**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 | Active |
| Core Processor | 8052 |
| Core Size | 8-Bit |
| Speed | 16MHz |
| Connectivity | EBI/EMI, I²C, SPI, UART/USART |
| Peripherals | PSM, Temp Sensor, WDT |
| Number of I/O | 34 |
| Program Memory Size | 62KB (62K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 4K x 8 |
| RAM Size | 2.25K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 5.5V |
| Data Converters | A/D 8x12b; D/A 2x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 52-QFP |
| Supplier Device Package | 52-MQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/analog-devices/aduc831bsz-reel |

# ADUC831* PRODUCT PAGE QUICK LINKS

**Last Content Update: 02/23/2017**

## COMPARABLE PARTS

View a parametric search of comparable parts.

## EVALUATION KITS

- ADuC831 QuickStart Development System

## DOCUMENTATION

**Application Notes**

- AN-1074: Understanding the Serial Download Protocol (Formerly uC004)
- AN-1139: Understanding the Parallel Programming Protocol
- AN-282: Fundamentals of Sampled Data Systems
- AN-660: XY-Matrix Keypad Interface to MicroConverter®
- AN-709: RTD Interfacing and Linearization Using an ADuC8xx MicroConverter®
- AN-759: Expanding the Number of DAC Outputs on the ADuC8xx and ADuC702x Families (uC012)
- UC-001: MicroConverter® I2C® Compatible Interface
- UC-006: A 4-wire UART-to-PC Interface
- UC-007: User Download (ULOAD) Mode
- UC-009: Addressing 16MB of External Data Memory
- UC-018: Uses of the Time Interval Counter
- UC-019: DMA To XRAM on the ADuC831/832
- UC-020: Migrating to the ADuC831 from the ADuC812

**Data Sheet**

- ADuC831: Errata Sheet
- ADuC831: MicroConverter®, 12-Bit ADCs and DACs with Embedded 62 kBytes Flash MCU Data Sheet

**User Guides**

- ADuC831 Quick Reference Guide
- UG-041: ADuC8xx Evaluation Kit Getting Started User Guide

## REFERENCE MATERIALS

**Technical Articles**

- Integrated Route Taken to Pulse Oximetry

## DESIGN RESOURCES

- ADUC831 Material Declaration
- PCN-PDN Information
- Quality And Reliability
- Symbols and Footprints

## DISCUSSIONS

View all ADUC831 EngineerZone Discussions.

## SAMPLE AND BUY

Visit the product page to see pricing options.

## TECHNICAL SUPPORT

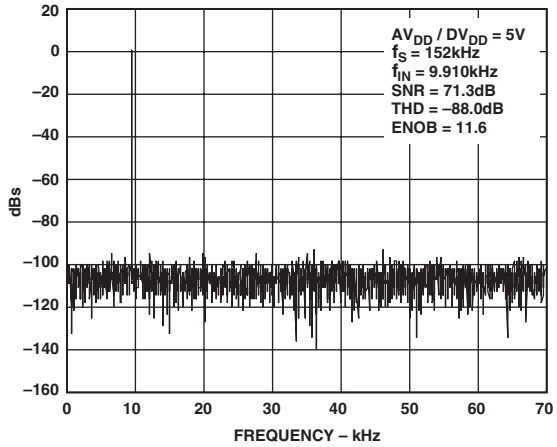
Submit a technical question or find your regional support number.

## DOCUMENT FEEDBACK

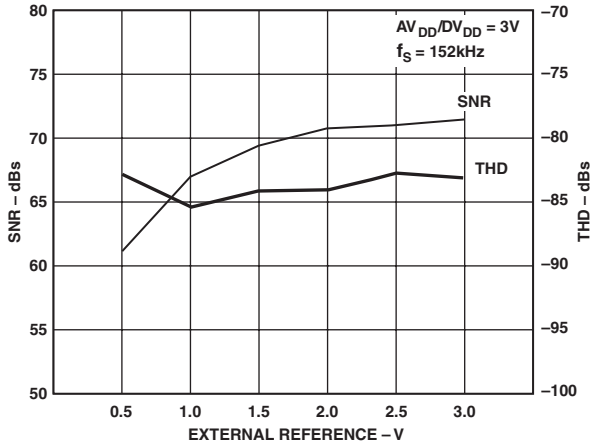
Submit feedback for this data sheet.

TABLE OF CONTENTS

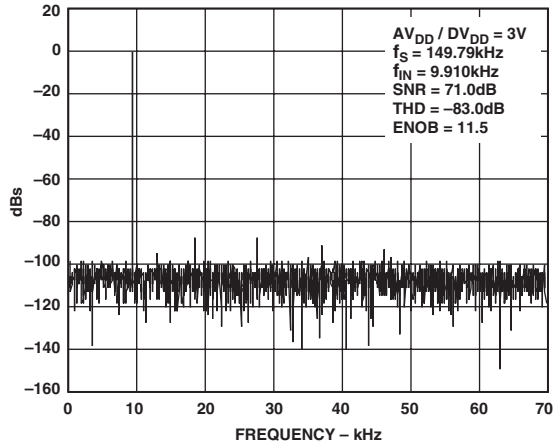
| Parameter | $V_{DD}$ = 5 V | $V_{DD}$ = 3 V | Unit | Test Conditions/Comments |
|---|---|---|---|---|
| SCLOCK and RESET Only[4] (Schmitt-Triggered Inputs) | | | | |
| $V_{T+}$ | 1.3 | 0.95 | V min | |
| | 3.0 | 2.5 | V max | |
| $V_{T-}$ | 0.8 | 0.4 | V min | |
| | 1.4 | 1.1 | V max | |
| $V_{T+} - V_{T-}$ | 0.3 | 0.3 | V min | |
| | 0.85 | 0.85 | V max | |
| CRYSTAL OSCILLATOR | | | | |
| Logic Inputs, XTAL1 Only | | | | |
| $V_{INL}$, Input Low Voltage | 0.8 | 0.4 | V typ | |
| $V_{INH}$, Input High Voltage | 3.5 | 2.5 | V typ | |
| XTAL1 Input Capacitance | 18 | 18 | pF typ | |
| XTAL2 Output Capacitance | 18 | 18 | pF typ | |
| MCU CLOCK RATE | 16 | 16 | MHz max | |
| DIGITAL OUTPUTS | | | | |
| Output High Voltage ($V_{OH}$) | 2.4 | | V min | $V_{DD}$ = 4.5 V to 5.5 V |
| | 4.0 | | V typ | $I_{SOURCE}$ = 80 μA |
| | | 2.4 | V min | $V_{DD}$ = 2.7 V to 3.3 V |
| | | 2.6 | V typ | $I_{SOURCE}$ = 20 μA |
| Output Low Voltage ($V_{OL}$) | | | | |
| ALE, Ports 0 and 2 | 0.4 | 0.4 | V max | $I_{SINK}$ = 1.6 mA |
| | 0.2 | 0.2 | V typ | $I_{SINK}$ = 1.6 mA |
| Port 3 | 0.4 | 0.4 | V max | $I_{SINK}$ = 4 mA |
| SCLOCK/SDATA | 0.4 | 0.4 | V max | $I_{SINK}$ = 8 mA, $I^2C$ Enabled |
| Floating State Leakage Current[4] | ±10 | ±10 | μA max | |
| | ±1 | ±1 | μA typ | |
| Floating State Output Capacitance | 10 | 10 | pF typ | |
| START UP TIME | | | | MCLKIN = 16 MHz |
| At Power-On | 500 | 500 | ms typ | |
| From Idle Mode | 100 | 100 | μs typ | |
| From Power-Down Mode | | | | |
| Wakeup with $\overline{INT0}$ Interrupt | 150 | 400 | μs typ | |
| Wakeup with SPI/$I^2C$ Interrupt | 150 | 400 | μs typ | |
| Wakeup with External RESET | 150 | 400 | μs typ | |
| After External RESET in Normal Mode | 30 | 30 | ms typ | |
| After WDT Reset in Normal Mode | 3 | 3 | ms typ | Controlled via WDCON SFR |



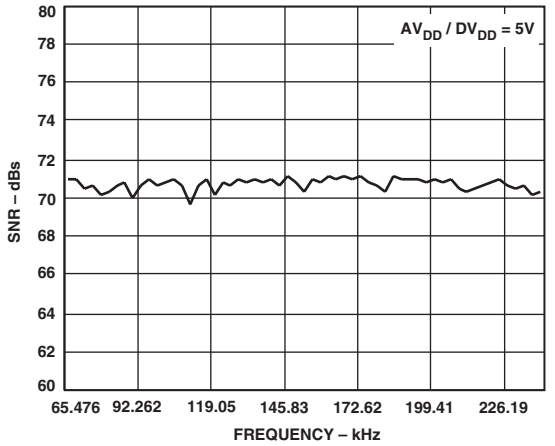

*TPC 11. Dynamic Performance at $V_{DD}$ = 5 V*



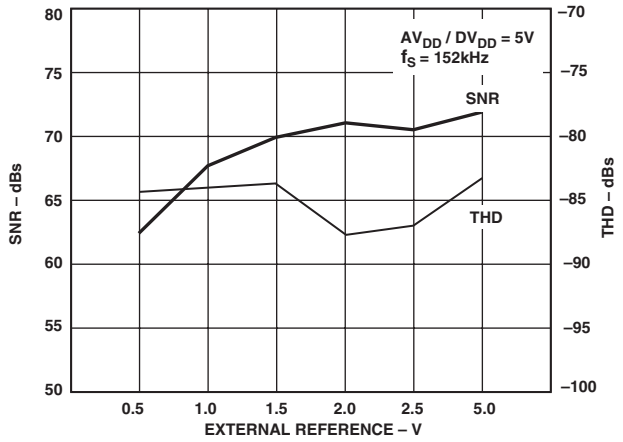

*TPC 14. Typical Dynamic Performance vs. $V_{REF}$, $V_{DD}$ = 3 V*



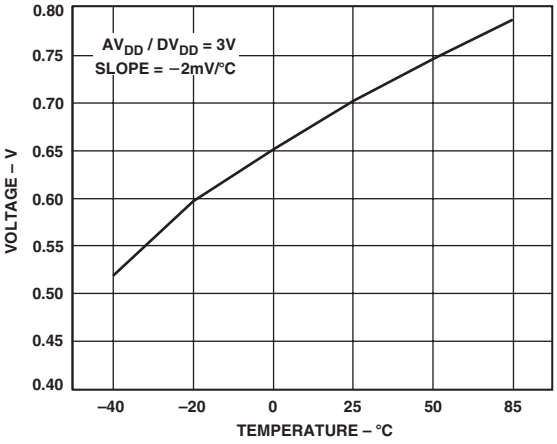

*TPC 12. Dynamic Performance at $V_{DD}$ = 3 V*




*TPC 15. Typical Dynamic Performance vs. Sampling Frequency*




*TPC 13. Typical Dynamic Performance vs. $V_{REF}$, $V_{DD}$ = 5 V*




*TPC 16. Typical Temperature Sensor Output vs. Temperature*

**External Data Memory (External XRAM)**
Just like a standard 8051 compatible core, the ADuC831 can access external data memory using a MOVX instruction. The MOVX instruction automatically outputs the various control strobes required to access the data memory.

The ADuC831, however, can access up to 16 MBytes of external data memory. This is an enhancement of the 64 kBytes external data memory space available on a standard 8051 compatible core.

The external data memory is discussed in more detail in the ADuC831 Hardware Design Considerations section.

**Internal XRAM**
2 kBytes of on-chip data memory exist on the ADuC831. This memory, although on-chip, is also accessed via the MOVX instruction. The 2 kBytes of internal XRAM are mapped into the bottom 2 kBytes of the external address space if the CFG831 bit is set. Otherwise, access to the external data memory will occur just like a standard 8051. When using the internal XRAM, ports 0 and 2 are free to be used as general-purpose I/O.

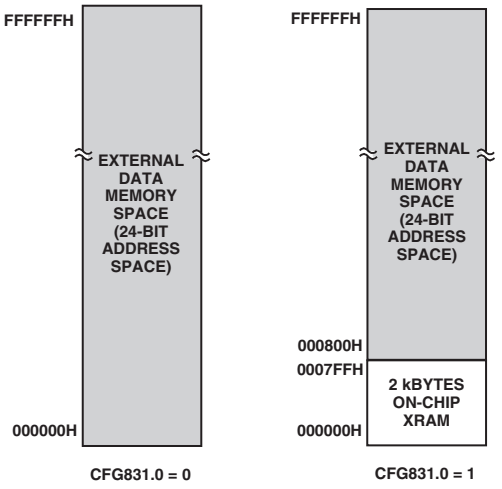



*Figure 4. Internal and External XRAM*

**SPECIAL FUNCTION REGISTERS (SFRS)**
The SFR space is mapped into the upper 128 bytes of internal data memory space and accessed by direct addressing only. It provides an interface between the CPU and all on-chip peripherals. A block diagram showing the programming model of the ADuC831 via the SFR area is shown in Figure 5.

All registers, except the Program Counter (PC) and the four general-purpose register banks, reside in the SFR area. The SFR registers include control, configuration, and data registers that provide an interface between the CPU and all on-chip peripherals.

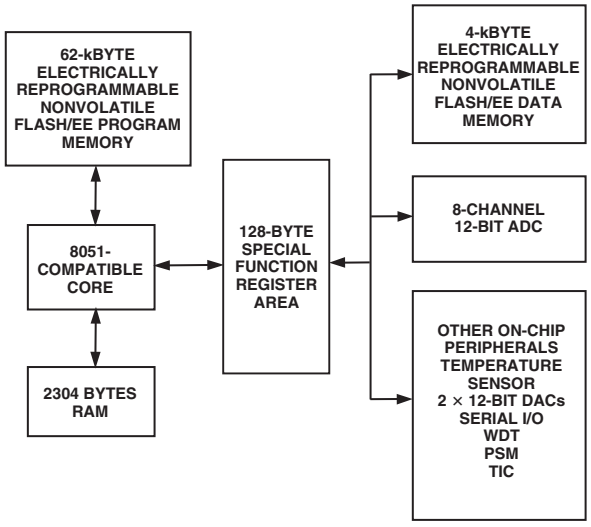



*Figure 5. Programming Model*

**Accumulator SFR (ACC)**
ACC is the Accumulator register and is used for math operations including addition, subtraction, integer multiplication and division, and Boolean bit manipulations. The mnemonics for accumulator-specific instructions refer to the Accumulator as A.

**B SFR (B)**
The B register is used with the ACC for multiplication and division operations. For other instructions it can be treated as a general-purpose scratchpad register.

**Stack Pointer (SP and SPH)**
The SP SFR is the stack pointer and is used to hold an internal RAM address that is called the *top of the stack*. The SP register is incremented before data is stored during PUSH and CALL executions. While the Stack may reside anywhere in on-chip RAM, the SP register is initialized to 07H after a reset. This causes the stack to begin at location 08H.

As mentioned earlier, the ADuC831 offers an extended 11-bit stack pointer. The three extra bits to make up the 11-bit stack pointer are the 3 LSBs of the SPH byte located at B7H.

**ADCCON1 – (ADC Control SFR #1)**
The ADCCON1 register controls conversion and acquisition times, hardware conversion modes and power-down modes as detailed below.

SFR Address: EFH
SFR Power-On Default Value: 00H
Bit Addressable: NO

**Table III. ADCCON1 SFR Bit Designations**

| Bit | Name | Description |
|---|---|---|
| ADCCON1.7 | MD1 | The Mode bit selects the active operating mode of the ADC.<br>Set by the user to power up the ADC.<br>Cleared by the user to power down the ADC. |
| ADCCON1.6 | EXT_REF | Set by the user to select an external reference.<br>Cleared by the user to use the internal reference. |
| ADCCON1.5<br>ADCCON1.4 | CK1<br>CK0 | The ADC clock divide bits (CK1, CK0) select the divide ratio for the master clock used to generate the ADC clock. To ensure correct ADC operation, the divider ratio must be chosen to reduce the ADC clock to 4.5 MHz and below. A typical ADC conversion will require 17 ADC clocks.<br>The divider ratio is selected as follows:<br>**CK1 CK0 MCLK Divider**<br>0 0 16<br>0 1 2<br>1 0 4<br>1 1 8 |
| ADCCON1.3<br>ADCCON1.2 | AQ1<br>AQ0 | The ADC acquisition select bits (AQ1, AQ0) select the time provided for the input track-and-hold amplifier to acquire the input signal. An acquisition of three or more ADC clocks is recommended; clocks are selected as follows:<br>**AQ1 AQ0 #ADC Clks**<br>0 0 1<br>0 1 2<br>1 0 3<br>1 1 4 |
| ADCCON1.1 | T2C | The Timer 2 conversion bit (T2C) is set by the user to enable the Timer 2 overflow bit be used as the ADC convert start trigger input. |
| ADCCON1.0 | EXC | The external trigger enable bit (EXC) is set by the user to allow the external Pin P3.5 (CONVST) to be used as the active low convert start input. This input should be an active low pulse (minimum pulsewidth >100 ns) at the required sample rate. |

The DMA logic operates from the ADC clock and uses pipelining to perform the ADC conversions and access the external memory at the same time. The time it takes to perform one ADC conversion is called a DMA cycle. The actions performed by the logic during a typical DMA cycle are shown in the following diagram.

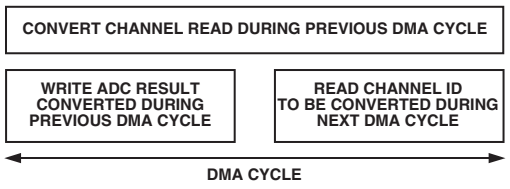



*Figure 16. DMA Cycle*

From the previous diagram, it can be seen that during one DMA cycle the following actions are performed by the DMA logic:

1. An ADC conversion is performed on the channel whose ID was read during the previous cycle.
2. The 12-bit result and the channel ID of the conversion performed in the previous cycle is written to the external memory.
3. The ID of the next channel to be converted is read from external memory.

For the previous example, the complete flow of events is shown in Figure 16. Because the DMA logic uses pipelining, it takes three cycles before the first correct result is written out.

**Micro Operation during ADC DMA Mode**

During ADC DMA mode the MicroConverter core is free to continue code execution, including general housekeeping and communication tasks. However, note that MCU core accesses to Ports 0 and 2 (which of course are being used by the DMA controller) are gated "OFF" during ADC DMA mode of operation. This means that even though the instruction that accesses the external Ports 0 or 2 will appear to execute, no data will be seen at these external ports as a result. Note that during DMA the internally contained XRAM Ports 0 and 2 are available for use.

The only case in which the MCU will be able to access XRAM during DMA, is when the internal XRAM is enabled and the section of RAM to which the DMA ADC results are being written to lies in an external XRAM. Then the MCU will be able to access the internal XRAM only. This is also the case for use of the extended stack pointer.

The MicroConverter core can be configured with an interrupt to be triggered by the DMA controller when it had finished filling the requested block of RAM with ADC results, allowing the service routine for this interrupt to postprocess data without any real-time timing constraints.

**ADC Offset and Gain Calibration Coefficients**

The ADuC831 has two ADC calibration coefficients, one for offset calibration and one for gain calibration. Both the offset and gain calibration coefficients are 14-bit words, and are each stored in two registers located in the Special Function Register (SFR) area. The offset calibration coefficient is divided into ADCOFSH (six bits) and ADCOFSL (eight bits) and the gain calibration coefficient is divided into ADCGAINH (six bits) and ADCGAINL (eight bits).

The offset calibration coefficient compensates for dc offset errors in both the ADC and the input signal. Increasing the offset coefficient compensates for positive offset, and effectively pushes the ADC transfer function down. Decreasing the offset coefficient compensates for negative offset, and effectively pushes the ADC transfer function up. The maximum offset that can be compensated is typically ±5% of $V_{REF}$, which equates to typically ±125 mV with a 2.5 V reference.

Similarly, the gain calibration coefficient compensates for dc gain errors in both the ADC and the input signal. Increasing the gain coefficient compensates for a smaller analog input signal range and scales the ADC transfer function up, effectively increasing the slope of the transfer function. Decreasing the gain coefficient, compensates for a larger analog input signal range and scales the ADC transfer function down, effectively decreasing the slope of the transfer function. The maximum analog input signal range for which the gain coefficient can compensate is $1.025 \times V_{REF}$ and the minimum input range is $0.975 \times V_{REF}$ which equates to typically ±2.5% of the reference voltage.

**CALIBRATING THE ADC**

There are two hardware calibration modes provided which can be easily initiated by user software. The ADCCON3 SFR is used to calibrate the ADC. Bit 1 (TYPICAL) and the CS3 to CS0 (ADCCON2) set up the calibration modes.

Device calibration can be initiated to compensate for significant changes in operating conditions frequency, analog input range, reference voltage and supply voltages. In this calibration mode, offset calibration uses internal AGND selected via ADCCON2 register bits CS3–CS0 (1011) and gain calibration uses internal $V_{REF}$ selected by CS3–CS0 (1100). Offset calibration should be executed first, followed by gain calibration.

System calibration can be initiated to compensate for both internal and external system errors. To perform system calibration using an external reference, tie system ground and reference to any two of the six selectable inputs. Enable external reference mode (ADCCON1.6). Select the channel connected to AGND via CS3–CS0 and perform system offset calibration. Select the channel connected to $V_{REF}$ via CS3–CS0 and perform system gain calibration.

The ADC should be configured to use settings for an ADCCLK of divide by 16 and 4 acquisition clocks.

## USER INTERFACE TO OTHER ON-CHIP ADuC831 PERIPHERALS

The following section gives a brief overview of the various peripherals also available on-chip. A summary of the SFRs used to control and configure these peripherals is also given.

### DAC

The ADuC831 incorporates two 12-bit, voltage output DACs on-chip. Each has a rail-to-rail voltage output buffer capable of driving 10 kΩ/100 pF. Each has two selectable ranges, 0 V to $V_{REF}$ (the internal band gap 2.5 V reference) and 0 V to $AV_{DD}$. Each can operate in 12-bit or 8-bit mode. Both DACs share a control register, DACCON, and four data registers, DAC1H/L, DAC0H/L. It should be noted that in 12-bit asynchronous mode, the DAC voltage output will be updated as soon as the DACL data SFR has been written; therefore, the DAC data registers should be updated as DACH first, followed by DACL. Note: for correct DAC operation on the 0 to $V_{REF}$ range, the ADC must be switched on. This results in the DAC using the correct reference value.

| **DACCON** | **DAC Control Register** |
|---|---|
| SFR Address | FDH |
| Power-On Default Value | 04H |
| Bit Addressable | No |

**Table IX. DACCON SFR Bit Designations**

| Bit | Name | Description |
|---|---|---|
| 7 | MODE | The DAC MODE bit sets the overriding operating mode for both DACs.<br>Set to "1" = 8-Bit Mode (Write 8 Bits to DACxL SFR).<br>Set to "0"= 12-Bit Mode. |
| 6 | RNG1 | DAC1 Range Select Bit.<br>Set to "1" = DAC1 Range 0–$V_{DD}$.<br>Set to "0" = DAC1 Range 0–$V_{REF}$. |
| 5 | RNG0 | DAC0 Range Select Bit.<br>Set to "1" = DAC0 Range 0–$V_{DD.}$<br>Set to "0" = DAC0 Range 0–$V_{REF.}$ |
| 4 | CLR1 | DAC1 Clear Bit.<br>Set to "0" = DAC1 Output Forced to 0 V.<br>Set to "1" = DAC1 Output Normal. |
| 3 | CLR0 | DAC0 Clear Bit.<br>Set to "0" = DAC1 Output Forced to 0 V.<br>Set to "1" = DAC1 Output Normal. |
| 2 | SYNC | DAC0/1 Update Synchronization Bit.<br>When set to "1" the DAC outputs update as soon as DACxL SFRs are written. The user can simultaneously update both DACs by first updating the DACxL/H SFRs while SYNC is "0." Both DACs will then update simultaneously when the SYNC bit is set to "1." |
| 1 | $\overline{PD1}$ | DAC1 Power-Down Bit.<br>Set to "1" = Power-On DAC1.<br>Set to "0" = Power-Off DAC1. |
| 0 | PD0 | DAC0 Power-Down Bit.<br>Set to "1" = Power-On DAC0.<br>Set to "0" = Power-Off DAC0. |

| **DACxH/L** | **DAC Data Registers** | | | |
|---|---|---|---|---|
| Function | DAC Data Registers, written by user to update the DAC output. | | | |
| SFR Address | DAC0L (DAC0 Data Low Byte) | → F9H; DAC1L (DAC1 Data Low Byte) | → FBH | |
| | DAC0H (DAC0 Data High Byte) | → FAH; DAC1H(DAC1 Data High Byte) | → FCH | |
| Power-On Default Value | 00H | → All four Registers | | |
| Bit Addressable | No | → All four Registers | | |

The 12-bit DAC data should be written into DACxH/L right-justified such that DACxL contains the lower eight bits, and the lower nibble of DACxH contains the upper four bits.

**PWM MODES OF OPERATION**

**MODE 0: PWM Disabled**

The PWM is disabled, allowing P2.6 and P2.7 to be used as normal.

**MODE 1: Single Variable Resolution PWM**

In Mode 1, both the pulse length and the cycle time (period) are programmable in user code, allowing the resolution of the PWM to be variable.

PWM1H/L sets the period of the output waveform. Reducing PWM1H/L reduces the resolution of the PWM output but increases the maximum output rate of the PWM.

(For example, setting PWM1H/L to 65536 gives a 16-bit PWM with a maximum output rate of 244 Hz (16 MHz/65536). Setting PWM1H/L to 4096 gives a 12-bit PWM with a maximum output rate of 3906 Hz (16 MHz/4096).)

PWM0H/L sets the duty cycle of the PWM output waveform, as shown in the diagram below.

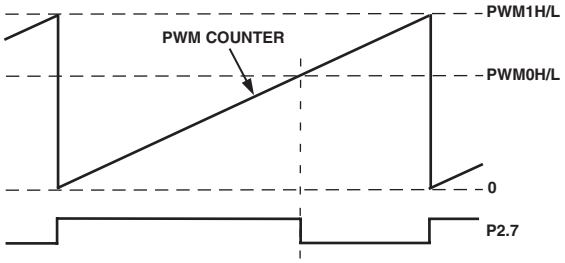



*Figure 27. ADuC831 PWM in Mode 1*

**MODE 2: Twin 8-Bit PWM**

In Mode 2, the duty cycle of the PWM outputs and the resolution of the PWM outputs are both programmable. The maximum resolution of the PWM output is eight bits.

PWM1L sets the period for both PWM outputs. Typically, this will be set to 255 (FFH) to give an 8-bit PWM, although it is possible to reduce this as necessary. A value of 100 could be loaded here to give a percentage PWM (i.e., the PWM is accurate to 1%).

The outputs of the PWM at P2.6 and P2.7 are shown in the diagram below. As can be seen, the output of PWM0 (P2.6) goes low when the PWM counter equals PWM0L. The output of PWM1 (P2.7) goes high when the PWM counter equals PWM1H, and goes low again when the PWM counter equals PWM0H. Setting PWM1H to 0 ensures that both PWM outputs start simultaneously.

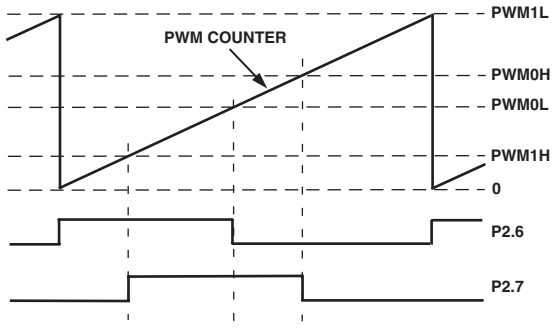



*Figure 28. PWM Mode 2*

**MODE 3: Twin 16-Bit PWM**

In Mode 3, the PWM counter is fixed to count from 0 to 65536 giving a fixed 16-bit PWM. Operating from the 16 MHz core clock results in a PWM output rate of 244 Hz. The duty cycle of the PWM outputs at P2.6 and P2.7 are independently programmable.

As shown in Figure 29, while the PWM counter is less than PWM0H/L, the output of PWM0 (P2.6) is high. Once the PWM counter equals PWM0H/L, then PWM0 (P2.6) goes low and remains low until the PWM counter rolls over.

Similarly while the PWM counter is less than PWM1H/L, the output of PWM1 (P2.7) is high. Once the PWM counter equals PWM1H/L, then PWM1 (P2.7) goes low and remains low until the PWM counter rolls over.

In this mode, both PWM outputs are synchronized. Once the PWM counter rolls over to 0, both PWM0 (P2.6) and PWM1 (P2.7) will go high.

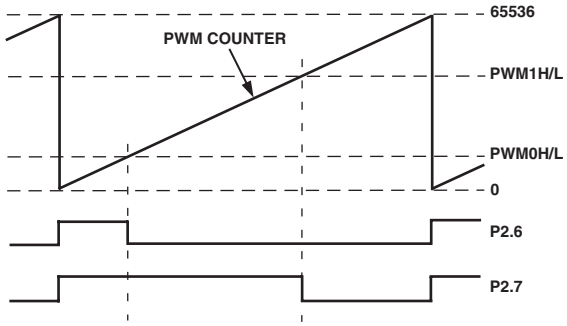



*Figure 29. PWM Mode 3*

The main features of the MicroConverter I²C interface are:

- Only two bus lines are required; a serial data line (SDATA) and a serial clock line (SCLOCK).
- An I²C master can communicate with multiple slave devices. Because each slave device has a unique 7-bit address, single master/slave relationships can exist at all times even in a multislave environment (Figure 34).
- On-Chip filtering rejects <50 ns spikes on the SDATA and the SCLOCK lines to preserve data integrity.

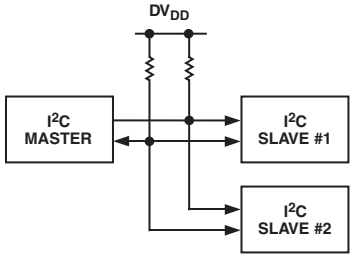



*Figure 34. Typical I²C System*

**Software Master Mode**

The ADuC831 can be used as an I²C master device by configuring the I²C peripheral in master mode and writing software to output the data bit by bit. This is referred to as a software master. Master mode is enabled by setting the I2CM bit in the I2CCON register.

To transmit data on the SDATA line, MDE must be set to enable the output driver on the SDATA pin. If MDE is set, then the SDATA pin will be pulled high or low depending on whether the MDO bit is set or cleared. MCO controls the SCLOCK pin and is always configured as an output in master mode. In master mode the SCLOCK pin will be pulled high or low depending on the whether MCO is set or cleared.

To receive data, MDE must be cleared to disable the output driver on SDATA. Software must provide the clocks by toggling the MCO bit and read the SDATA pin via the MDI bit. If MDE is cleared MDI can be used to read the SDATA pin. The value of the SDATA pin is latched into MDI on a rising edge of SCLOCK. MDI is set if the SDATA pin was high on the last rising edge of SCLOCK. MDI is cleared if the SDATA pin was low on the last rising edge of SCLOCK.

Software must control MDO, MCO, and MDE appropriately to generate the START condition, slave address, acknowledge bits, data bytes, and STOP conditions appropriately. These functions are provided in technical note uC001.

**Hardware Slave Mode**

After reset the ADuC831 defaults to hardware slave mode. The I²C interface is enabled by clearing the SPE bit in SPICON. Slave mode is enabled by clearing the I2CM bit in I2CCON. The ADuC831 has a full hardware slave. In slave mode the I²C address is stored in the I2CADD register. Data received or to be transmitted is stored in the I2CDAT register.

Once enabled in I²C slave mode the slave controller waits for a START condition. If the ADuC831 detects a valid start condition, followed by a valid address, followed by the R/$\overline{W}$ bit, the I2CI interrupt bit will get set by the hardware automatically.

The I²C peripheral will only generate a core interrupt if the user has preconfigured the I²C interrupt enable bit in the IEIP2 SFR as well as the global interrupt bit EA in the IE SFR.

```
; Enabling I2C Interrupts for the ADuC831
MOV  IEIP2,#01H     ; enable I2C interrupt
SETB EA
```

On the ADuC831 an autoclear of the I2CI bit is implemented so this bit is cleared automatically on a read or write access to the I2CDAT SFR.

```
MOV     I2CDAT, A   ; I2CI auto cleared
MOV     A, I2CDAT   ; I2CI auto cleared
```

If for any reason the user tries to clear the interrupt more than once, i.e., access the data SFR more than once per interrupt then the I²C controller will halt. The interface will then have to be reset using the I2CRS bit.

The user can choose to poll the I2CI bit or enable the interrupt. In the case of the interrupt, the PC counter will vector to 003BH at the end of each complete byte. For the first byte when the user gets to the I2CI ISR, the 7-bit address and the R/$\overline{W}$ bit will appear in the I2CDAT SFR.

The I2CTX bit contains the R/$\overline{W}$ bit sent from the master. If I2CTX is set then the master would like to receive a byte. Thus the slave will transmit data by writing to the I2CDAT register. If I2CTX is cleared, the master would like to transmit a byte. Therefore, the slave will receive a serial byte. Software can interrogate the state of I2CTX to determine whether it should write to or read from I2CDAT.

Once the ADuC831 has received a valid address, hardware will hold SCLOCK low until the I2CI bit is cleared by software. This allows the master to wait for the slave to be ready before transmitting the clocks for the next byte.

The I2CI interrupt bit will be set every time a complete data byte is received or transmitted, provided it is followed by a valid ACK. If the byte is followed by a NACK an interrupt is NOT generated. The ADuC831 will continue to issue interrupts for each complete data byte transferred until a STOP condition is received or the interface is reset.

When a STOP condition is received, the interface will reset to a state where it is waiting to be addressed (idle). Similarly, if the interface receives a NACK at the end of a sequence it also returns to the default idle state. The I2CRS bit can be used to reset the I²C interface. This bit can be used to force the interface back to the default idle state.

It should be noted that there is no way (in hardware) to distinguish between an interrupt generated by a received START + valid address and an interrupt generated by a received data byte. User software must be used to distinguish between these interrupts.

**TIME INTERVAL COUNTER (TIC)**

A time interval counter is provided on-chip for counting longer intervals than the standard 8051 compatible timers are capable of. The TIC is capable of timeout intervals ranging from 1/128 second to 255 hours. Furthermore, this counter is clocked by an internal R/C oscillator rather than the external crystal and has the ability to remain active in power-down mode and time long power-down intervals. This has obvious applications for remote battery-powered sensors where regular widely spaced readings are required. The R/C oscillator is accurate to ±10% at 25°C. Note: Instructions to the TIC SFRs are also clocked at 32 kHz, sufficient time must be allowed for in user code for these instructions to execute.

Six SFRs are associated with the time interval counter, TIMECON being its control register. Depending on the configuration of the IT0 and IT1 bits in TIMECON, the selected time counter register overflow will clock the interval counter. When this counter is equal to the time interval value loaded in the INTVAL SFR, the TII bit (TIMECON.2) is set and generates an interrupt if enabled. If the ADuC831 is in power-down mode, again with TIC interrupt enabled, the TII bit will wake up the device and resume code execution by vectoring directly to the TIC interrupt service vector address at 0053H. The TIC-related SFRs are described below. Note also that the time-base SFRs can be written initially with the current time, the TIC can then be controlled and accessed by user software. In effect, this facilitates the implementation of a real-time clock. A block diagram of the TIC is shown in Figure 35.

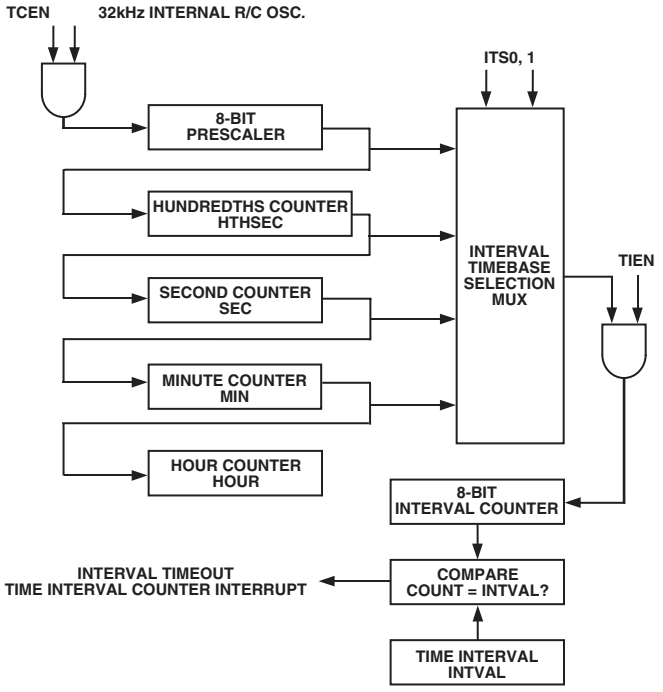



*Figure 35. TIC, Simplified Block Diagram*

| **TIMECON** | **TIC Control Register** |
|---|---|
| SFR Address | A1H |
| Power-On Default Value | 00H |
| Bit Addressable | No |

**Table XVI. TIMECON SFR Bit Designations**

| Bit | Name | Description |
|---|---|---|
| 7 | ---- | Reserved for Future Use. |
| 6 | TFH | Twenty-Four Hour Select Bit.<br>Set by the user to enable the Hour counter to count from 0 to 23.<br>Cleared by the user to enable the Hour counter to count from 0 to 255. |
| 5 | ITS1 | Interval Timebase Selection Bits. |
| 4 | ITS0 | Written by user to determine the interval counter update rate.<br>ITS1 ITS0 Interval Timebase<br>0 0 1/128 Second<br>0 1 Seconds<br>1 0 Minutes<br>1 1 Hours |
| 3 | STI | Single Time Interval Bit.<br>Set by user to generate a single interval timeout. If set, a timeout will clear the TIEN bit.<br>Cleared by user to allow the interval counter to be automatically reloaded and start counting again at each interval timeout. |
| 2 | TII | TIC Interrupt Bit.<br>Set when the 8-bit Interval Counter matches the value in the INTVAL SFR.<br>Cleared by user software. |
| 1 | TIEN | Time Interval Enable Bit.<br>Set by user to enable the 8-bit time interval counter.<br>Cleared by user to disable the interval counter. |
| 0 | TCEN | Time Clock Enable Bit.<br>Set by user to enable the time clock to the time interval counters.<br>Cleared by user to disable the clock to the time interval counters and reset the time interval SFRs to the last value written to them by the user. The time registers (HTHSEC, SEC, MIN, and HOUR) can be written while TCEN is low. |

**INTVAL** — **User Time Interval Select Register**

| | |
|---|---|
| Function | User code writes the required time interval to this register. When the 8-bit interval counter is equal to the time interval value loaded in the INTVAL SFR, the TII bit (TIMECON.2) is set and generates an interrupt if enabled. |
| SFR Address | A6H |
| Power-On Default Value | 00H |
| Bit Addressable | No |
| Valid Value | 0 to 255 decimal |

**HTHSEC** — **Hundredths Seconds Time Register**

| | |
|---|---|
| Function | This register is incremented in 1/128 second intervals once TCEN in TIMECON is active. The HTHSEC SFR counts from 0 to 127 before rolling over to increment the SEC time register. |
| SFR Address | A2H |
| Power-On Default Value | 00H |
| Bit Addressable | No |
| Valid Value | 0 to 127 decimal |

**SEC** — **Seconds Time Register**

| | |
|---|---|
| Function | This register is incremented in 1-second intervals once TCEN in TIMECON is active. The SEC SFR counts from 0 to 59 before rolling over to increment the MIN time register. |
| SFR Address | A3H |
| Power-On Default Value | 00H |
| Bit Addressable | No |
| Valid Value | 0 to 59 decimal |

**MIN** — **Minutes Time Register**

| | |
|---|---|
| Function | This register is incremented in 1-minute intervals once TCEN in TIMECON is active. The MIN counts from 0 to 59 before rolling over to increment the HOUR time register. |
| SFR Address | A4H |
| Power-On Default Value | 00H |
| Bit Addressable | No |
| Valid Value | 0 to 59 decimal |

**HOUR** — **Hours Time Register**

| | |
|---|---|
| Function | This register is incremented in 1-hour intervals once TCEN in TIMECON is active. The HOUR SFR counts from 0 to 23 before rolling over to 0. |
| SFR Address | A5H |
| Power-On Default Value | 00H |
| Bit Addressable | No |
| Valid Value | 0 to 23 decimal |

**8052 COMPATIBLE ON-CHIP PERIPHERALS**

This section gives a brief overview of the various secondary peripheral circuits that are also available to the user on-chip. These remaining functions are mostly 8052 compatible (with a few additional features) and are controlled via standard 8052 SFR bit definitions.

**Parallel I/O**

The ADuC831 uses four input/output ports to exchange data with external devices. In addition to performing general-purpose I/O, some ports are capable of external memory operations while others are multiplexed with alternate functions for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general-purpose I/O pin.

**Port 0**

Port 0 is an 8-bit, open-drain, bidirectional I/O port that is directly controlled via the Port 0 SFR. Port 0 is also the multiplexed low-order address and data bus during accesses to external program or data memory.

Figure 36 shows a typical bit latch and I/O buffer for a Port 0 port pin. The bit latch (one bit in the port's SFR) is represented as a Type D flip-flop, which will clock in a value from the internal bus in response to a "write to latch" signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a "read latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read pin" signal from the CPU. Some instructions that read a port activate the "read latch" signal, and others activate the "read pin" signal. See the following Read-Modify-Write Instructions section for more details.

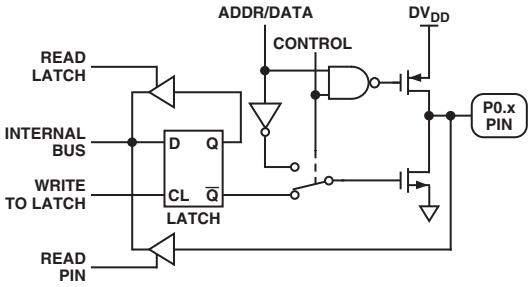



*Figure 36. Port 0 Bit Latch and I/O Buffer*

As shown in Figure 36, the output drivers of Port 0 pins are switchable to an internal ADDR and ADDR/DATA bus by an internal CONTROL signal for use in external memory accesses. During external memory accesses the P0 SFR gets 1s written to it (i.e., all of its bit latches become 1). When accessing external memory, the CONTROL signal in Figure 36 goes high, enabling push-pull operation of the output pin from the internal address or data bus (ADDR/DATA line). Therefore, no external pull-ups are required on Port 0 in order for it to access external memory.

In general-purpose I/O port mode, Port 0 pins that have 1s written to them via the Port 0 SFR will be configured as "open drain" and will therefore float. In this state, Port 0 pins can be used as high impedance inputs. This is represented in Figure 36 by the NAND gate whose output remains high as long as the CONTROL signal is low, thereby disabling the top FET. External pull-up resistors are therefore required when Port 0 pins are used as general-purpose outputs. Port 0 pins with 0s written to them will drive a logic low output voltage ($V_{OL}$) and will be capable of sinking 1.6 mA.

**Port 1**

Port 1 is also an 8-bit port directly controlled via the P1 SFR. Port 1 digital output capability is not supported on this device. Port 1 pins can be configured as digital inputs or analog inputs.

By (power-on) default these pins are configured as analog inputs, i.e., 1 written in the corresponding Port 1 register bit. To configure any of these pins as digital inputs, the user should write a 0 to these port bits to configure the corresponding pin as a high impedance digital input.

These pins also have various secondary functions described in Table XVII.

**Table XVII. Port 1, Alternate Pin Functions**

| Pin | Alternate Function |
|---|---|
| P1.0 | T2 (Timer/Counter 2 External Input) |
| P1.1 | T2EX (Timer/Counter 2 Capture/Reload Trigger) |
| P1.5 | $\overline{SS}$ (Slave Select for the SPI Interface) |

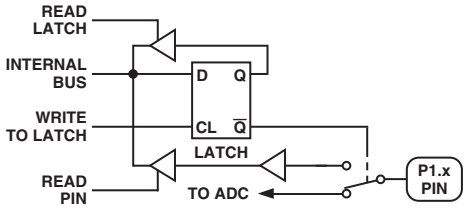



*Figure 37. Port 1 Bit Latch and I/O Buffer*

**Port 2**

Port 2 is a bidirectional port with internal pull-up resistors directly controlled via the P2 SFR. Port 2 also emits the high order address bytes during fetches from external program memory and middle and high order address bytes during accesses to the 24-bit external data memory space.

As shown in Figure 38, the output drivers of Ports 2 are switchable to an internal ADDR and ADDR/DATA bus by an internal CONTROL signal for use in external memory accesses (as for Port 0). In external memory addressing mode (CONTROL = 1) the port pins feature push-pull operation controlled by the internal address bus (ADDR line). However, unlike the P0 SFR during external memory accesses, the P2 SFR remains unchanged.

**Timer/Counter Operation Modes**

The following paragraphs describe the operating modes for Timer/Counter 2. The operating modes are selected by bits in the T2CON SFR as shown in Table XXII.

**Table XXII. T2CON Operating Modes**

| RCLK (or) TCLK | CAP2 | TR2 | Mode |
|---|---|---|---|
| 0 | 0 | 1 | 16-Bit Autoreload |
| 0 | 1 | 1 | 16-Bit Capture |
| 1 | X | 1 | Baud Rate |
| X | X | 0 | OFF |

**16-Bit Autoreload Mode**

In Autoreload mode, there are two options, which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then when Timer 2 rolls over it not only sets TF2, but also causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2 = 1, then Timer 2 still performs the above, but with the added feature that a 1-to-0 transition at external input T2EX will also trigger the 16-bit reload and set EXF2. The Autoreload mode is illustrated in Figure 49.

**16-Bit Capture Mode**

In the Capture mode, there are again two options, which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then Timer 2 is a 16-bit timer or counter which, upon overflowing, sets bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2 = 1, then Timer 2 still performs the above, but a l-to-0 transition on external input T2EX causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set, and EXF2, like TF2, can generate an interrupt. The Capture mode is illustrated in Figure 50.

The baud rate generator mode is selected by RCLK = 1 and/or TCLK = 1.

In either case, if Timer 2 is being used to generate the baud rate, the TF2 interrupt flag will not occur. Therefore, Timer 2 interrupts will not occur so they do not have to be disabled. In this mode the EXF2 flag, however, can still cause interrupts and this can be used as a third external interrupt.

Baud rate generation will be described as part of the UART serial port operation in the following pages.

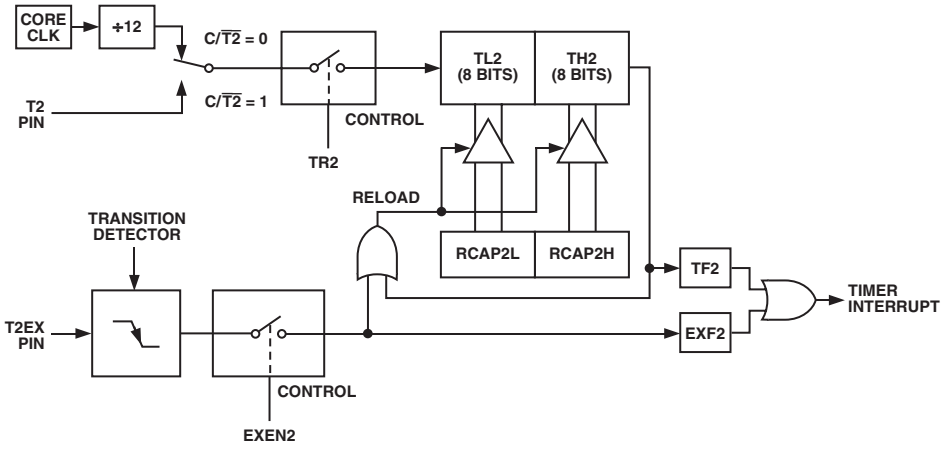



*Figure 49. Timer/Counter 2, 16-Bit Autoreload Mode*

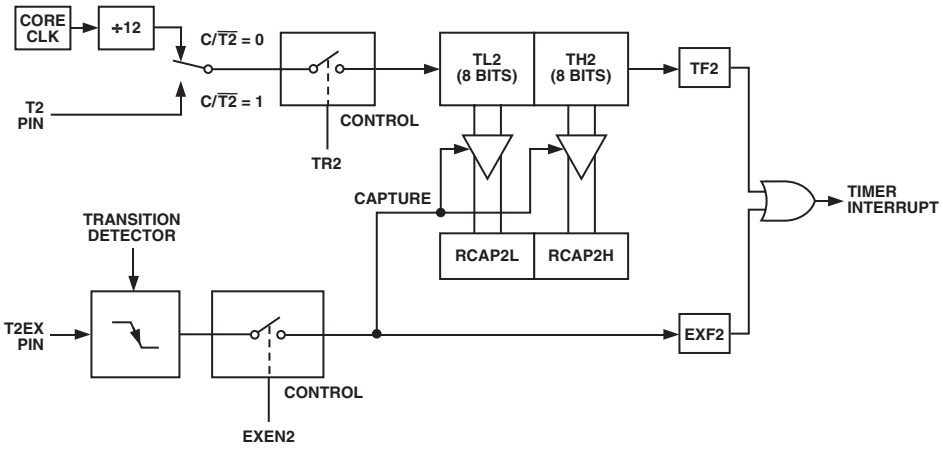



*Figure 50. Timer/Counter 2, 16-Bit Capture Mode*

## UART SERIAL INTERFACE

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the receive register. However, if the first byte still has not been read by the time reception of the second byte is complete, the first byte will be lost. The physical interface to the serial data network is via pins RXD(P3.0) and TXD(P3.1), while the SFR interface to the UART is comprised of SBUF and SCON, as described below.

### SBUF

The serial port receive and transmit registers are both accessed through the SBUF SFR (SFR address = 99H). Writing to SBUF loads the transmit register and reading SBUF accesses a physically separate receive register.

**SCON** **UART Serial Port Control Register**

| | |
|---|---|
| SFR Address | 98H |
| Power-On Default Value | 00H |
| Bit Addressable | Yes |

**Table XXIII. SCON SFR Bit Designations**

| Bit | Name | Description |
|---|---|---|
| 7 | SM0 | UART Serial Mode Select Bits. |
| 6 | SM1 | These bits select the Serial Port operating mode as follows:<br>SM0 SM1 Selected Operating Mode<br>0 0 Mode 0: Shift Register, fixed baud rate (Core_Clk/2)<br>0 1 Mode 1: 8-bit UART, variable baud rate<br>1 0 Mode 2: 9-bit UART, fixed baud rate (Core_Clk/64) or (Core_Clk/32)<br>1 1 Mode 3: 9-bit UART, variable baud rate |
| 5 | SM2 | Multiprocessor Communication Enable Bit.<br>Enables multiprocessor communication in Modes 2 and 3. In Mode 0, SM2 should be cleared. In Mode 1, if SM2 is set, RI will not be activated if a valid stop bit was not received. If SM2 is set, cleared, RI will be set as soon as the byte of data has been received. In Modes 2 or 3, if SM2 is RI will not be activated if the received ninth data bit in RB8 is 0. If SM2 is cleared, RI will be set as soon as the byte of data has been received. |
| 4 | REN | Serial Port Receive Enable Bit.<br>Set by user software to enable serial port reception.<br>Cleared by user software to disable serial port reception. |
| 3 | TB8 | Serial Port Transmit (Bit 9).<br>The data loaded into TB8 will be the ninth data bit that will be transmitted in Modes 2 and 3. |
| 2 | RB8 | Serial Port Receiver Bit 9.<br>The ninth data bit received in Modes 2 and 3 is latched into RB8. For Mode 1 the stop bit is latched into RB8. |
| 1 | TI | Serial Port Transmit Interrupt Flag.<br>Set by hardware at the end of the eighth bit in Mode 0, or at the beginning of the stop bit in Modes 1, 2, and 3. TI must be cleared by user software. |
| 0 | RI | Serial Port Receive Interrupt Flag.<br>Set by hardware at the end of the eighth bit in mode 0, or halfway through the stop bit in Modes 1, 2, and 3. RI must be cleared by software. |

**Mode 0: 8-Bit Shift Register Mode**
Mode 0 is selected by clearing both the SM0 and SM1 bits in the SFR SCON. Serial data enters and exits through RxD. TxD outputs the shift clock. Eight data bits are transmitted or received. Transmission is initiated by any instruction that writes to SBUF. The data is shifted out of the RxD line. The eight bits are transmitted with the least-significant bit (LSB) first, as shown in Figure 51.

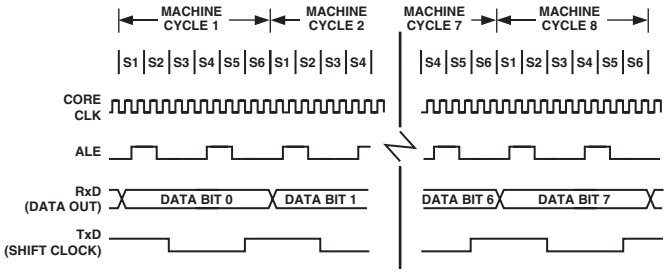



*Figure 51. UART Serial Port Transmission, Mode 0*

Reception is initiated when the receive enable bit (REN) is 1 and the receive interrupt bit (RI) is 0. When RI is cleared the data is clocked into the RxD line and the clock pulses are output from the TxD line.

**Mode 1: 8-Bit UART, Variable Baud Rate**
Mode 1 is selected by clearing SM0 and setting SM1. Each data byte (LSB first) is preceded by a start bit (0) and followed by a stop bit (1). Therefore, 10 bits are transmitted on TxD or received on RxD. The baud rate is set by the Timer 1 or Timer 2 overflow rate, or a combination of the two (one for transmission and the other for reception).

Transmission is initiated by writing to SBUF. The "write to SBUF" signal also loads a 1 (stop bit) into the ninth bit position of the transmit shift register. The data is output bit by bit until the stop bit appears on TxD and the transmit interrupt flag (TI) is automatically set as shown in Figure 52.

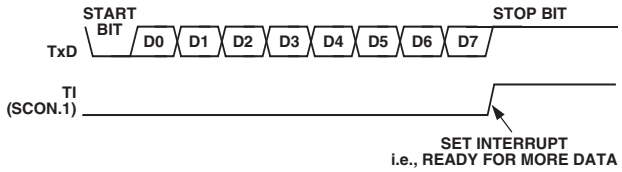



*Figure 52. UART Serial Port Transmission, Mode 0*

Reception is initiated when a 1-to-0 transition is detected on RxD. Assuming a valid start bit was detected, character reception continues. The start bit is skipped and the eight data bits are clocked into the serial port shift register. When all eight bits have been clocked in, the following events occur:

The eight bits in the receive shift register are latched into SBUF.

The ninth bit (Stop bit) is clocked into RB8 in SCON.

The Receiver Interrupt flag (RI) is set.

This will be the case if, and only if, the following conditions are met at the time the final shift pulse is generated:

RI = 0, and either SM2 = 0, or SM2 = 1 and the received stop bit = 1.

If either of these conditions is not met, the received frame is irretrievably lost, and RI is not set.

**Mode 2: 9-Bit UART with Fixed Baud Rate**
Mode 2 is selected by setting SM0 and clearing SM1. In this mode the UART operates in 9-bit mode with a fixed baud rate. The baud rate is fixed at Core_Clk/64 by default, although by setting the SMOD bit in PCON, the frequency can be doubled to Core_Clk/32. Eleven bits are transmitted or received, a start bit (0), eight data bits, a programmable ninth bit, and a stop bit (1). The ninth bit is most often used as a parity bit, although it can be used for anything, including a ninth data bit if required.

To transmit, the eight data bits must be written into SBUF. The ninth bit must be written to TB8 in SCON. When transmission is initiated, the eight data bits (from SBUF) are loaded onto the transmit shift register (LSB first). The contents of TB8 are loaded into the ninth bit position of the transmit shift register. The transmission will start at the next valid baud rate clock. The TI flag is set as soon as the stop bit appears on TxD.

Reception for Mode 2 is similar to that of Mode 1. The eight data bytes are input at RxD (LSB first) and loaded onto the receive shift register. When all eight bits have been clocked in, the following events occur:

The eight bits in the receive shift register are latched into SBUF.

The ninth data bit is latched into RB8 in SCON.

The Receiver Interrupt flag (RI) is set.

This will be the case if, and only if, the following conditions are met at the time the final shift pulse is generated:

RI = 0, and either SM2 = 0, or SM2 = 1 and the received stop bit = 1.

If either of these conditions is not met, the received frame is irretrievably lost, and RI is not set.

**Mode 3: 9-Bit UART with Variable Baud Rate**
Mode 3 is selected by setting both SM0 and SM1. In this mode, the 8051 UART serial port operates in 9-bit mode with a variable baud rate determined by either Timer 1 or Timer 2. The operation of the 9-bit UART is the same as for Mode 2 but the baud rate can be varied as for Mode 1.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

**UART Serial Port Baud Rate Generation**
***Mode 0 Baud Rate Generation***
The baud rate in Mode 0 is fixed:

$$Mode\ 0\ Baud\ Rate = (Core\ Clock\ Frequency/12)$$

***Mode 2 Baud Rate Generation***
The baud rate in Mode 2 depends on the value of the SMOD bit in the PCON SFR. If SMOD = 0, the baud rate is 1/64 of the core clock. If SMOD = 1, the baud rate is 1/32 of the core clock:

$$Mode\ 2\ Baud\ Rate = (2^{SMOD}/64) \times (Core\ Clock\ Frequency)$$

***Mode 1 and 3 Baud Rate Generation***
The baud rates in Modes 1 and 3 are determined by the overflow rate in Timer 1 or Timer 2, or both (one for transmit and the other for receive).

| Parameter | | 12 MHz Min | 12 MHz Max | Variable Clock Min | Variable Clock Max | Unit | Figure |
|---|---|---|---|---|---|---|---|
| **EXTERNAL PROGRAM MEMORY READ CYCLE** | | | | | | | |
| $t_{LHLL}$ | ALE Pulsewidth | 127 | | $2t_{CK} - 40$ | | ns | 70 |
| $t_{AVLL}$ | Address Valid to ALE Low | 43 | | $t_{CK} - 40$ | | ns | 70 |
| $t_{LLAX}$ | Address Hold after ALE Low | 53 | | $t_{CK} - 30$ | | ns | 70 |
| $t_{LLIV}$ | ALE Low to Valid Instruction In | | 234 | | $4t_{CK} - 100$ | ns | 70 |
| $t_{LLPL}$ | ALE Low to $\overline{PSEN}$ Low | 53 | | $t_{CK} - 30$ | | ns | 70 |
| $t_{PLPH}$ | $\overline{PSEN}$ Pulsewidth | 205 | | $3t_{CK} - 45$ | | ns | 70 |
| $t_{PLIV}$ | $\overline{PSEN}$ Low to Valid Instruction In | | 145 | | $3t_{CK} - 105$ | ns | 70 |
| $t_{PXIX}$ | Input Instruction Hold after $\overline{PSEN}$ | 0 | | 0 | | ns | 70 |
| $t_{PXIZ}$ | Input Instruction Float after $\overline{PSEN}$ | | 59 | | $t_{CK} - 25$ | ns | 70 |
| $t_{AVIV}$ | Address to Valid Instruction In | | 312 | | $5t_{CK} - 105$ | ns | 70 |
| $t_{PLAZ}$ | $\overline{PSEN}$ Low to Address Float | | 25 | | 25 | ns | 70 |
| $t_{PHAX}$ | Address Hold after $\overline{PSEN}$ High | 0 | | 0 | | ns | 70 |

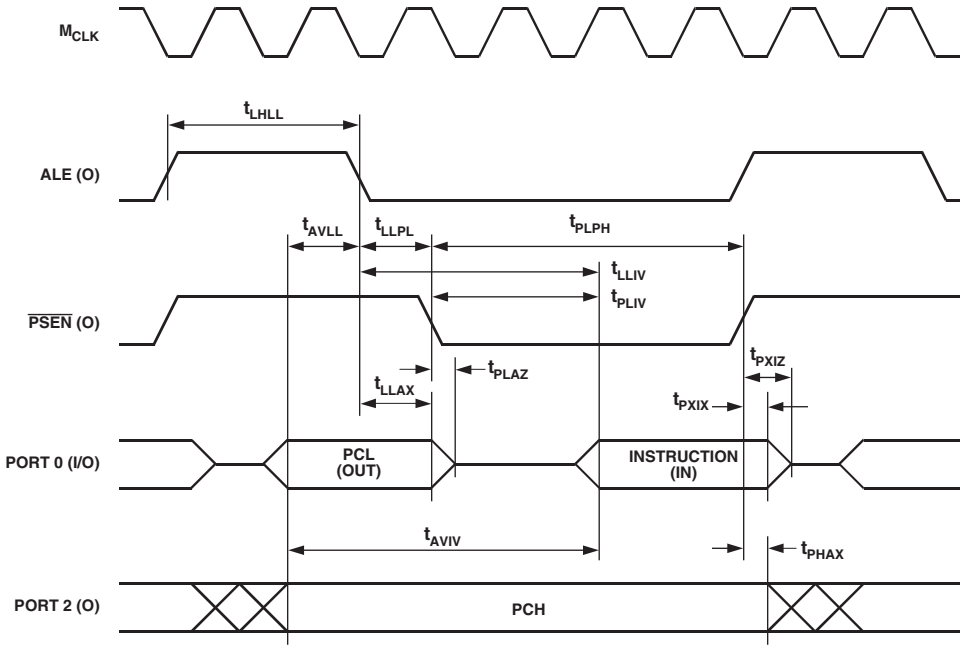



*Figure 70. External Program Memory Read Cycle*

| Parameter | | 12 MHz Min | 12 MHz Max | Variable Clock Min | Variable Clock Max | Unit | Figure |
|---|---|---|---|---|---|---|---|
| **EXTERNAL DATA MEMORY WRITE CYCLE** | | | | | | | |
| $t_{WLWH}$ | $\overline{WR}$ Pulsewidth | 400 | | $6t_{CK} - 100$ | | ns | 72 |
| $t_{AVLL}$ | Address Valid after ALE Low | 43 | | $t_{CK} - 40$ | | ns | 72 |
| $t_{LLAX}$ | Address Hold after ALE Low | 48 | | $t_{CK} - 35$ | | ns | 72 |
| $t_{LLWL}$ | ALE Low to $\overline{RD}$ or $\overline{WR}$ Low | 200 | 300 | $3t_{CK} - 50$ | $3t_{CK} + 50$ | ns | 72 |
| $t_{AVWL}$ | Address Valid to $\overline{RD}$ or $\overline{WR}$ Low | 203 | | $4t_{CK} - 130$ | | ns | 72 |
| $t_{QVWX}$ | Data Valid to $\overline{WR}$ Transition | 33 | | $t_{CK} - 50$ | | ns | 72 |
| $t_{QVWH}$ | Data Setup before $\overline{WR}$ | 433 | | $7t_{CK} - 150$ | | ns | 72 |
| $t_{WHQX}$ | Data and Address Hold after $\overline{WR}$ | 33 | | $t_{CK} - 50$ | | ns | 72 |
| $t_{WHLH}$ | $\overline{RD}$ or $\overline{WR}$ High to ALE High | 43 | 123 | $t_{CK} - 40$ | $6t_{CK} - 100$ | ns | 72 |

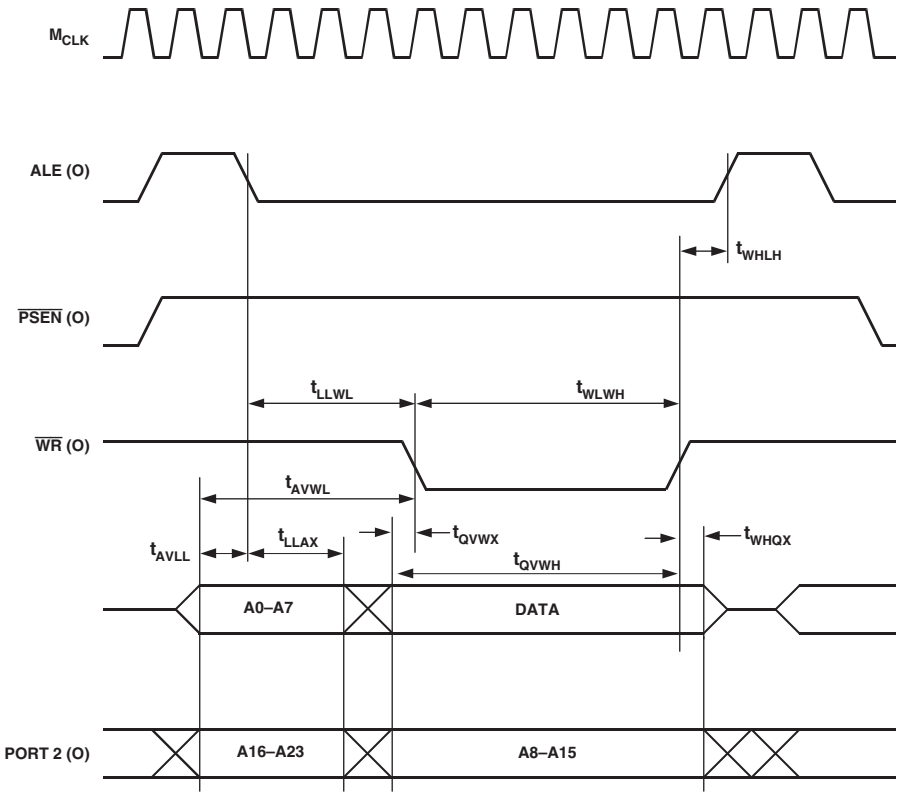



*Figure 72. External Data Memory Write Cycle*

| Parameter | | 12 MHz Min | 12 MHz Typ | 12 MHz Max | Variable Clock Min | Variable Clock Typ | Variable Clock Max | Unit | Figure |
|---|---|---|---|---|---|---|---|---|---|
| **UART TIMING (Shift Register Mode)** | | | | | | | | | |
| $t_{XLXL}$ | Serial Port Clock Cycle Time | | 1.0 | | | $12t_{CK}$ | | μs | 73 |
| $t_{QVXH}$ | Output Data Setup to Clock | 700 | | | $10t_{CK} - 133$ | | | ns | 73 |
| $t_{DVXH}$ | Input Data Setup to Clock | 300 | | | $2t_{CK} + 133$ | | | ns | 73 |
| $t_{XHDX}$ | Input Data Hold after Clock | 0 | | | 0 | | | ns | 73 |
| $t_{XHQX}$ | Output Data Hold after Clock | 50 | | | $2t_{CK} - 117$ | | | ns | 73 |

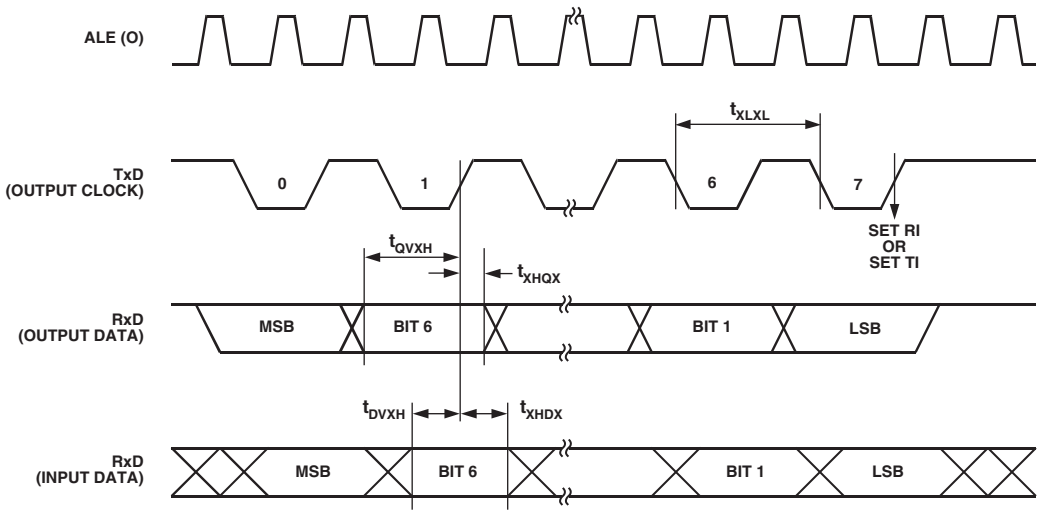



*Figure 73. UART Timing in Shift Register Mode*

| Parameter | | Min | Typ | Max | Unit | Figure |
|---|---|---|---|---|---|---|
| **SPI SLAVE MODE TIMING (CPHA = 0)** | | | | | | |
| $t_{SS}$ | $\overline{SS}$ to SCLOCK Edge | 0 | | | ns | 78 |
| $t_{SL}$ | SCLOCK Low Pulsewidth | | 330 | | ns | 78 |
| $t_{SH}$ | SCLOCK High Pulsewidth | | 330 | | ns | 78 |
| $t_{DAV}$ | Data Output Valid after SCLOCK Edge | | | 50 | ns | 78 |
| $t_{DSU}$ | Data Input Setup Time before SCLOCK Edge | 100 | | | ns | 78 |
| $t_{DHD}$ | Data Input Hold Time after SCLOCK Edge | 100 | | | ns | 78 |
| $t_{DF}$ | Data Output Fall Time | | 10 | 25 | ns | 78 |
| $t_{DR}$ | Data Output Rise Time | | 10 | 25 | ns | 78 |
| $t_{SR}$ | SCLOCK Rise Time | | 10 | 25 | ns | 78 |
| $t_{SF}$ | SCLOCK Fall Time | | 10 | 25 | ns | 78 |
| $t_{DOSS}$ | Data Output Valid after $\overline{SS}$ Edge | | | 20 | ns | 78 |
| $t_{SFS}$ | $\overline{SS}$ High after SCLOCK Edge | 0 | | | ns | 78 |

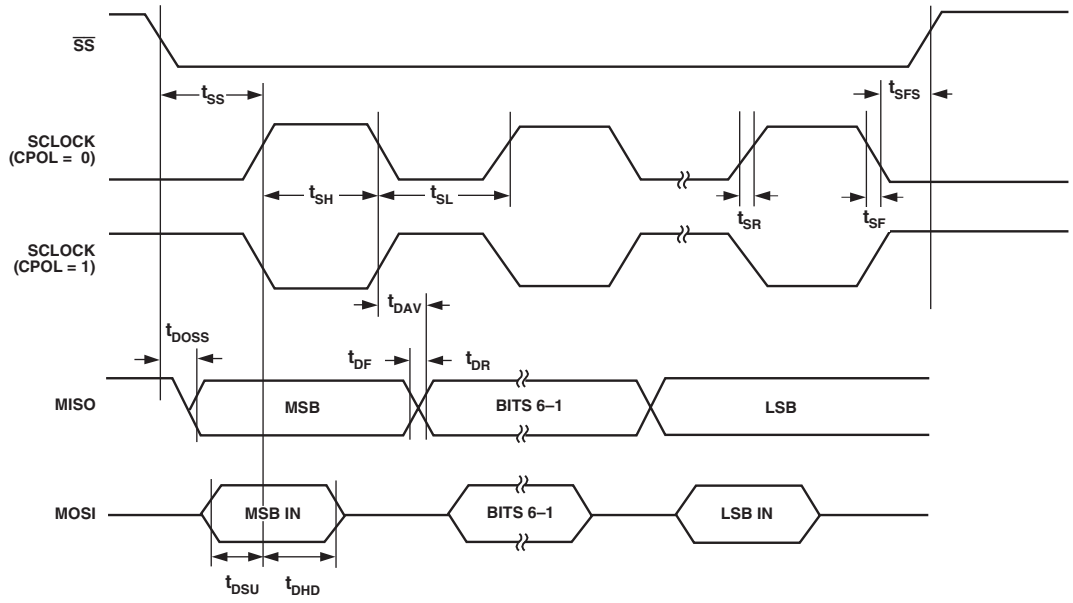



*Figure 78. SPI Slave Mode Timing (CPHA = 0)*