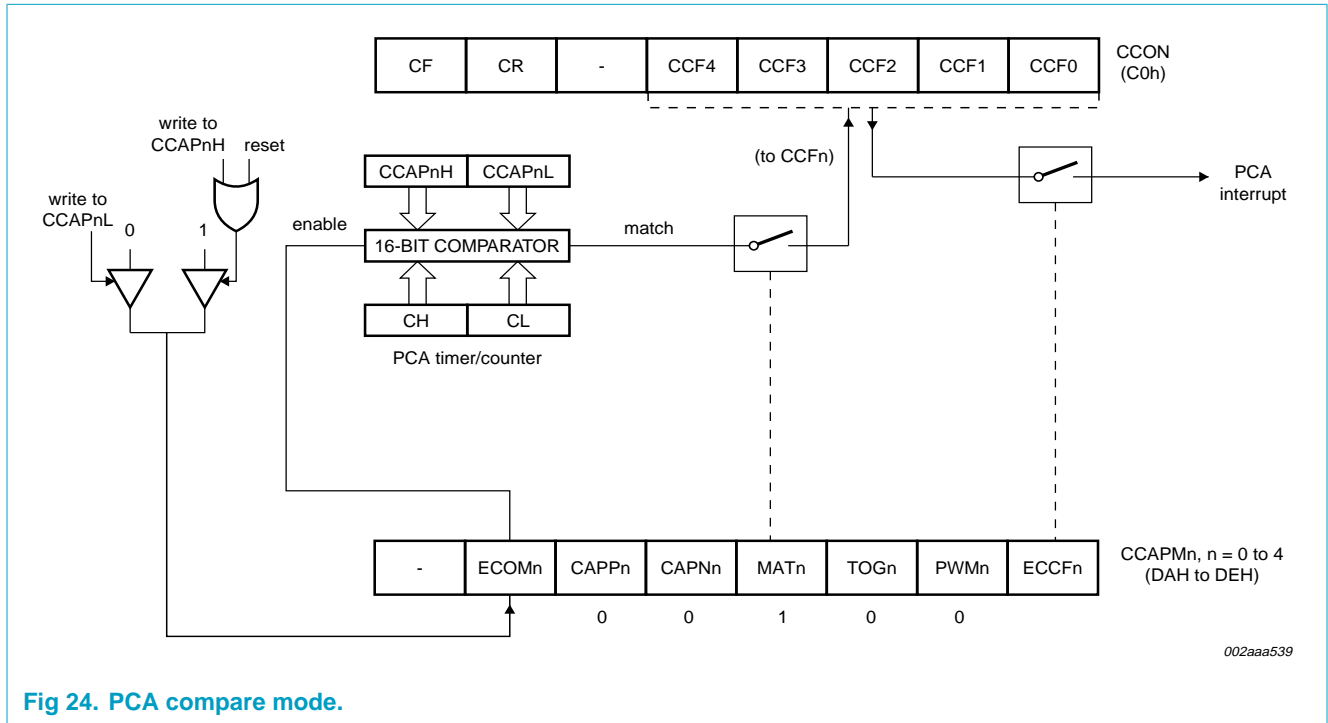


Fig 24. PCA compare mode.

7.8.3 High speed output mode

In this mode (Figure 25) 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.

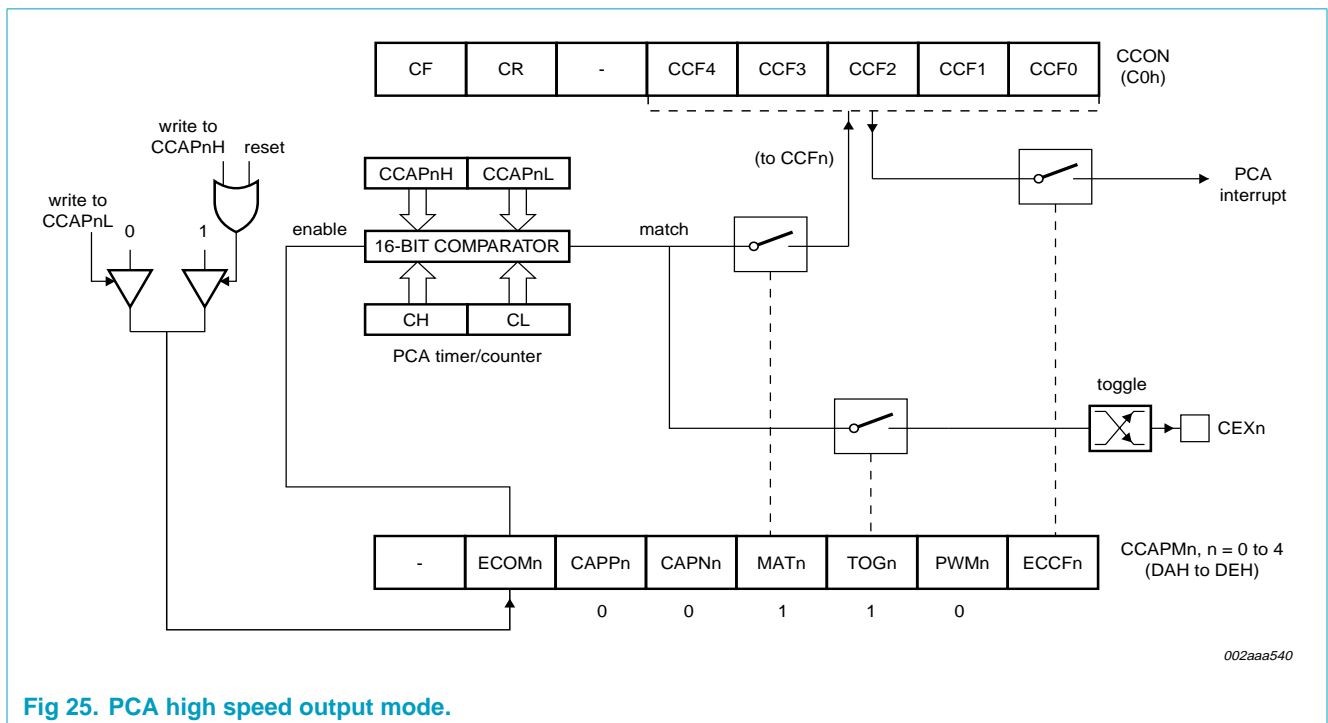


Fig 25. PCA high speed output mode.

### 7.8.4 Pulse width modulator mode

All of the PCA modules can be used as PWM outputs (Figure 26). Output frequency depends on the source for the PCA timer.

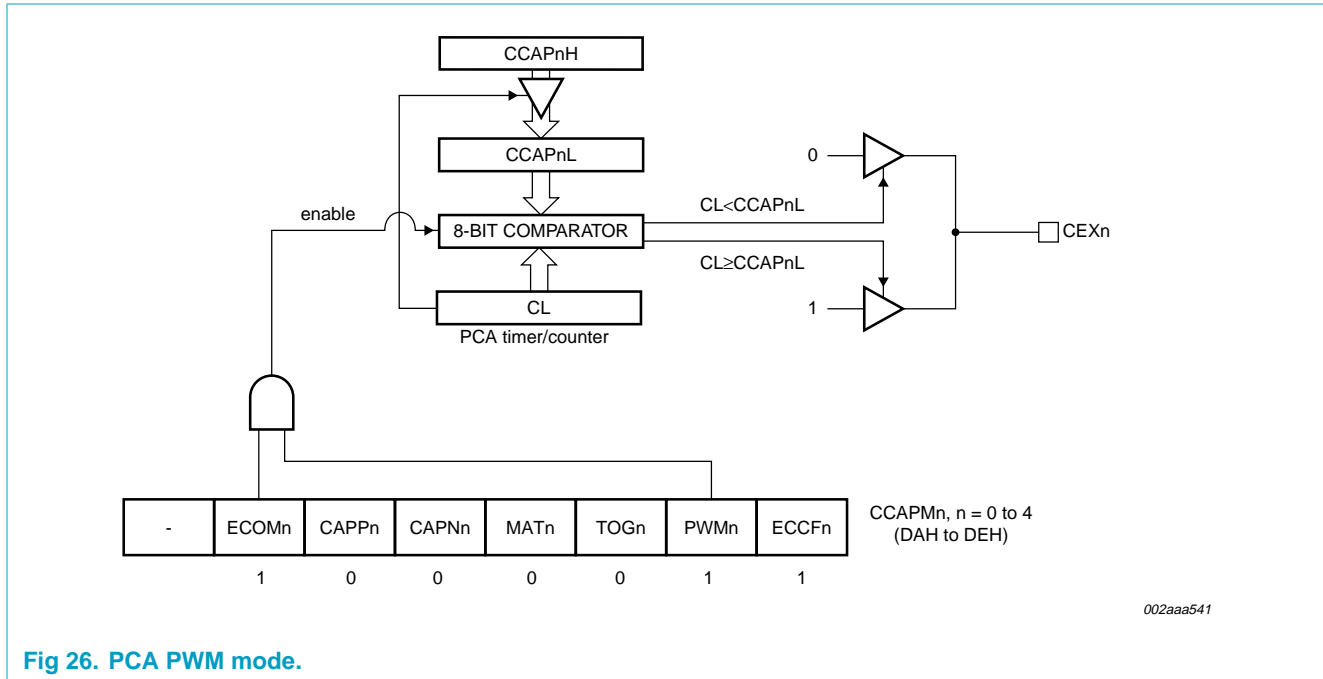


Fig 26. PCA PWM mode.

All of the modules will have the same frequency of output because they all share one and only PCA timer. The duty cycle of each module is independently variable using the module's capture register CCAPnL. When the value of the PCA CL SFR is less than the value in the module's CCAPnL 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, CCAPnL is reloaded with the value in CCAPnH. This 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.

### 7.8.5 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 26 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.

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 shown above.

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.
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.

```
;CALL the following WATCHDOG subroutine periodically.
CLR    EA                ;Hold off interrupts
MOV    CCAP4L,#00       ;Next compare value is within 255 counts of
                          ;current PCA timer value

MOV    CCAP4H,CH
SETB   EA                ;Re-enable interrupts
RET
```

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.

## 7.9 Security Bit

The Security Bit protects against software piracy and prevents the contents of the flash from being read by unauthorized parties in Parallel Programmer Mode. It also protects against code corruption resulting from accidental erasing and programming to the internal flash memory.

When the Security Bit is activated all parallel programming commands except for Chip-Erase are ignored (thus the device cannot be read). However, ISP reading, writing, or erasing of the user's code can still be performed if the serial number and length has not been programmed. **Therefore, when a user requests to program the Security Bit, the programmer should prompt the user and program a serial number into the device.**

## 7.10 Interrupt priority and polling sequence

The device supports eight interrupt sources under a four level priority scheme.

**Table 45** summarizes the polling sequence of the supported interrupts. Note that the SPI serial interface and the UART share the same interrupt vector. (See **Figure 27**).

**Table 45: Interrupt polling sequence**

Description	Interrupt Flag	Vector Address	Interrupt Enable	Interrupt Priority	Service Priority	Wake-Up Power-down
Ext. Int0	IE0	0003H	EX0	PX0/H	1 (highest)	yes
Brown-out	-	004BH	EBO	PBO/H	2	no
T0	TF0	000BH	ET0	PT0/H	3	no

Table 45: Interrupt polling sequence...continued

Description	Interrupt Flag	Vector Address	Interrupt Enable	Interrupt Priority	Service Priority	Wake-Up Power-down
Ext. Int1	IE1	0013H	EX1	PX1/H	4	yes
T1	TF1	001BH	ET1	PT1/H	5	no
PCA	CF/CCFn	0033H	EC	PPCH	6	no
UART/SPI	TI/RI/SPIF	0023H	ES	PS/H	7	no
T2	TF2, EXF2	002BH	ET2	PT2/H	8	no

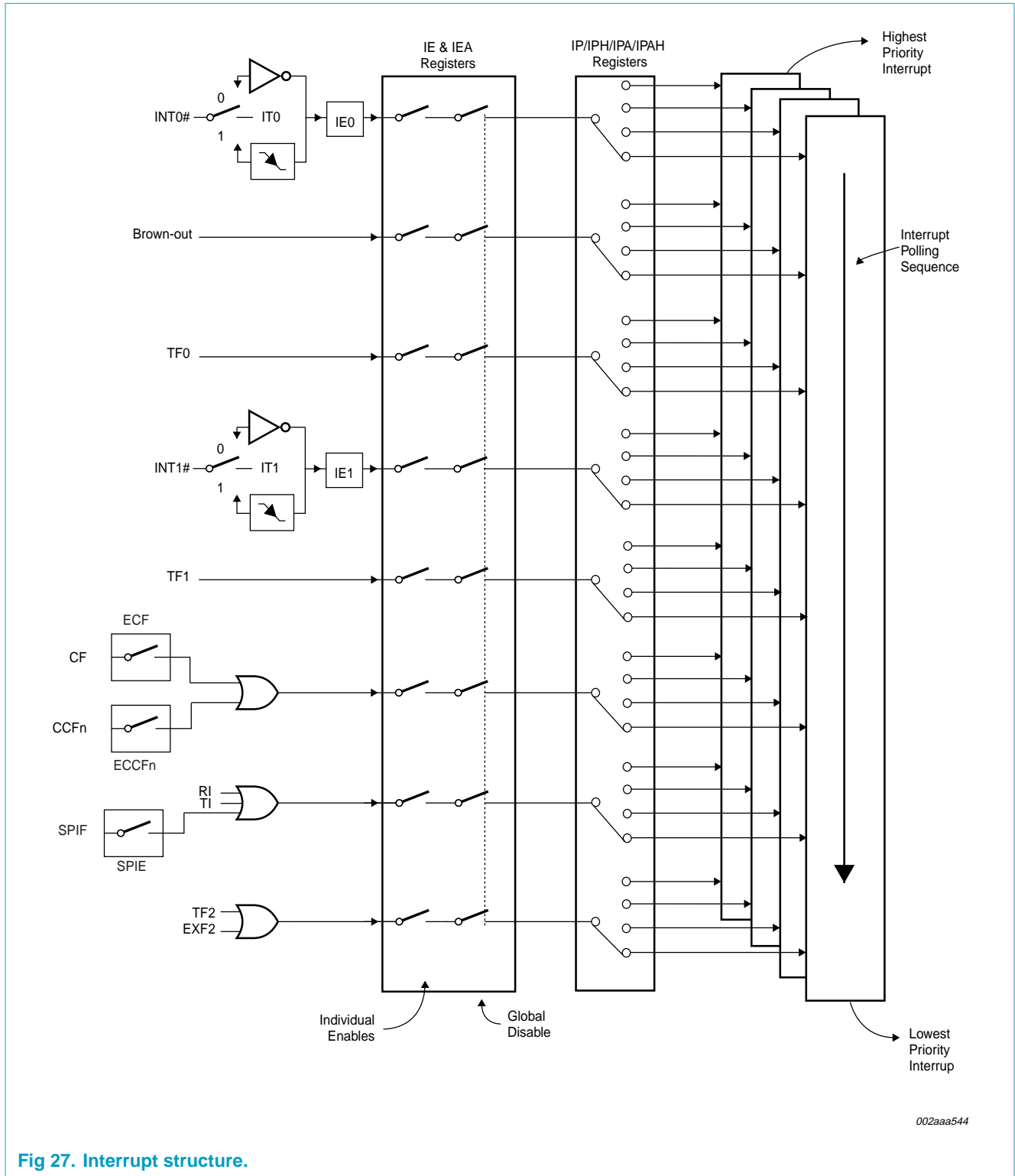


Fig 27. Interrupt structure.



**Table 46: IEN0 - Interrupt enable register 0 (address A8H) bit allocation***Bit addressable; Reset value: 00H*

Bit	7	6	5	4	3	2	1	0
Symbol	EA	EC	ET2	ES	ET1	EX1	ET0	EX0

**Table 47: IEN0 - Interrupt enable register 0 (address A8H) bit description**

Bit	Symbol	Description
7	EA	Interrupt Enable Bit: EA = 1 interrupt(s) can be serviced, EA = 0 interrupt servicing disabled.
6	EC	PCA Interrupt Enable bit.
5	ET2	Timer 2 Interrupt Enable.
4	ES	Serial Port Interrupt Enable
3	ET1	Timer 1 Overflow Interrupt Enable.
2	EX1	External Interrupt 1 Enable.
1	ET0	Timer 0 Overflow Interrupt Enable.
0	EX0	External Interrupt 0 Enable.

**Table 48: IEN1 - Interrupt enable register 1 (address E8H) bit allocation***Bit addressable; Reset value: 00H*

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	EBO	-	-	-

**Table 49: IEN1 - Interrupt enable register 1 (address E8H) bit description**

Bit	Symbol	Description
7 to 4	-	Reserved for future use. Should be set to '0' by user programs.
3	EBO	Brown-out Interrupt Enable. 1 = enable, 0 = disable.
2 to 0	-	Reserved for future use. Should be set to '0' by user programs.

**Table 50: IP0 - Interrupt priority 0 low register (address B8H) bit allocation***Bit addressable; Reset value: 00H*

Bit	7	6	5	4	3	2	1	0
Symbol	-	PPC	PT2	PS	PT1	PX1	PT0	PX0

**Table 51: IP0 - Interrupt priority 0 low register (address B8H) bit description**

Bit	Symbol	Description
7	-	Reserved for future use. Should be set to '0' by user programs.
6	PPC	PCA Interrupt Priority Low Bit.
5	PT2	Timer 2 Interrupt Priority Low Bit.
4	PS	Serial Port Interrupt Priority Low Bit.
3	PT1	Timer 1 Interrupt Priority Low Bit.
2	PX1	External Interrupt 1 Priority Low Bit.
1	PT0	Timer 0 Interrupt Priority Low Bit.
0	PX0	External Interrupt 0 Priority Low Bit.

**Table 52: IP0H - Interrupt priority 0 high register (address B7H) bit allocation***Not bit addressable; Reset value: 00H*

Bit	7	6	5	4	3	2	1	0
Symbol	-	PPCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H

**Table 53: IP0H - Interrupt priority 0 high register (address B7H) bit description**

Bit	Symbol	Description
7	-	Reserved for future use. Should be set to '0' by user programs.
6	PPCH	PCA Interrupt Priority High Bit.
5	PT2H	Timer 2 Interrupt Priority High Bit.
4	PSH	Serial Port Interrupt Priority High Bit.
3	PT1H	Timer 1 Interrupt Priority High Bit.
2	PX1H	External Interrupt 1 Priority High Bit.
1	PT0H	Timer 0 Interrupt Priority High Bit.
0	PX0H	External Interrupt 0 Priority High Bit.

**Table 54: IP1 - Interrupt priority 1 register (address F8H) bit allocation***Bit addressable; Reset value: 00H*

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	-	PBO	-	-	-	-

**Table 55: IP1 - Interrupt priority 1 register (address F8H) bit description**

Bit	Symbol	Description
7 to 5	-	Reserved for future use. Should be set to '0' by user programs.
4	PBO	Brown-out interrupt priority bit.
3 to 0	-	Reserved for future use. Should be set to '0' by user programs.

**Table 56: IP1H - Interrupt priority 1 high register (address F7H) bit allocation***Not bit addressable; Reset value: 00H*

Bit	7	6	5	4	3	2	1	0
Symbol	-	-	-	PBOH	-	-	-	-

**Table 57: IP1H - Interrupt priority 1 high register (address F7H) bit description**

Bit	Symbol	Description
7 to 5	-	Reserved for future use. Should be set to '0' by user programs.
4	PBOH	Brown-out interrupt priority bit.
3 to 0	-	Reserved for future use. Should be set to '0' by user programs.

## 7.11 Power-saving modes

The device provides two power saving modes of operation for applications where power consumption is critical. The two modes are idle and Power-down, see [Table 58](#).

### 7.11.1 Idle mode

Idle mode is entered setting the IDL bit in the PCON register. In idle mode, the program counter (PC) is stopped. The system clock continues to run and all interrupts and peripherals remain active. The on-chip RAM and the special function registers hold their data during this mode.

The device exits idle mode through either a system interrupt or a hardware reset. Exiting idle mode via system interrupt, the start of the interrupt clears the IDL bit and exits idle mode. After exit the Interrupt Service Routine, the interrupted program resumes execution beginning at the instruction immediately following the instruction which invoked the idle mode. A hardware reset starts the device similar to a power-on reset.

### 7.11.2 Power-down mode

The Power-down mode is entered by setting the PD bit in the PCON register. In the Power-down mode, the clock is stopped and external interrupts are active for level sensitive interrupts only. SRAM contents are retained during Power-down, the minimum  $V_{DD}$  level is 2.0 V.

The device exits Power-down mode through either an enabled external level sensitive interrupt or a hardware reset. The start of the interrupt clears the PD bit and exits Power-down. Holding the external interrupt pin low restarts the oscillator, the signal must hold low at least 1024 clock cycles before bringing back high to complete the exit. Upon interrupt signal restored to logic  $V_{IH}$ , the interrupt service routine program execution resumes beginning at the instruction immediately following the instruction which invoked Power-down mode. A hardware reset starts the device similar to power-on reset.

To exit properly out of Power-down, the reset or external interrupt should not be executed before the  $V_{DD}$  line is restored to its normal operating voltage. Be sure to hold  $V_{DD}$  voltage long enough at its normal operating level for the oscillator to restart and stabilize (normally less than 10 ms).

Table 58: Power-saving modes

Mode	Initiated by	State of MCU	Exited by
Idle Mode	Software (Set IDL bit in PCON) MOV PCON, #01H;	CLK is running. Interrupts, serial port and timers/counters are active. Program Counter is stopped. ALE and $\overline{\text{PSEN}}$ signals at a HIGH level during Idle. All registers remain unchanged.	Enabled interrupt or hardware reset. Start of interrupt clears IDL bit and exits idle mode, after the ISR RETI instruction, program resumes execution beginning at the instruction following the one that invoked idle mode. A user could consider placing two or three NOP instructions after the instruction that invokes idle mode to eliminate any problems. A hardware reset restarts the device similar to a power-on reset.
Power-down Mode	Software (Set PD bit in PCON) MOV PCON, #02H;	CLK is stopped. On-chip SRAM and SFR data is maintained. ALE and $\overline{\text{PSEN}}$ signals at a LOW level during power -down. External Interrupts are only active for level sensitive interrupts, if enabled.	Enabled external level sensitive interrupt or hardware reset. Start of interrupt clears PD bit and exits Power-down mode, after the ISR RETI instruction program resumes execution beginning at the instruction following the one that invoked Power-down mode. A user could consider placing two or three NOP instructions after the instruction that invokes Power-down mode to eliminate any problems. A hardware reset restarts the device similar to a power-on reset.

## 7.12 System clock and clock options

### 7.12.1 Clock Input Options and Recommended Capacitor Values for Oscillator

Shown in [Figure 28](#) are the input and output of an internal inverting amplifier (XTAL1, XTAL2), which can be configured for use as an on-chip oscillator.

When driving the device from an external clock source, XTAL2 should be left disconnected and XTAL1 should be driven.

At start-up, the external oscillator may encounter a higher capacitive load at XTAL1 due to interaction between the amplifier and its feedback capacitance. However, the capacitance will not exceed 15 pF once the external signal meets the  $V_{IL}$  and  $V_{IH}$  specifications.

Crystal manufacturer, supply voltage, and other factors may cause circuit performance to differ from one application to another. C1 and C2 should be adjusted appropriately for each design. [Table 59](#) shows the typical values for C1 and C2 vs. crystal type for various frequencies

Table 59: Recommended values for C1 and C2 by crystal type

Crystal	C1 = C2
Quartz	20 pF to 30 pF
Ceramic	40 pF to 50 pF

More specific information about on-chip oscillator design can be found in the *FlashFlex51 Oscillator Circuit Design Considerations* application note.

7.12.2 Clock doubling option

By default, the device runs at 12 clocks per machine cycle (x1 mode). The device has a clock doubling option to speed up to 6 clocks per machine cycle (please see Table 60). Clock double mode can be enabled either by an external programmer or using IAP. When set, the EDC bit in FST register will indicate 6 clock mode.

The clock double mode is only for doubling the internal system clock and the internal flash memory, i.e.  $\overline{EA} = 1$ . To access the external memory and the peripheral devices, careful consideration must be taken. Also note that the crystal output (XTAL2) will not be doubled.

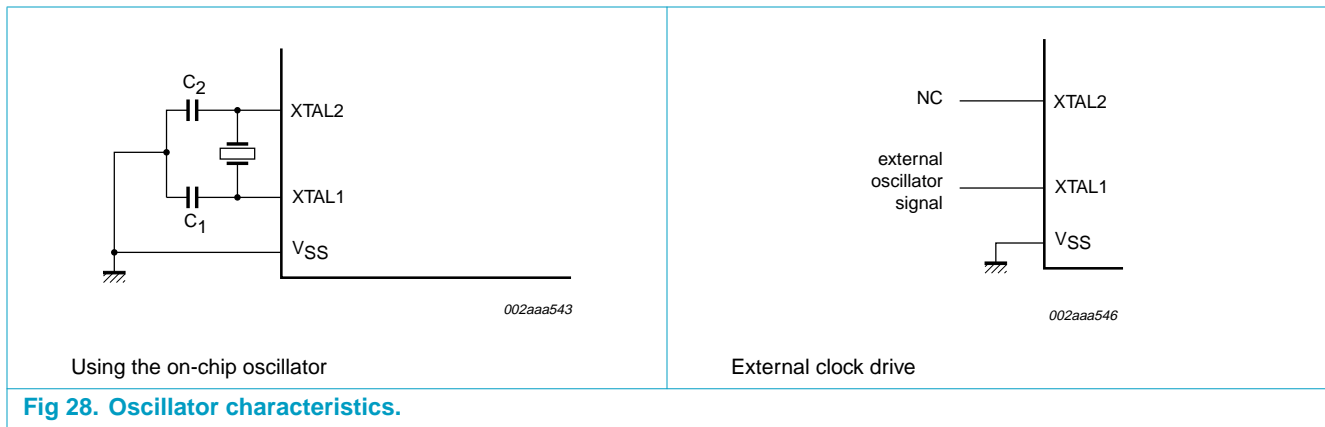


Fig 28. Oscillator characteristics.

Table 60: Clock doubling features

Device	Standard mode (x1)		Clock double mode (x2)	
	Clocks per machine cycle	Max. external clock frequency (MHz)	Clocks per machine cycle	Max. external clock frequency (MHz)
P89LV51RD2	12	33	6	16

Table 61: FST - Flash status register (address B6) bit allocation

Not Bit addressable; Reset value: xxxxx0xxB

Bit	7	6	5	4	3	2	1	0
Symbol	-	SB	-	-	EDC	-	-	-

Table 62: FST - Flash status register (address B6) bit description

Bit	Symbol	Description
7	-	Reserved for future use. Should be set to '0' by user programs.
6	SB	Security bit.
5 to 4	-	Reserved for future use. Should be set to '0' by user programs.
3	EDC	Enable double clock.
2 to 0	-	Reserved for future use. Should be set to '0' by user programs.

## 8. Limiting values

**Table 63: Limiting values**

In accordance with the Absolute Maximum Rating System (IEC 60134).

Parameters are valid over operating temperature range unless otherwise specified. All voltages are with respect to  $V_{SS}$  unless otherwise noted.

Symbol	Parameter	Conditions	Min	Max	Unit
$T_{amb(bias)}$	operating bias ambient temperature		-55	+125	°C
$T_{stg}$	storage temperature range		-65	+150	°C
$V_{\bar{E}A}$	voltage on $\bar{E}A$ pin to $V_{SS}$		-0.5	14	V
$V_n$	DC voltage on any pin to ground potential		-0.5	$V_{DD} + 0.5$	V
$V_{it}$	transient voltage (<20 ns) on any other pin to $V_{SS}$		-1.0	$V_{DD} + 1.0$	V
$I_{OL(I/O)}$	maximum $I_{OL}$ per I/O pins P1.5, P1.6, P1.7		-	20	mA
$I_{OL(I/O)}$	maximum $I_{OL}$ per I/O for all other pins		-	15	mA
$P_{tot(pack)}$	total power dissipation per package	$T_{amb} = 25\text{ °C}$	-	1.5	W
	through hole lead soldering temperature	10 seconds	-	300	°C
	surface mount lead soldering temperature	3 seconds	-	240	°C
	output short circuit current		[1] -	50	mA

[1] Outputs shorted for no more than one second. No more than one output shorted at a time. (Based on package heat transfer limitations, not device power consumption.)

## 9. Recommended operating conditions

**Table 64: Operating range**

Symbol	Description	Min	Max	Unit
$T_{amb}$	ambient temperature under bias			
	commercial	0	+70	°C
	industrial	-40	+85	°C
$V_{DD}$	supply voltage	2.7	3.6	V
$f_{osc}$	oscillator frequency	0	33	MHz
	oscillator frequency for in-application programming	0.25	33	MHz

**Table 65: Reliability characteristics**

Symbol	Parameter	Minimum specification	Units	Test method
$N_{END}^{[1]}$	endurance	10,000	cycles	JEDEC Standard A117
$T_{DR}^{[1]}$	data retention	100	years	JEDEC Standard A103
$I_{LTH}^{[1]}$	latch up	$100 + I_{DD}$	mA	JEDEC Standard 78

[1] This parameter is measured only for initial qualification and after a design or process change that could affect this parameter.

**Table 66: AC conditions of test<sup>[1]</sup>**

Input rise/fall time	10 ns
Output load	$C_L = 100$ pf

[1] See Figure 35 and Figure 37.

**Table 67: Recommended system power-up timings**

Symbol	Parameter	Minimum	Unit
$T_{PU-READ}^{[1]}$	Power-up to read operation	100	$\mu$ s
$T_{PU-WRITE}^{[1]}$	Power-up to write operation	100	$\mu$ s

[1] This parameter is measured only for initial qualification and after a design or process change that could affect this parameter.

**Table 68: Pin impedance**

( $V_{DD} = 3.3$  V,  $T_{amb} = 25$  °C,  $f = 1$  MHz, other pins open)

Parameter	Description	Test condition	Maximum	Unit
$C_{I/O}^{[1]}$	I/O pin capacitance	$V_{I/O} = 0$ V	15	pF
$C_{IN}^{[1]}$	input capacitance	$V_{IN} = 0$ V	12	pF
$L_{PIN}$	pin inductance		20	nH

[1] This parameter is measured only for initial qualification and after a design or process change that could affect this parameter.

## 10. Static characteristics

**Table 69: DC electrical characteristics**

$T_{amb} = 0\text{ }^{\circ}\text{C to }+70\text{ }^{\circ}\text{C or }-40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}; V_{DD} = 2.7\text{ V to }3.6\text{ V}; V_{SS} = 0\text{ V}$

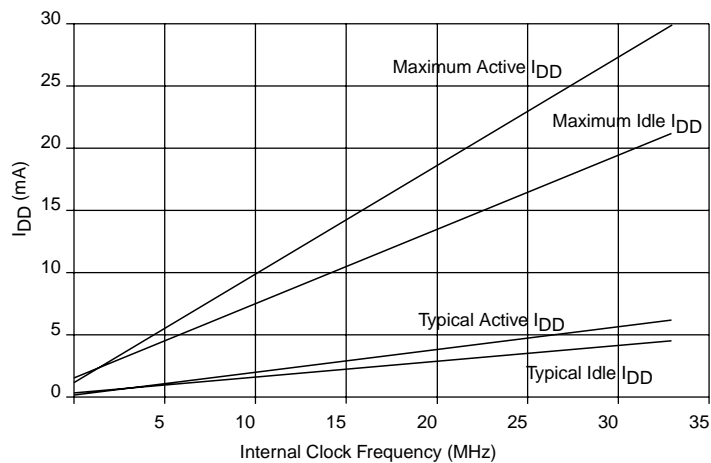
Symbol	Parameter	Conditions	Min	Max	Unit	
$V_{IL}$	LOW-level input voltage	$2.7\text{ V} < V_{DD} < 3.6\text{ V}$	-0.5	0.7	V	
$V_{IH}$	HIGH-level input voltage	$2.7\text{ V} < V_{DD} < 3.6\text{ V}$	$0.2V_{DD} + 0.9$	$V_{DD} + 0.5$	V	
$V_{IH1}$	HIGH-level input voltage (XTAL1, RST)	$2.7\text{ V} < V_{DD} < 3.6\text{ V}$	$0.7V_{DD}$	$V_{DD} + 0.5$	V	
$V_{OL}$	LOW-level output voltage (ports 1.5, 1.6, 1.7)	$V_{DD} = 2.7\text{ V}; I_{OL} = 16\text{ mA}$	-	1.0	V	
$V_{OL}$	LOW-level output voltage (ports 1, 2, 3) <sup>[1]</sup>	$V_{DD} = 2.7\text{ V}$				
		$I_{OL} = 100\text{ }\mu\text{A}$	-	0.3	V	
		$I_{OL} = 1.6\text{ mA}$	-	0.45	V	
		$I_{OL} = 3.5\text{ mA}$	-	1.0	V	
$V_{OL1}$	LOW-level output voltage (Port 0, ALE, PSEN) <sup>[1][3]</sup>	$V_{DD} = 2.7\text{ V}$				
		$I_{OL} = 200\text{ }\mu\text{A}$	-	0.3	V	
		$I_{OL} = 3.2\text{ mA}$	-	0.45	V	
$V_{OH}$	HIGH-level output voltage (ports 1, 2, 3, ALE, PSEN) <sup>[4]</sup>	$V_{DD} = 2.7\text{ V}$				
		$I_{OH} = -10\text{ }\mu\text{A}$	$V_{DD} - 0.3$	-	V	
		$I_{OH} = -30\text{ }\mu\text{A}$	$V_{DD} - 0.7$	-	V	
		$I_{OH} = -60\text{ }\mu\text{A}$	$V_{DD} - 1.5$	-	V	
$V_{OH1}$	HIGH-level output voltage (Port 0 in External Bus Mode) <sup>[4]</sup>	$V_{DD} = 2.7\text{ V}$				
		$I_{OH} = -200\text{ }\mu\text{A}$	$V_{DD} - 0.3$	-	V	
		$I_{OH} = -3.2\text{ mA}$	$V_{DD} - 0.7$	-	V	
$V_{BOD}$	brown-out detection voltage		2.35	2.55	V	
$I_{IL}$	logic 0 input current (ports 1, 2, 3)	$V_{IN} = 0.4\text{ V}$	-	-75	$\mu\text{A}$	
$I_{TL}$	logic 1-to-0 transition current (ports 1, 2, 3) <sup>[5]</sup>	$V_{IN} = 2\text{ V}$	-	-650	$\mu\text{A}$	
$I_{LI}$	input leakage current (Port 0)	$0.45\text{ V} < V_{IN} < V_{DD} - 0.3\text{ V}$	-	$\pm 10$	$\mu\text{A}$	
$R_{RST}$	RST pull-down resistor		-	225	$\text{k}\Omega$	
$C_{IO}$	pin capacitance <sup>[6]</sup>	@ 1 MHz, $T_{amb} = 25\text{ }^{\circ}\text{C}$	-	15	pF	
$I_{DD}$	power supply current	active mode	@ 12 MHz	-	11.5	mA
			@ 33 MHz	-	30	mA
	idle mode	@ 12 MHz	-	8.5	mA	
		@ 33 MHz	-	21	mA	
	Power-down mode (min. $V_{DD} = 2\text{ V}$ )	$T_{amb} = 0\text{ }^{\circ}\text{C to }+70\text{ }^{\circ}\text{C}$	-	45	$\mu\text{A}$	
		$T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$	-	55	$\mu\text{A}$	

[1] Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:

- Maximum  $I_{OL}$  per 8-bit port: 26 mA
- Maximum  $I_{OL}$  total for all outputs: 71 mA
- If  $I_{OL}$  exceeds the test condition,  $V_{OH}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.



- [2] Capacitive loading on Ports 0 and 2 may cause spurious noise to be superimposed on the  $V_{OLS}$  of ALE and Ports 1 and 3. The noise due to external bus capacitance discharging into the Port 0 and 2 pins when the 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.
- [3] Load capacitance for Port 0, ALE and  $\overline{PSEN}$  = 100 pF, load capacitance for all other outputs = 80 pF.
- [4] Capacitive loading on Ports 0 and 2 may cause the  $V_{OH}$  on ALE and  $\overline{PSEN}$  to momentarily fall below the  $V_{DD} - 0.7$  specification when the address bits are stabilizing.
- [5] 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.
- [6] Pin capacitance is characterized but not tested.  $\overline{EA}$  = 25 pF (max).



002aaa547

Fig 29.  $I_{DD}$  vs. frequency.

## 11. Dynamic characteristics

**Table 70: AC characteristics**

Over operating conditions: load capacitance for Port 0, ALE, and  $\overline{\text{PSEN}}$  = 100 pF; load capacitance for all other outputs = 80 pF

$T_{amb} = 0^\circ\text{C}$  to  $+70^\circ\text{C}$  or  $-40^\circ\text{C}$  to  $+85^\circ\text{C}$ ;  $V_{DD} = 2.7\text{ V}$  to  $3.6\text{ V}$  @ 33 MHz;  $V_{SS} = 0\text{ V}$

Symbol	Parameter	33 MHz (X1 mode) 16 MHz (X2 mode) <sup>[1]</sup>		Variable		Unit
		Min	Max	Min	Max	
$1/T_{\text{CLCL}}$	X1 Mode oscillator frequency	0	33	0	33	MHz
$1/2T_{\text{CLCL}}$	X2 Mode oscillator frequency	0	16	0	16	MHz
$t_{\text{LHLL}}$	ALE pulse width	46	-	$2T_{\text{CLCL}} - 15$	-	ns
$t_{\text{AVLL}}$	address valid to ALE LOW	5	-	$T_{\text{CLCL}} - 25$	-	ns
$t_{\text{LLAX}}$	address hold after ALE LOW	5	-	$T_{\text{CLCL}} - 25$	-	ns
$t_{\text{LLIV}}$	ALE LOW to valid instruction in	-	56	-	$4T_{\text{CLCL}} - 65$	ns
$t_{\text{LLPL}}$	ALE LOW to $\overline{\text{PSEN}}$ LOW	5	-	$T_{\text{CLCL}} - 25$	-	ns
$t_{\text{PLPH}}$	$\overline{\text{PSEN}}$ pulse width	66	-	$T_{\text{CLCL}} - 25$	-	ns
$t_{\text{PLIV}}$	$\overline{\text{PSEN}}$ LOW to valid instruction in	-	35	-	$3T_{\text{CLCL}} - 55$	ns
$t_{\text{PXIX}}$	input instruction hold after $\overline{\text{PSEN}}$	-	-	0	-	ns
$t_{\text{PXIZ}}$	input instruction float after $\overline{\text{PSEN}}$	-	25	-	$T_{\text{CLCL}} - 5$	ns
$t_{\text{PXAV}}$	$\overline{\text{PSEN}}$ to address valid	22	-	$T_{\text{CLCL}} - 8$	-	ns
$t_{\text{AVIV}}$	address to valid instruction in	-	72	-	$5T_{\text{CLCL}} - 80$	ns
$t_{\text{PLAZ}}$	$\overline{\text{PSEN}}$ LOW to address float	-	10	-	10	ns
$t_{\text{RLRH}}$	$\overline{\text{RD}}$ pulse width	142	-	$6T_{\text{CLCL}} - 40$	-	ns
$t_{\text{WLWH}}$	write pulse width ( $\overline{\text{WR}}$ )	142	-	$6T_{\text{CLCL}} - 40$	-	ns
$t_{\text{RLDV}}$	$\overline{\text{RD}}$ LOW to valid data in	-	62	-	$5T_{\text{CLCL}} - 90$	ns
$t_{\text{RHDX}}$	data hold after $\overline{\text{RD}}$	0	-	0	-	ns
$t_{\text{RHDZ}}$	data float after $\overline{\text{RD}}$	-	36	-	$2T_{\text{CLCL}} - 25$	ns
$t_{\text{LLDV}}$	ALE LOW to valid data in	-	152	-	$8T_{\text{CLCL}} - 90$	ns
$t_{\text{AVDV}}$	address to valid data in	-	183	-	$9T_{\text{CLCL}} - 90$	ns
$t_{\text{LLWL}}$	ALE LOW to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ LOW	66	116	$3T_{\text{CLCL}} - 25$	$3T_{\text{CLCL}} + 25$	ns
$t_{\text{AVWL}}$	address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ LOW	46	-	$4T_{\text{CLCL}} - 75$	-	ns
$t_{\text{WHQX}}$	data hold after $\overline{\text{WR}}$	3	-	$T_{\text{CLCL}} - 27$	-	ns
$t_{\text{QVWH}}$	data valid to $\overline{\text{WR}}$ HIGH	142	-	$7T_{\text{CLCL}} - 70$	-	ns
$t_{\text{RLAZ}}$	$\overline{\text{RD}}$ LOW to address float	-	0	-	0	ns
$t_{\text{WHLH}}$	$\overline{\text{RD}}$ to $\overline{\text{WR}}$ HIGH to ALE HIGH	5	55	$T_{\text{CLCL}} - 25$	$T_{\text{CLCL}} + 25$	ns

[1] Calculated values are for X1 mode only.

11.1 Explanation of symbols

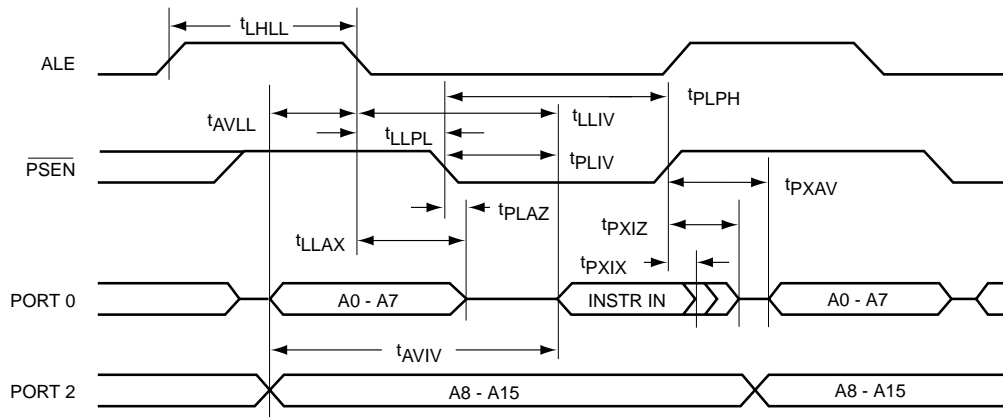
Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

- A — Address
- C — Clock
- D — Input data
- H — Logic level HIGH
- I — Instruction (program memory contents)
- L — Logic level LOW or ALE
- P —  $\overline{\text{PSEN}}$
- Q — Output data
- R —  $\overline{\text{RD}}$  signal
- T — Time
- V — Valid
- W —  $\overline{\text{WR}}$  signal
- X — No longer a valid logic level
- Z — High impedance (Float)

Example:

$T_{AVLL}$  = Time from Address Valid to ALE LOW

$T_{LLPL}$  = Time from ALE LOW to  $\overline{\text{PSEN}}$  LOW



002aaa548

Fig 30. External program memory read cycle.

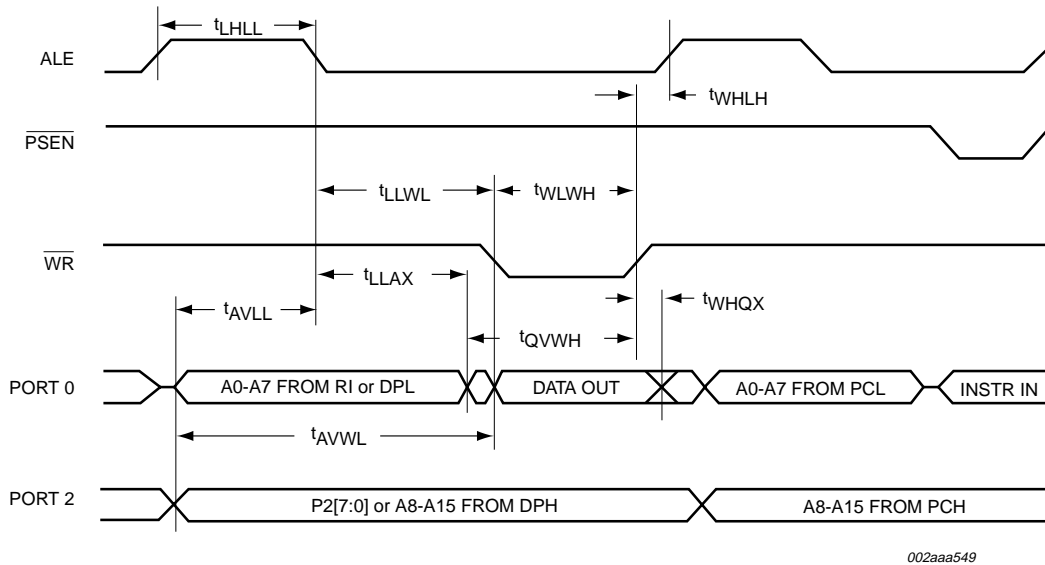


Fig 31. External data memory read cycle.

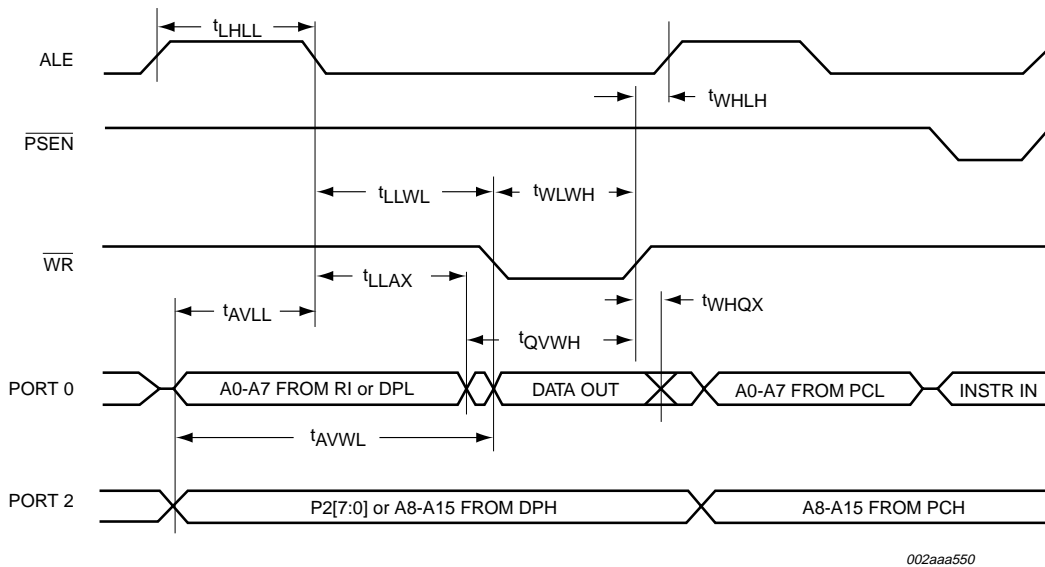


Fig 32. External data memory write cycle.

Table 71: External clock drive

Symbol	Parameter	Oscillator				Unit
		12 MHz		Variable		
		Min	Max	Min	Max	
$1/T_{CLCL}$	oscillator frequency	-	-	0	33	MHz
$t_{CLCL}$		83	-	-	-	ns
$t_{CHCX}$	high time	-	-	$0.35T_{CLCL}$	$0.65T_{CLCL}$	ns

Table 71: External clock drive...continued

Symbol	Parameter	Oscillator				Unit
		12 MHz		Variable		
		Min	Max	Min	Max	
t <sub>CLCX</sub>	low time	-	-	0.35T <sub>CLCL</sub>	0.65T <sub>CLCL</sub>	ns
t <sub>CLCH</sub>	rise time	-	20	-	-	ns
t <sub>CHCL</sub>	fall time	-	20	-	-	ns

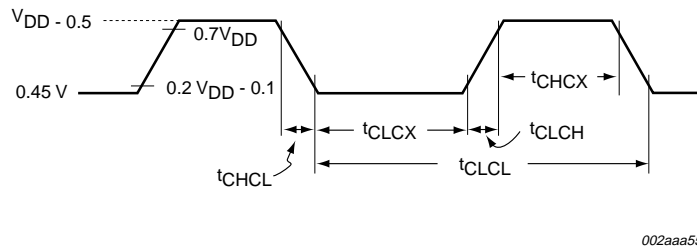


Fig 33. External clock drive waveform.

Table 72: Serial port timing

Symbol	Parameter	Oscillator				Unit
		12 MHz		Variable		
		Min	Max	Min	Max	
t <sub>XLXL</sub>	serial port clock cycle time	1.0	-	12t <sub>CLCL</sub>	-	μs
t <sub>QVXH</sub>	output data set-up to clock rising edge	700	-	10t <sub>CLCL</sub> - 133	-	ns
t <sub>XHQX</sub>	output data hold after clock rising edge	50	-	2t <sub>CLCL</sub> - 117	-	ns
t <sub>XHDX</sub>	input data hold after clock rising edge	0	-	0	-	ns
t <sub>XHDV</sub>	clock rising edge to input data valid	-	700	-	10t <sub>CLCL</sub> - 133	ns

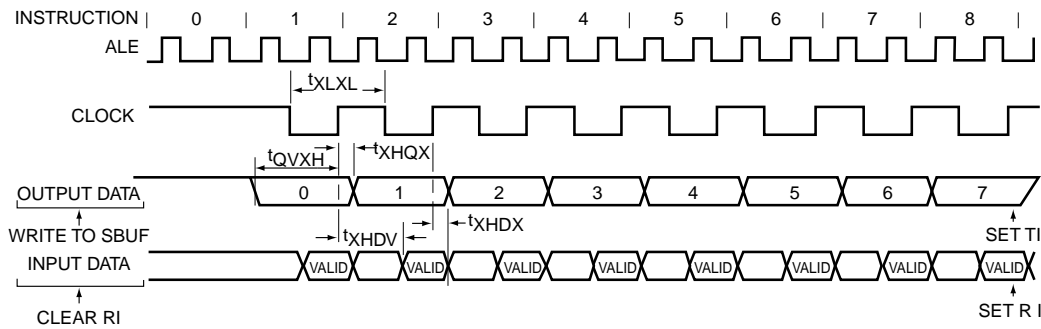
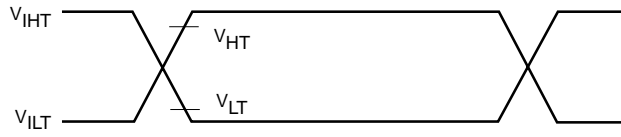


Fig 34. Shift register mode timing waveforms.

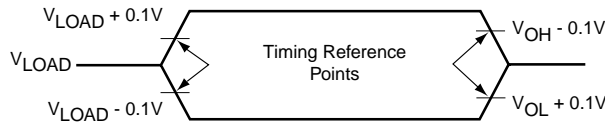


Note:  $V_{HT}$ -  $V_{HIGH}$  Test  
 $V_{LT}$ -  $V_{LOW}$  Test  
 $V_{IHT}$ - $V_{INPUT}$  HIGH Test  
 $V_{ILT}$ -  $V_{INPUT}$  LOW Test

002aaa553

AC inputs during testing are driven at  $V_{IHT}$  ( $V_{DD} - 0.5$  V) for logic 1 and  $V_{ILT}$  (0.45 V) for a logic 0. Measurement reference points for inputs and outputs are at  $V_{HT}$  ( $0.2V_{DD} + 0.9$ ) and  $V_{LT}$  ( $0.2V_{DD} - 0.1$ )

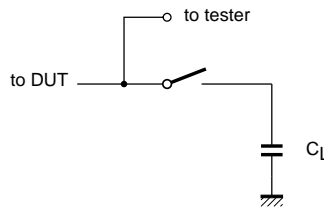
**Fig 35. AC testing input/output test waveform.**



002aaa554

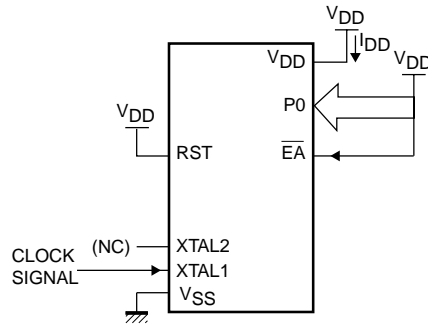
For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded  $V_{OH}/V_{OL}$  level occurs.  $I_{OH}/I_{OL} = \pm 20$  mA.

**Fig 36. Float waveform.**



002aaa555

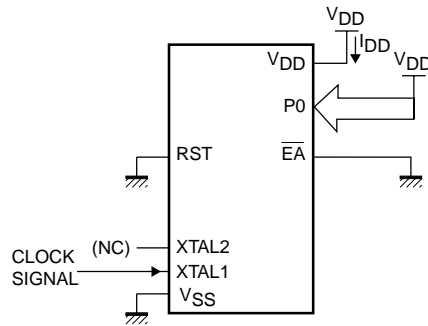
**Fig 37. Test load example.**



002aaa556

All other pins disconnected

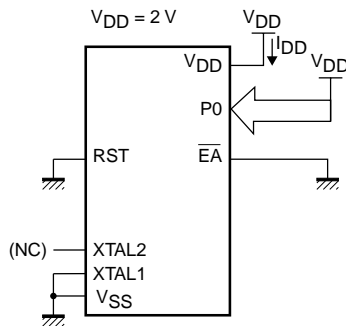
**Fig 38.  $I_{DD}$  test condition, active mode.**



002aaa557

All other pins disconnected

**Fig 39.  $I_{DD}$  test condition, idle mode.**



002aaa558

All other pins disconnected

**Fig 40.  $I_{DD}$  test condition, Power-down mode.**

12. Package outline

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

SOT129-1

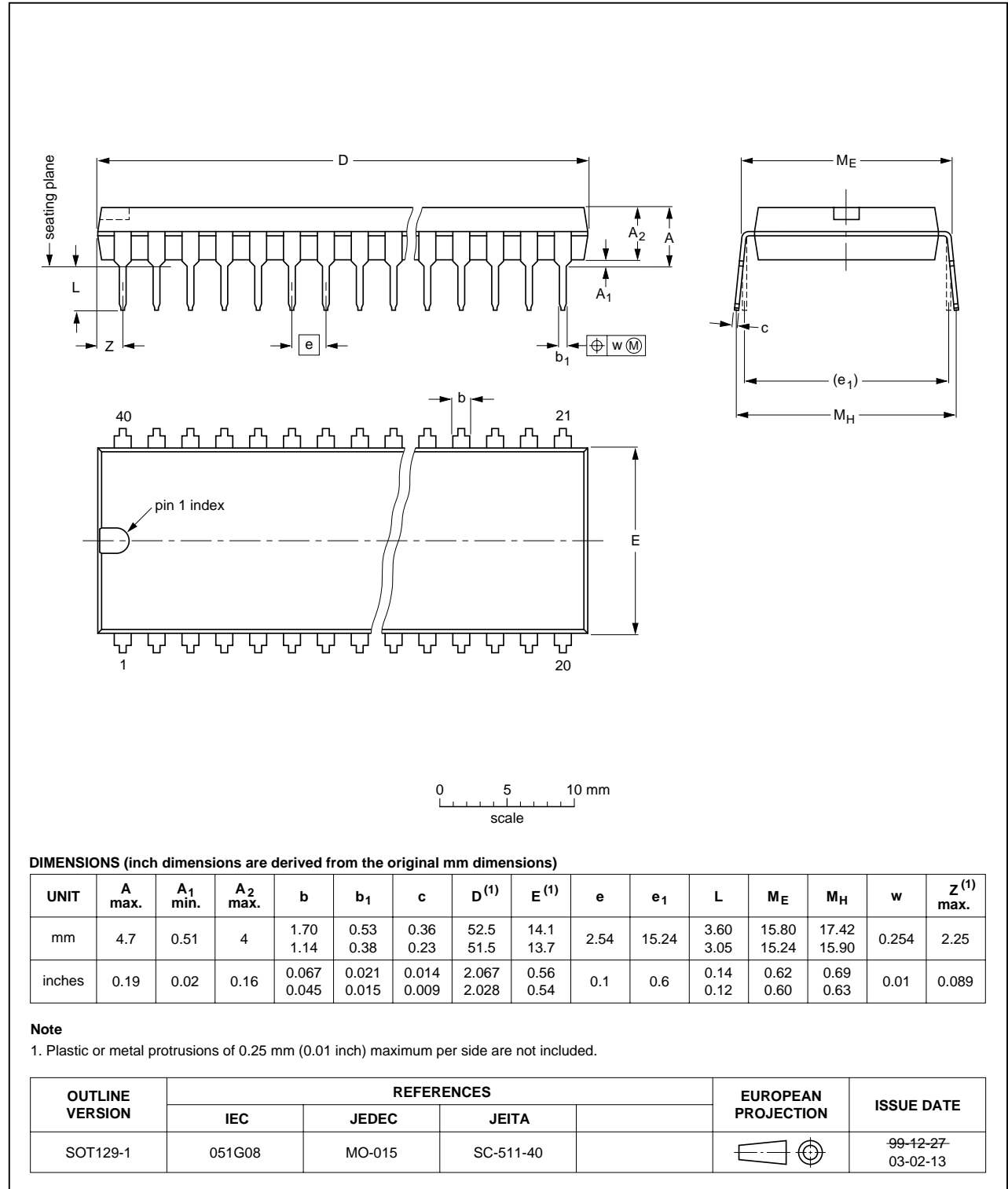


Fig 41. SOT129-1 (DIP40) package outline.



TQFP44: plastic thin quad flat package; 44 leads; body 10 x 10 x 1.0 mm

SOT376-1

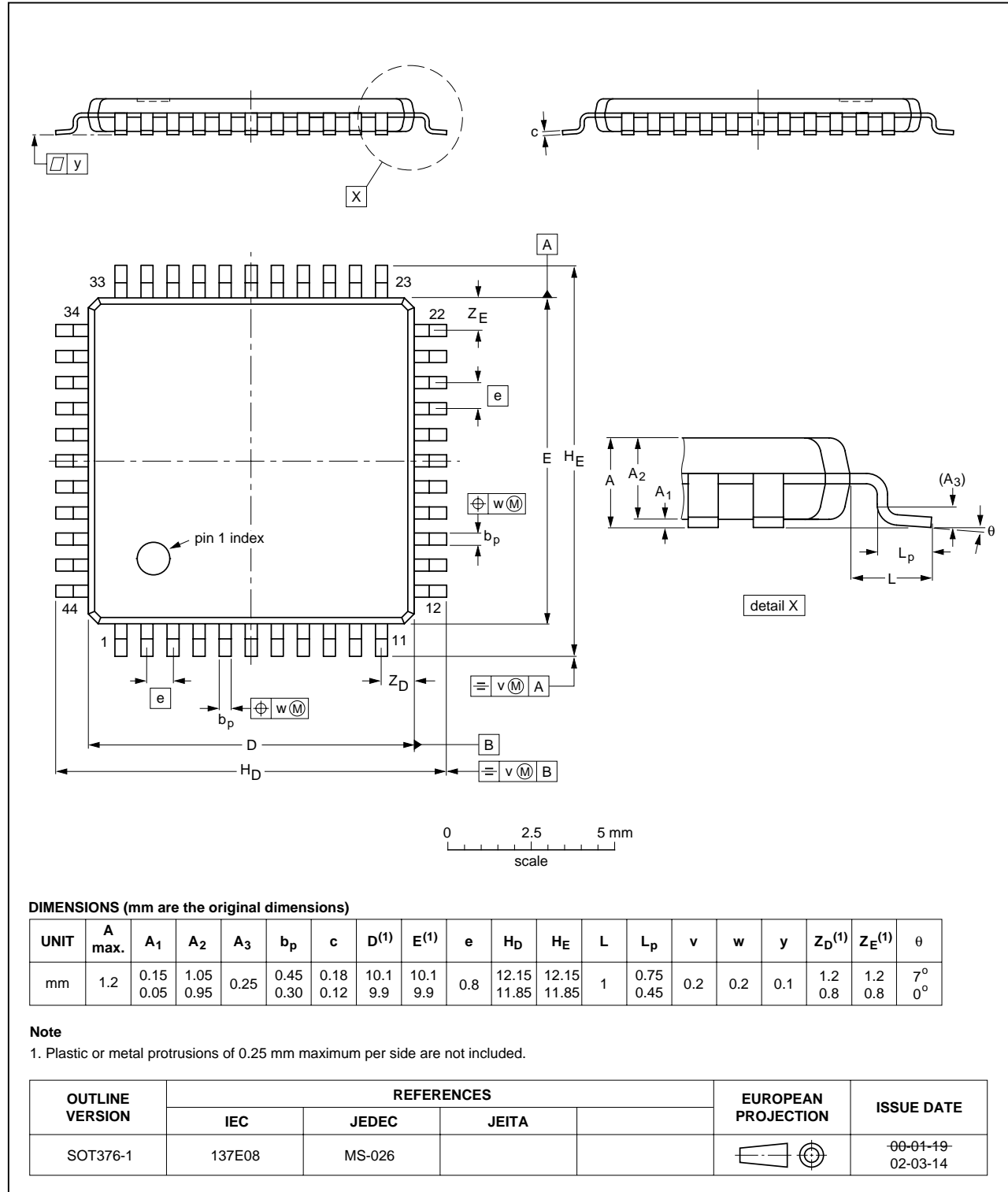


Fig 42. SOT376-1 (TQFP44) package outline.

PLCC44: plastic leaded chip carrier; 44 leads

SOT187-2

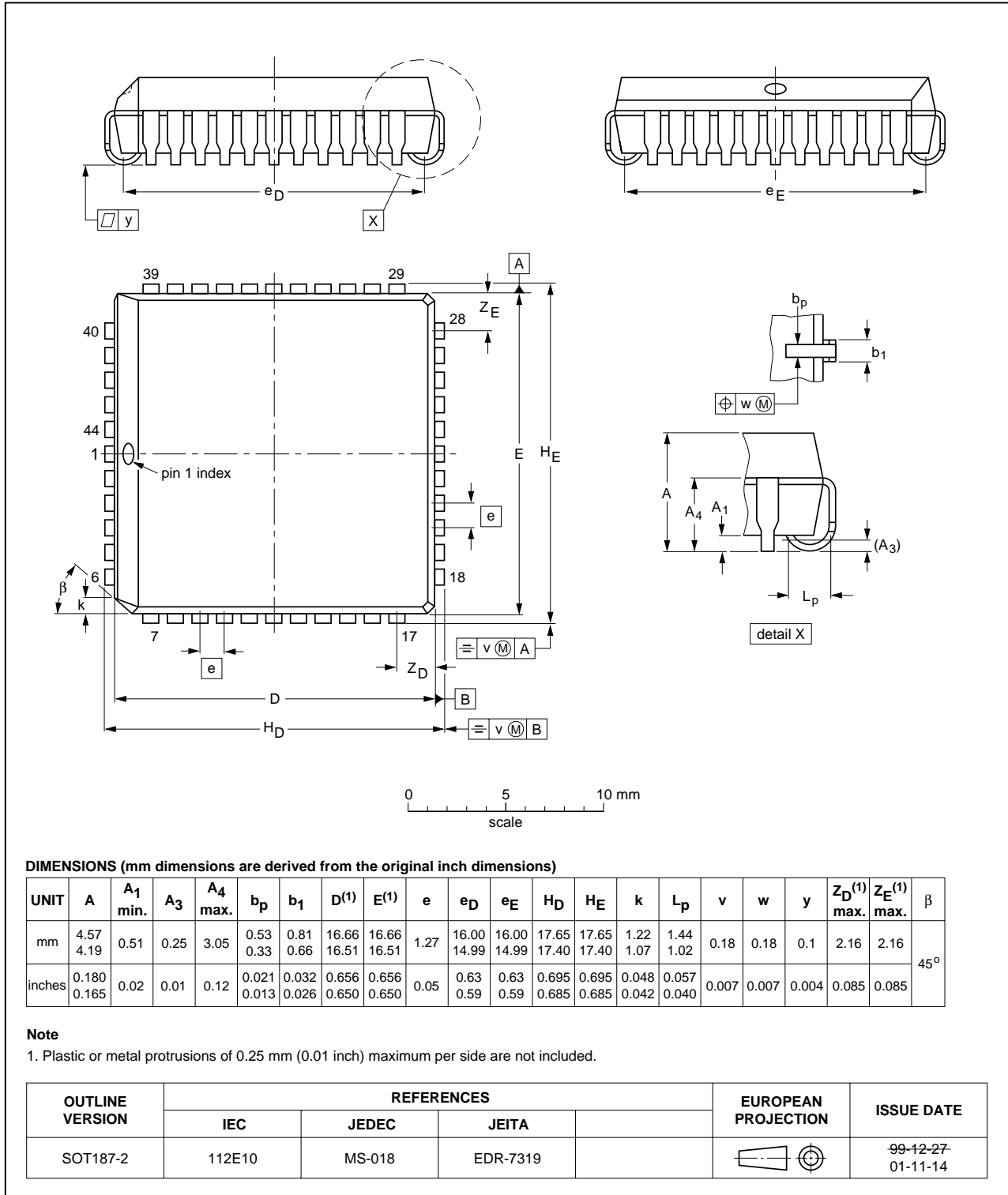


Fig 43. SOT187-2 (PLCC44) package outline.

## 13. Revision history

Table 73: Revision history

Rev	Date	CPCN	Description
04	20041202	-	<b>Product data (9397 750 14342)</b> Modifications: <ul style="list-style-type: none"><li>• Added devices P89LV51RB2 and P89LV51RC2.</li></ul>
03	20041011	-	<b>Product data (9397 750 14101)</b>
02	20031113	-	<b>Product data (9397 750 11783)</b>
01	20030630	-	<b>Product data (9397 750 11669); ECN 853-2432 30075 dated 27 June 2003</b>

## 14. Data sheet status

Level	Data sheet status <sup>[1]</sup>	Product status <sup>[2][3]</sup>	Definition
I	Objective data	Development	This data sheet contains data from the objective specification for product development. Philips Semiconductors reserves the right to change the specification in any manner without notice.
II	Preliminary data	Qualification	This data sheet contains data from the preliminary specification. Supplementary data will be published at a later date. Philips Semiconductors reserves the right to change the specification without notice, in order to improve the design and supply the best possible product.
III	Product data	Production	This data sheet contains data from the product specification. Philips Semiconductors reserves the right to make changes at any time in order to improve the design, manufacturing and supply. Relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN).

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

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

[3] For data sheets describing multiple type numbers, the highest-level product status determines the data sheet status.

## 15. Definitions

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

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

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

## 16. 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 in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

## Contact information

For additional information, please visit <http://www.semiconductors.philips.com>.

For sales office addresses, send e-mail to: [sales.addresses@www.semiconductors.philips.com](mailto:sales.addresses@www.semiconductors.philips.com).

Fax: +31 40 27 24825

## Contents

<b>1</b>	<b>General description</b> . . . . .	<b>1</b>	7.6.2	SPI description . . . . .	42
<b>2</b>	<b>Features</b> . . . . .	<b>1</b>	7.7	Watchdog timer . . . . .	45
<b>3</b>	<b>Ordering information</b> . . . . .	<b>2</b>	7.8	Programmable Counter Array (PCA) . . . . .	46
3.1	Ordering options . . . . .	2	7.8.1	PCA capture mode . . . . .	50
<b>4</b>	<b>Block diagram</b> . . . . .	<b>3</b>	7.8.2	16-bit software timer mode . . . . .	51
<b>5</b>	<b>Pinning information</b> . . . . .	<b>4</b>	7.8.3	High speed output mode . . . . .	52
5.1	Pinning . . . . .	4	7.8.4	Pulse width modulator mode . . . . .	53
5.2	Pin description . . . . .	7	7.8.5	PCA Watchdog timer . . . . .	53
<b>6</b>	<b>Special function registers</b> . . . . .	<b>10</b>	7.9	Security Bit . . . . .	54
<b>7</b>	<b>Functional description</b> . . . . .	<b>14</b>	7.10	Interrupt priority and polling sequence . . . . .	54
7.1	Memory organization . . . . .	14	7.11	Power-saving modes . . . . .	58
7.1.1	Flash program memory bank selection . . . . .	14	7.11.1	Idle mode . . . . .	59
7.1.2	Power-on reset code execution . . . . .	14	7.11.2	Power-down mode . . . . .	59
7.1.3	Software reset . . . . .	15	7.12	System clock and clock options . . . . .	60
7.1.4	Brown-out detect reset . . . . .	16	7.12.1	Clock Input Options and Recommended Capacitor Values for Oscillator . . . . .	60
7.1.5	Watchdog reset . . . . .	16	7.12.2	Clock doubling option . . . . .	61
7.1.6	Data RAM memory . . . . .	16	<b>8</b>	<b>Limiting values</b> . . . . .	<b>62</b>
7.1.7	Expanded data RAM addressing . . . . .	16	<b>9</b>	<b>Recommended operating conditions</b> . . . . .	<b>62</b>
7.1.8	Dual data pointers . . . . .	19	<b>10</b>	<b>Static characteristics</b> . . . . .	<b>64</b>
7.2	Flash memory In-Application Programming . . . . .	20	<b>11</b>	<b>Dynamic characteristics</b> . . . . .	<b>66</b>
7.2.1	Flash organization . . . . .	20	11.1	Explanation of symbols . . . . .	67
7.2.2	Boot block (Block 1) . . . . .	20	<b>12</b>	<b>Package outline</b> . . . . .	<b>72</b>
7.2.3	In-System Programming (ISP) . . . . .	20	<b>13</b>	<b>Revision history</b> . . . . .	<b>75</b>
7.2.4	Using the In-System Programming . . . . .	21	<b>14</b>	<b>Data sheet status</b> . . . . .	<b>76</b>
7.2.5	Using the serial number . . . . .	25	<b>15</b>	<b>Definitions</b> . . . . .	<b>76</b>
7.2.6	In-Application Programming method . . . . .	25	<b>16</b>	<b>Disclaimers</b> . . . . .	<b>76</b>
7.3	Timers/counters 0 and 1 . . . . .	27			
7.3.1	Mode 0 . . . . .	29			
7.3.2	Mode 1 . . . . .	29			
7.3.3	Mode 2 . . . . .	30			
7.3.4	Mode 3 . . . . .	30			
7.4	Timer 2 . . . . .	31			
7.4.1	Capture mode . . . . .	32			
7.4.2	Auto-reload mode (up or down counter) . . . . .	33			
7.4.3	Programmable clock-out . . . . .	35			
7.4.4	Baud rate generator mode . . . . .	36			
7.4.5	Summary of baud rate equations . . . . .	37			
7.5	UARTs . . . . .	37			
7.5.1	Mode 0 . . . . .	38			
7.5.2	Mode 1 . . . . .	38			
7.5.3	Mode 2 . . . . .	38			
7.5.4	Mode 3 . . . . .	38			
7.5.5	Framing error . . . . .	39			
7.5.6	More about UART mode 1 . . . . .	39			
7.5.7	More about UART modes 2 and 3 . . . . .	40			
7.5.8	Multiprocessor communications . . . . .	40			
7.5.9	Automatic address recognition . . . . .	40			
7.6	Serial peripheral interface . . . . .	42			
7.6.1	SPI features . . . . .	42			



# PHILIPS

*Let's make things better.*