



Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	40MHz
Connectivity	SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	32
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-DIP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/p89v51rd2fn-112

- Brownout 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	
P89V51RB2FA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2
P89V51RB2FN	DIP40	plastic dual in-line package; 40 leads (600 mil)	SOT129-1
P89V51RB2BBC	TQFP44	plastic thin quad flat package; 44 leads; body 10 × 10 × 1.0 mm	SOT376-1
P89V51RC2FA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2
P89V51RC2FBC	TQFP44	plastic thin quad flat package; 44 leads; body 10 × 10 × 1.0 mm	SOT376-1
P89V51RC2FN	DIP40	plastic dual in-line package; 40 leads (600 mil)	SOT129-1
P89V51RD2FA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2
P89V51RD2FBC	TQFP44	plastic thin quad flat package; 44 leads; body 10 × 10 × 1.0 mm	SOT376-1
P89V51RD2BN	DIP40	plastic dual in-line package; 40 leads (600 mil)	SOT129-1
P89V51RD2FN	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
P89V51RB2FA	16 kB	−40 °C to +85 °C	0 MHz to 40 MHz
P89V51RB2FN	16 kB	−40 °C to +85 °C	
P89V51RB2BBC	16 kB	0 °C to +70 °C	
P89V51RC2FA	32 kB	−40 °C to +85 °C	
P89V51RC2FBC	32 kB	−40 °C to +85 °C	
P89V51RC2FN	32 kB	−40 °C to +85 °C	
P89V51RD2FA	64 kB	−40 °C to +85 °C	
P89V51RD2FBC	64 kB	−40 °C to +85 °C	
P89V51RD2BN	64 kB	0 °C to +70 °C	
P89V51RD2FN	64 kB	−40 °C to +85 °C	

5. Pinning information

5.1 Pinning

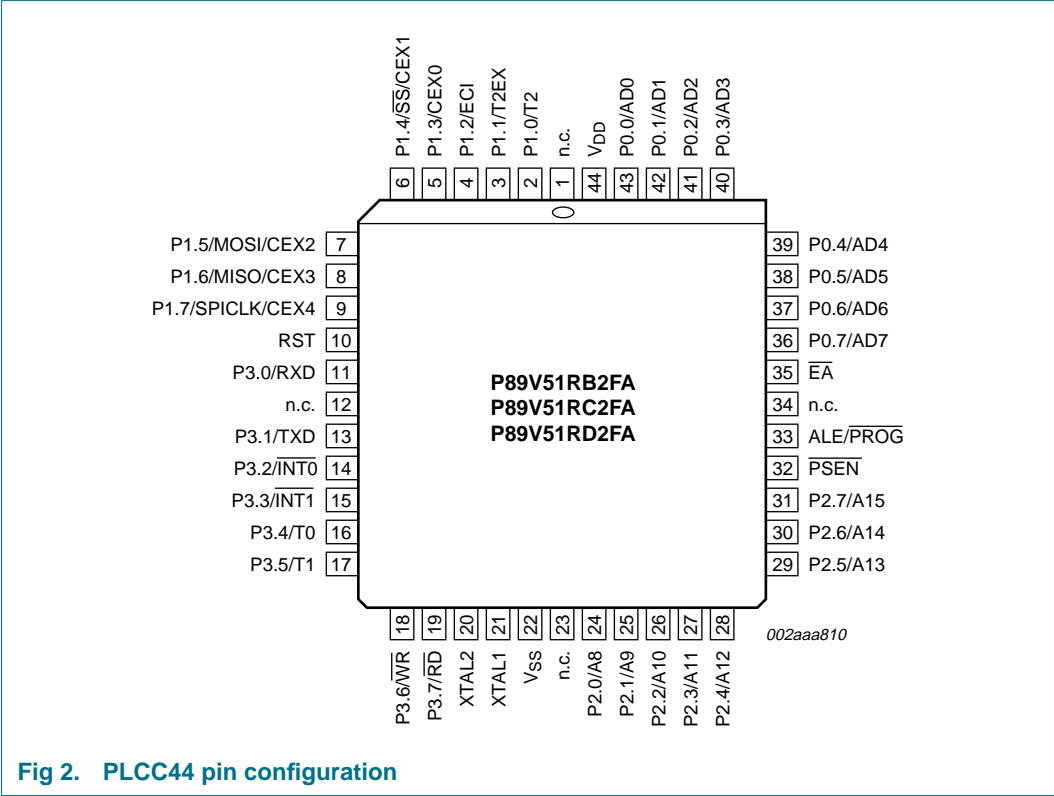


Fig 2. PLCC44 pin configuration

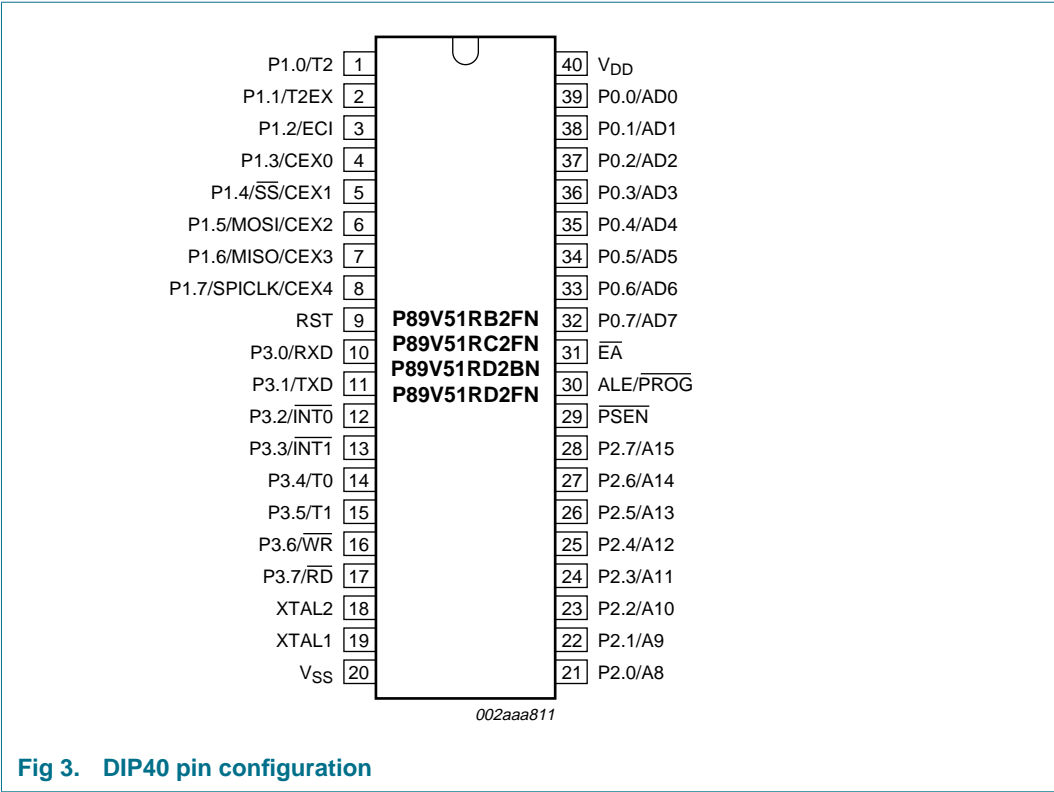


Fig 3. DIP40 pin configuration

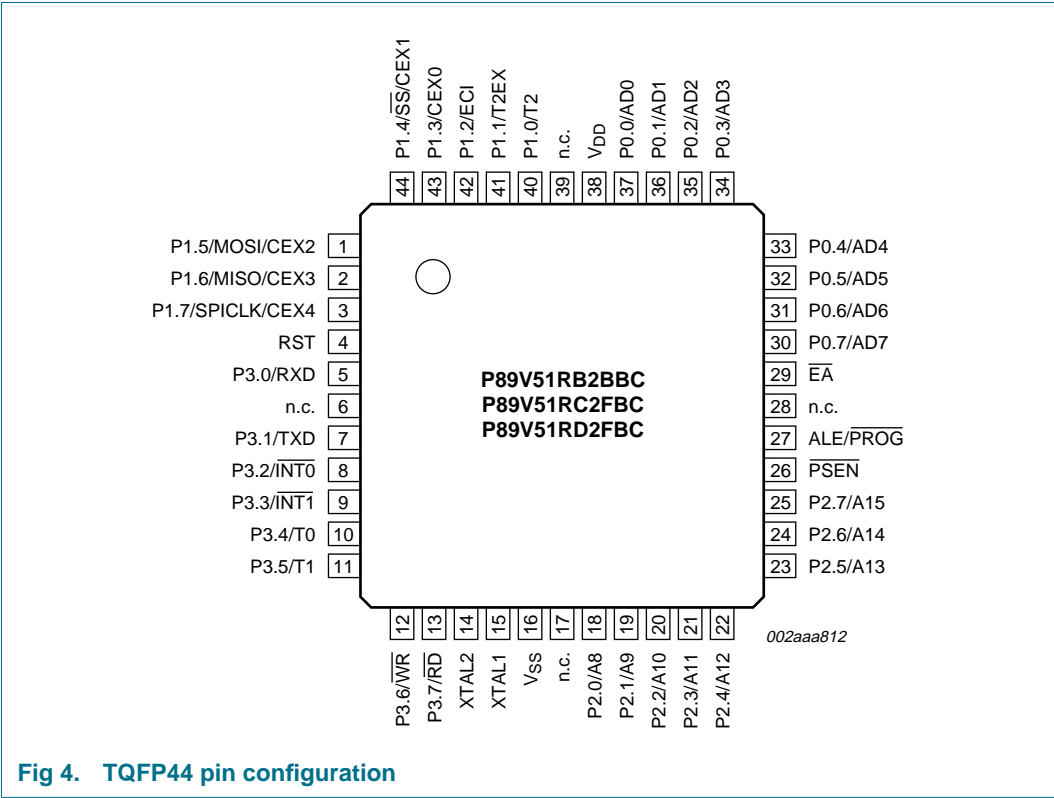


Fig 4. TQFP44 pin configuration

Table 3. P89V51RB2/RC2/RD2 pin description ...continued

Symbol	Pin			Type	Description
	DIP40	TQFP44	PLCC44		
P1.2/ECI	3	42	4	I/O	P1.2 — Port 1 bit 2.
				I	ECI — External clock input. This signal is the external clock input for the PCA.
P1.3/CEX0	4	43	5	I/O	P1.3 — Port 1 bit 3.
				I/O	CEX0 — 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/ \overline{SS} /CEX1	5	44	6	I/O	P1.4 — Port 1 bit 4.
				I	\overline{SS} — Slave port select input for SPI.
				I/O	CEX1 — Capture/compare external I/O for PCA Module 1.
P1.5/MOSI/CEX2	6	1	7	I/O	P1.5 — Port 1 bit 5.
				I/O	MOSI — Master Output Slave Input for SPI.
				I/O	CEX2 — Capture/compare external I/O for PCA Module 2.
P1.6/MISO/CEX3	7	2	8	I/O	P1.6 — Port 1 bit 6.
				I/O	MISO — Master Input Slave Output for SPI.
				I/O	CEX3 — Capture/compare external I/O for PCA Module 3.
P1.7/SPICLK/CEX4	8	3	9	I/O	P1.7 — Port 1 bit 7.
				I/O	SPICLK — Serial clock input/output for SPI.
				I/O	CEX4 — Capture/compare external I/O for PCA Module 4.
P2.0 to P2.7				I/O with internal pull-up	Port 2: Port 2 is an 8-bit bidirectional 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.
P2.0/A8	21	18	24	I/O	P2.0 — Port 2 bit 0.
				O	A8 — Address bit 8.
P2.1/A9	22	19	25	I/O	P2.1 — Port 2 bit 1.
				O	A9 — Address bit 9.
P2.2/A10	23	20	26	I/O	P2.2 — Port 2 bit 2.
				O	A10 — Address bit 10.
P2.3/A11	24	21	27	I/O	P2.3 — Port 2 bit 3.
				O	A11 — Address bit 11.
P2.4/A12	25	22	28	I/O	P2.4 — Port 2 bit 4.
				O	A12 — Address bit 12.

Table 3. P89V51RB2/RC2/RD2 pin description ...continued

Symbol	Pin			Type	Description
	DIP40	TQFP44	PLCC44		
RST	9	4	10	I	Reset: While the oscillator is running, a HIGH logic state on this pin for two machine cycles will reset the device. If the $\overline{\text{PSEN}}$ 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.
$\overline{\text{EA}}$	31	29	35	I	External Access Enable: $\overline{\text{EA}}$ must be connected to V_{SS} in order to enable the device to fetch code from the external program memory. $\overline{\text{EA}}$ must be strapped to V_{DD} for internal program execution. The $\overline{\text{EA}}$ pin can tolerate a high voltage of 12 V.
ALE/ $\overline{\text{PROG}}$	30	27	33	I/O	Address Latch Enable: 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 ($\overline{\text{PROG}}$) for flash programming. Normally the ALE ^[1] is emitted at a constant rate of $\frac{1}{6}$ the crystal frequency ^[2] 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.
n.c.	-	6, 17, 28, 39	1, 12, 23, 34	I/O	not connected
XTAL1	19	15	21	I	Crystal 1: Input to the inverting oscillator amplifier and input to the internal clock generator circuits.
XTAL2	18	14	20	O	Crystal 2: Output from the inverting oscillator amplifier.
V_{DD}	40	38	44	I	Power supply
V_{SS}	20	16	22	I	Ground

[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 Ω to 50 k Ω to V_{DD} , e.g., for ALE pin.

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

6. Functional description

6.1 Special function registers

Remark: SFR 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 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 B (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:

```
MOV90H, #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 B of memory is physically located on the chip and logically occupies the first 768 B 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 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 B of internal RAM (lower 128 B and upper 128 B). 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^[1]

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 OSPI to 0FFH; cannot access 100H to 02FFH.

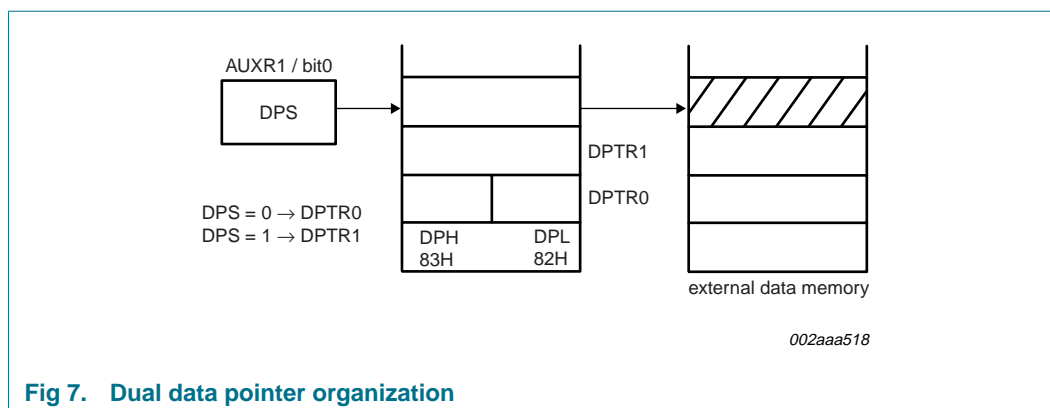


Fig 7. Dual data pointer organization

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.

6.3 Flash memory IAP

6.3.1 Flash organization

The P89V51RB2/RC2/RD2 program memory consists of a 16/32/64 kB block. 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 bootloader may be invoked. This ISP bootloader 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.

6.3.2 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.

Table 12. 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 boot code 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>:02xxxx06HHLLcc</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 13. IAP function calls ...continued

IAP function	IAP call parameters
Program Security Bit, Double Clock	Input parameters: R1 = 05H DPL = 01H = security bit DPL = 05H = Double Clock Return parameter(s): ACC = 00 = pass ACC = !00 = fail
Read Security Bit, Double Clock, SoftICE	Input parameters: ACC = 07H Return parameter(s): ACC = 00 SoftICE S/N-match 0 SB 0 DBL_CLK
Erase sector	Input parameters: R1 = 08H DPH = sector address high byte DPL = sector address low byte Return parameter(s): ACC = 00 = pass ACC = !00 = fail

6.4 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.

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.

6.5.5 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_{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

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

6.6 UARTs

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

6.6.1 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.

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.

6.6.8 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.

6.6.9 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.

Table 29. SPCR - SPI control register (address D5H) bit description ...continued

Bit	Symbol	Description
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 SPICLK rate of the device when a master. SPR1 and SPR0 have no effect on the slave. See Table 30 below.
0	SPR0	SPI Clock Rate Select bit 0. Along with SPR1 controls the SPICLK rate of the device when a master. SPR1 and SPR0 have no effect on the slave. See Table 30 below.

Table 30. SPCR - SPI control register (address D5H) clock rate selection

SPR1	SPR0	SPICLK = f_{osc} divided by
0	0	4
0	1	16
1	0	64
1	1	128

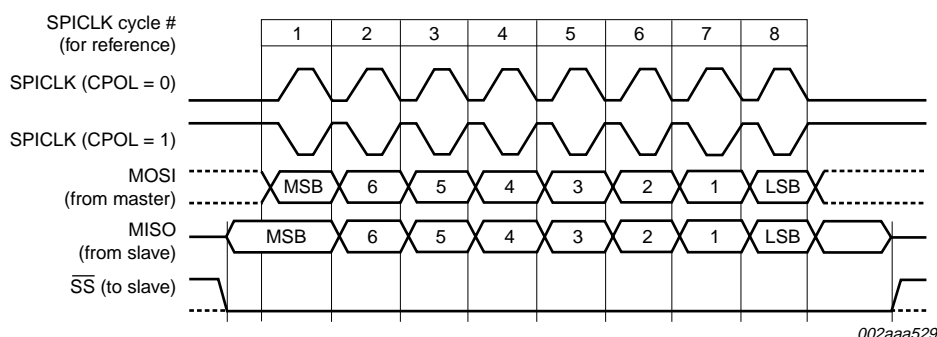
Table 31. SPSR - SPI status register (address AAH) bit allocation

Bit addressable; Reset source(s): any reset; Reset value: 0000 0000B

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

Table 32. 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.

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

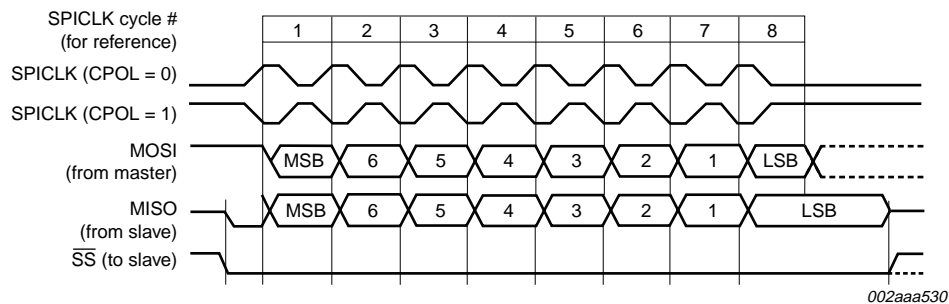


Fig 19. SPI transfer format with CPHA = 1

6.8 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 WDT. The WDT register will increment every 344064 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 WDT 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_{osc} is the oscillator frequency.

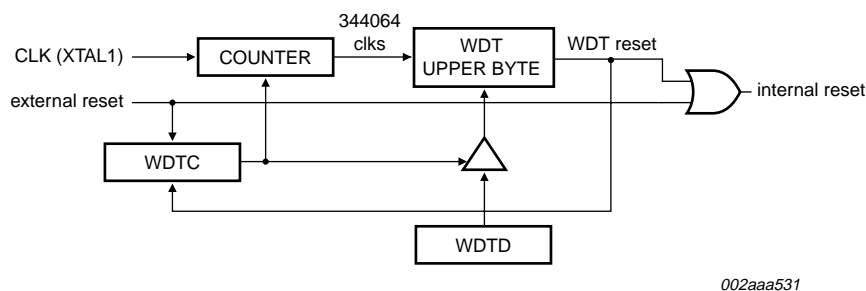
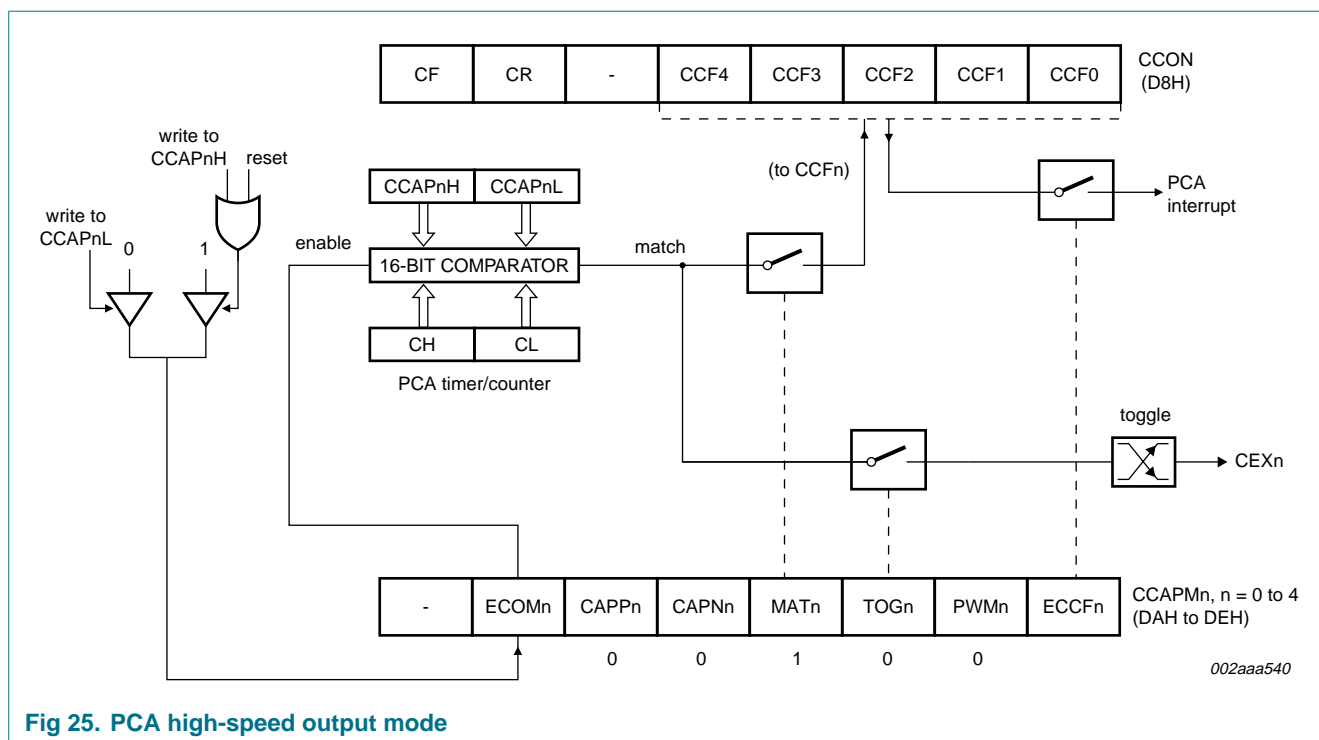
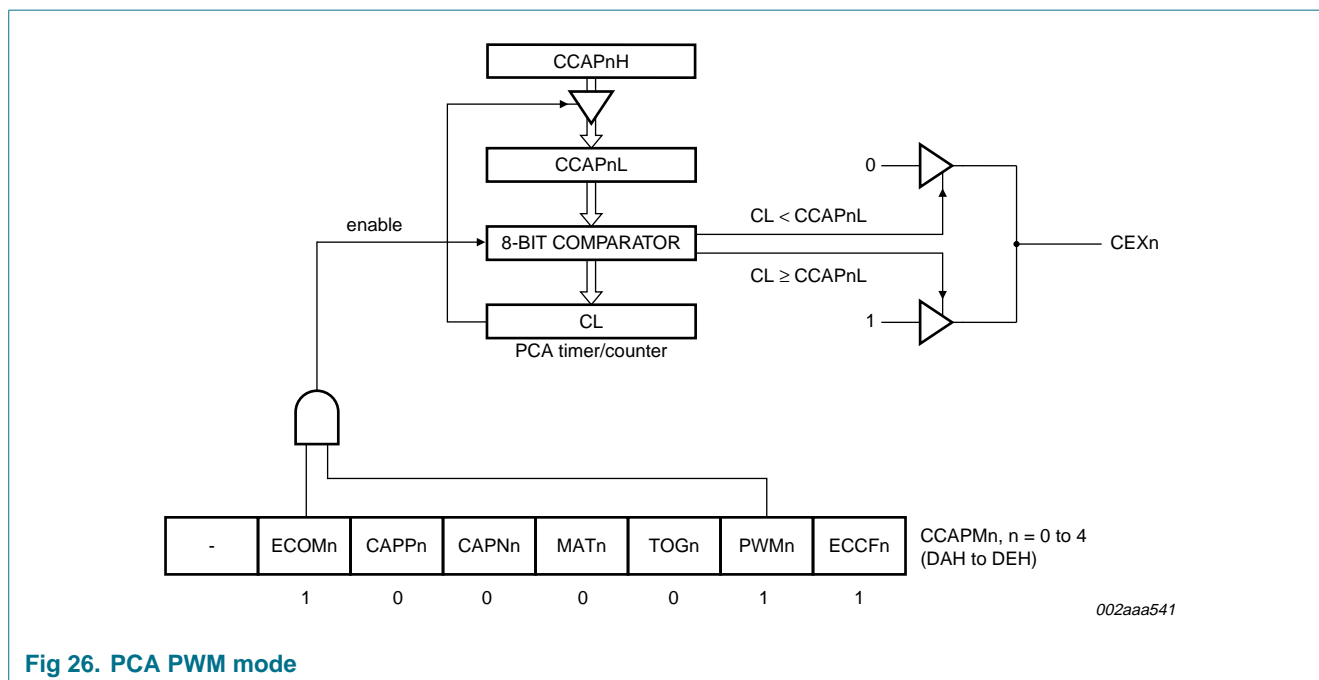


Fig 20. Block diagram of programmable WDT



6.9.4 PWM mode

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



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

Table 45. 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 46. 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 47. 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	Brownout Interrupt Enable. 1 = enable, 0 = disable.
2 to 0	-	Reserved for future use. Should be set to '0' by user programs.

Table 48. 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 49. IP0 - Interrupt priority 0 low register (address B8H) bit description

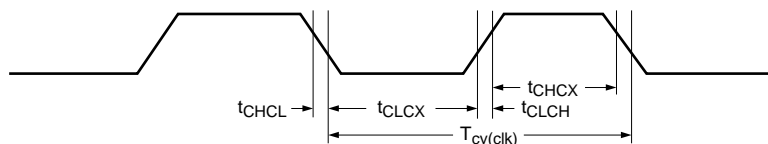
Bit	Symbol	Description
7	-	Reserved for future use. Should be set to '0' by user programs.
6	PPC	PCA interrupt priority LOW bit.
5	PT2	Timer 2 interrupt priority LOW bit.
4	PS	Serial Port interrupt priority LOW bit.
3	PT1	Timer 1 interrupt priority LOW bit.
2	PX1	External interrupt 1 priority LOW bit.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

Table 50. 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 64. External clock drive

Symbol	Parameter	Oscillator				Unit
		40 MHz		Variable		
		Min	Max	Min	Max	
f _{osc}	oscillator frequency	-	-	0	40	MHz
T _{cy(clk)}	clock cycle time	25	-	-	-	ns
t _{CHCX}	clock HIGH time	8.75	-	0.35T _{cy(clk)}	0.65T _{cy(clk)}	ns
t _{CLCX}	clock LOW time	8.75	-	0.35T _{cy(clk)}	0.65T _{cy(clk)}	ns
t _{CLCH}	clock rise time	-	10	-	-	ns
t _{CHCL}	clock fall time	-	10	-	-	ns

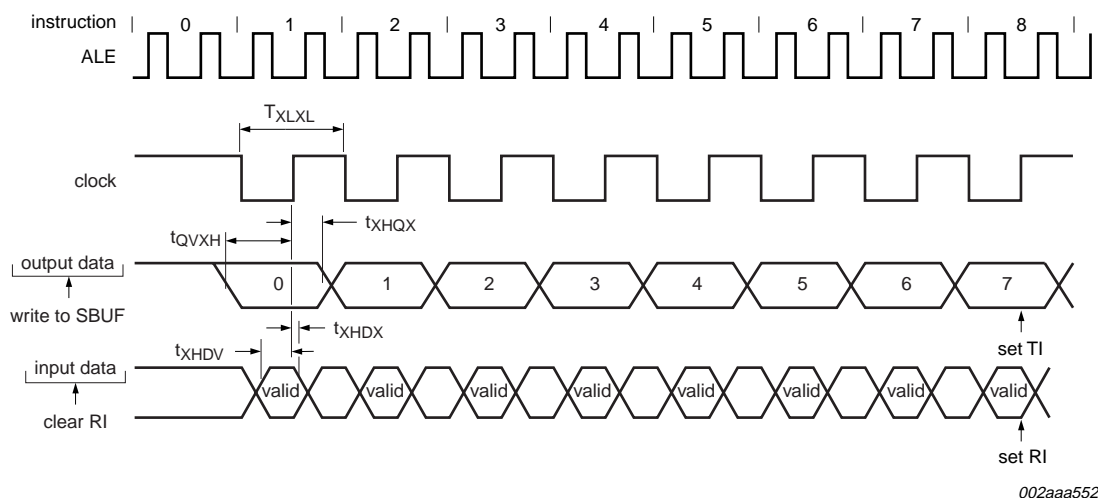


002aaa907

Fig 34. External clock drive waveform (with an amplitude of at least $V_{i(RMS)} = 200$ mV)

Table 65. Serial port timing

Symbol	Parameter	Oscillator				Unit
		40 MHz		Variable		
		Min	Max	Min	Max	
T _{XLXL}	serial port clock cycle time	0.3	-	12T _{cy(clk)}	-	μs
t _{QVXH}	output data set-up to clock rising edge time	117	-	10T _{cy(clk)} – 133	-	ns
t _{XHQX}	output data hold after clock rising edge time	0	-	2T _{cy(clk)} – 50	-	ns
t _{XHDX}	input data hold after clock rising edge time	0	-	0	-	ns
t _{XHDV}	input data valid to clock rising edge time	-	117	-	10T _{cy(clk)} – 133	ns



002aaa552

Fig 35. Shift register mode timing waveforms

Table 66. SPI interface timing

Symbol	Parameter	Conditions	Variable clock		$f_{osc} = 18 \text{ MHz}$		Unit
			Min	Max	Min	Max	
f_{SPI}	SPI operating frequency		0	$T_{cy(clk)} / 4$	0	10	MHz
T_{SPICYC}	SPI cycle time	see Figure 36 , 37 , 38 , 39	$4T_{cy(clk)}$	-	222	-	ns
$t_{SPILEAD}$	SPI enable lead time	see Figure 38 , 39	250	-	250	-	ns
t_{SPILAG}	SPI enable lag time	see Figure 38 , 39	250	-	250	-	ns
$t_{SPICLK H}$	SPICLK HIGH time	see Figure 36 , 37 , 38 , 39	$2T_{cy(clk)}$	-	111	-	ns
$t_{SPICLK L}$	SPICLK LOW time	see Figure 36 , 37 , 38 , 39	$2T_{cy(clk)}$	-	111	-	ns
t_{SPIDSU}	SPI data set-up time	master or slave; see Figure 36 , 37 , 38 , 39	100	-	100	-	ns
t_{SPIDH}	SPI data hold time	master or slave; see Figure 36 , 37 , 38 , 39	100	-	100	-	ns
t_{SPIA}	SPI access time	see Figure 38 , 39	0	80	0	80	ns
t_{SPIDIS}	SPI disable time	see Figure 38 , 39	0	160	-	160	ns
t_{SPIDV}	SPI enable to output data valid time	see Figure 36 , 37 , 38 , 39	-	111	-	111	ns
t_{SPIOH}	SPI output data hold time	see Figure 36 , 37 , 38 , 39	0	-	0	-	ns
t_{SPIR}	SPI rise time	see Figure 36 , 37 , 38 , 39					
	SPI outputs (SPICLK, MOSI, MISO)		-	100	-	100	ns
	SPI inputs (SPICLK, MOSI, MISO, SS)		-	2000	-	2000	ns
t_{SPIF}	SPI fall time	see Figure 36 , 37 , 38 , 39					
	SPI outputs (SPICLK, MOSI, MISO)		-	100	-	100	ns
	SPI inputs (SPICLK, MOSI, MISO, SS)		-	2000	-	2000	ns

