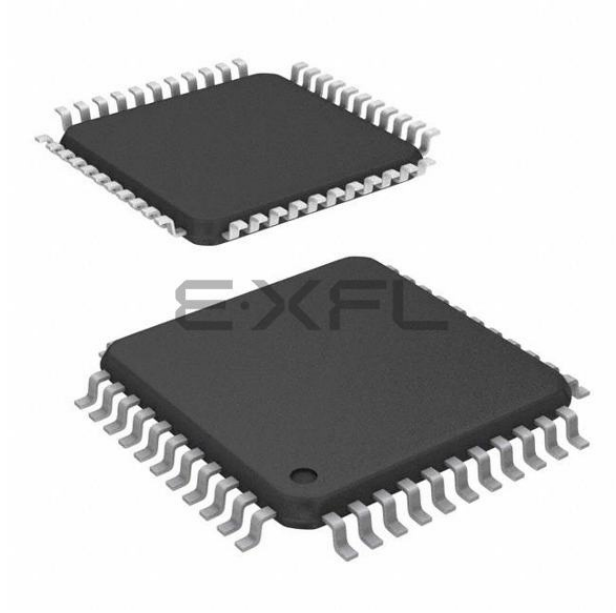


**Welcome to E-XFL.COM**

## 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 | I²C, SPI, UART/USART |
| Peripherals | POR, PWM, WDT |
| Number of I/O | 36 |
| Program Memory Size | 16KB (16K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 512 x 8 |
| Voltage - Supply (Vcc/Vdd) | 4.5V ~ 5.5V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-TQFP |
| Supplier Device Package | 44-TQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/p89v660fbc-557 |

**Table 3. Pin description** *…continued*

| Symbol | Pin | | Type | Description |
|---|---|---|---|---|
| | TQFP44 | PLCC44 | | |
| P1[4]/CEX1 | 44 | 6 | I/O | **P1[4] —** Port 1 bit 4. |
| | | | I/O | **CEX1 —** Capture/compare external I/O for PCA Module 1 |
| P1[5]/CEX2 | 1 | 7 | I/O | **P1[5] —** Port 1 bit 5. |
| | | | I/O | **CEX2 —** Capture/compare external I/O for PCA Module 2 |
| P1[6]/SCL | 2 | 8 | I/O | **P1[6] —** Port 1 bit 6. |
| | | | I/O | **SCL —** $I^2C$-bus serial clock input/output |
| P1[7]/SDA | 3 | 9 | I/O | **P1[7] —** Port 1 bit 7. |
| | | | I/O | **SDA —** $I^2C$-bus serial data input/output |
| P2[0] to P2[7][1] | | | 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 making the transition to ‘1’s. |
| P2[0]/A8 | 18 | 24 | I/O | **P2[0] —** Port 2 bit 0. |
| | | | O | **A8 —** Address bit 8. |
| P2[1]/A9 | 19 | 25 | I/O | **P2[1] —** Port 2 bit 1. |
| | | | O | **A9 —** Address bit 9. |
| P2[2]/A10 | 20 | 26 | I/O | **P2[2] —** Port 2 bit 2. |
| | | | O | **A10 —** Address bit 10. |
| P2[3]/A11 | 21 | 27 | I/O | **P2[3] —** Port 2 bit 3. |
| | | | O | **A11 —** Address bit 11. |
| P2[4]/A12 | 22 | 28 | I/O | **P2[4] —** Port 2 bit 4. |
| | | | O | **A12 —** Address bit 12. |
| P2[5]/A13 | 23 | 29 | I/O | **P2[5] —** Port 2 bit 5. |
| | | | O | **A13 —** Address bit 13. |
| P2[6]/A14 | 24 | 30 | I/O | **P2[6] —** Port 2 bit 6. |
| | | | O | **A14 —** Address bit 14. |
| P2[7]/A15 | 25 | 31 | I/O | **P2[7] —** Port 2 bit 7. |
| | | | O | **A15 —** Address bit 15. |
| P3[0] to P3[7][1] | | | I/O with internal pull-up | **Port 3**: 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. |
| P3[0]/RXD | 5 | 11 | I | **P3[0] —** Port 3 bit 0. |
| | | | I | **RXD —** Serial input port. |
| P3[1]/TXD | 7 | 13 | O | **P3[1] —** Port 3 bit 1. |
| | | | O | **TXD —** Serial output port. |

**Table 3. Pin description** *…continued*

| Symbol | Pin | | Type | Description |
|---|---|---|---|---|
| | TQFP44 | PLCC44 | | |
| P3[2]/$\overline{\text{INT0}}$ | 8 | 14 | I | **P3[2] —** Port 3 bit 2. |
| | | | I | **INT0 —** External interrupt 0 input. |
| P3[3]/$\overline{\text{INT1}}$ | 9 | 15 | I | **P3[3] —** Port 3 bit 3. |
| | | | I | **INT1 —** External interrupt 1 input |
| P3[4]/T0/CEX3 | 10 | 16 | I/O | **P3[4] —** Port 3 bit 4. |
| | | | I | **T0 —** External count input to Timer/Counter 0. |
| | | | I/O | **CEX3 —** Capture/compare external I/O for PCA Module 3. |
| P3[5]/T1/CEX4 | 11 | 17 | I/O | **P3[5] —** Port 3 bit 5. |
| | | | I | **T1 —** External count input to Timer/Counter 1 |
| | | | I/O | **CEX4 —** Capture/compare external I/O for PCA Module 4 |
| P3[6]/$\overline{\text{WR}}$ | 12 | 18 | O | **P3[6] —** Port 3 bit 6. |
| | | | O | **$\overline{\text{WR}}$ —** External data memory write strobe |
| P3[7]/$\overline{\text{RD}}$ | 13 | 19 | O | **P3[7] —** Port 3 bit 7. |
| | | | O | **$\overline{\text{RD}}$ —** External data memory read strobe. |
| P4[0] to P4[3][1] | | | I/O with internal pull-up | **Port 4**: Port 4 is a 4-bit bidirectional I/O port with internal pull-ups. Port 4 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 4 pins that are externally pulled LOW will source current ($I_{IL}$) because of the internal pull-ups. |
| P4[0]/SCL_1/ SPICLK | 17 | 23 | I/O | **P4[0] —** Port 4 bit 0. |
| | | | I/O | **SCL_1 —** Second I²C-bus serial clock input/output |
| | | | I/O | **SPICLK —** Serial clock input/output for SPI |
| P4[1]/SDA_1/ MISO | 28 | 34 | I/O | **P4[1] —** Port 4 bit 1. |
| | | | I/O | **SDA_1 —** Second I²C-bus serial data input/output |
| | | | I/O | **MISO —** Master input/slave output for SPI |
| P4[2]/MOSI | 39 | 1 | I/O | **P4[2] —** Port 4 bit 2. |
| | | | I/O | **MOSI —** Master output/slave input for SPI |
| P4[3]/$\overline{\text{SS}}$ | 6 | 12 | I | **P4[3] —** Port 4 bit 3. |
| | | | I | **$\overline{\text{SS}}$ —** Slave select input for SPI |
| $\overline{\text{PSEN}}$ | 26 | 32 | I/O | **Program Store Enable**: $\overline{\text{PSEN}}$ is the read strobe for external program memory. When the device is executing from internal program memory, $\overline{\text{PSEN}}$ is inactive (HIGH). When the device is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory. |
| RST | 4 | 10 | I | **Reset**: While the oscillator is running, a HIGH logic state on this pin for two machine cycles will reset the device. |
| $\overline{\text{EA}}$ | 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. |

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

** indicates Special Function Registers (SFRs) that are bit addressable.*

| Name | Description | SFR addr. | Bit functions and addresses | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | MSB | | | | | | | LSB |
| | | **Bit address** | **AF** | **AE** | **AD** | **AC** | **AB** | **AA** | **A9** | **A8** |
| IEN0* | Interrupt Enable 0 | A8H | EA | EC | ES1 | ES0 | ET1 | EX1 | ET0 | EX0 |
| | | **Bit address** | **EF** | **EE** | **ED** | **EC** | **EB** | **EA** | **E9** | **E8** |
| IEN1* | Interrupt Enable 1 | E8H | - | - | - | - | - | ES3 | ES2 | ET2 |
| | | **Bit address** | **BF** | **BE** | **BD** | **BC** | **BB** | **BA** | **B9** | **B8** |
| IP0* | Interrupt Priority 0 | B8H | PT2 | PPC | PS1 | PS0 | PT1 | PX1 | PT0 | PX0 |
| IP0H | Interrupt Priority 0 HIGH | B7H | PT2H | PPCH | PS1H | PS0H | PT1H | PX1H | PT0H | PX0H |
| | | **Bit address** | **FF** | **FE** | **FD** | **FC** | **FB** | **FA** | **F9** | **F8** |
| IP1* | Interrupt Priority 1 | 91H | - | - | - | - | - | - | PS3 | PS2 |
| IP1H | Interrupt Priority 1 HIGH | 92H | - | - | - | - | - | - | PS3 | PS2 |
| | | **Bit address** | **87** | **86** | **85** | **84** | **83** | **82** | **81** | **80** |
| P0* | Port 0 | 80H | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | AD1 | AD0 |
| | | **Bit address** | **97** | **96** | **95** | **94** | **93** | **92** | **91** | **90** |
| P1* | Port 1 | 90H | SDA | SCL | CEX2 | CEX1 | CEX0 | ECI | T2EX | T2 |
| | | **Bit address** | **A7** | **A6** | **A5** | **A4** | **A3** | **A2** | **A1** | **A0** |
| P2* | Port 2 | A0H | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |
| | | **Bit address** | **B7** | **B6** | **B5** | **B4** | **B3** | **B2** | **B1** | **B0** |
| P3* | Port 3 | B0H | $\overline{RD}$ | $\overline{WR}$ | CEX4/T1 | CEX3/T0 | $\overline{INT1}$ | $\overline{INT0}$ | TXD | RXD |
| P4 | Port 4 | A1H | - | - | - | - | $\overline{SS}$ | MOSI | MISO/ SDA_1 | SPICLK/ SCL_1 |
| PCON | Power Control Register | 87H | SMOD1 | SMOD0 | - | POF | GF1 | GF0 | PD | IDL |
| | | **Bit address** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** |
| PSW* | Program Status Word | D0H | CY | AC | F0 | RS1 | RS0 | OV | F1 | P |
| RCAP2H | Timer2 Capture HIGH | CBH | | | | | | | | |
| RCAP2L | Timer2 Capture LOW | CAH | | | | | | | | |
| | | **Bit address** | **9F** | **9E** | **9D** | **9C** | **9B** | **9A** | **99** | **98** |
| S0CON* | Serial Port Control | 98H | SM0/FE_ | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
| S0BUF | Serial Port Data Buffer Register | 99H | | | | | | | | |
| SADDR | Serial Port Address Register | A9H | | | | | | | | |

for the 89V662, above 6FFH for the 89V664) will access external off-chip memory 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.

**Table 6. 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 '1', accesses internal XRAM with address specified in MOVX instruction. If address supplied with this instruction exceeds on-chip available XRAM, off-chip RAM is accessed. When '0', every MOVX instructions 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 EXTRAM = 0, 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 7 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 7. 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 < 0100H (89V660)** | **ADDR ≥ 0100H (89V660)** | **ADDR = any** |
| | **ADDR < 0300H (89V662)** | **ADDR ≥ 0300H (89V662)** | |
| | **ADDR < 0700H (89V664)** | **ADDR ≥ 0700H (89V664)** | |
| EXTRAM = 0 | $\overline{RD}$/$\overline{WR}$ asserted | $\overline{RD}$/$\overline{WR}$ asserted | $\overline{RD}$/$\overline{WR}$ asserted |
| EXTRAM = 1 | $\overline{RD}$/$\overline{WR}$ not asserted | $\overline{RD}$/$\overline{WR}$ asserted | $\overline{RD}$/$\overline{WR}$ not asserted |

### 6.3.3 Boot block

When the microcontroller programs its own flash memory, all of the low level details are handled by code (bootloader) that is contained in a Boot block. A user program calls the common entry point in the Boot block with appropriate parameters to accomplish the desired operation. Boot block operations include erase user code, program user code, program security bits, chip erase, etc. The Boot block logically overlays the program memory space from FC00H to FFFFH, when it is enabled. The Boot block may be disabled on-the-fly so that the upper 1 kB of user code is available to the user's program.

### 6.3.4 Power-on reset code execution

The P89V660/662/664 contains two special flash elements: the Boot Vector and the Boot Status bit. Following reset, the P89V660/662/664 examines the contents of the Boot Status bit. If the Boot Status bit is set to zero, power-up execution starts at location 0000H, which is the normal start address of the user's application code. When the Boot Status bit is set to a value other than zero, the contents of the Boot Vector are used as the high byte of the execution address and the low byte is set to 00H

Table 10 shows the factory default Boot Vector setting for this device. A factory-provided bootloader is pre-programmed into the address space indicated and uses the indicated boot loader entry point to perform ISP functions.

**Table 10. Default boot vector values and ISP entry points**

| Device | Default boot vector | Default bootloader entry point | Default bootloader code range |
|---|---|---|---|
| P89V660/662/664 | FCH | FC00H | FC00H to FFFFH |

### 6.3.5 Hardware activation of the bootloader

The bootloader can also be executed by forcing the device into ISP mode during a power-on sequence. This has the same effect as having a non-zero status byte. This allows an application to be built that will normally execute user code but can be manually forced into ISP operation. If the factory default setting for the boot vector (FCH) is changed, it will no longer point to the factory pre-programmed ISP bootloader code. After programming the flash, the status byte should be programmed to zero in order to allow execution of the user's application code beginning at address 0000H.

### 6.3.6 ISP

ISP is performed without removing the microcontroller from the system. The ISP facility consists of a series of internal hardware resources coupled with internal firmware to facilitate remote programming of the P89V660/662/664 through the serial port. This firmware is provided by NXP and embedded within each P89V660/662/664 device. The NXP ISP facility has made in-circuit programming in an embedded application possible with a minimum of additional expense in components and circuit board area. The ISP function uses five pins ($V_{DD}$, $V_{SS}$, 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.

### 6.3.7 Using ISP

The ISP feature allows for a wide range of baud rates to be used in your application, independent of the oscillator frequency. It is also adaptable to a wide range of oscillator frequencies. This is accomplished by measuring the bit-time of a single bit in a received character. This information is then used to program the baud rate in terms of timer counts

based on the oscillator frequency. The ISP feature requires that an initial character (an uppercase U) be sent to the P89V660/662/664 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 P89V660/662/664 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 11. As a record is received by the P89V660/662/664, 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 P89V660/662/664 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 11. ISP hex record formats**

| Record type | Command/data function |
|---|---|
| 00 | Program User Code Memory<br>:nnaaaa00dd..ddcc<br>Where:<br>nn = number of bytes to program<br>aaaa = address<br>dd..dd = data bytes<br>cc = checksum<br>Example:<br>:0900000001020304050607080 9CA |
| 01 | End of File (EOF), no operation<br>:xxxxxx01cc<br>Where:<br>xxxxxx = required field but value is a ‘don’t care’<br>cc = checksum<br>Example:<br>:00000001FF |
| 02 | not used |

### 6.3.8 IAP method

Several IAP calls are available for use by an application program to permit selective erasing, reading and programming of flash pages, security bits, security bits, Status bit, 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 FFF0H. The IAP calls are shown in Table 12.

**Table 12. IAP function calls**

| IAP function | IAP call parameters |
|---|---|
| Read Id | **Input parameters:** |
| | R1 = 00H or 80H (WDT feed) |
| | DPH = 00H |
| | DPL = 00H = manufacturer id |
| | DPL = 01H = device id 1 |
| | DPL = 02H = device id 2 |
| | DPL = 03H = 6x/12x bit (bit 7 = 1 = 6x) |
| | DPL = 80H = ISP version number |
| | **Return parameter(s):** |
| | ACC = requested parameter |
| Erase 8 kB/16 kB code block | **Input parameters:** |
| | R1 = 01H or 81H (WDT feed) |
| | DPL = 00H, block 0, 0 kB to 8 kB |
| | DPL = 20H, block 1, 8 kB to 16 kB |
| | DPL = 40H, block 2, 16 kB to 32 kB |
| | DPL = 80H, block 3, 32 kB to 48 kB |
| | DPL = C0H, block 4, 48 kB to 64 kB |
| | **Return parameter(s):** |
| | ACC = 00 = pass |
| | ACC = !00 = fail |
| Program User Code | **Input parameters:** |
| | R1 = 02H or 82H (WDT feed) |
| | DPH = memory address MSB |
| | DPL = memory address LSB |
| | ACC = byte to program |
| | **Return parameter(s):** |
| | ACC = 00 = pass |
| | ACC = !00 = fail |
| Read User Code | **Input parameters:** |
| | R1 = 03H or 83H (WDT feed) |
| | DPH = memory address MSB |
| | DPL = memory address LSB |
| | **Return parameter(s):** |
| | ACC = device data |

**Table 12. IAP function calls** *…continued*

| IAP function | IAP call parameters |
|---|---|
| Erase Status bit and Boot vector | **Input parameters:**<br>R1 = 04H or 84H (WDT feed)<br>DPL = don't care<br>DPH = don't care<br>**Return parameter(s):**<br>ACC = 00 = pass<br>ACC = !00 = fail |
| Program Security bits | **Input parameters:**<br>R1 = 05H or 85H (WDT feed)<br>DPL = 00H = security bit 1<br>DPL = 01H = security bit 2<br>DPL = 02H = security bit 3<br>**Return parameter(s):**<br>ACC = 00 = pass<br>ACC = !00 = fail |
| Program Status bit, Boot vector, 6x/12x bit | **Input parameters:**<br>R1 = 06H or 86H (WDT feed)<br>DPL = 00H = program Status bit<br>DPL = 01H = program Boot vector<br>DPL = 02H = 6x/12x bit<br>ACC = Boot vector value to program<br>**Return parameter(s):**<br>ACC = 00 = pass<br>ACC = !00 = fail |
| Read Security bits, Status bit, Boot vector | **Input parameters:**<br>ACC = 07H or 87H (WDT feed)<br>DPL = 00H = security bits<br>DPL = 01H = Status bit<br>DPL = 02H = Boot vector<br>**Return parameter(s):**<br>ACC = 00 SoftICE S/N-match 0 SB 0 DBL_CLK |
| Erase page | **Input parameters:**<br>R1 = 08H or 88H (WDT feed)<br>DPH = page address high byte<br>DPL = page address low byte<br>**Return parameter(s):**<br>ACC = 00 = pass<br>ACC = !00 = fail |

## 6.4 I2C-bus interface

The I2C-bus uses two wires, Serial Clock (SCL) and Serial Data (SDA) to transfer information between devices connected to the bus, and has the following features:

- Bidirectional data transfer between masters and slaves

### 6.4.1 I2C-bus data register

S1DAT register contains the data to be transmitted or the data received. The CPU can read and write to this 8-bit register while it is not in the process of shifting a byte. Thus this register should only be accessed when the SI bit is set. Data in S1DAT remains stable as long as the SI bit is set. Data in S1DAT is always shifted from right to left: the first bit to be transmitted is the MSB (bit 7), and after a byte has been received, the first bit of received data is located at the MSB of S1DAT.

### 6.4.2 I2C-bus slave address register

The S1ADR register is readable and writable, and is only used when the I2C-bus interface is set to slave mode. In master mode, this register has no effect. The LSB of S1ADR is general call bit. When this bit is set, the general call address (00H) is recognized.

**Table 13. I2C-bus slave address register (S1ADR - address DBH) bit allocation**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Symbol | S1ADR.6 | S1ADR.5 | S1ADR.4 | S1ADR.3 | S1ADR.2 | S1ADR.1 | S1ADR.0 | S1GC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 14. I2C-bus slave address register (S1ADR - address DBH) bit description**

| Bit | Symbol | Description |
|---|---|---|
| 7:1 | S1ADR7:1 | 7 bit own slave address. When in master mode, the contents of this register has no effect. |
| 0 | S1GC | General call bit. When set, the general call address (00H) is recognized, otherwise it is ignored. |

### 6.4.3 I2C-bus control register

The CPU can read and write this register. There are two bits are affected by hardware: the SI bit and the STO bit. The SI bit is set by hardware and the STO bit is cleared by hardware.

CR2:0 determines the SCL source and frequency when the I2C-bus is in master mode. In slave mode these bits are ignored and the bus will automatically synchronize with any clock frequency up to 100 kHz from the master I2C-bus device. Timer 1 should be programmed by the user in 8 bit auto-reload mode (Mode 2) when used as the SCL source. See Table 17.

The STA bit is START flag. Setting this bit causes the I2C-bus interface to enter master mode and attempt transmitting a START condition or transmitting a repeated START condition when it is already in master mode.

The STO bit is STOP flag. Setting this bit causes the I2C-bus interface to transmit a STOP condition in master mode, or recovering from an error condition in slave mode.

If the STA and STO are both set, then a STOP condition is transmitted to the I2C-bus if it is in master mode, and transmits a START condition afterwards. If it is in slave mode, an internal STOP condition will be generated, but it is not transmitted to the bus.

**Table 24. Slave Receiver mode** *…continued*

| Status code (S1STA) | Status of the I²C-bus hardware | Application software response | | | | | Next action taken by I²C-bus hardware |
|---|---|---|---|---|---|---|---|
| | | to/from S1DAT | to S1CON | | | | |
| | | | STA | STO | SI | AA | |
| 88H | Previously addressed with own SLA address; Data has been received; NOT ACK has been returned. | Read data byte or | 0 | 0 | 0 | 0 | Switched to not addressed SLA mode; no recognition of own SLA or general address. |
| | | read data byte or | 0 | 0 | 0 | 1 | Switched to not addressed SLA mode; Own SLA will be recognized; general call address will be recognized if S1ADR.0 = 1. |
| | | read data byte or | 1 | 0 | 0 | 0 | Switched to not addressed SLA mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free. |
| | | read data byte | 1 | 0 | 0 | 1 | Switched to not addressed SLA mode; Own slave address will be recognized; General call address will be recognized if S1ADR.0 = 1. A START condition will be transmitted when the bus becomes free. |
| 90H | Previously addressed with General call; Data has been received; ACK has been returned. | Read data byte or | x | 0 | 0 | 0 | Data byte will be received and NOT ACK will be returned. |
| | | read data byte | x | 0 | 0 | 1 | Data byte will be received and ACK will be returned. |
| 98H | Previously addressed with General call; Data has been received; NOT ACK has been returned. | Read data byte | 0 | 0 | 0 | 0 | Switched to not addressed SLA mode; no recognition of own SLA or General call address. |
| | | read data byte | 0 | 0 | 0 | 1 | Switched to not addressed SLA mode; Own slave address will be recognized; General call address will be recognized if S1ADR.0 = 1. |
| | | read data byte | 1 | 0 | 0 | 0 | Switched to not addressed SLA mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free. |
| | | read data byte | 1 | 0 | 0 | 1 | Switched to not addressed SLA mode; Own slave address will be recognized; General call address will be recognized if S1ADR.0 = 1. A START condition will be transmitted when the bus becomes free. |

**Table 24. Slave Receiver mode** *…continued*

| Status code (S1STA) | Status of the I²C-bus hardware | Application software response | | | | | Next action taken by I²C-bus hardware |
|---|---|---|---|---|---|---|---|
| | | to/from S1DAT | to S1CON | | | | |
| | | | STA | STO | SI | AA | |
| A0H | A STOP condition or repeated START condition has been received while still addressed as SLA/REC or SLA/TRX. | No S1DAT action | 0 | 0 | 0 | 0 | Switched to not addressed SLA mode; no recognition of own SLA or General call address. |
| | | no S1DAT action | 0 | 0 | 0 | 1 | Switched to not addressed SLA mode; Own slave address will be recognized; General call address will be recognized if S1ADR.0 = 1. |
| | | no S1DAT action | 1 | 0 | 0 | 0 | Switched to not addressed SLA mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free. |
| | | no S1DAT action | 1 | 0 | 0 | 1 | Switched to not addressed SLA mode; Own slave address will be recognized; General call address will be recognized if S1ADR.0 = 1. A START condition will be transmitted when the bus becomes free. |

**Table 25. Slave transmitter mode**

| Status code (S1STA) | Status of the I²C-bus hardware | Application software response | | | | | Next action taken by I²C-bus hardware |
|---|---|---|---|---|---|---|---|
| | | to/from S1DAT | to S1CON | | | | |
| | | | STA | STO | SI | AA | |
| A8H | Own SLA+R has been received; ACK has been returned. | Load data byte or | x | 0 | 0 | 0 | Last data byte will be transmitted and ACK bit will be received. |
| | | load data byte | x | 0 | 0 | 1 | Data byte will be transmitted; ACK will be received. |
| B0H | Arbitration lost in SLA+R/W as master; Own SLA+R has been received, ACK has been returned. | Load data byte or | x | 0 | 0 | 0 | Last data byte will be transmitted and ACK bit will be received. |
| | | load data byte | x | 0 | 0 | 1 | Data byte will be transmitted; ACK bit will be received. |
| B8H | Data byte in S1DAT has been transmitted; ACK has been received. | Load data byte or | x | 0 | 0 | 0 | Last data byte will be transmitted and ACK bit will be received |
| | | load data byte | x | 0 | 0 | 1 | Data byte will be transmitted; ACK will be received. |

**Table 25. Slave transmitter mode** *…continued*

| Status code (S1STA) | Status of the I²C-bus hardware | Application software response | | | | | Next action taken by I²C-bus hardware |
|---|---|---|---|---|---|---|---|
| | | to/from S1DAT | to S1CON | | | | |
| | | | STA | STO | SI | AA | |
| C0H | Data byte in S1DAT has been transmitted; NOT ACK has been received. | No S1DAT action or | 0 | 0 | 0 | 0 | Switched to not addressed SLA mode; no recognition of own SLA or General call address. |
| | | no S1DAT action or | 0 | 0 | 0 | 1 | Switched to not addressed SLA mode; Own slave address will be recognized; General call address will be recognized if S1ADR.0 = 1. |
| | | no S1DAT action or | 1 | 0 | 0 | 0 | Switched to not addressed SLA mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free. |
| | | no S1DAT action | 1 | 0 | 0 | 1 | Switched to not addressed SLA mode; Own slave address will be recognized; General call address will be recognized if S1ADR.0 = 1. A START condition will be transmitted when the bus becomes free. |
| C8H | Last data byte in S1DAT has been transmitted (AA = 0); ACK has been received. | No S1DAT action or | 0 | 0 | 0 | 0 | Switched to not addressed SLA mode; no recognition of own SLA or General call address. |
| | | no S1DAT action or | 0 | 0 | 0 | 1 | Switched to not addressed SLA mode; Own slave address will be recognized; General call address will be recognized if S1ADR.0 = 1. |
| | | no S1DAT action or | 1 | 0 | 0 | 0 | Switched to not addressed SLA mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free. |
| | | no S1DAT action | 1 | 0 | 0 | 1 | Switched to not addressed SLA mode; Own slave address will be recognized; General call address will be recognized if S1ADR.0 = 1. A START condition will be transmitted when the bus becomes free. |

## 6.5 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 26 and Table 27).

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.

**Table 31. Timer 2 operating mode**

| RCLK+TCLK | CP/$\overline{RL2}$ | 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 32. 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/$\overline{T2}$ | CP/$\overline{RL2}$ |

**Table 33. 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/$\overline{T2}$ | Timer or counter select. (Timer 2)<br>0 = internal timer ($f_{osc}$ / 6)<br>1 = external event counter (falling edge triggered; external clock’s maximum rate = $f_{osc}$ / 12. |
| 0 | CP/$\overline{RL2}$ | 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 34. T2MOD - Timer 2 mode control register (address C9H) bit allocation**
*Not bit addressable; Reset value: XX00 0000B*

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

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

### 6.6.5 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:

Baud rate = Timer 2 overflow rate / 16

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

Baud rate = $f_{osc}$ / (16 × (65536 − (RCAP2H, RCAP2L)))

Where $f_{osc}$ = oscillator frequency

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

RCAP2H, RCAP2L = 65536 − $f_{osc}$ / (16 × baud rate)

**Table 36. Timer 2 generated commonly used baud rates**

| Rate | Oscillator frequency | 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 |

## 6.7 UARTs

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

**Table 41. 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 42. |
| 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 42. |

**Table 42. SPCR - SPI control register (address D5H) clock rate selection**

| SPR1 | SPR0 | SPICLK = $f_{osc}$ divided by | |
|---|---|---|---|
| | | 6-clock mode | 12-clock mode |
| 0 | 0 | 2 | 4 |
| 0 | 1 | 8 | 16 |
| 1 | 0 | 32 | 64 |
| 1 | 1 | 64 | 128 |

**Table 43. 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 44. 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 ES3 = 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. |

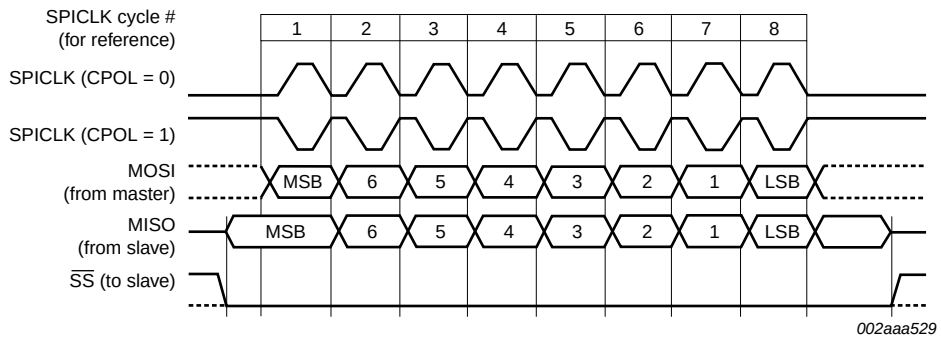



**Fig 24. SPI transfer format with CPHA = 0**

**Table 55. 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 | ES1 | ES0 | ET1 | EX1 | ET0 | EX0 |

**Table 56. 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 | ES1 | I²C-bus Interrupt Enable (primary). |
| 4 | ES0 | 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 57. IEN1 - Interrupt enable register 1 (address E8H) bit allocation**
*Bit addressable; Reset value: 00H*

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Symbol | - | - | - | - | - | ES3 | ES2 | ET2 |

**Table 58. IEN1 - Interrupt enable register 1 (address E8H) bit description**

| Bit | Symbol | Description |
|---|---|---|
| 7 to 3 | - | Reserved for future use. Should be set to ‘0’ by user programs. |
| 2 | ES3 | SPI Interrupt Enable. |
| 1 | ES2 | I²C-bus Interrupt Enable (secondary). |
| 0 | ET2 | Timer 2 Interrupt Enable. |

**Table 59. 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 | PT2 | PPC | PS1 | PS0 | PT1 | PX1 | PT0 | PX0 |

**Table 60. IP0 - Interrupt priority 0 low register (address B8H) bit description**

| Bit | Symbol | Description |
|---|---|---|
| 7 | PT2 | Timer 2 Interrupt Priority Low Bit. |
| 6 | PPC | PCA Interrupt Priority Low Bit. |
| 5 | PS1 | I²C-bus Interrupt Priority Low Bit. |
| 4 | PS0 | 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 61. 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 | PT2H | PPCH | PS1H | PS0H | PT1H | PX1H | PT0H | PX0H |

**Table 62. IP0H - Interrupt priority 0 high register (address B7H) bit description**

| Bit | Symbol | Description |
|---|---|---|
| 7 | PT2H | Timer 2 Interrupt Priority High Bit. |
| 6 | PPCH | PCA Interrupt Priority High Bit. |
| 5 | PS1H | I²C-bus Interrupt Priority High Bit (primary). |
| 4 | PS0H | 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 63. IP1 - Interrupt priority 1 register (address 91H) bit allocation**
*Bit addressable; Reset value: 00H*

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Symbol | - | - | - | - | - | - | PS3 | PS2 |

**Table 64. IP1 - Interrupt priority 1 register (address 91H) bit description**

| Bit | Symbol | Description |
|---|---|---|
| 7 to 2 | - | Reserved for future use. Should be set to ‘0’ by user programs. |
| 1 | PS3 | SPI Interrupt Priority Low Bit. |
| 0 | PS2 | I²C-bus Interrupt Priority 1 Low Bit (secondary). |

**Table 65. IP1H - Interrupt priority 1 high register (address 92H) bit allocation**
*Not bit addressable; Reset value: 00H*

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Symbol | - | - | - | - | - | - | PS3H | PS2H |

**Table 66. IP1H - Interrupt priority 1 high register (address 92H) bit description**

| Bit | Symbol | Description |
|---|---|---|
| 7 to 2 | - | Reserved for future use. Should be set to ‘0’ by user programs. |
| 1 | PS3H | SPI Interrupt Priority High Bit. |
| 0 | PS2H | I²C-bus Interrupt Priority High Bit (secondary). |

## 6.13 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 67.

# 7. Limiting values

**Table 69. 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 |
|---|---|---|---|---|---|
| $T_{amb(bias)}$ | bias ambient temperature | | –55 | +125 | °C |
| $T_{stg}$ | storage temperature | | –65 | +150 | °C |
| $V_I$ | input voltage | on $\overline{EA}$ pin to $V_{SS}$ | –0.5 | 14 | V |
| $V_n$ | voltage on any other pin | except $V_{SS}$, with respect to $V_{DD}$ | –0.5 | $V_{DD}$ + 0.5 | V |
| $I_{OL(I/O)}$ | LOW-level output current per input/output pin | | - | 15 | mA |
| $P_{tot(pack)}$ | total power dissipation (per package) | based on package heat transfer, not device power consumption | - | 1.5 | W |

# 8. Static characteristics

**Table 70. Static characteristics**

*$T_{amb}$ = –40 °C to +85 °C; $V_{DD}$ = 4.5 V to 5.5 V; $V_{SS}$ = 0 V*

| Symbol | Parameter | Conditions | | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| $n_{endu(fl)}$ | endurance of flash memory | JEDEC Standard A117 | [1] | 10000 | - | - | cycles |
| $t_{ret(fl)}$ | flash memory retention time | JEDEC Standard A103 | [1] | 100 | - | - | years |
| $I_{latch}$ | I/O latch-up current | JEDEC Standard 78 | [1] | 100 + $I_{DD}$ | - | - | mA |
| $V_{th(HL)}$ | HIGH-LOW threshold voltage | except SCL, SDA | | –0.5 | - | $0.2V_{DD}$ – 0.1 | V |
| $V_{IL}$ | LOW-level input voltage | SCL, SDA | | –0.5 | - | $0.3V_{DD}$ | V |
| $V_{th(LH)}$ | LOW-HIGH threshold voltage | except SCL, SDA, XTAL1, RST | | $0.2V_{DD}$ + 0.9 | - | $V_{DD}$ + 0.5 | V |
| $V_{IH}$ | HIGH-level input voltage | SCL, SDA | | $0.7V_{DD}$ | - | $V_{DD}$ + 0.5 | V |
| | | XTAL1, RST | | $0.7V_{DD}$ | - | 6.0 | V |
| $V_{OL}$ | LOW-level output voltage | $V_{DD}$ = 4.5 V, except, $\overline{PSEN}$, ALE, SCL, SDA | [2][3][4] | | | | |
| | | $I_{OL}$ = 1.6 mA | | - | - | 0.4 | V |
| | | $V_{DD}$ = 4.5 V, ALE, $\overline{PSEN}$ | | | | | |
| | | $I_{OL}$ = 3.2 mA | | - | - | 0.45 | V |
| | | $V_{DD}$ = 4.5 V, SCL, SDA | | | | | |
| | | $I_{OL}$ = 3.0 mA | | - | - | 0.4 | V |
| $V_{OH}$ | HIGH-level output voltage | $V_{DD}$ = 4.5 V, ports 1, 2, 3, 4 | [5] | | | | |
| | | $I_{OH}$ = –30 μA | | $V_{DD}$ – 0.7 | - | - | V |
| | | $V_{DD}$ = 4.5 V, Port 0 in External Bus mode, ALE, $\overline{PSEN}$ | | | | | |
| | | $I_{OH}$ = –3.2 mA | | $V_{DD}$ – 0.7 | - | - | V |
| $I_{IL}$ | LOW-level input current | $V_I$ = 0.4 V, ports 1, 2, 3, 4 | | –1 | - | –75 | μA |

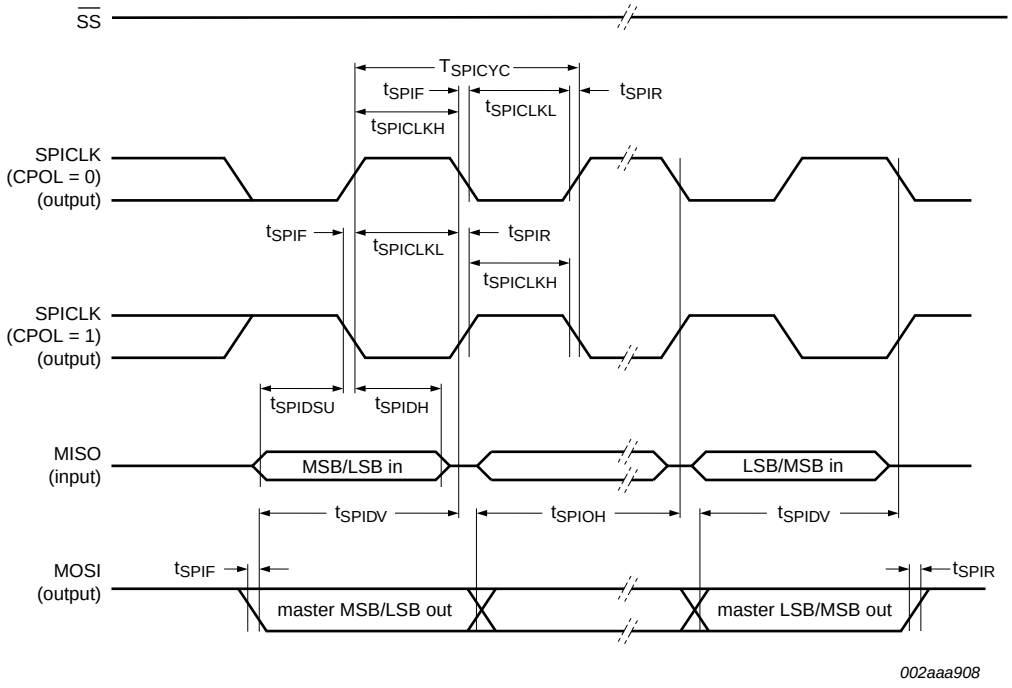



**Fig 42. SPI master timing (CPHA = 0)**

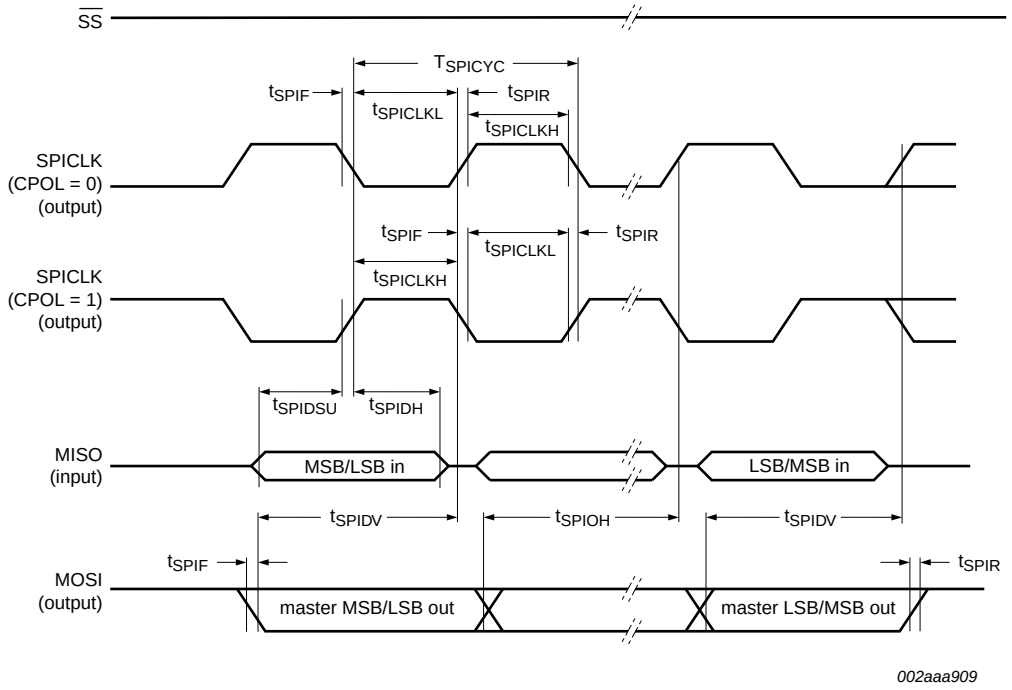



**Fig 43. SPI master timing (CPHA = 1)**

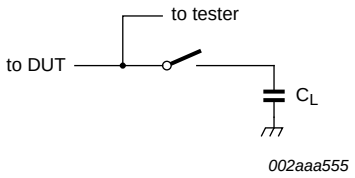
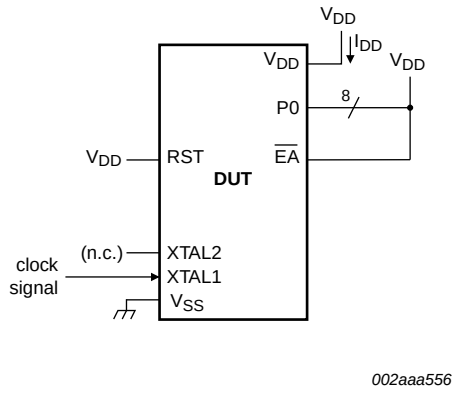
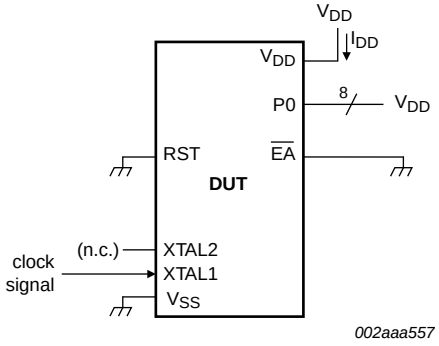



**Fig 46. Test load example**




All other pins disconnected

**Fig 47. $I_{DD}$ test condition, active mode**




All other pins disconnected

**Fig 48. $I_{DD}$ test condition, Idle mode**