**Welcome to E-XFL.COM**

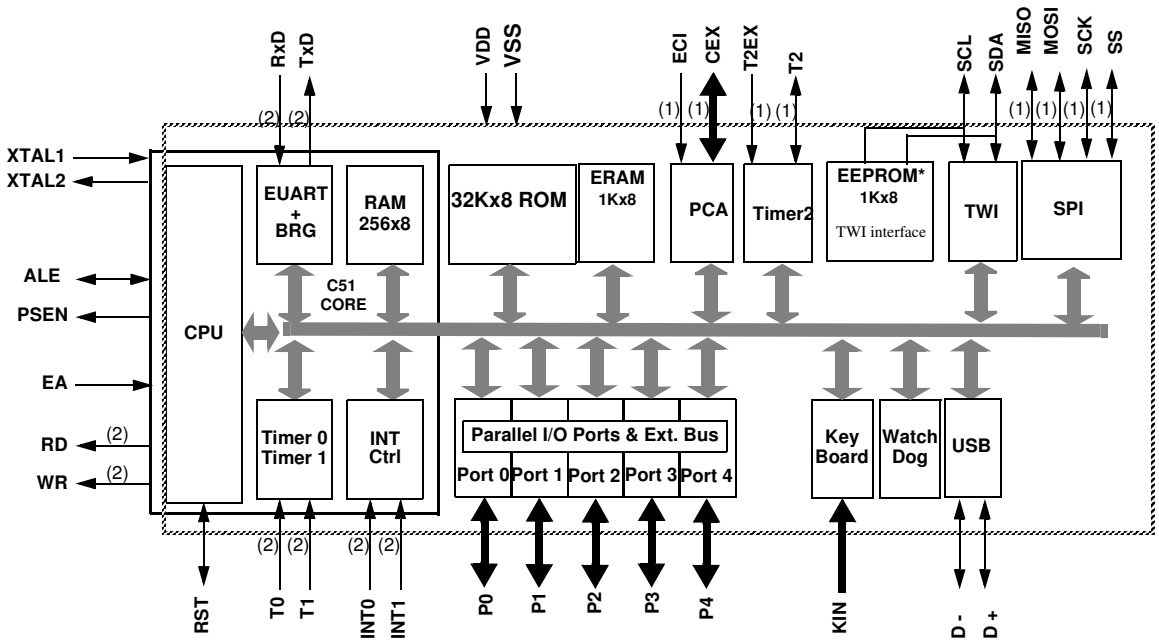
## What is "Embedded - Microcontrollers"?

"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 | Discontinued at Digi-Key |
| Core Processor | 80C52 |
| Core Size | 8-Bit |
| Speed | 48MHz |
| Connectivity | EBI/EMI, I²C, SPI, UART/USART, USB |
| Peripherals | LED, POR, PWM, WDT |
| Number of I/O | 18 |
| Program Memory Size | 32KB (32K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 3.6V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | - |
| Package / Case | - |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/at83c5136-tisul |

## 3. Block Diagram




* EEPROM only available in MLF48

Notes:
1. Alternate function of Port 1
2. Alternate function of Port 3
3. Alternate function of Port 4

| Signal Name | Type | Description | Alternate Function |
|---|---|---|---|
| P3[7:0] | I/O | **Port 3**<br>P3 is an 8-bit bidirectional I/O port with internal pull-ups. | LED[3:0]<br>RxD<br>TxD<br>$\overline{INT0}$<br>$\overline{INT1}$<br>T0<br>T1<br>$\overline{WR}$<br>$\overline{RD}$ |
| P4[1:0] | I/O | **Port 4**<br>P4 is an 2-bit open port. | SCL<br>SDA |

**Table 4-9.** Clock Signal Description

| Signal Name | Type | Description | Alternate Function |
|---|---|---|---|
| XTAL1 | I | **Input to the on-chip inverting oscillator amplifier**<br>To use the internal oscillator, a crystal/resonator circuit is connected to this pin. If an external oscillator is used, its output is connected to this pin. | - |
| XTAL2 | O | **Output of the on-chip inverting oscillator amplifier**<br>To use the internal oscillator, a crystal/resonator circuit is connected to this pin. If an external oscillator is used, leave XTAL2 unconnected. | - |
| PLLF | I | **PLL Low Pass Filter input**<br>Receives the RC network of the PLL low pass filter (See Figure 5-1 on page 11 ). | - |

**Table 4-10.** USB Signal Description

| Signal Name | Type | Description | Alternate Function |
|---|---|---|---|
| D+ | I/O | USB Data + signal<br>Set to high level under reset. | - |
| D- | I/O | USB Data - signal<br>Set to low level under reset. | - |
| VREF | O | **USB Reference Voltage**<br>Connect this pin to D+ using a 1.5 kΩ resistor to use the Detach function. | - |

**Table 4-11.** System Signal Description

| Signal Name | Type | Description | Alternate Function |
|---|---|---|---|
| AD[7:0] | I/O | Multiplexed Address/Data LSB for external access<br>Data LSB for Slave port access (used for 8-bit and 16-bit modes) | P0[7:0] |
| A[15:8] | I/O | Address Bus MSB for external access<br>Data MSB for Slave port access (used for 16-bit mode only) | P2[7:0] |

# 10. Stacked EEPROM

## 10.1 Overview

The AT83C5134/35/36 features a stacked 2-wire serial data EEPROM. The data EEPROM allows to save from 512 Byte for AT24C04 version up to 32 Kbytes for AT24C256 version. The EEPROM is internally connected to the microcontroller on SDA and SCL pins.

## 10.2 Protocol

In order to access this memory, it is necessary to use software subroutines according to the AT24Cxx datasheet. Nevertheless, because the internal pull-up resistors of the AT83C5134/35/36 is quite high (around 100KΩ), the protocol should be slowed in order to be sure that the SDA pin can rise to the high level before reading it.

Another solution to keep the access to the EEPROM in specification is to work with a software pull-up.

Using a software pull-up, consists of forcing a low level at the output pin of the microcontroller before configuring it as an input (high level).

The C51 the ports are “quasi-bidirectional” ports. It means that the ports can be configured as output low or as input high. In case a port is configured as an output low, it can sink a current and all internal pull-ups are disconnected. In case a port is configured as an input high, it is pulled up with a strong pull-up (a few hundreds Ohms resistor) for 2 clock periods. Then, if the port is externally connected to a low level, it is only kept high with a weak pull up (around 100KΩ), and if not, the high level is latched high thanks to a medium pull (around 10kΩ).

Thus, when the port is configured as an input, and when this input has been read at a low level, there is a pull-up of around 100KΩ, which is quite high, to quickly load the SDA capacitance. So in order to help the reading of a high level just after the reading of a low level, it is possible to force a transition of the SDA port from an input state (1), to an output low state (0), followed by a new transition from this output low state to input state; In this case, the high pull-up has been replaced with a low pull-up which warranties a good reading of the data.

**Table 12-2.** T2MOD Register

T2MOD - Timer 2 Mode Control Register (C9h)

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

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 6 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 5 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 4 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 3 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 2 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 1 | T2OE | **Timer 2 Output Enable bit**<br>Cleared to program P1.0/T2 as clock input or I/O port.<br>Set to program P1.0/T2 as clock output. |
| 0 | DCEN | **Down Counter Enable bit**<br>Cleared to disable Timer 2 as up/down counter.<br>Set to enable Timer 2 as up/down counter. |

Reset Value = xxxx xx00b

Not bit addressable

# 13. Programmable Counter Array (PCA)

The PCA provides more timing capabilities with less CPU intervention than the standard timer/counters. Its advantages include reduced software overhead and improved accuracy. The PCA consists of a dedicated timer/counter which serves as the time base for an array of five compare/capture modules. Its clock input can be programmed to count any one of the following signals:

- Peripheral clock frequency ($F_{CLK\ PERIPH}$) ÷ 6
- Peripheral clock frequency ($F_{CLK\ PERIPH}$) ÷ 2
- Timer 0 overflow
- External input on ECI (P1.2)

Each compare/capture modules can be programmed in any one of the following modes:

- rising and/or falling edge capture,
- software timer
- high-speed output, or
- pulse width modulator

Module 4 can also be programmed as a watchdog timer (see Section "PCA Watchdog Timer", page 48).

When the compare/capture modules are programmed in the capture mode, software timer, or high speed output mode, an interrupt can be generated when the module executes its function. All five modules plus the PCA timer overflow share one interrupt vector.

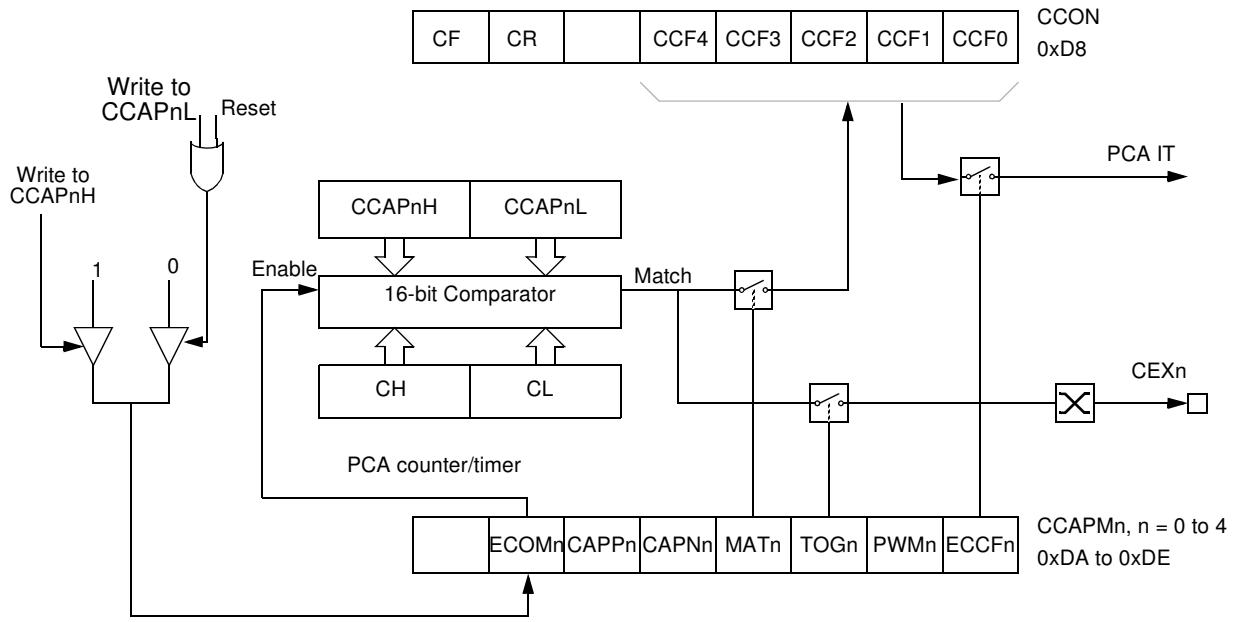
The PCA timer/counter and compare/capture modules share Port 1 for external I/O. These pins are listed below. If the port pin is not used for the PCA, it can still be used for standard I/O.

| PCA Component | External I/O Pin |
|---|---|
| 16-bit Counter | P1.2/ECI |
| 16-bit Module 0 | P1.3/CEX0 |
| 16-bit Module 1 | P1.4/CEX1 |
| 16-bit Module 2 | P1.5/CEX2 |
| 16-bit Module 3 | P1.6/CEX3 |
| 16-bit Module 4 | P1.7/CEX4 |

The PCA timer is a common time base for all five modules (see Figure 13-1). The timer count source is determined from the CPS1 and CPS0 bits in the CMOD register (Table 13-1) and can be programmed to run at:

- 1/6 the peripheral clock frequency ($F_{CLK\ PERIPH}$).
- 1/2 the peripheral clock frequency ($F_{CLK\ PERIPH}$).
- The Timer 0 overflow
- The input on the ECI pin (P1.2)

**Figure 13-5.** PCA High-speed Output Mode




Before enabling ECOM bit, CCAPnL and CCAPnH should be set with a non zero value, otherwise an unwanted match could happen.

Once ECOM set, writing CCAPnL will clear ECOM so that an unwanted match doesn't occur while modifying the compare value. Writing to CCAPnH will set ECOM. For this reason, user software should write CCAPnL first, and then CCAPnH. Of course, the ECOM bit can still be controlled by accessing to CCAPMn register.

## 13.4 Pulse Width Modulator Mode

All of the PCA modules can be used as PWM outputs. Figure 13-6 shows the PWM function. The frequency of the output depends on the source for the PCA timer. All of the modules will have the same frequency of output because they all share the PCA timer. The duty cycle of each module is independently variable using the module's capture register CCAPLn. When the value of the PCA CL SFR is less than the value in the module's CCAPLn SFR the output will be low, when it is equal to or greater than the output will be high. When CL overflows from FF to 00, CCAPLn is reloaded with the value in CCAPHn. 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.

Example of computed value when X2 = 1, SMOD1 = 1, SPD = 1

| Baud Rates | $F_{osc}$ = 16.384 MHz | | $F_{osc}$ = 24 MHz | |
|---|---|---|---|---|
| | BRL | Error (%) | BRL | Error (%) |
| 115200 | 247 | 1.23 | 243 | 0.16 |
| 57600 | 238 | 1.23 | 230 | 0.16 |
| 38400 | 229 | 1.23 | 217 | 0.16 |
| 28800 | 220 | 1.23 | 204 | 0.16 |
| 19200 | 203 | 0.63 | 178 | 0.16 |
| 9600 | 149 | 0.31 | 100 | 0.16 |
| 4800 | 43 | 1.23 | - | - |

Example of computed value when X2 = 0, SMOD1 = 0, SPD = 0

| Baud Rates | $F_{osc}$ = 16.384 MHz | | $F_{osc}$ = 24 MHz | |
|---|---|---|---|---|
| | BRL | Error (%) | BRL | Error (%) |
| 4800 | 247 | 1.23 | 243 | 0.16 |
| 2400 | 238 | 1.23 | 230 | 0.16 |
| 1200 | 220 | 1.23 | 202 | 3.55 |
| 600 | 185 | 0.16 | 152 | 0.16 |

The baud rate generator can be used for mode 1 or 3 (refer to Figure 14-4.), but also for mode 0 for UART, thanks to the bit SRC located in BDRCON register (Table 14-4.)

## 14.4 UART Registers

SADEN - Slave Address Mask Register for UART (B9h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |

Reset Value = 0000 0000b

SADDR - Slave Address Register for UART (A9h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |

Reset Value = 0000 0000b

SBUF - Serial Buffer Register for UART (99h)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |

Reset Value = XXXX XXXXb

**Table 16-4.** IPH0 Register

IPH0 - Interrupt Priority High Register (B7h)

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

| Bit Number | Bit Mnemonic | Description |
|---|---|---|
| 7 | - | **Reserved**<br>The value read from this bit is indeterminate. Do not set this bit. |
| 6 | PPCH | **PCA interrupt Priority high bit.**<br>PPCH PPCL Priority Level<br>0 0 Lowest<br>0 1<br>1 0<br>1 1 Highest |
| 5 | PT2H | **Timer 2 overflow interrupt Priority High bit**<br>PT2H PT2L Priority Level<br>0 0 Lowest<br>0 1<br>1 0<br>1 1 Highest |
| 4 | PSH | **Serial port Priority High bit**<br>PSH PSL Priority Level<br>0 0 Lowest<br>0 1<br>1 0<br>1 1 Highest |
| 3 | PT1H | **Timer 1 overflow interrupt Priority High bit**<br>PT1H PT1L Priority Level<br>0 0 Lowest<br>0 1<br>1 0<br>1 1 Highest |
| 2 | PX1H | **External interrupt 1 Priority High bit**<br>PX1H PX1L Priority Level<br>0 0 Lowest<br>0 1<br>1 0<br>1 1 Highest |
| 1 | PT0H | **Timer 0 overflow interrupt Priority High bit**<br>PT0H PT0L Priority Level<br>0 0 Lowest<br>0 1<br>1 0<br>1 1 Highest |
| 0 | PX0H | **External interrupt 0 Priority High bit**<br>PX0H PX0L Priority Level<br>0 0 Lowest<br>0 1<br>1 0<br>1 1 Highest |

Reset Value = x000 0000b

Not bit addressable

**Figure 19-4.** Data Transmission Format (CPHA = 0)

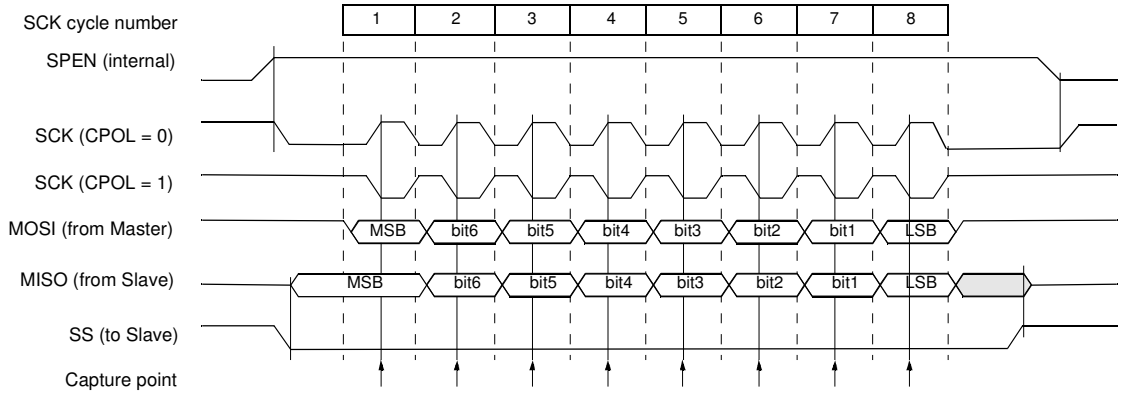



**Figure 19-5.** Data Transmission Format (CPHA = 1)

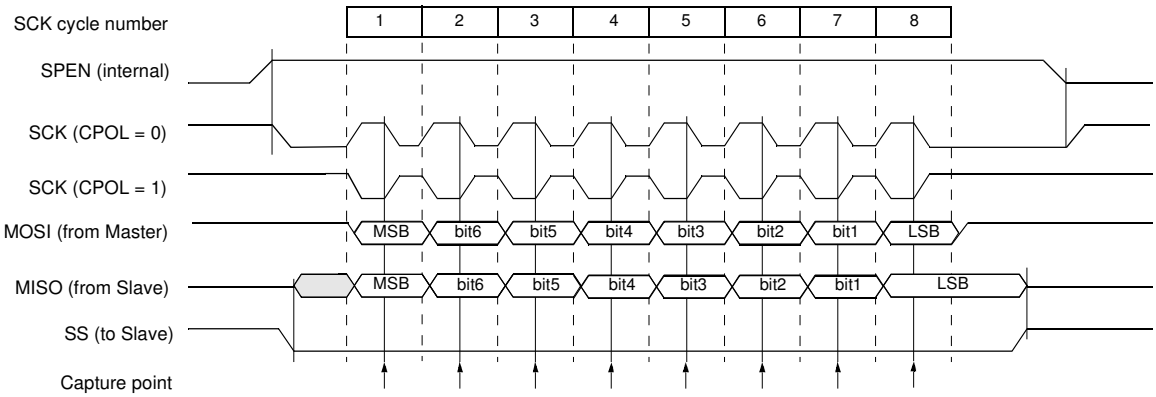
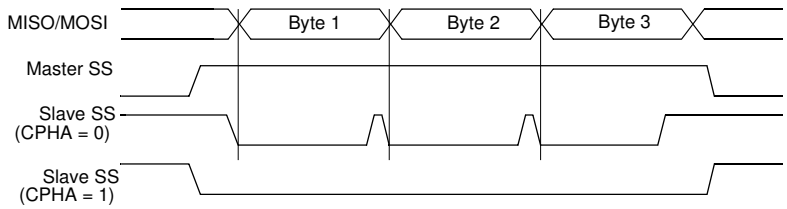



**Figure 19-6.** CPHA/$\overline{SS}$ Timing




As shown in Figure 19-5, the first SCK edge is the MSB capture strobe. Therefore the Slave must begin driving its data before the first SCK edge, and a falling edge on the $\overline{SS}$ pin is used to start the transmission. The $\overline{SS}$ pin must be toggled high and then low between each byte transmitted (Figure 19-2).

Figure 19-6 shows an SPI transmission in which CPHA is'1'. In this case, the Master begins driving its MOSI pin on the first SCK edge. Therefore the Slave uses the first SCK edge as a start transmission signal. The $\overline{SS}$ pin can remain low between transmissions (Figure 19-1). This format may be preferable in systems having only one Master and only one Slave driving the MISO data line.

### 19.3.3 Error Conditions

The following flags in the SPSTA signal SPI error conditions:

**Figure 20-2.** Block Diagram

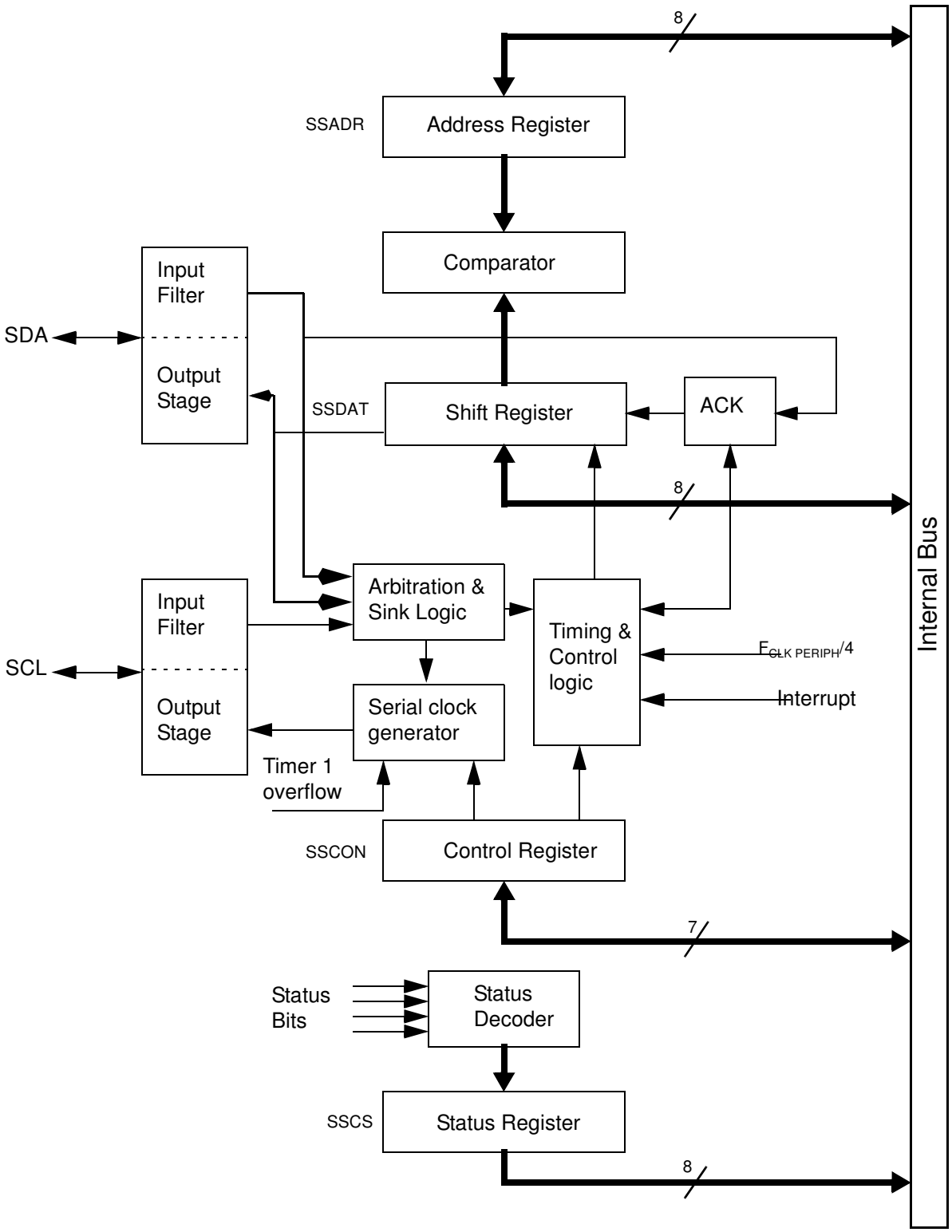


**Table 20-7.** Status in Slave Receiver Mode (Continued)

| Status Code (SSCS) | Status of the 2-wire bus and 2-wire hardware | Application Software Response | | | | | Next Action Taken By 2-wire Software |
|---|---|---|---|---|---|---|---|
| | | To/from SSDAT | To SSCON | | | | |
| | | | STA | STO | SI | AA | |
| 98h | Previously addressed with general call; data has been received; NOT ACK has been returned | Read data byte or | 0 | 0 | 0 | 0 | Switched to the not addressed slave mode; no recognition of own SLA or GCA |
| | | Read data byte or | 0 | 0 | 0 | 1 | Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1 |
| | | Read data byte or | 1 | 0 | 0 | 0 | Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free |
| | | Read data byte | 1 | 0 | 0 | 1 | Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1. A START condition will be transmitted when the bus becomes free |
| A0h | A STOP condition or repeated START condition has been received while still addressed as slave | No SSDAT action or | 0 | 0 | 0 | 0 | Switched to the not addressed slave mode; no recognition of own SLA or GCA |
| | | No SSDAT action or | 0 | 0 | 0 | 1 | Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1 |
| | | No SSDAT action or | 1 | 0 | 0 | 0 | Switched to the not addressed slave mode; no recognition of own SLA or GCA. A START condition will be transmitted when the bus becomes free |
| | | No SSDAT action | 1 | 0 | 0 | 1 | Switched to the not addressed slave mode; own SLA will be recognised; GCA will be recognised if GC=logic 1. A START condition will be transmitted when the bus becomes free |

- Set configuration

  The CONFG bit in the USBCON register has to be set after a SET_CONFIGURATION request with a non-zero value. Otherwise, this bit has to be cleared.

### 21.2.2 Endpoint Configuration

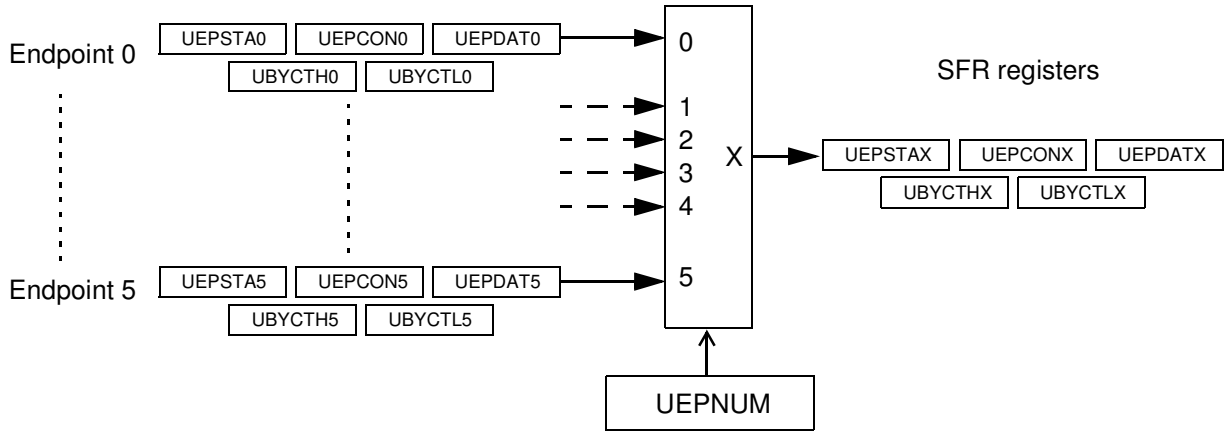
- Selection of an Endpoint

  The endpoint register access is performed using the UEPNUM register. The registers

  – UEPSTAX
  – UEPCONX
  – UEPDATX
  – UBYCTLX
  – UBYCTHX

  These registers correspond to the endpoint whose number is stored in the UEPNUM register. To select an Endpoint, the firmware has to write the endpoint number in the UEPNUM register.

**Figure 21-5.** Endpoint Selection




- Endpoint enable

  Before using an endpoint, this one will be enabled by setting the EPEN bit in the UEPCONX register.

  An endpoint which is not enabled won't answer to any USB request. The Default Control Endpoint (Endpoint 0) will always be enabled in order to answer to USB standard requests.

- Endpoint type configuration

  All Standard Endpoints can be configured in Control, Bulk, Interrupt or Isochronous mode. The Ping-pong Endpoints can be configured in Bulk, Interrupt or Isochronous mode. The configuration of an endpoint is performed by setting the field EPTYPE with the following values:

  – Control:EPTYPE = 00b
  – Isochronous:EPTYPE = 01b
  – Bulk:EPTYPE = 10b
  – Interrupt:EPTYPE = 11b

### 21.7.2 STALL Handshake

This function is only available for Control, Bulk, and Interrupt endpoints.

The firmware has to set the STALLRQ bit in the UEPSTAX register to send a STALL handshake at the next request of the Host on the endpoint selected with the UEPNUM register. The RXSETUP, TXRDY, TXCMPL, RXOUTB0 and RXOUTB1 bits must be first reset to 0. The bit STLCRC is set at 1 by the USB controller when a STALL has been sent. This triggers an interrupt if enabled.
The firmware will clear the STALLRQ and STLCRC bits after each STALL sent.

The STALLRQ bit is cleared automatically by hardware when a valid SETUP PID is received on a CONTROL type endpoint.

Important note: when a Clear Halt Feature occurs for an endpoint, the firmware will reset this endpoint using the UEPRST register in order to reset the data toggle management.

### 21.7.3 Start of Frame Detection

The SOFINT bit in the USBINT register is set when the USB controller detects a Start of Frame PID. This triggers an interrupt if enabled. The firmware will clear the SOFINT bit to allow the next Start of Frame detection.

### 21.7.4 Frame Number

When receiving a Start of Frame, the frame number is automatically stored in the UFNUML and UFNUMH registers. The CRCOK and CRCERR bits indicate if the CRC of the last Start of Frame is valid (CRCOK set at 1) or corrupted (CRCERR set at 1). The UFNUML and UFNUMH registers are automatically updated when receiving a new Start of Frame.

### 21.7.5 Data Toggle Bit

The Data Toggle bit is set by hardware when a DATA0 packet is received and accepted by the USB controller and cleared by hardware when a DATA1 packet is received and accepted by the USB controller. This bit is reset when the firmware resets the endpoint FIFO using the UEPRST register.

For Control endpoints, each SETUP transaction starts with a DATA0 and data toggling is then used as for Bulk endpoints until the end of the Data stage (for a control write transfer). The Status stage completes the data transfer with a DATA1 (for a control read transfer).

For Isochronous endpoints, the device firmware will ignore the data-toggle.

## 21.8 Suspend/Resume Management

### 21.8.1 Suspend

The Suspend state can be detected by the USB controller if all the clocks are enabled and if the USB controller is enabled. The bit SPINT is set by hardware when an idle state is detected for more than 3 ms. This triggers a USB interrupt if enabled.

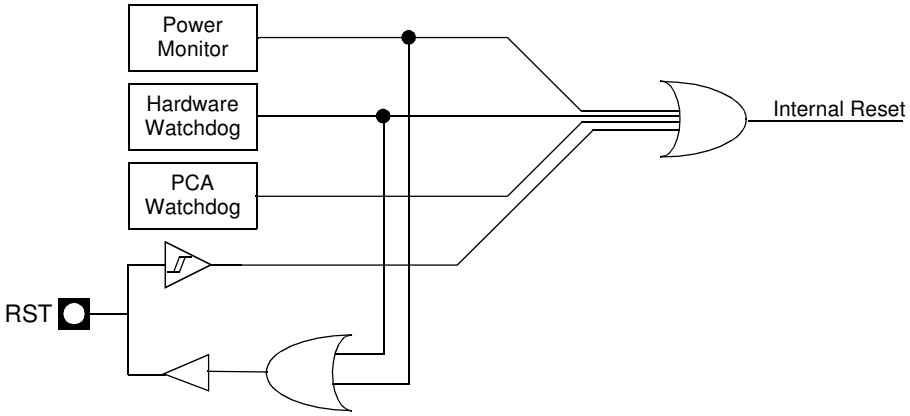
In order to reduce current consumption, the firmware can put the USB PAD in idle mode, stop the clocks and put the C51 in Idle or Power-down mode. The Resume detection is still active.
The USB PAD is put in idle mode when the firmware clear the SPINT bit. In order to avoid a new suspend detection 3ms later, the firmware has to disable the USB clock input using the SUSPCLK bit in the USBCON Register. The USB PAD automatically exits of idle mode when a wake-up event is detected.

# 22. Reset

## 22.1 Introduction

The reset sources are: Power Management, Hardware Watchdog, PCA Watchdog and Reset input.

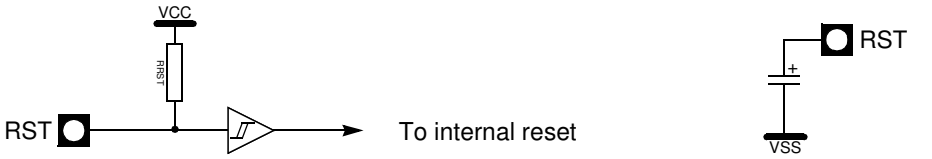
**Figure 22-1.** Reset schematic




## 22.2 Reset Input

The Reset input can be used to force a reset pulse longer than the internal reset controlled by the Power Monitor. RST input has a pull-up resistor allowing power-on reset by simply connecting an external capacitor to $V_{SS}$ as shown in Figure 22-2. Resistor value and input characteristics are discussed in the Section “DC Characteristics” of the AT83C5134/35/36 datasheet.

**Figure 22-2.** Reset Circuitry and Power-On Reset



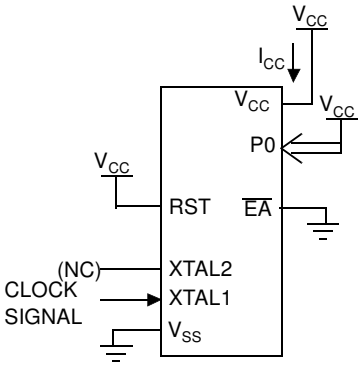

a. RST input circuitry

b. Power-on Reset

## 22.3 Reset Output

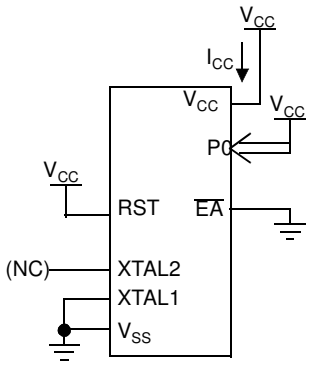
As detailed in Section “Hardware Watchdog Timer”, page 138, the WDT generates a 96-clock period pulse on the RST pin. In order to properly propagate this pulse to the rest of the application in case of external capacitor or power-supply supervisor circuit, a 1 kΩ resistor must be added as shown Figure 22-3.

**Figure 27-2.** $I_{CC}$ Test Condition, Idle Mode



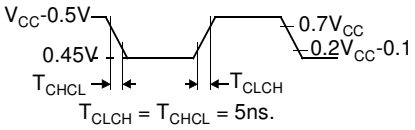

All other pins are disconnected.

**Figure 27-3.** $I_{CC}$ Test Condition, Power-down Mode




All other pins are disconnected.

**Figure 27-4.** Clock Signal Waveform for $I_{CC}$ Tests in Active and Idle Modes




### 27.2.1 LED's

**Table 27-1.** LED Outputs DC Parameters

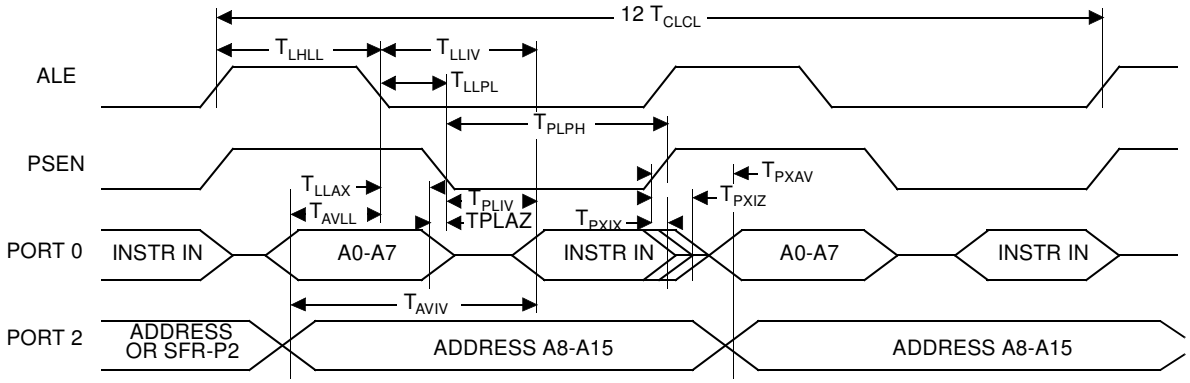
| Symbol | Parameter | Min | Typ | Max | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| $I_{OL}$ | Output Low Current, P3.6 and P3.7 LED modes | 1 | 2 | 4 | mA | 2 mA configuration |
| | | 2 | 4 | 8 | mA | 4 mA configuration |
| | | 5 | 10 | 20 | mA | 10 mA configuration |

Note: 1. ($T_A$ = -20°C to +50°C, $V_{CC}$ - $V_{OL}$ = 2 V ± 20%)

**Table 27-4.** AC Parameters for a Variable Clock

| Symbol | Type | Standard Clock | X2 Clock | X Parameter | Units |
|---|---|---|---|---|---|
| $T_{LHLL}$ | Min | 2 T - x | T - x | 10 | ns |
| $T_{AVLL}$ | Min | T - x | 0.5 T - x | 15 | ns |
| $T_{LLAX}$ | Min | T - x | 0.5 T - x | 15 | ns |
| $T_{LLIV}$ | Max | 4 T - x | 2 T - x | 30 | ns |
| $T_{LLPL}$ | Min | T - x | 0.5 T - x | 10 | ns |
| $T_{PLPH}$ | Min | 3 T - x | 1.5 T - x | 20 | ns |
| $T_{PLIV}$ | Max | 3 T - x | 1.5 T - x | 40 | ns |
| $T_{PXIX}$ | Min | x | x | 0 | ns |
| $T_{PXIZ}$ | Max | T - x | 0.5 T - x | 7 | ns |
| $T_{AVIV}$ | Max | 5 T - x | 2.5 T - x | 40 | ns |
| $T_{PLAZ}$ | Max | x | x | 10 | ns |

### 27.4.3 External Program Memory Read Cycle



### 27.6.0.2 Timings

Test conditions: capacitive load on all pins= 50 pF.

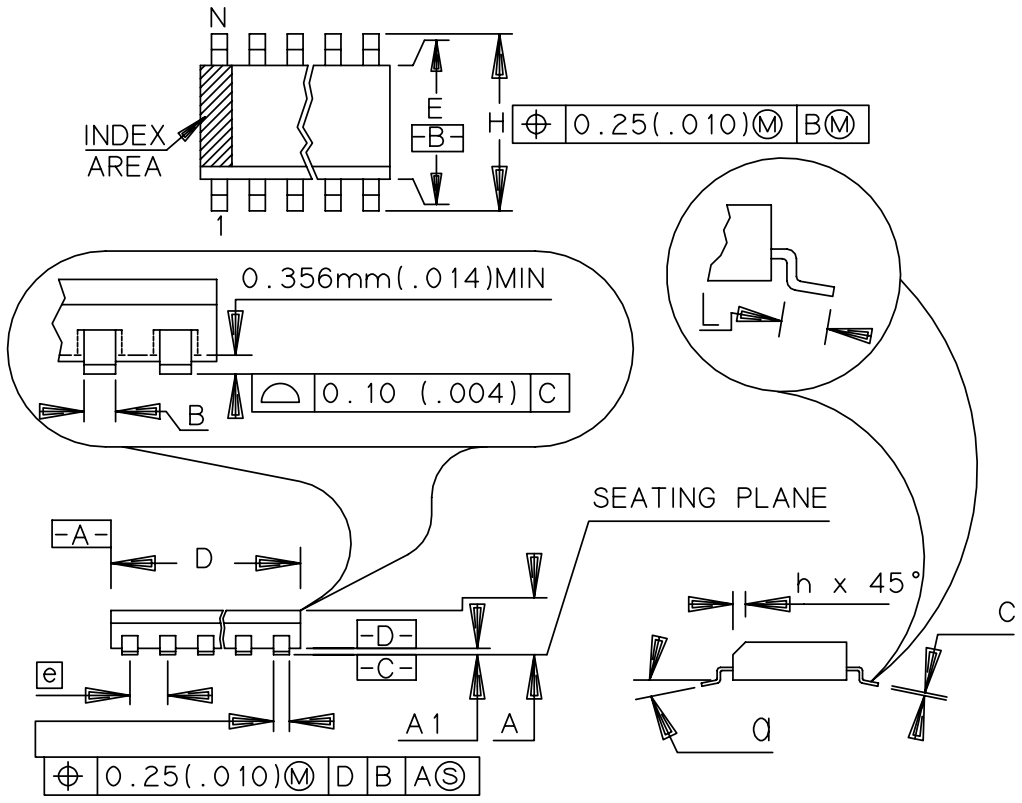
**Table 27-15.** SPI Interface Master AC Timing

**$V_{DD}$ = 2.7 to 5.5 V, $T_A$ = -40 to +85°C**

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| **Slave Mode** | | | | |
| $T_{CHCH}$ | Clock Period | 2 | | $T_{PER}$ |
| $T_{CHCX}$ | Clock High Time | 0.8 | | $T_{PER}$ |
| $T_{CLCX}$ | Clock Low Time | 0.8 | | $T_{PER}$ |
| $T_{SLCH}$, $T_{SLCL}$ | $\overline{SS}$ Low to Clock edge | 100 | | ns |
| $T_{IVCL}$, $T_{IVCH}$ | Input Data Valid to Clock Edge | 50 | | ns |
| $T_{CLIX}$, $T_{CHIX}$ | Input Data Hold after Clock Edge | 50 | | ns |
| $T_{CLOV,}$ $T_{CHOV}$ | Output Data Valid after Clock Edge | | 50 | ns |
| $T_{CLOX}$, $T_{CHOX}$ | Output Data Hold Time after Clock Edge | 0 | | ns |
| $T_{CLSH}$, $T_{CHSH}$ | $\overline{SS}$ High after Clock Edge | 0 | | ns |
| $T_{SLOV}$ | $\overline{SS}$ Low to Output Data Valid | | $4T_{PER}$+20 | ns |
| $T_{SHOX}$ | Output Data Hold after $\overline{SS}$ High | | $2T_{PER}$+100 | ns |
| $T_{SHSL}$ | $\overline{SS}$ High to $\overline{SS}$ Low | $2T_{PER}$+120 | | |
| $T_{ILIH}$ | Input Rise Time | | 2 | µs |
| $T_{IHIL}$ | Input Fall Time | | 2 | µs |
| $T_{OLOH}$ | Output Rise time | | 100 | ns |
| $T_{OHOL}$ | Output Fall Time | | 100 | ns |
| **Master Mode** | | | | |
| $T_{CHCH}$ | Clock Period | 4 | | $T_{PER}$ |
| $T_{CHCX}$ | Clock High Time | $2T_{PER}$-20 | | ns |
| $T_{CLCX}$ | Clock Low Time | $2T_{PER}$-20 | | ns |
| $T_{IVCL}$, $T_{IVCH}$ | Input Data Valid to Clock Edge | 50 | | ns |
| $T_{CLIX}$, $T_{CHIX}$ | Input Data Hold after Clock Edge | 50 | | ns |
| $T_{CLOV,}$ $T_{CHOV}$ | Output Data Valid after Clock Edge | | 20 | ns |
| $T_{CLOX}$, $T_{CHOX}$ | Output Data Hold Time after Clock Edge | 0 | | ns |

Note: $T_{PER}$ is XTAL period when SPI interface operates in X2 mode or twice XTAL period when SPI interface operates in X1 mode.

## 29.3 28-lead SO




| | MM | | INCH | |
|---|---|---|---|---|
| A | 2.35 | 2.65 | .093 | .104 |
| A1 | 0.10 | 0.30 | .004 | .012 |
| B | 0.35 | 0.49 | .014 | .019 |
| C | 0.23 | 0.32 | .009 | .013 |
| D | 17.70 | 18.10 | .697 | .713 |
| E | 7.40 | 7.60 | .291 | .299 |
| e | 1.27 | BSC | .050 | BSC |
| H | 10.00 | 10.65 | .394 | .419 |
| h | 0.25 | 0.75 | .010 | .029 |
| L | 0.40 | 1.27 | .016 | .050 |
| N | 28 | | 28 | |
| α | 0° | | 8° | |

07/27/07

| | TITLE | DRAWING No. | REV. |
|---|---|---|---|
| Atmel Nantes S.A.<br>La Chantrerie - BP 70602<br>44306 Nantes Cedex 3 - France | TI, 28 - Lead, 0.300" Body Width<br>Plastic Gull Wing Small Outline Package (SOIC) | TI | N |

# 30. Document Revision History

## 30.1 Changes from Rev A. to Rev. B

1. Added QFN32 package.

## 30.2 Changes from Rev B. to Rev. C

1. Updated package drawings.