



Z80C30/Z85C30

***CMOS SCC Serial Com-
munications Controller***

Product Specification

PS011703-0102



This publication is subject to replacement by a later edition. To determine whether a later edition exists, or to request copies of publications, contact:

Windows is a registered trademark of Microsoft Corporation.

Document Disclaimer

© 2002 by ZiLOG, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. Except with the express written approval ZILOG, use of information, devices, or technology as critical components of life support systems is not authorized. No licenses or other rights are conveyed, implicitly or otherwise, by this document under any intellectual property rights.



Table of Contents

| | |
|--|------------|
| Table of Contents | iii |
| List of Figures | v |
| List of Tables | vii |
| Overview | 1 |
| Other Features for Z85C30 Only | 2 |
| General Description | 5 |
| Pin Descriptions | 7 |
| Z85C30/Z80C30 Common Pin Functions | 7 |
| Z85C30 | 11 |
| Z80C30 | 12 |
| Functional Description | 17 |
| I/O Interface Capabilities | 19 |
| Programming | 40 |
| Z85C30 | 40 |
| Z80C30 | 40 |
| Z85C30/Z80C30 Setup | 41 |
| Z85C30 Timing | 48 |
| Z80C30 Timing | 51 |
| Electrical Characteristics | 55 |
| Absolute Maximum Ratings | 55 |
| Standard Test Conditions | 55 |
| Capacitance | 57 |
| Miscellaneous | 57 |
| DC Characteristics | 57 |
| Z80C30/Z85C30 | 57 |
| AC Characteristics | 59 |
| Z85C30 Read/Write Timing Diagrams | 59 |
| Packaging and Ordering | 83 |
| Package Information | 83 |
| Ordering Information | 84 |
| Z80C30/Z85C30 | 84 |

Z80C30/Z85C30 CMOS SCC
Serial Communications Controller

iv





List of Figures

| | |
|---|----|
| Figure 1. SCC Block Diagram | 6 |
| Figure 2. Z85C30 and Z80C30 DIP Pin Assignments | 13 |
| Figure 3. Z85C30 and Z80C30 PLCC Pin Assignments | 14 |
| Figure 4. Z85C30 Pin Functions | 15 |
| Figure 5. Z80C30 Pin Functions | 16 |
| Figure 6. SCC Transmit Data Path | 18 |
| Figure 7. SCC Receive Data Path | 19 |
| Figure 8. SCC Interrupt Priority Schedule | 23 |
| Figure 9. Some SCC Protocols | 27 |
| Figure 10. Detecting 5- or 7-Bit Synchronous Characters | 28 |
| Figure 11. An SDLC Loop | 31 |
| Figure 12. Data Encoding Methods | 35 |
| Figure 13. SDLC Frame Status FIFO | 38 |
| Figure 14. SDLC Byte Counting Detail | 39 |
| Figure 15. Write Register Bit Functions | 42 |
| Figure 16. Write Register Bit Functions | 43 |
| Figure 17. Write Register Bit Functions | 44 |
| Figure 18. Write Register Bit Functions | 45 |
| Figure 19. Read Register Bit Functions | 47 |
| Figure 20. Read Register Bit Functions | 48 |
| Figure 21. Read Cycle Timing | 49 |
| Figure 22. Write Cycle Timing | 50 |
| Figure 23. Interrupt Acknowledge Cycle Timing | 51 |
| Figure 24. Read Cycle Timing | 52 |
| Figure 25. Write Cycle Timing | 53 |
| Figure 26. Interrupt Acknowledge Cycle Timing | 54 |
| Figure 27. Standard Test Load | 56 |
| Figure 28. Open-Drain Test Load | 56 |
| Figure 29. Z85C30 Read/Write Timing Diagram | 60 |
| Figure 30. Z85C30 Interrupt Acknowledge Timing Diagram | 61 |
| Figure 31. Z85C30 Cycle Timing Diagram | 61 |
| Figure 32. Z85C30 Reset Timing Diagram | 62 |
| Figure 33. Z85C30 General Timing Diagram | 67 |
| Figure 34. Z85C30 System Timing Diagram | 70 |



| | |
|---|----|
| Figure 35. Z80C30 Read/Write Timing Diagram | 73 |
| Figure 36. Z80C30 Interrupt Acknowledge Timing Diagram..... | 74 |
| Figure 37. Z80C30 Reset Timing Diagram | 74 |
| Figure 38. Z80C30 General Timing Diagram | 79 |
| Figure 39. Z80C30 System Timing Diagram | 82 |
| Figure 40. 40-Pin DIP Package Diagram | 83 |
| Figure 41. 44-Pin PLCC Package Diagram | 84 |



List of Tables

| | | |
|-----------|--|----|
| Table 1. | SCC Read Register Functions | 20 |
| Table 2. | SCC Write Register Functions | 20 |
| Table 3. | Capacitance | 57 |
| Table 4. | Z80C30/Z85C30 DC Characteristics | 58 |
| Table 5. | Z85C30 Read/Write Timing | 62 |
| Table 6. | Z85C30 General Timing Table | 68 |
| Table 7. | Z85C30 System Timing Table | 71 |
| Table 8. | Z85C30 Read/Write Timing | 72 |
| Table 9. | Z80C30 Read/Write Timing | 75 |
| Table 10. | Z80C30 General Timing | 80 |
| Table 11. | Z80C30 System Timing | 83 |
| Table 12. | Z80C30/Z85C30 Ordering Information | 85 |

**Z80C30/Z85C30 CMOS SCC
Serial Communications Controller**



viii



Overview

- Z85C30: Optimized for Non-Multiplexed Bus Microprocessors.
Z80C30: Optimized for Multiplexed Bus Microprocessors
- Pin Compatible to NMOS Versions
- Two Independent, 0 to 4.1 Mbit/Second, Full-Duplex Channels. Each channel with Separate Crystal Oscillator, Baud Rate Generator, and Digital Phase-Locked Loop (DPLL) for Clock Recovery.
- Multi-Protocol Operation under Program Control; Programmable for NRZ, NRZI, or FM Data Encoding.
- Asynchronous Mode with Five to Eight Bits and One, One and One-Half, or Two Stop Bits Per Character, Programmable Clock Factor, Break Detection and Generation; Parity, Overrun, and Framing Error Detection.
- Synchronous Mode with Internal or External Character Synchronization on One or Two Synchronous Characters and CRC Generation and Checking with CRC-16 or CRC-CCITT Preset to either 1s or 0s.
- SDLC/HDLC Mode with Comprehensive Frame-Level Control, Automatic Zero Insertion and Deletion, I-Field Residue Handling, Abort Generation and Detection, CRC Generation and Checking, and SDLC Loop.
- Software Interrupt Acknowledge Feature (not available with NMOS)
- Local Loopback and Auto Echo Modes
- Supports T1 Digital Trunk
- Enhanced DMA Support (not available with NMOS) 10 x 19-Bit Status FIFO 14-Bit Byte Counter



- Speeds:
 - Z85C30-8.5, 10, 16.384 MHz
 - Z80C30-8, 10 MHz

Other Features for Z85C30 Only

Some of the features listed below are available by default. Some of them (features with *) are disabled on default to maintain compatibility with the existing SCC design, and “program to enable through WR7”.

- New programmable WR7' (write register 7 prime) to enable new features
- Improvements to support SDLC mode of synchronous communication
 - Improve functionality to ease sending back-to-back frames
 - Automatic SDLC opening Flag transmission*
 - Automatic Tx Underrun/EOM Latch reset in SDLC mode*
 - Automatic $\overline{\text{RTS}}$ deactivation*
 - TxD pin forced High in SDLC NRZI mode after closing flag*
 - Complete CRC reception*
 - Improved response to Abort sequence in status FIFO
 - Automatic Tx CRC generator preset/reset
 - Extended read for write registers*
 - Write data set-up timing improvement
- Improved AC timing
 - Three to 3.6 PCLK access recovery time.
 - Programmable $\overline{\text{DTR/REQ}}$ timing*



- Write data to falling edge of \overline{WR} set-up time requirement is now eliminated
- Reduced \overline{INT} timing
- Other features include.
 - Extended read function to read back the written value to the write registers.*
 - Latching RRO during read
 - RRO, bit D7 and RR10, bit D6 now has reset default value.

Z80C30/Z85C30 CMOS SCC
Serial Communications Controller



4



General Description

The ZiLOG Z80C30/Z85C30 Serial Communications Controller (SCC), is a pin and software compatible CMOS member of the SCC family introduced by ZiLOG in 1981. It is a dual channel, multi-protocol data communications peripheral that easily interfaces to CPU's with either multiplexed or non-multiplexed address/data buses. The advanced CMOS process offers lower power consumption, higher performance, and superior noise immunity. The programming flexibility of the internal registers allows the SCC to be configured to satisfy a wide variety of serial communications applications. Figure 1 illustrates a block diagram of the SCC.

The many on-chip features such as Baud Rate Generators (BRG), Digital Phase Locked Loops (DPLL), and crystal oscillators reduce the need for external logic. Additional features include a 10 x 19-bit status FIFO and 14-bit byte counter to support high speed SDLC transfers using DMA controllers.

The SCC handles asynchronous formats, synchronous byte-oriented protocols such as IBM Bisync, and synchronous bit-oriented protocols such as HDLC and IBM SDLC. This versatile device supports virtually any serial data transfer application (cassette, diskette, tape drives, etc.).

The device generates and checks CRC codes in any synchronous mode and can be programmed to check data integrity in various modes. The SCC also contains facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

The daisy-chain interrupt hierarchy is also supported.

Z80C30/Z85C30 CMOS SCC Serial Communications Controller

6



Channel A
Exploded View

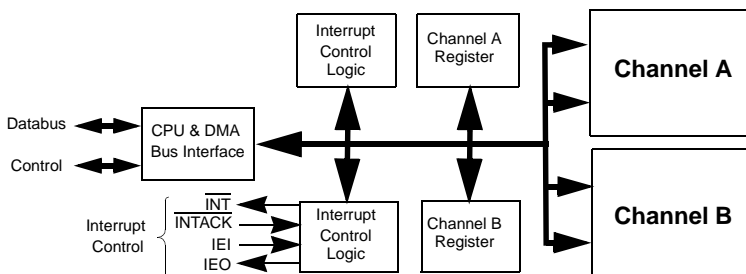
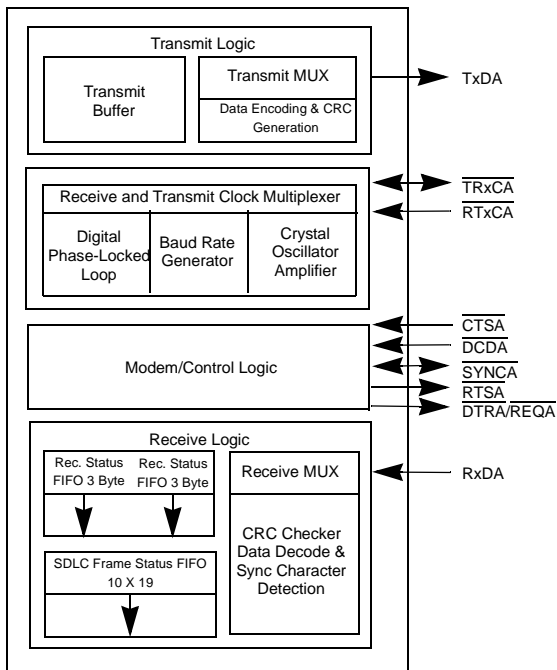


Figure 1. SCC Block Diagram



Pin Descriptions

Z85C30/Z80C30 Common Pin Functions

The following section describes the pin functions common to the Z85C30 and the Z80C30. Figures 2 and 3 detail the respective pin assignments and Figures 4 and 5 designate the pin functions.

$\overline{\text{CTSA}}$, $\overline{\text{CTSB}}$

Clear To Send (inputs, active Low). If these pins are programmed for Auto Enable, a Low on the inputs enables the respective transmitters. If not programmed as Auto Enable, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC detects pulses on these inputs and can interrupt the CPU on both logic level transitions.

$\overline{\text{DCDA}}$, $\overline{\text{DCDB}}$

Data Carrier Detect (inputs, active Low). These pins function as receiver enables if they are programmed for Auto Enable. Otherwise, these pins are used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow rise-time signals. The SCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.

$\overline{\text{DTR/REQA}}$, $\overline{\text{DTR/REQB}}$

Data Terminal Ready/Request (outputs, active Low). These outputs follow the state programmed into the DTR bit. They can also be used as general-purpose outputs or as Request lines for a DMA controller.



IEI

Interrupt Enable In (input, active High). IEI is used with IEO to form an interrupt daisy-chain when there is more than one interrupt driven device. A high IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.

IEO

Interrupt Enable Out (output, active High). IEO is High only if IEI is High and the CPU is not servicing the SCC interrupt or the SCC is not requesting an interrupt (interrupt Acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.

$\overline{\text{INT}}$

Interrupt Request (output, open-drain, active Low). This signal activates when the SCC requests an interrupt.

$\overline{\text{INTACK}}$

Interrupt Acknowledge (input, active Low). This signal indicates an active Interrupt Acknowledge cycle. During this cycle, the SCC interrupt daisy chain settles. When $\overline{\text{RD}}$ or $\overline{\text{DS}}$ becomes active, the SCC places an interrupt vector on the data bus (if IEI is High). $\overline{\text{INTACK}}$ is latched by the rising edge of PCLK.

PCLK

Clock (input). This is the master SCC clock used to synchronize internal signals. PCLK is a TTL level signal. PCLK is not required to have any phase relationship with the master system clock.



RxDA, RxDB

Receive Data (inputs, active High). These signals receive serial data at standard TTL levels.

RTXCA, RTXCB

Receive/Transmit Clocks (inputs, active Low). These pins can be programmed in several different operating modes. In each channel, \overline{RTxC} may supply the receive clock, the transmit clock, the clock for the Baud Rate Generator, or the clock for the Digital Phase-Locked Loop. These pins can also be programmed for use with the respective \overline{SYNC} pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in Asynchronous modes.

RTSA, RTSB

Request To Send (outputs, active Low). When the Request To Send (RTS) bit in Write Register 5 (Figure 9) sets, the IRTS signal goes Low. When the RTS bit is reset in the Asynchronous mode and Auto Enable is on, the signal goes High after the transmitter is empty. In Synchronous mode it strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.

SYNCA, SYNCB

Synchronization (inputs or outputs, active Low). These pins function as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to \overline{CTS} and \overline{DCD} . In this mode, transitions on these lines affect the state of the Synchronous/Hunt status bits in Read Register 0 (Figure 8) but have no other function.



In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, $\overline{\text{SYNC}}$ must be driven Low for two receive clock cycles after the last bit in the synchronous character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of $\overline{\text{SYNC}}$.

In the Internal Synchronization mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which synchronous characters are recognized. This synchronous condition is not latched. These outputs are active each time a synchronization pattern is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.

TxDA, TxDB

Transmit Data (outputs, active High). These output signals transmit serial data at standard TTL levels.

$\overline{\text{TRXCA}}$, $\overline{\text{TRXCB}}$

Transmit/Receive Clocks (inputs or outputs, active low). These pins can be programmed in several different operating modes. $\overline{\text{TRxC}}$ may supply the receive clock or the transmit clock in the input mode or supply the output of the Digital Phase-locked loop, the crystal oscillator, the Baud Rate Generator, or the transmit clock in the output mode.

$\overline{\text{W/REQA}}$, $\overline{\text{W/REQB}}$

Wait/Request (outputs, open-drain when programmed for a Wait function, driven High or low when programmed for a Request function). These dual-purpose outputs may be programmed as



Request lines for a DMA controller or as Wait lines to synchronize the CPU to the SCC data rate. The reset state is Wait.

Z85C30

A/\overline{B}

Channel A/Channel B (input). This signal selects the channel in which the read or write operation occurs.

\overline{CE}

Chip Enable (input, active Low). This signal selects the SCC for a read or write operation.

D7–D0

Data Bus (bidirectional, tri-state). These lines carry data and command to and from the SCC.

D/\overline{C}

Data/Control Select (input). This signal defines the type of information transferred to or from the SCC. A High indicates a data transfer; a Low indicates a command.

\overline{RD}

Read (input, active Low). This signal indicates a read operation and when the SCC is selected, enables the SCC's bus drivers. During the Interrupt Acknowledge cycle, this signal gates the interrupt vector onto the bus if the SCC is the highest priority device requesting an interrupt.



\overline{WR}

Write (input, active Low). When the SCC is selected, this signal indicates a write operation. The coincidence of IAD and \overline{WR} is interpreted as a reset.

Z80C30

AD7–AD0

Address/Data Bus (bidirectional, active High, Tri-state). These multiplexed lines carry register addresses to the SCC as well as data or control information.

\overline{AS}

Address Strobe (input, active Low). Addresses on AD7–AD0 are latched by the rising edge of this signal.

$\overline{CS0}$

Chip Select 0 (input, active Low). This signal is latched concurrently with the addresses on AD7–AD0 and must be active for the intended bus transaction to occur.

CS1

Chip Select 1 (input, active High). This second select signal must also be active before the intended bus transaction can occur. CS1 must remain active throughout the transaction.



\overline{DS}

Data strobe (input, active Low). This signal provides timing for the transfer of data into and out of the SCC. If AS and \overline{DS} coincide, this confluence is interpreted as a reset.

$\overline{R/\overline{W}}$

Read/Write (input). This signal specifies whether the operation to be performed is a read or a write.

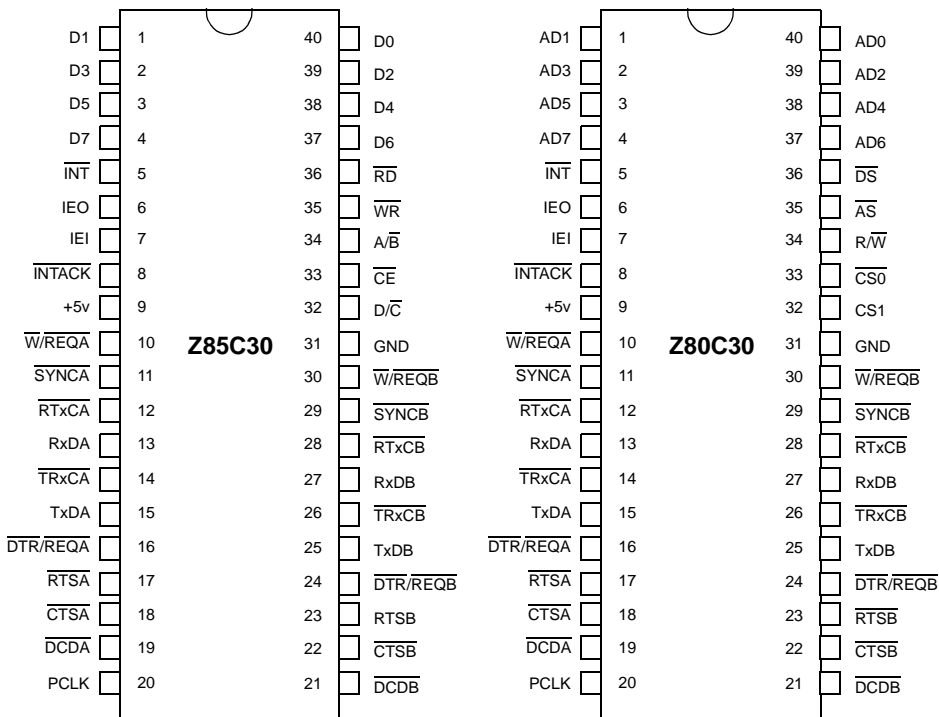


Figure 2. Z85C30 and Z80C30 DIP Pin Assignments

Z80C30/Z85C30 CMOS SCC Serial Communications Controller

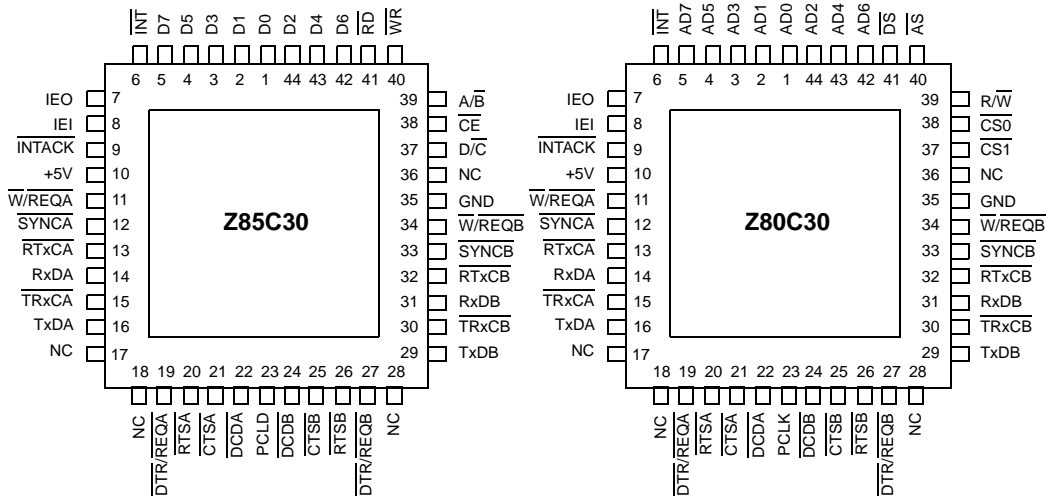


Figure 3. Z85C30 and Z80C30 PLCC Pin Assignments

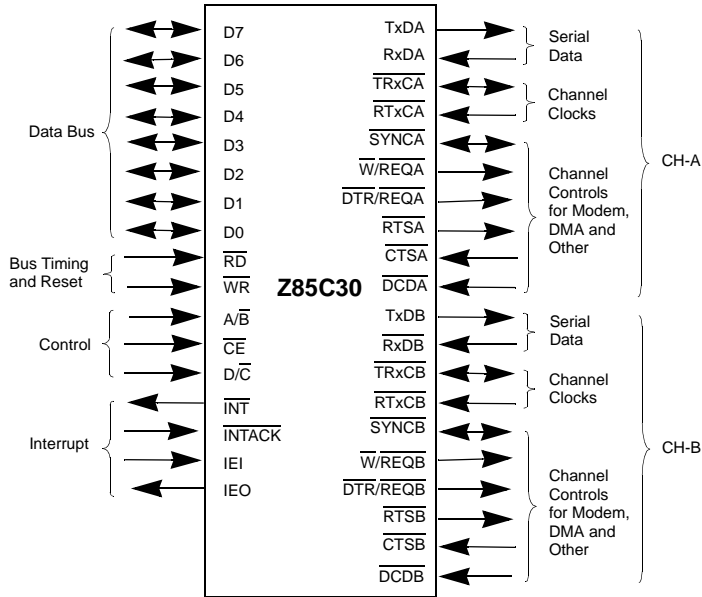


Figure 4. Z85C30 Pin Functions

Z80C30/Z85C30 CMOS SCC Serial Communications Controller

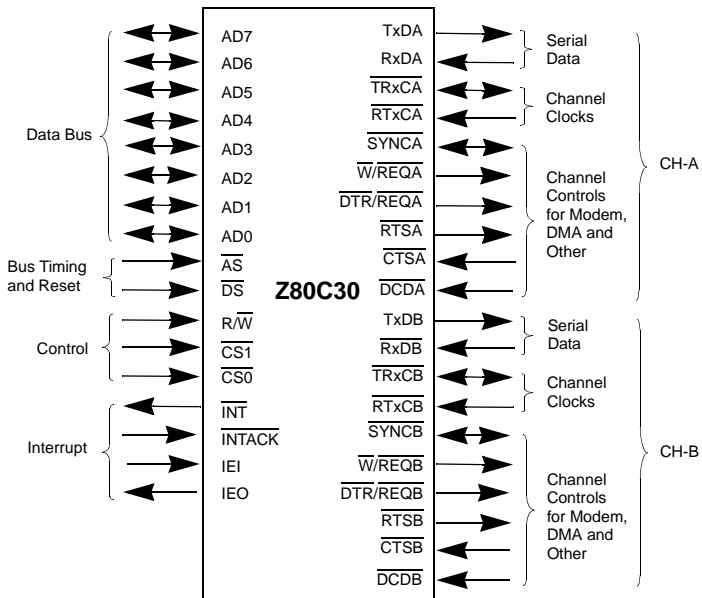


Figure 5. Z80C30 Pin Functions



Functional Description

The architecture of the SCC is described from two points of view:

- As a data communications device which transmits and receives data in a wide variety of protocols;
- As a microprocessor peripheral in which the SCC offers valuable features such as vectored interrupts and DMA support.

The SCC's peripheral and data communication are described in the following sections. Figure 1 on page 6 illustrates the SCC block diagram. Figures 6 and 7 show the details of the communications between the receive and transmit logic to the system bus. The features and data path for each of the SCC's A and B channels is identical.

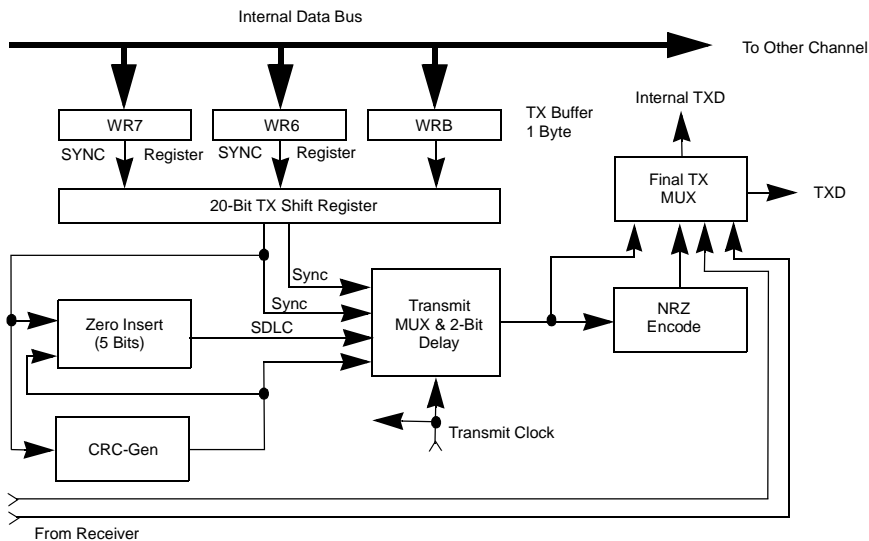


Figure 6. SCC Transmit Data Path

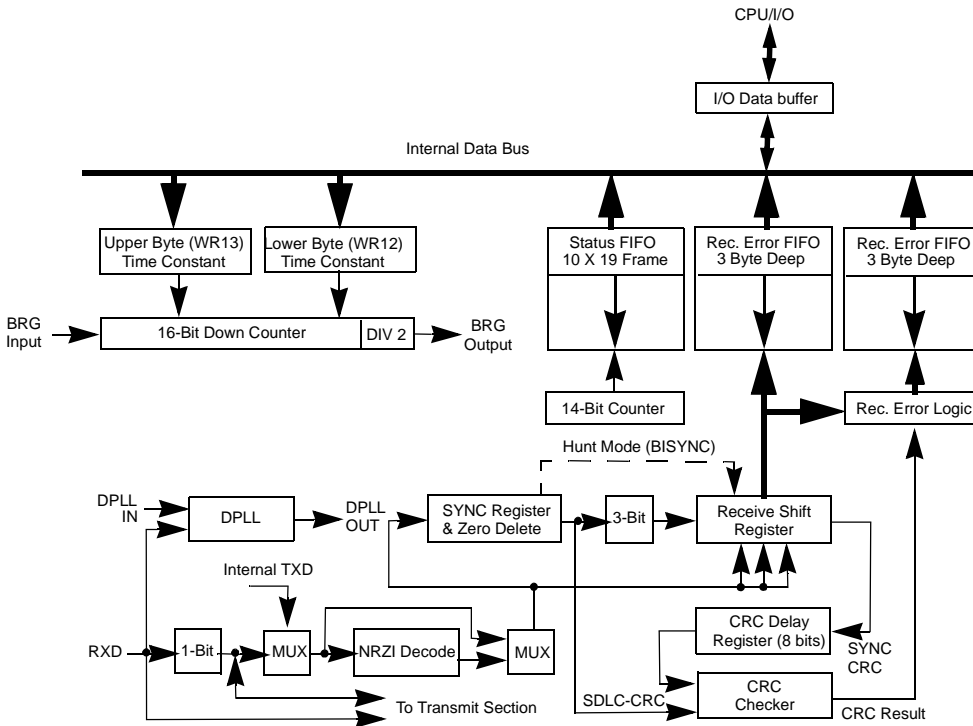


Figure 7. SCC Receive Data Path

I/O Interface Capabilities

System communication to and from the SCC is performed through the SCC's register set. There are sixteen write registers and eight read registers. Tables 1 and 2 list all SCC registers and provide a brief description of their functions.



Throughout this document, the write and read registers are referenced with the following notation “WR” for Write Register and “RR” for Read Register. For example:

WR4A Write Register 4 for channel A
RR3 Read Register 3 for either/both channels

Table 1. SCC Read Register Functions

| Register | Function |
|-----------------|---|
| RR0 | Transmit/Receive buffer status and External status |
| RR1 | Special Receive Condition status |
| RR2 | Modified interrupt vector (Channel B only) Unmodified interrupt vector (Channel A only) |
| RR3 | Interrupt Pending bits (Channel A only) |
| RR8 | Receive Buffer |
| RR10 | Miscellaneous status |
| RR12 | Lower byte of Baud Rate Generator time constant |
| RR13 | Upper byte of Baud Rate Generator time constant |
| RR15 | External/Status interrupt information |

Table 2. SCC Write Register Functions

| Register | Function |
|-----------------|--|
| WR0 | CRC initialize, initialization commands for the various modes, register pointers |
| WR1 | Transmit/Receive interrupt and data transfer mode definition |



Table 2. SCC Write Register Functions (continued)

| Register | Function |
|-----------------|--|
| WR2 | Interrupt vector (accessed through either channel) |
| WR3 | Receive parameters and control |
| WR4 | Transmit/Receive miscellaneous parameters and modes |
| WR5 | Transmit parameters and controls |
| WR6 | Sync characters or SDLC address field |
| WR7 | Sync character or SDLC flag |
| WR7* | Extended Feature and FIFO Control (WR7 Prime) 85C30 Only |
| WR8 | Transmit buffer |
| WR9 | Master interrupt control and reset (accessed through either channel) |
| WR10 | Miscellaneous transmitter/receiver control bits |
| WR11 | Clock mode control |
| WR12 | Lower byte of Baud Rate Generator time constant |
| WR13 | Upper byte of Baud Rate Generator time constant |
| WR14 | Miscellaneous control bits |
| WR15 | External/Status interrupt control |

There are three methods to move data into and out of the SCC:

- Polling
- Interrupt (vectored and non-vectored)
- Block Transfer. The Block Transfer mode can be implemented under CPU or DMA control.



Polling

When polling, all interrupts are disabled. Three status registers in the SCC are automatically updated whenever any function is performed. For example, End-Of-Frame in SDLC mode sets a bit in one of these status registers. The purpose of polling is for the CPU to periodically read a status register until the register contents indicate the need for data to be transferred. Only one register is read. Depending on its contents, the CPU either writes data, reads data, or continues. Two bits in the register indicate the need for data transfer. An alternative is a poll of the Interrupt Pending register to determine the source of an interrupt. The status for both channels resides in one register.

Interrupts

The SCC's interrupt structure supports vectored and nested interrupts. Nested interrupts are supported with the interrupt acknowledge feature (INTACK pin) of the SCC. This allows the CPU to recognize the occurrence of an interrupt, and re-enable higher priority interrupts. Because an INTACK cycle releases the INT pin from the active state, a higher priority SCC interrupt or another higher priority device can interrupt the CPU. When an SCC responds to an Interrupt Acknowledge signal (INTACK) from the CPU, an interrupt vector can be placed on the data bus. This vector is written in WR2 and may be read in RR2. To speed interrupt response time, the SCC can modify three bits in this vector to indicate status. If the vector is read in Channel A, status is never included. If the vector is read in Channel B, status is always included.

Each of the six sources of interrupts in the SCC (Transmit, Receive, and External/Status interrupts in both channels) has three bits associated with the interrupt source. Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). Operation



of the IE bit is straight forward. If the IE bit is set for a given interrupt source, then that source can request interrupts. The exception is when the MIE (Master Interrupt Enable) bit in WR9 is reset and no interrupts can be requested. The IE bits are write only.

The other two bits are related to the interrupt priority chain (Figure 8). As a microprocessor peripheral, the SCC may request an interrupt only when no higher priority device is requesting one, that is, when IEI is High. If the device in question requests an interrupt, it pulls down $\overline{\text{INT}}$. The CPU responds with $\overline{\text{INTACK}}$, and the interrupting device places the vector on the data bus.

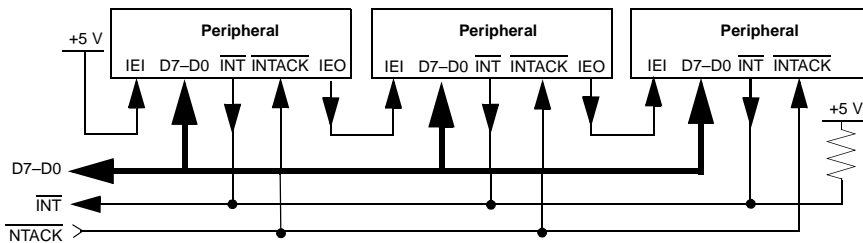


Figure 8. SCC Interrupt Priority Schedule

The SCC can also execute an interrupt acknowledge cycle through software. In some CPU environments, it is difficult to create the $\overline{\text{INTACK}}$ signal with the necessary timing to acknowledge interrupts and allow the nesting of interrupts. In these cases, the $\overline{\text{INTACK}}$ signal can be created with a software command to the SCC.

In the SCC, the Interrupt Pending (IP) bit signals a need for interrupt servicing. When an IP bit is 1 and the IEI input is High, the $\overline{\text{INT}}$ output is pulled Low, requesting an interrupt. In the SCC, if the IE bit is not set by enabling interrupts, then the IP for that source is never set. The IP bits are readable in RR3A.



The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the SCC and external to the SCC are prevented from requesting interrupts. The internal interrupt sources are inhibited by the state of the internal daisy chain, while lower priority devices are inhibited by the IEO output of the SCC being pulled Low and propagated to subsequent peripherals. An IUS bit is set during an Interrupt Acknowledge cycle if there are no higher priority devices requesting interrupts.

There are three types of interrupts: Transmit, Receive, and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receiver, Transmit, and External/Status interrupts prioritized in that order within each channel.

When enabled, the receiver interrupts the CPU in one of three ways.

- Interrupt on First Receive Character or Special Receive Condition
- Interrupt on All Receive Characters or Special Receive Conditions
- Interrupt on Special Receive Conditions Only

Interrupt on First Character or Special Condition and Interrupt on Special Condition Only are typically used with the Block Transfer mode. A special Receive Condition is one of the following: receiver overrun, framing error in Asynchronous mode, end-of-frame in SDLC mode and, optionally, a parity error. The Special Receive Condition interrupt is different from an ordinary receive character available interrupt only by the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt occurs from Special Receive Conditions anytime after the first receive character interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the CTS, DCD, and SYNC pins, however, an External/Status interrupt is also caused by a Transmit Underrun



condition; a zero count in the Baud Rate Generator; by the detection of a Break (Asynchronous mode), Abort (SDLC mode) or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the SCC to interrupt when the Abort or EOP sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Abort condition in external logic in SDLC mode. In SDLC Loop mode, this feature allows secondary stations to recognize the primary station regaining control of the loop during a poll sequence.

Software Interrupt Acknowledge

On the CMOS version of the SCC, the SCC interrupt acknowledge cycle can be initiated through software. If Write Register 9 (WR9) bit D5 is set, Read Register 2 (RR2) results in an interrupt acknowledge cycle to be executed internally. Like a hardware $\overline{\text{INTACK}}$ cycle, a software acknowledge causes the $\overline{\text{INT}}$ pin to return High, the IEO pin to go low and set the IUS latch for the highest priority interrupt pending.

Similar to using the hardware $\overline{\text{INTACK}}$ signal, a software acknowledge cycle requires that a Reset Highest IUS command be issued in the interrupt service routine. Whenever an interrupt acknowledge cycle is used, hardware or software, a reset highest IUS command is required. If RR2 is read from channel A, the unmodified vector is returned. If RR2 is read from channel B, then the vector is modified to indicate the source of the interrupt. The Vector Includes Status (VIS) and No Vector (NV) bits in WR9 are ignored when bit 05 is set to 1.

When the $\overline{\text{INTACK}}$ and IEI pins are not being used, they should be pulled up to V_{CC} through a resistor (10 K ohm typical).



CPU/DMA Block Transfer.

The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and OMA controllers. The Block Transfer mode uses the WAIT/REOUEST output in conjunction with the Wait/Request bits in WR1. The WAIT/REOUEST output can be defined under software control as a WAIT line in the CPU Block Transfer mode or as a REQUEST line in the DMA Block Transfer mode.

To a DMA controller, the SCC REQUEST output indicates that the SCC is ready to transfer data to or from memory. To the CPU, the WAIT line indicates that the ESCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The DTR/REQUEST line allows full-duplex operation under DMA control.

SCC Data Communications Capabilities

The SCC provides two independent full-duplex programmable channels for use in any common asynchronous or synchronous data communication protocols (Figure 9). Each of the data communication channels has identical features and capabilities.

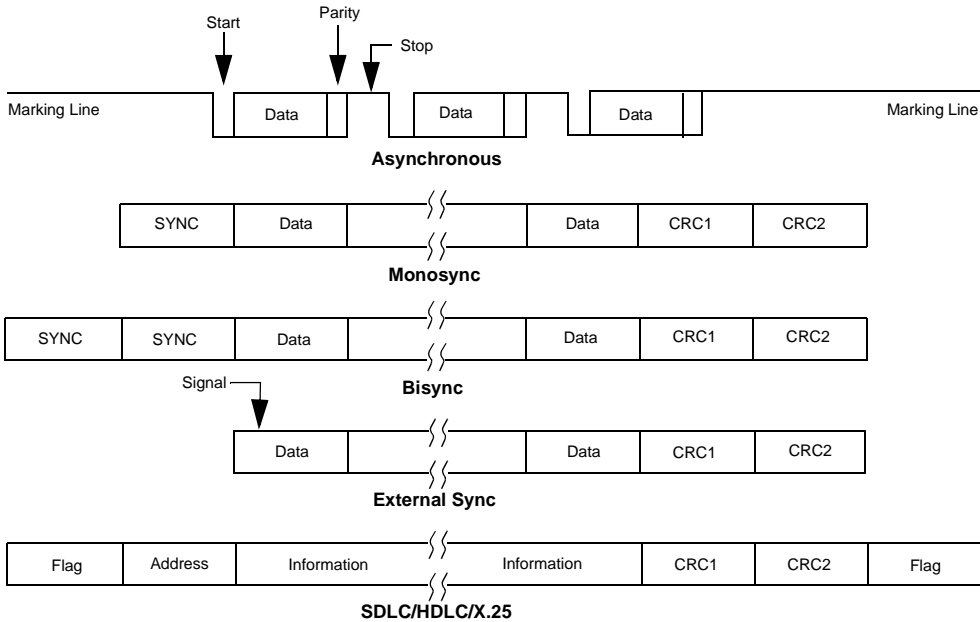


Figure 9. Some SCC Protocols

Asynchronous Modes

Send and Receive is accomplished independently on each channel with five to eight bits per character, plus optional even or odd parity. The transmitters can supply one, one-and-a-half, or two stop bits per character and can provide a break output at any time. The receiver break-detection logic interrupts the CPU both at the start and at the end of a received break. Reception is protected from spikes by a transient spike-rejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxD A or RxD B pins). If the Low does not persist (a transient), the character assembly process does not start.



Framing errors and overrun errors are detected and buffered together with the partial character on which they occur. Vectored interrupts allow fast servicing of error conditions using dedicated routines. A built-in checking process avoids the interpretation of a framing error as a new start bit. A framing error results in the addition of one-half a bit time to the point at which the search for the next start bit begins.

The SCC does not require symmetric transmit and receive clock signals - a feature allowing use of the wide variety of clock sources. The transmitter and receiver handle data at a rate supplied to the receive and transmit clock inputs. In Asynchronous modes, the SYNC pin may be programmed as an input used for functions such as monitoring a ring indicator.

Synchronous Modes

The SCC supports both byte and bit-oriented synchronous communication. Synchronous byte-oriented protocols are handled in several modes. They allow character synchronization with a 6-bit or 8-bit sync character (Monosync), and a 12-bit or 16-bit synchronization pattern (Bisync), or with an external sync signal. Leading sync characters are removed without interrupting the CPU.

5- or 7-bit synchronous characters are detected with 8- or 16-bit patterns in the SCC by overlapping the larger pattern across multiple incoming synchronous characters as indicated in Figure 10.

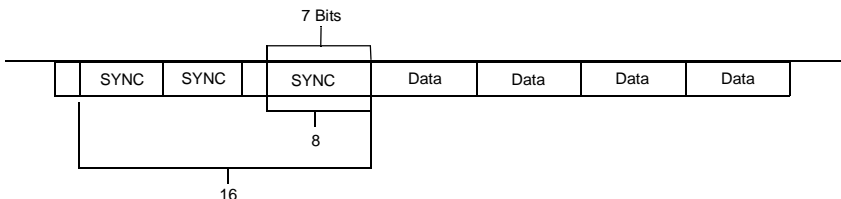


Figure 10. Detecting 5- or 7-Bit Synchronous Characters



CRC checking for Synchronous byte-oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This feature permits the implementation of protocols such as IBM Bisync.

Both CRC-16 ($X^{16} + X^{15} + X^2 + 1$) and CCITT ($X^{16} + X^{12} + X^5 + 1$) error-checking polynomials are supported. Either polynomial may be selected in all Synchronous modes. Users may preset the CRC generator and checker to all 1's or all 0's. The SCC also provides a feature that automatically transmits CRC data when no other data is available for transmission. This feature allows for high speed transmissions under DMA control, with no need for CPU intervention at the end of a message. When there is no data or CRC to send in Synchronous modes, the transmitter inserts 6-, 8-, or 16-bit sync characters, regardless of the programmed character length.

SDLC Mode

The SCC supports Synchronous bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero insertion, and CRC generation. A special command is used to abort a frame in transmission. At the end of a message, the SCC automatically transmits the CRC and trailing flag when the transmitter underruns. The transmitter may also be programmed to send an idle line consisting of continuous flag characters or a steady marking condition.

If a transmit underrun occurs in the middle of a message, an external/status interrupt warns the CPU of this status change, issuing an abort. The SCC may also be programmed to send an abort itself in case of an underrun, relieving the CPU of this task. One to eight bits per character can be sent, allowing reception of a message with no prior information about the character structure in the information field of a frame.



The receiver automatically acquires synchronization on the leading flag of a frame in SDLC or HDLC and provides a synchronization signal on the $\overline{\text{SYNC}}$ pin (an interrupt can also be programmed). The receiver can be programmed to search for frames addressed by a single byte (or four bits within a byte) of a user-selected address or to a global broadcast address. In this mode, frames not matching either the user-selected or broadcast address are ignored.

The number of address bytes are extended under software control. For receiving data, an interrupt on the first received character, or an interrupt on every character, or on special condition only (end-of-frame) can be selected. The receiver automatically deletes all 0's inserted by the transmitter during character assembly. CRC is also calculated and is automatically checked to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers. In SDLC mode, the SCC must be programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all 1's or all 0's. The CRC inverts before transmission and the receiver checks against the bit pattern 0001110100001111.

NRZ, NRZI or FM coding may be used in any 1 x mode. The parity options available in Asynchronous modes are available in Synchronous modes.

SDLC Loop Mode

The SCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, a primary controller station manages the message traffic flow on the loop and any number of secondary stations. In SDLC Loop mode, the SCC performs the functions of a secondary station while an SCC operating in regular SDLC mode acts as a controller (Figure 11). SDLC loop mode can be selected by setting WR10 bit D1.

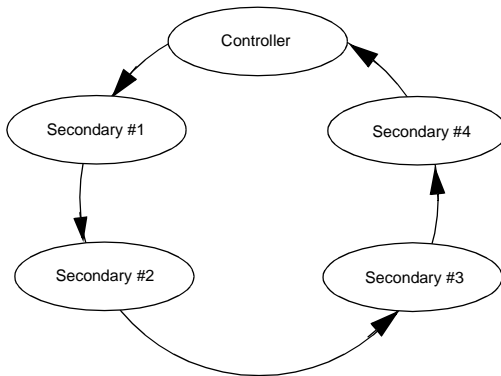


Figure 11. An SDLC Loop

A secondary station in an SDLC Loop is always listening to the messages sent around the loop and, in fact, passes these messages to the rest of the loop by retransmitting them with a one-bit-time delay. The secondary station places its own message on the loop only at specific times. The controller signals that secondary stations can transmit messages by sending a special character, called an EOP (End Of Poll), around the loop. The EOP character is the bit pattern 11111110. Because of zero insertion during messages, this bit pattern is unique and easily recognized.

When a secondary station contains a message to transmit and recognizes an EOP on the line, it changes the last binary 1 of the EOP to a 0 before transmission. This change has the effect of turning the EOP into a flag sequence. The secondary station now places its message on the loop and terminates the message with an EOP. Any secondary stations further down the loop with messages to transmit append their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send echo the incoming message and are



prohibited from placing messages on the loop (except when recognizing an EOP). In SDLC Loop mode, NRZ, NRZI, and FM coding may all be used.

The SCC's ability to receive high speed back-to-back SDLC frames is maximized by a 10- deep by 19-bit wide status FIFO. When enabled (through WR15, bit D2), it provides the DMA the ability to continue to transfer data into memory so that the CPU can examine the message later. For each SDLC frame, a 14-bit byte count and 5 status/error bits are stored. The byte count and status bits are accessed through Read Registers 6 and 7. Read Registers 6 and 7 are only accessible when the SDLC FIFO is enabled. The 10 x 19 status FIFO is separate from the 3-byte receive data FIFO.

Baud Rate Generator

Each channel in the SCC contains a programmable Baud Rate Generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On start-up, the output flip-flop is set in a High state, the value in the time constant register is loaded into the counter, and the counter starts counting down. The output of the Baud Rate Generator toggles when reaching 0, the value in the time constant register is loaded into the counter, and the process is repeated. The time constant may be changed at any time, but the new value does not take effect until the next load of the counter.

The output of the Baud Rate Generator may be used as either the transmit clock, the receive clock, or both. It can also drive the Digital Phase-locked loop (see next section).

If the receive clock or transmit clock is not programmed to come from the TRxC pin, the output of the Baud Rate Generator may be echoed out through the TRxC pin. The following formula relates the time constant to the baud rate where PCLK or RTxC is the Baud



Rate Generator input frequency in Hertz. The clock mode is 1, 16, 32, or 64, as selected in Write Register 4, bits D6 and D7. Synchronous operation modes select 1 and Asynchronous modes select 16, 32 or 64.

$$\text{Time Constant} = \frac{\text{PCLK or RTxC Frequency}}{2(\text{Baud Rate})(\text{Clock Rate})} - 2$$

Digital Phase-Locked Loop

The SCC contains a Digital Phase-Locked Loop (DPLL) to recover clock information from a data stream with NRZI or FM encoding. The DPLL is driven by a clock that is nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a clock for the data. This clock is used as the SCC receive clock, the transmit clock, or both. When the DPLL is selected as the transmit clock source, it provides a jitter-free clock output that is the DPLL input frequency divided by the appropriate divisor for the selected encoding technique.

For NRZI encoding, the DPLL counts the 32x clock to create nominal bit times. As the 32x clock is counted, the DPLL is searching the incoming data stream for edges (either 1 to 0, or 0 to 1). Whenever an edge is detected, the DPLL makes a count adjustment (during the next counting cycle), producing a terminal count closer to the center of the bit cell.

For FM encoding, the DPLL again counts from 0 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the data stream occur between counts 15 and 16 and between counts 31 and 0. The DPLL looks for edges only during a time centered on the 15 to 16 counting transition.



The 32x clock for the DPLL can be programmed to come from either the RTxC input or the output of the Baud Rate Generator. The DPLL output may be programmed to be echoed out of the SCC through the TRxC pin (if this pin is not being used as an input).

Data Encoding

The SCC may be programmed to encode and decode the serial data in four different methods (Figure 12). In NRZ encoding, a 1 is represented by a High level and a 0 is represented by a Low level. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level.

In FM1 (more properly, bi-phase mark), a transition occurs at the beginning of every bit cell. A 1 is represented by an additional transition at the center of the bit cell and a 0 is represented by no additional transition at the center of the bit cell.

In FM0 (bi-phase space), a transition occurs at the beginning of every bit cell. A 0 is represented by an additional transition at the center of the bit cell, and a 1 is represented by no additional transition at the center of the bit cell.

In addition to these four methods, the SCC can be used to decode Manchester (bi-phase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is 0 to 1, the bit is a 0. If the transition is 1 to 0, the bit is a 1.

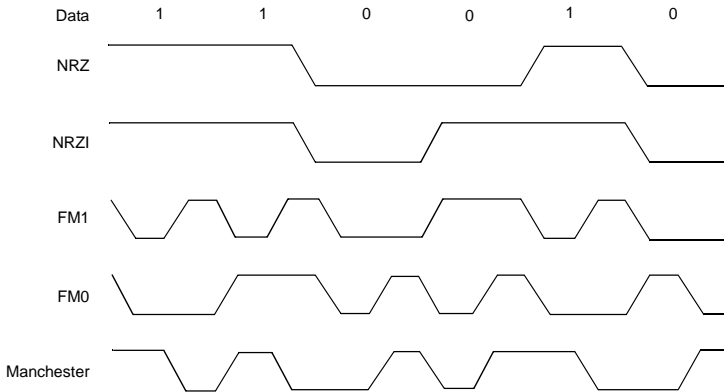


Figure 12. Data Encoding Methods

Auto Echo and Local Loopback

The SCC is capable of automatically echoing everything it receives. This feature is useful mainly in Asynchronous modes, but works in Synchronous and SDLC modes as well. Auto Echo mode (Tx0 is Rx0) is used with NRZI or FM encoding with no additional delay because the data stream is not decoded before retransmission. In Auto Echo mode, the $\overline{\text{CTS}}$ input is ignored as a transmitter enable (although transitions on this input can still cause interrupts if programmed to do so). In this mode, the transmitter is actually bypassed and the programmer is responsible for disabling transmitter interrupts and WAIT/REQUEST on transmit.

The SCC is also capable of local loopback. In this mode, Tx0 or Rx0 is similar to Auto Echo mode. However, in Local Loopback mode the internal transmit data is tied to the internal receive data and Rx0 is ignored (except to be echoed out through Tx0). The CTS and DCD inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause inter-



rupts. Local Loopback works in Asynchronous, Synchronous and SDLC modes with NRZ, NRZI or FM coding of the data stream.

SDLC FIFO Frame Status FIFO Enhancement

The SCC's ability to receive high speed back-to-back SDLC frames is maximized by a 10- deep by 19-bit wide status FIFO. When enabled (through WR15, bit D2), it provides the DMA the ability to continue to transfer data into memory so that the CPU can examine the message later. For each SDLC frame, a 14-bit byte count and 5 status/error bits are stored. The byte count and status bits are accessed through Read Registers 6 and 7. Read Registers 6 and 7 are only accessible when the SDLC FIFO is enabled. The 10x19 status FIFO is separate from the 3-byte receive data FIFO.

When the enhancement is enabled, the status in Read Register 1 (RR1) and byte count for the SDLC frame are stored in the 10 x 19 bit status FIFO. This arrangement allows the DMA controller to transfer the next frame into memory while the CPU verifies that the message was properly received.

Summarizing the operation; data is received, assembled, and loaded into the eight byte FIFO before being transferred to memory by the DMA controller. When a flag is received at the end of an SDLC frame, the frame byte count from the 14-bit counter and five status bits are loaded into the status FIFO for verification by the CPU. The CRC checker automatically resets in preparation for the next frame which can begin immediately. Since the byte count and status are saved for each frame, the message integrity is verified at a later time. Status information for up to 10 frames is stored before a status FIFO overrun occurs.

If a frame is terminated with an ABORT, the byte count is loaded to the status FIFO and the counter resets for the next frame.



FIFO Detail

For a better understanding of the FIFO operation details, refer to the block diagram in Figure 13.

Enable/Disable

This FIFO is implemented is enabled when WR15, bit D2, is set and the SCC is in the SDLC/HDLC mode. Otherwise, the status register contents bypass the FIFO and go directly to the bus interface (the FIFO pointer logic is reset either when disabled or through a channel or power-on reset). When the FIFO mode is disabled, the SCC is downward compatible with the NMOS Z8530. The FIFO mode is disabled on power-up (WR15 D2 is set to 0 on reset). The effects of backward compatibility on the register set are that RR4 is an image of RR0, RR5 is an image of RR1, RR6 is an image of RR2 and RR7 is an image of RR3. For the details of the added registers, refer to Figure 16. The status of the FIFO Enable signal is obtained by reading RR15, bit D2. If the FIFO is enabled, the bit is set to 1; otherwise, it resets.

Read Operation

When WR15 bit D2 sets and the FIFO is not empty, the next read to status register RR1 or registers RR7 and RR6, is from the FIFO. Reading status register RR1 causes one location of the FIFO to become empty. Status is read after reading the byte count, otherwise the count is incorrect. Before the FIFO underflows, it is disabled. In this case, the multiplexer is switched allowing status to read directly from the status register. Reads from RR7 and RR6 contain bits that are undefined. Bit D6 of RR7 (FIFO Data Available) determines if status data is coming from the FIFO or directly from the status register, which sets to 1 when the FIFO is not empty. Not all status bits are stored in the FIFO. The All Sent, Par-



ity, and EOF bits bypass the FIFO. Status bits sent through the FIFO are Residue Bits (3), Overflow, and CRC Error.

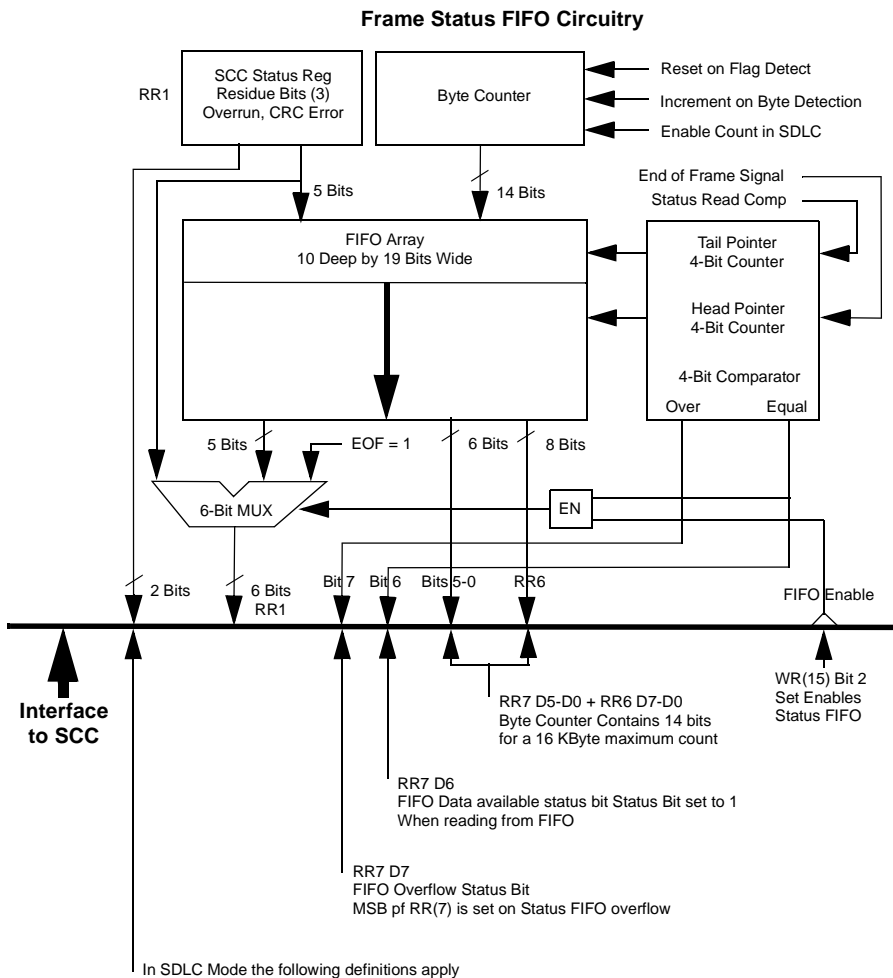


Figure 13. SDLC Frame Status FIFO



The sequence for operation of the byte count and FIFO logic is to read the registers in the following order. RR1, RR6, and RR1 (reading RR6 is optional). Additional logic prevents the FIFO from being emptied by multiple reads from RR1. The read from RR1 latches the FIFO empty/full status bit (06) and steers the status multiplexer to read from the SCC megacell instead of the status FIFO (since the status FIFO is empty). The read from RR1 allows an entry to be read from the FIFO (if the FIFO was empty, logic was added to prevent a FIFO underflow condition).

Write Operation

When the end of an SDLC frame (EOF) is received and the FIFO is enabled, the contents of the status and byte-count registers are loaded into the FIFO. The EOF signal is used to increment the FIFO. If the FIFO overflows, RR7, bit D7 (FIFO Overflow) sets to indicate the overflow. This bit and the FIFO control logic is reset by disabling and re-enabling the FIFO control bit (WR15, bit 02). For details of FIFO control timing during an SDLC frame, refer to Figure 14.

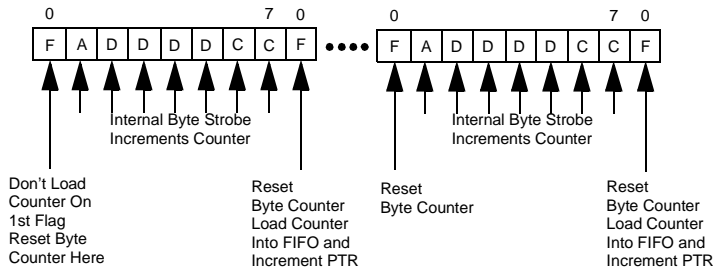


Figure 14. SDLC Byte Counting Detail



Programming

The SCC contains write registers in each channel that are programmed by the system separately to configure the functional personality of the channels.

Z85C30

In the SCC, the data registers are directly addressed by selecting a High on the $\overline{D/C}$ pin. With all other registers (except WR0 and RR0), programming the write registers requires two write operations and reading the read registers requires both a write and a read operation. The first write is to WR0 and contains three bits that point to the selected register. The second write is the actual control word for the selected register, and if the second operation is read, the selected read register is accessed. All of the SCC registers, including the data registers, may be accessed in this fashion. The pointer bits are automatically cleared after the read or write operation so that WR0 (or RR0) is addressed again.

Z80C30

All SCC registers are directly addressable. A command issued in WR0B controls how the SCC decodes the address placed on the address/data bus at the beginning of a Read or Write cycle. In the Shift Right mode, the channel select A/B is taken from AD0 and the state of AD5 is ignored. In the Shift Left mode, the channel select A/B is taken from AD5 and the state of AD0 is ignored. AD7 and AD6 are always ignored as address bits and the register address occupies AD4-AD1.



Z85C30/Z80C30 Setup

Initialization

The system program first issues a series of commands to initialize the basic mode of operation. This is followed by other commands to qualify conditions within the selected mode. For example, in the Asynchronous mode, character length, clock rate, number of stop bits, and even or odd parity must be set first. The interrupt mode is set, and finally, the receiver and transmitter are enabled.

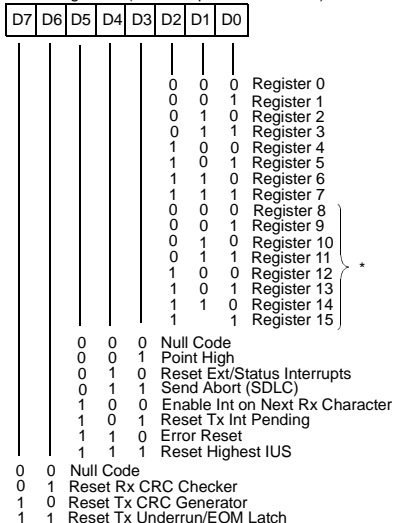
Write Registers

The SCC contains 15 write registers for the 80C30, while there are 16 for the 85C30 (one more additional write register if counting the transmit buffer) in each channel. These write registers are programmed separately to configure the functional “personality” of the channels. There are two registers (WR2 and WR9) shared by the two channels that are accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits and reset commands. Figures 15 through 18 depict the format of each write register.

Z80C30/Z85C30 CMOS SCC Serial Communications Controller

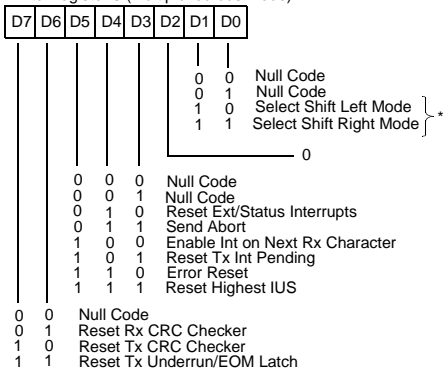


Write Register 0 (non-multiplexed bus mode)



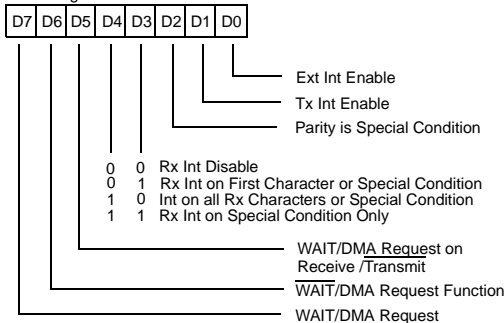
* With Point High Command

Write Register 0 (multiplexed bus mode)

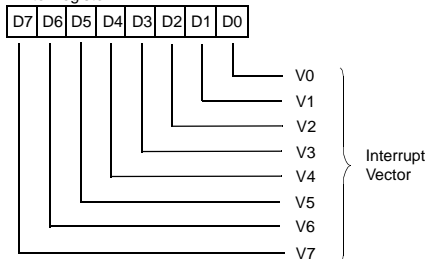


* B Channel Only

Write Register 1



Write Register 2



Write Register 3

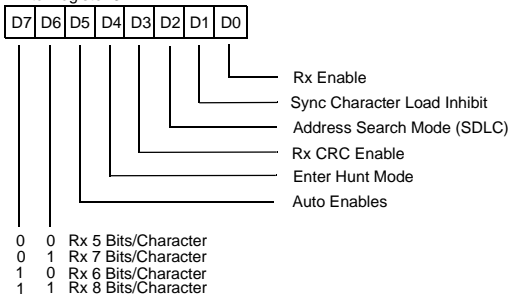


Figure 15. Write Register Bit Functions

Z80C30/Z85C30 CMOS SCC Serial Communications Controller

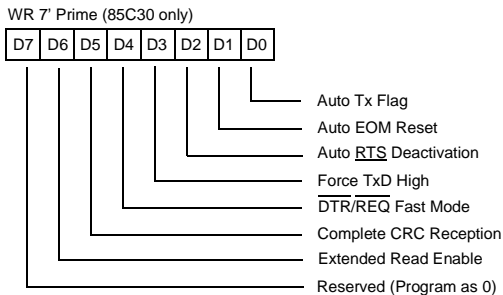
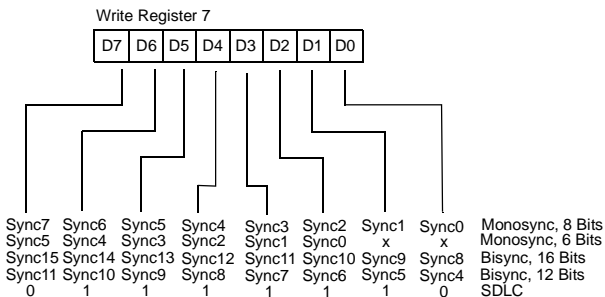
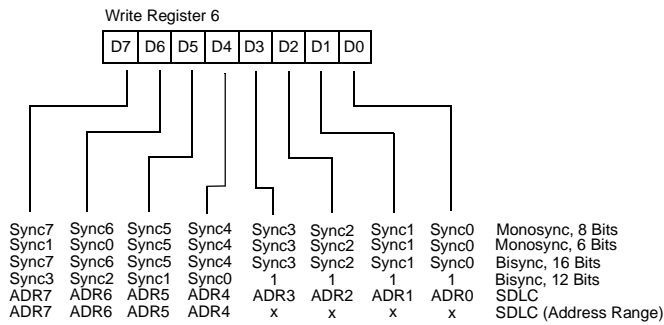


Figure 17. Write Register Bit Functions

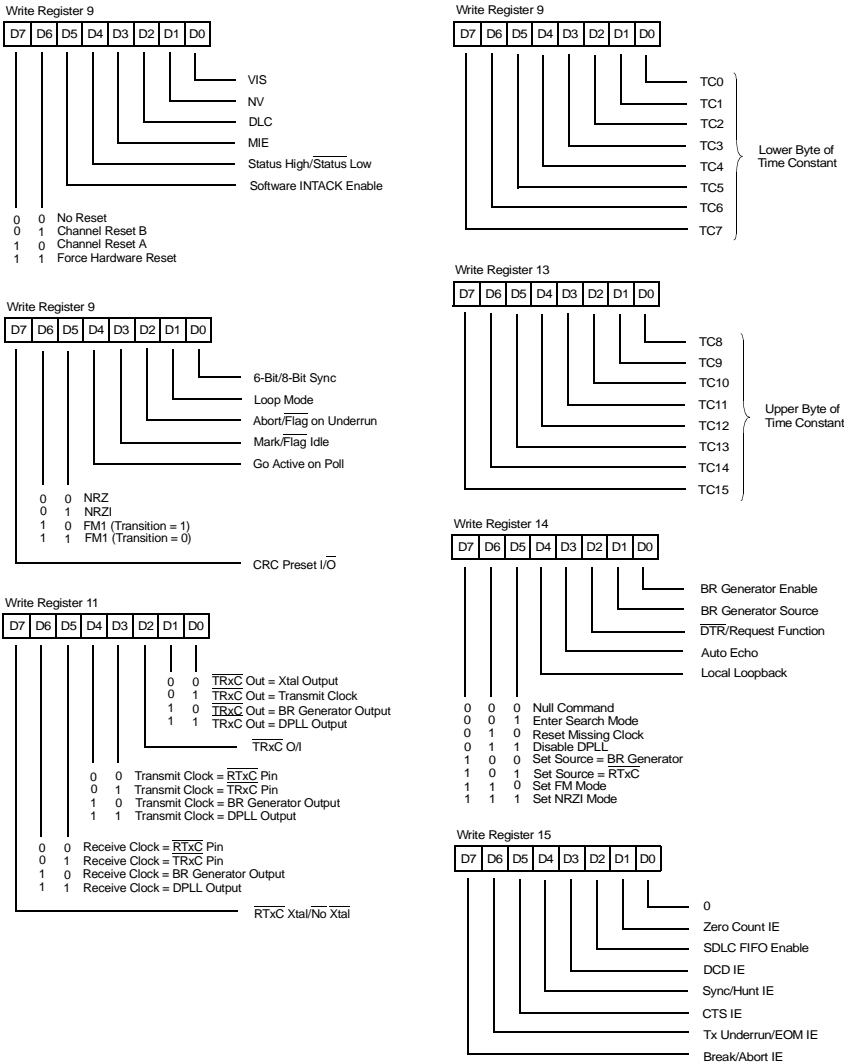


Figure 18. Write Register Bit Functions



Read Registers

The SCC contains ten read registers (eleven, counting the receive buffer (RR8) in each channel). Four of these may be read to obtain status information (RR0, RR1, RR10, and RR15). Two registers (RR12 and RR13) are read to learn the Baud Rate Generator time constant. RR2 contains either the unmodified interrupt vector (Channel A) or the vector modified by status information (Channel B). RR3 contains the Interrupt Pending (IP) bits (Channel A only –, Figure 19). RR6 and RR7 contain the information in the SDLC Frame Status FIFO, but is only read when WR15 D2 is set (Figures 19 and 20).

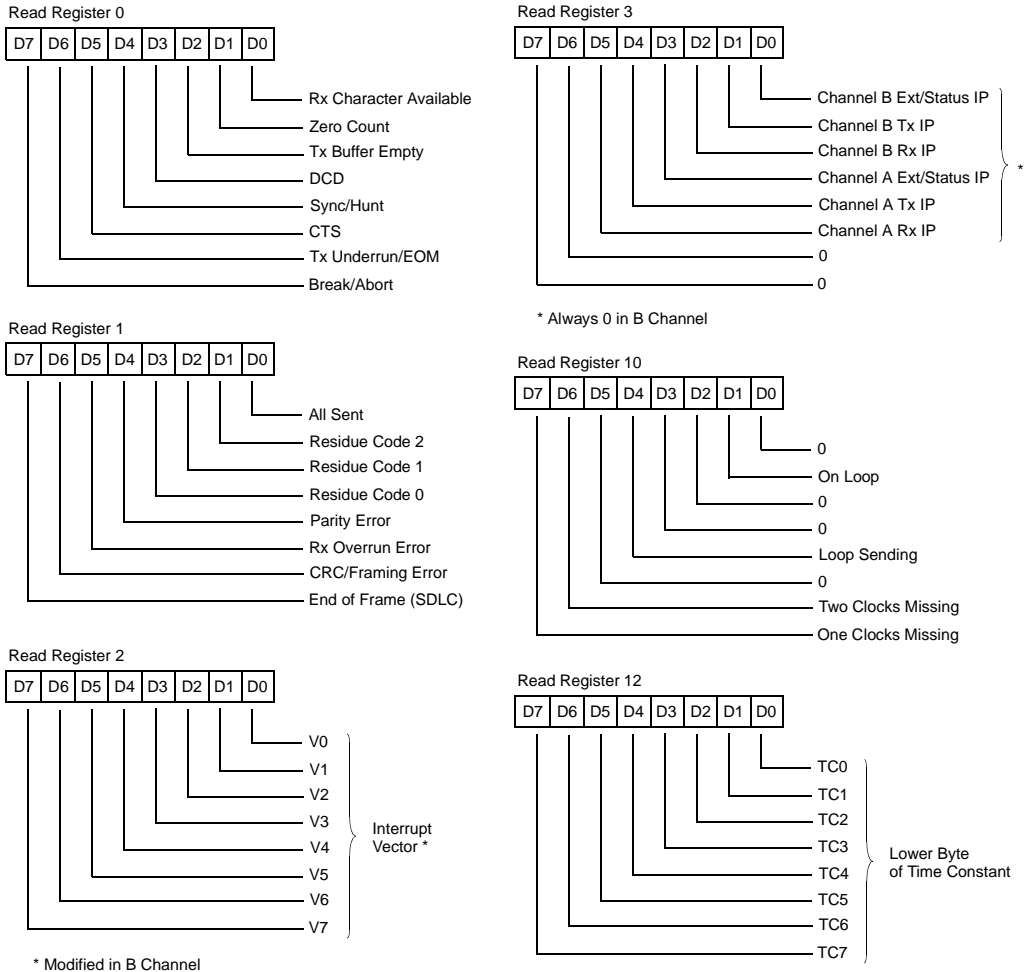


Figure 19. Read Register Bit Functions

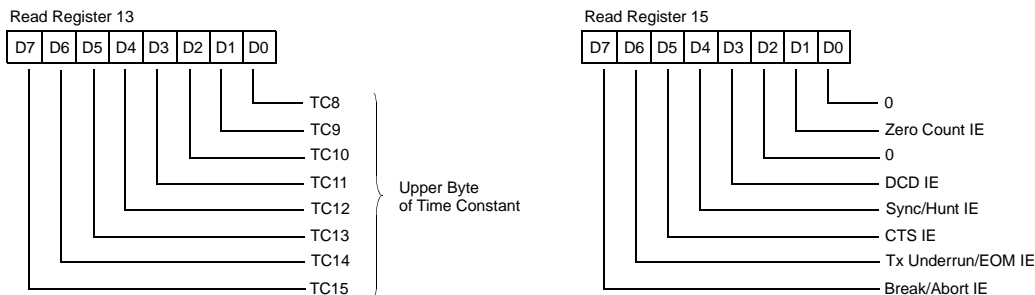


Figure 20. Read Register Bit Functions

Z85C30 Timing

The SCC generates internal control signals from the \overline{WR} and \overline{RD} that are related to PCLK. PCLK has no phase relationship with \overline{WR} and \overline{RD} , the circuitry generating the internal control signals provides time for meta-stable conditions to disappear. This gives rise to a recovery time related to PCLK. The recovery time applies only between bus transactions involving the SCC. The recovery time required for proper operation is specified from the falling edge of \overline{WR} or \overline{RD} in the first transaction involving the SCC to the falling edge of \overline{WR} or \overline{RD} in the second transaction involving the SCC. This time must be at least 3 PCLKs regardless of which register or channel is being accessed.

Read Cycle Timing

Figure 21 illustrates Read cycle timing. Addresses on A/\overline{B} and D/\overline{C} and the status on \overline{INTACK} must remain stable throughout the cycle. If \overline{CE} falls after \overline{RD} falls, or if \overline{CE} rises before \overline{RD} rises, the effective \overline{RD} is shortened.

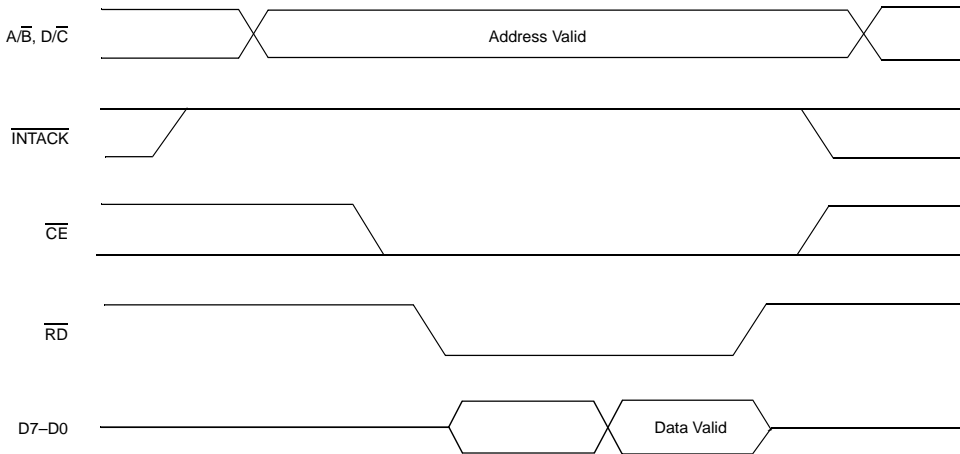


Figure 21. Read Cycle Timing

Write Cycle Timing

Figure 22 illustrates Write cycle timing. Addresses on A/\overline{B} and D/\overline{C} and the status on \overline{INTACK} must remain stable throughout the cycle. If \overline{CE} falls after \overline{WR} falls, or if \overline{CE} rises before \overline{WR} rises, the effective \overline{WR} is shortened.

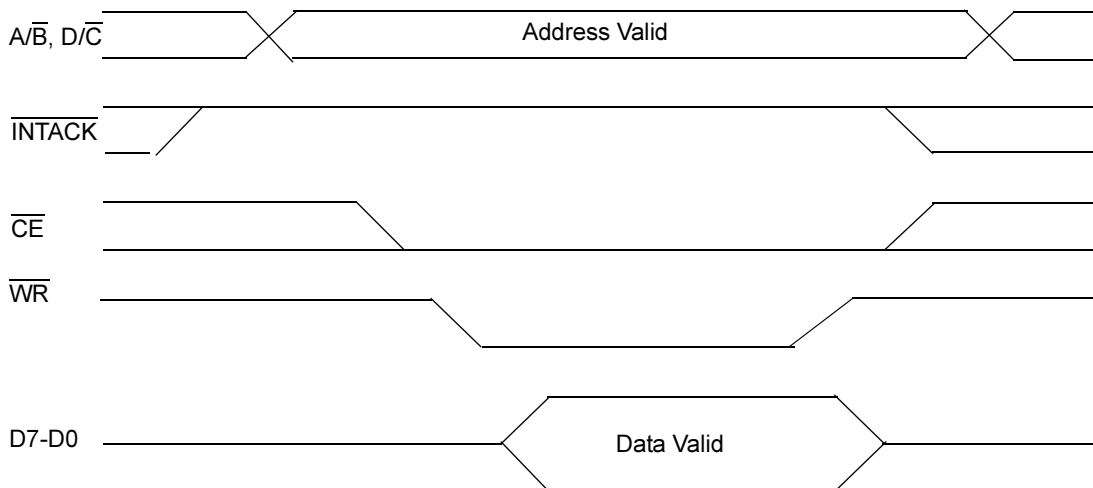


Figure 22. Write Cycle Timing

Interrupt Acknowledge Cycle Timing

Figure 23 illustrates Interrupt Acknowledge cycle timing. Between the time \overline{INTACK} goes Low and the falling edge of \overline{RD} , the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the SCC and IEI is High when \overline{RD} falls, the Acknowledge cycle is intended for the SCC. In this case, the SCC may be programmed to respond to \overline{RD} Low by placing its interrupt vector on D7-D0. It then sets the appropriate Interrupt-Under-Service latch internally. If the external daisy chain is not used, AC parameter #38 is required to settle the interrupt priority daisy chain internal to the SCC. If the external daisy chain is used, the user should follow the equation in AC Characteristics, Read/Write Timing Table 5,



Note 5 starting on Page 62 for calculating the required daisy-chain settle time.

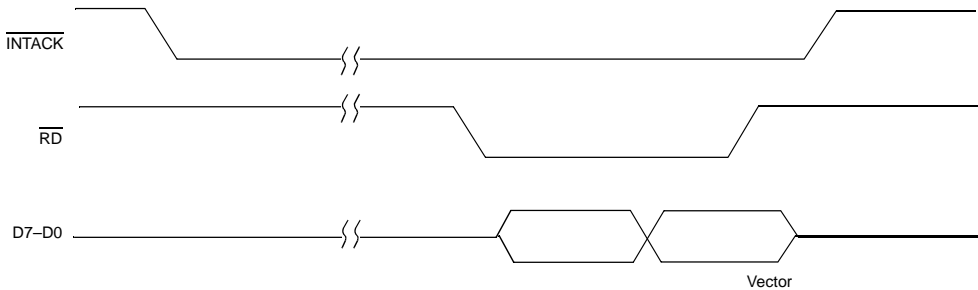


Figure 23. Interrupt Acknowledge Cycle Timing

Z80C30 Timing

The SCC generates internal control signals from \overline{AS} and \overline{DS} that are related to $PCLK$. Because $PCLK$ has no phase relationship with AS and DS , the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to $PCLK$. The recovery time applies only between bus transactions involving the SCC. The recovery time required for proper operation is specified from the falling edge of \overline{DS} in the first transaction involving the SCC to the falling edge of \overline{DS} in the second transaction involving the SCC.

Read Cycle Timing

Figure 24 illustrates Read cycle timing. The address on $AD7-AD0$ and the state of $\overline{CS0}$ and \overline{INTACK} are latched by the rising edge of \overline{AS} . R/W must be High to indicate a Read cycle. $\overline{CS1}$ must also be High for the Read cycle to occur. The data bus drivers in the SCC are then enabled while \overline{DS} is Low.

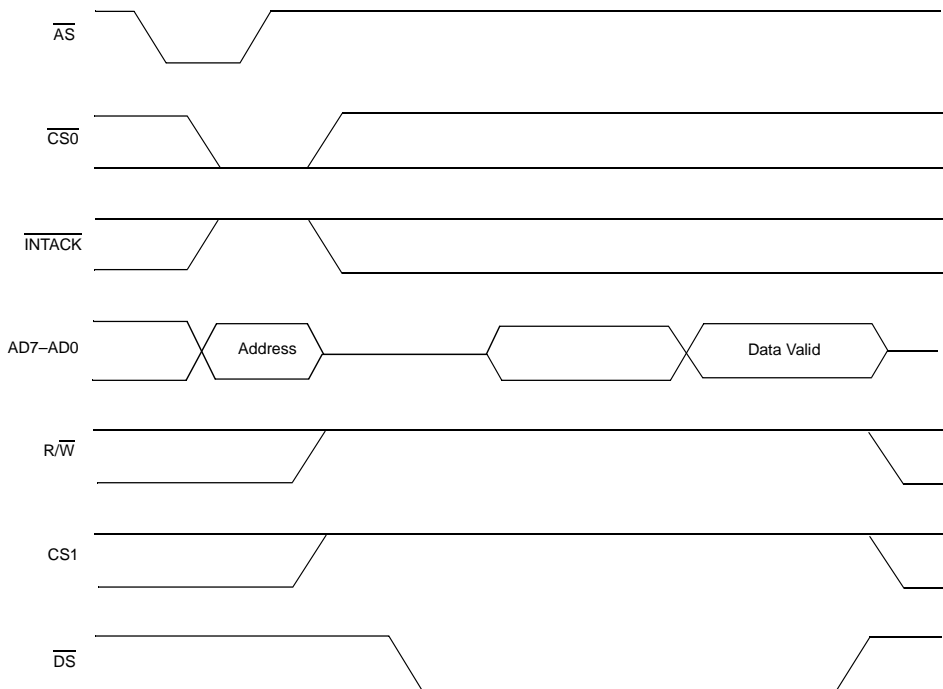


Figure 24. Read Cycle Timing

Write Cycle Timing

Figure 25 illustrates Write cycle timing. The address on $AD7-AD0$ and the state of $\overline{CS0}$ and \overline{INTACK} are latched by the rising edge of \overline{AS} . R/\overline{W} must be Low to indicate a Write cycle. $CS1$ must be High for the Write cycle to occur. \overline{DS} Low strobes the data into the SCC.

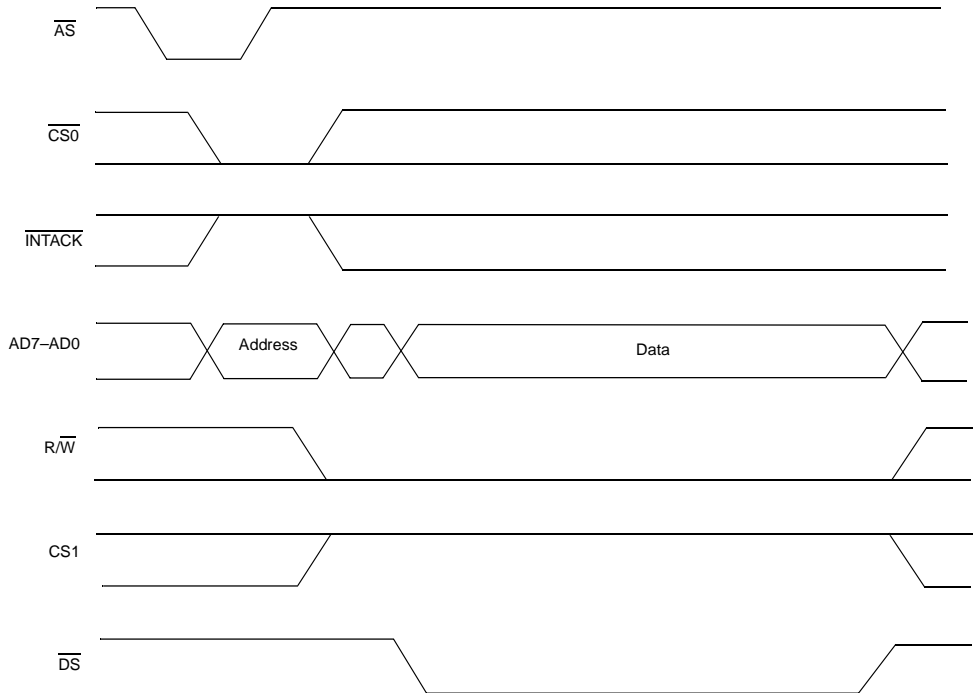


Figure 25. Write Cycle Timing

Interrupt Acknowledge Cycle Timing

Figure 26 illustrates Interrupt Acknowledge cycle timing. The address on $AD7-AD0$ and the state of $\overline{CS0}$ and \overline{INTACK} are latched by the rising edge of \overline{AS} . If \overline{INTACK} is Low, the address and $\overline{CS0}$ are ignored. The state of the R/\overline{W} and $CS1$ are also ignored for the duration of the Interrupt Acknowledge cycle. Between the rising edge of \overline{AS} and the falling edge of \overline{DS} , the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the SCC, and IEI is High when \overline{DS} falls, the Acknowledge cycle was intended for the SCC. In this case, the SCC is pro-



grammed to respond to RD Low by placing its interrupt vector on D7-D0 and internally setting the appropriate Interrupt-Under-Service latch.

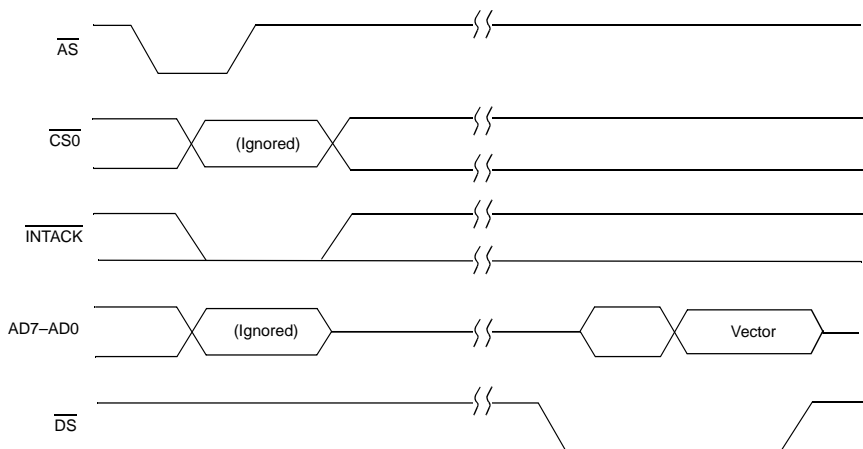


Figure 26. Interrupt Acknowledge Cycle Timing



Electrical Characteristics

Absolute Maximum Ratings

| | |
|--|------------------------------|
| V _{CC} Supply Voltage range | -0.3V to +7.0V |
| Voltages on all pins with respect to GND | -3V to V _{CC} +0.3V |
| T _A Operating Ambient Temperature | See Ordering Information |
| Storage Temperature | -65°C to +150°C |

Stresses greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only. Operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Standard Test Conditions

The DC Characteristics and capacitance sections below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Refer to Figures 27 and 28.

- $+4.50V \leq V_{CC} \leq +5.50V$
- GND = 0V
- T_A as specified in Ordering Information

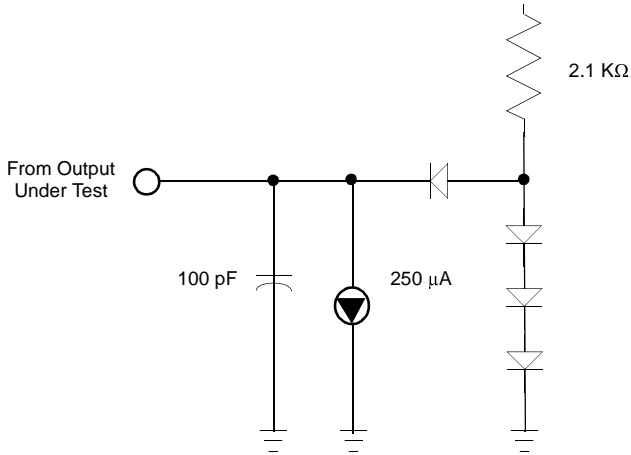


Figure 27. Standard Test Load

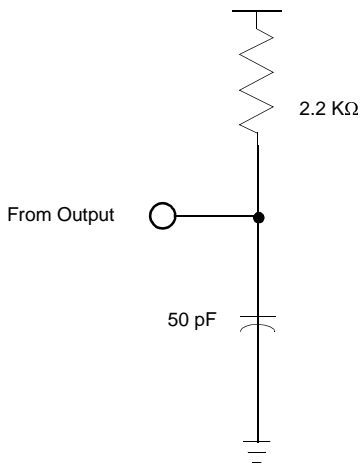


Figure 28. Open-Drain Test Load



Capacitance

Table 3 lists the input, output, and bidirectional capacitance.

Table 3. Capacitance

| Symbol | Parameter | Min | Max | Unit | Test Condition |
|-----------|---------------------------|-----|-----|-----------------|--|
| C_{IN} | Input Capacitance | | 10 | pF ^a | Unmeasured Pins Returned to Ground ^b |
| C_{OUT} | Output Capacitance | | 15 | pF | |
| $C_{I/O}$ | Bidirectional Capacitance | | 20 | pF | |

a. pF = 1 MHz, over specified temperature range.

b. Unmeasured pins returned to Ground.

Miscellaneous

Gate Count is 6800

DC Characteristics

Z80C30/Z85C30

Table 4 lists the dc characteristics for the Z80C30/Z85C30 devices.

Z80C30/Z85C30 CMOS SCC
Serial Communications Controller



Table 4. Z80C30/Z85C30 DC Characteristics

| Symbol | Parameter | Min | Typ | Max | Unit | Condition |
|-------------|--------------------------------------|----------------|-----|------------------|---------------|---|
| V_{IH} | Input High Voltage | 2.2 | | $V_{CC} + 0.3^a$ | V | |
| V_{IL} | Input Low Voltage | -0.3 | | 0.8 | V | |
| V_{OH1} | Output High Voltage | 2.4 | | | V | $I_{OH} = -1.6 \text{ mA}$ |
| V_{OH2} | Output High Voltage | $V_{CC} - 0.8$ | | | V | $I_{OH} = -250 \mu\text{A}$ |
| V_{OL} | Output Low Voltage | | | 0.4 | V | $I_{OL} = +2.0 \text{ mA}$ |
| I_{IL} | Input Leakage | | | ± 10.0 | μA | $0.4 V_{IN} + 2.4V$ |
| I_{OL} | Output Leakage | | | ± 10.0 | μA | $0.4 V_{OUT} + 2.4V$ |
| I_{CC1} | V_{CC} Supply Current ^b | | 7 | 12 (10 MHz) | mA | $V_{CC} = 5V$ $V_{IH} = 4.8$ $V_{IL} = 0$ |
| | | | 9 | 15 (16.384 MHz) | mA | Crystal Oscillator off |
| I_{CCOSC} | Crystal OSC Current ^c | | 4 | | mA | Current for each OSC in addition to I_{CC1} |

- a. $V_{CC} = 5V \pm 10\%$ unless otherwise specified, over specified temperature range.
- b. Typical I_{CC} was measured with oscillator off.
- c. No I_{CC} (OSC) max is specified due to dependency on external circuit and frequency of oscillation.



AC Characteristics

Z85C30 Read/Write Timing Diagrams

Figures 29 through 32 illustrate the Z85C30 read/write timing diagrams. Table 5 lists the Z85C30 timing parameters.

**Z80C30/Z85C30 CMOS SCC
Serial Communications Controller**

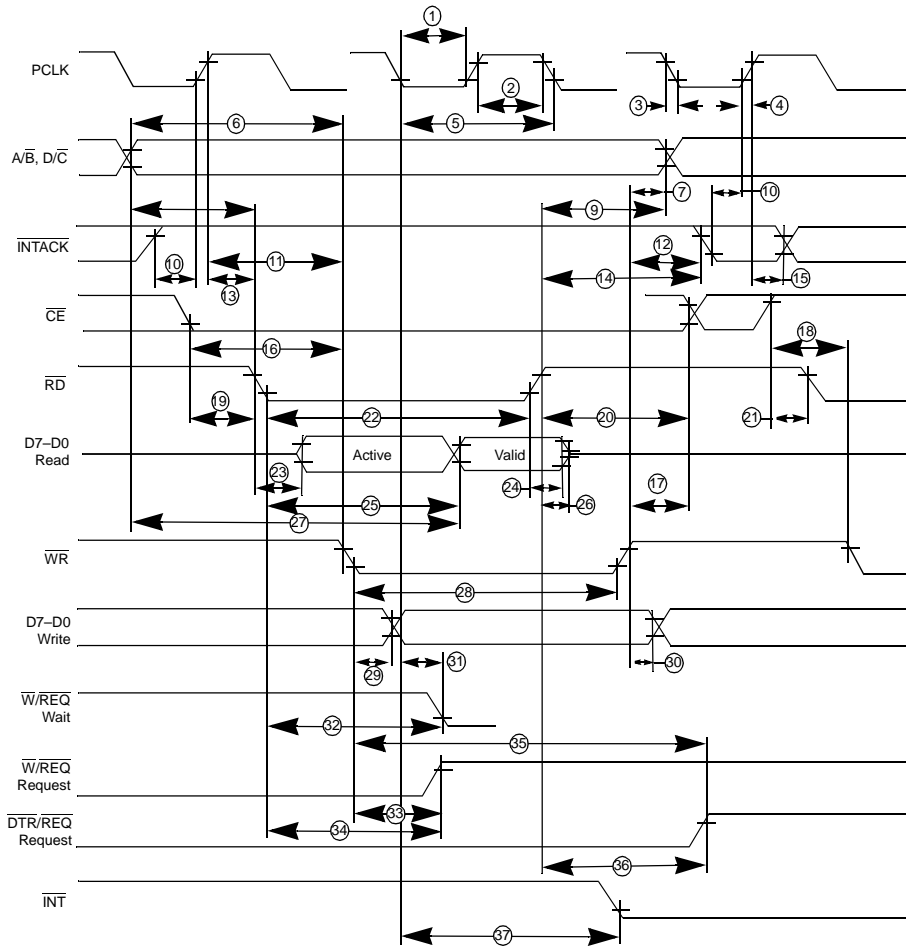


Figure 29. Z85C30 Read/Write Timing Diagram

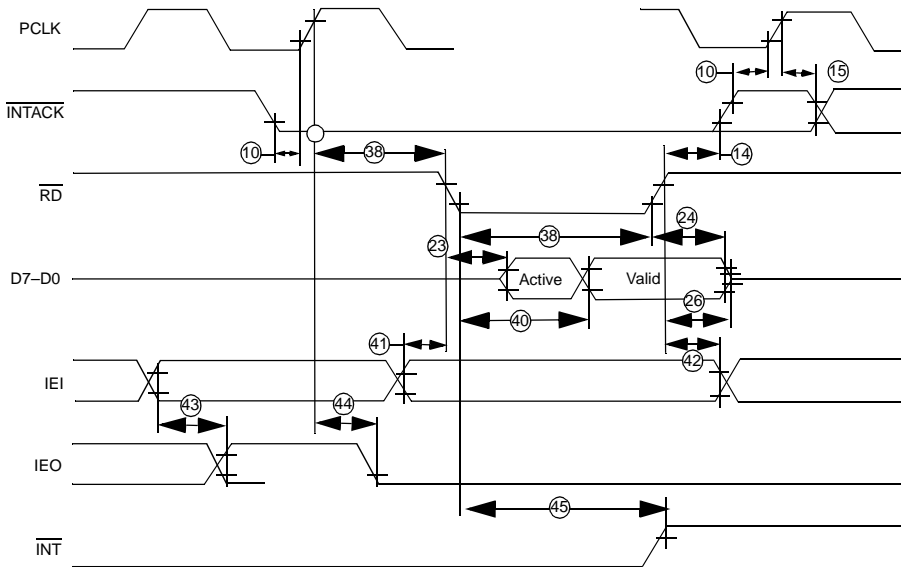


Figure 30. Z85C30 Interrupt Acknowledge Timing Diagram

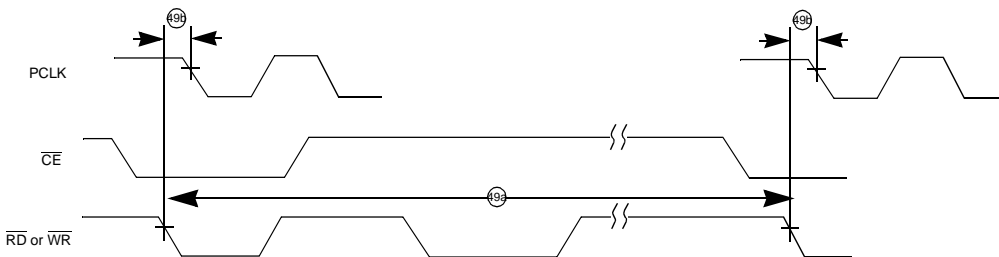


Figure 31. Z85C30 Cycle Timing Diagram

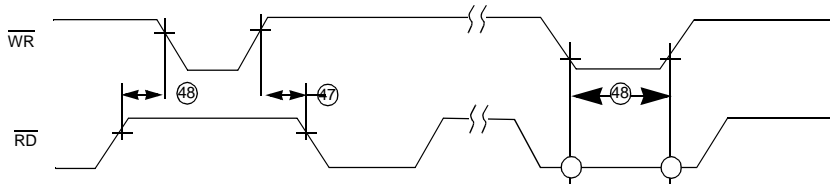


Figure 32. Z85C30 Reset Timing Diagram

Table 5. Z85C30 Read/Write Timing

| No | Symbol | Parameter | 8.5 MHz | | 10 MHz | | 16 MHz | |
|----|----------|--|---------|------|--------|------|--------|------|
| | | | Min | Max | Min | Max | Min | Max |
| 1 | TwPCI | PCLK Low Width | 45 | 2000 | 40 | 2000 | 26 | 2000 |
| 2 | TwPCh | PCLK High Width | 45 | 2000 | 40 | 2000 | 26 | 2000 |
| 3 | TfPC | PCLK Fall Time | | 10 | | 10 | | 5 |
| 4 | TrPC | PCLK Rise Time | | 10 | | 10 | | 5 |
| 5 | TcPC | PCLK Cycle Time | 118 | 4000 | 100 | 4000 | 61 | 4000 |
| 6 | TsA(WR) | Address to \overline{WR} Fall Setup Time | 66 | | 50 | | 35 | |
| 7 | ThA(WR) | Address to \overline{WR} Rise Hold Time | 0 | | 0 | | 0 | |
| 8 | TsA(RD) | Address to \overline{RD} Fall Setup Time | 66 | | 50 | | 35 | |
| 9 | ThA(RD) | Address to \overline{RD} Rise Hold Time | 0 | | 0 | | 0 | |
| 10 | TsiA(PC) | \overline{INTACK} to PCLK Rise Setup Time | 20 | | 20 | | 15 | |



Table 5. Z85C30 Read/Write Timing (continued)

| No | Symbol | Parameter | 8.5 MHz | | 10 MHz | | 16 MHz | |
|----|------------------------|--|---------|-----|--------|-----|--------|-----|
| | | | Min | Max | Min | Max | Min | Max |
| 11 | TsiAi(WR) ^a | $\overline{\text{INTACK}}$ to $\overline{\text{WR}}$ Fall Setup Time | 140 | | 120 | | 70 | |
| 12 | ThIA(WR) | $\overline{\text{INTACK}}$ to $\overline{\text{WR}}$ Rise Hold Time | 0 | | 0 | | 0 | |
| 13 | TsiAi(RD) ¹ | $\overline{\text{INTACK}}$ to RD Fall Setup Time | 140 | | 120 | | 70 | |
| 14 | ThIA(RD) | $\overline{\text{INTACK}}$ to RD Rise Hold Time | 0 | | 0 | | 0 | |
| 15 | ThIA(PC) | $\overline{\text{INTACK}}$ to PCLK Rise Hold Time | 38 | | 30 | | 15 | |
| 16 | TsCEI(WR) | $\overline{\text{CE}}$ Low to $\overline{\text{WR}}$ Fall Setup Time | 0 | | 0 | | 0 | |
| 17 | ThCE(WR) | $\overline{\text{CE}}$ to $\overline{\text{WR}}$ Rise Hold Time | 0 | | 0 | | 0 | |
| 18 | TsCEh(WR) | $\overline{\text{CE}}$ High to $\overline{\text{WR}}$ Fall Setup Time | 58 | | 50 | | 30 | |
| 19 | TsCEI(RD) ¹ | $\overline{\text{CE}}$ Low to $\overline{\text{RD}}$ Fall Setup Time | 0 | | 0 | | 0 | |
| 20 | ThCE(RD) ¹ | $\overline{\text{CE}}$ to $\overline{\text{RD}}$ Rise Hold Time | 0 | | 0 | | 0 | |
| 21 | TsCEh(RD) ¹ | $\overline{\text{CE}}$ High to $\overline{\text{RD}}$ Fall Setup Time | 58 | | 50 | 30 | | |
| 22 | TwRDI ¹ | $\overline{\text{RD}}$ Low Width | 145 | | 125 | | 70 | |
| 23 | TdRD(DRA) | $\overline{\text{RD}}$ Fall to Read Data Active Delay | 0 | | 0 | | 0 | |

Z80C30/Z85C30 CMOS SCC
Serial Communications Controller



Table 5. Z85C30 Read/Write Timing (continued)

| No | Symbol | Parameter | 8.5 MHz | | 10 MHz | | 16 MHz | |
|-----|-------------------------|--|---------|-------|--------|-------|--------|-------|
| | | | Min | Max | Min | Max | Min | Max |
| 24 | TdRDr(DR) | \overline{RD} Rise to Data Not Valid Delay | 0 | | 0 | | 0 | |
| 25 | TdRDI(DR) | \overline{RD} Fall to Read Data Valid Delay | | 135 | | 120 | | 70 |
| 26 | TdRD(DRz) | \overline{RD} Rise to Read Data Float Delay | | 38 | | 35 | | 30 |
| 27 | TdA(DR) | Addr to Read Data Valid Delay | | 210 | | 160 | | 100 |
| 28 | TwWRI | \overline{WR} Low Width | 145 | | 125 | | 75 | |
| 29 | TdWR(DW) | \overline{WR} Fall to Write Data Valid Delay | | 35 | | 35 | | 20 |
| 30 | ThDW(WR) | Write Data to \overline{WR} Rise Hold Time | 0 | | 0 | | 0 | |
| 31 | TdWR(W) ^b | \overline{WR} Fall to Wait Valid Delay | | 168 | | 100 | | 50 |
| 32 | TdRD(W) ² | \overline{RD} Fall to Wait Valid Delay | | 168 | | 100 | | 50 |
| 33 | TdWRf(REQ) | \overline{WR} Fall to $\overline{W/REQ}$ Not Valid Delay | | 168 | | 120 | | 70 |
| 34 | TdRDf(REQ) ^c | \overline{RD} Fall to $\overline{W/REQ}$ Not Valid Delay | | 168 | | 120 | | 70 |
| 35a | TdWRr(REQ) | \overline{WR} Fall to $\overline{DTR/REQ}$ Not Valid | | 4TcPc | | 4TcPc | | 4TcPc |
| 35b | TdWRr(REQ) ³ | \overline{WR} Fall to $\overline{DTR/REQ}$ Not Valid | | 168 | | 100 | | 70 |



Table 5. Z85C30 Read/Write Timing (continued)

| No | Symbol | Parameter | 8.5 MHz | | 10 MHz | | 16 MHz | |
|----|-------------------------|---|---------|-----|--------|-----|--------|-----|
| | | | Min | Max | Min | Max | Min | Max |
| 36 | TdRDrrREQ) | \overline{RD} Rise to $\overline{DTR}/\overline{REQ}$ Not Valid Delay | | NA | | NA | | NA |
| 37 | TdPC(INT) | PCLK Fall to \overline{INT} Valid Delay | | 500 | | 320 | | 175 |
| 38 | TdIAi(RD) ^d | \overline{INTACK} to \overline{RD} Fall (Ack) Delay | 145 | | 90 | | 50 | |
| 39 | TwrDA | \overline{RD} (Acknowledge) Width | 145 | | 125 | | 75 | |
| 40 | TdRDA(DR) | \overline{RD} Fall (Ack) to Read Data Valid Delay | 135 | | 120 | | 70 | |
| 41 | TsiEI(RDA) | IEI to \overline{RD} Fall (Ack) Setup Time | 95 | | 80 | | 50 | |
| 42 | ThIEI(RDA) | IEI to \overline{RD} Rise (Ack) Hold Time | 0 | | 0 | | 0 | |
| 43 | TdIEIrlEO) | IEI to IEO Delay Time | | 95 | | 80 | | 45 |
| 44 | TdPC(IEO) | PCLK Rise to IEO Delay | | 195 | | 175 | | 80 |
| 45 | TdRDA(INT) ² | \overline{RD} Fall to \overline{INT} Inactive Delay | | 480 | | 320 | | 200 |
| 46 | TdRDrrWRQ) | \overline{RD} Rise to \overline{WR} Fall Delay for No Reset | 15 | | 15 | | 10 | |



Table 5. Z85C30 Read/Write Timing (continued)

| No | Symbol | Parameter | 8.5 MHz | | 10 MHz | | 16 MHz | |
|-----|-------------------|---|---------|---------|---------|-----|--------|-----|
| | | | Min | Max | Min | Max | Min | Max |
| 47 | TdWRQ(RD) | \overline{WR} Rise to \overline{RD} Fall Delay for No Reset | 15 | | 15 | | 10 | |
| 48 | TwRES | \overline{WR} and \overline{RD} Low for Reset | 145 | | 100 | | 75 | |
| 49a | Trc ^e | Valid Access Recovery Time | 3.5TcPc | 3.5TcPc | 3.5TcPc | | | |
| 49b | Trci ^f | \overline{RD} or \overline{WR} Fall to PC Fall Setup Time | 0 | | 0 | | 0 | |

- a. Parameter does not apply to Interrupt Acknowledge transactions.
- b. Open-drain output, measured with open-drain test load.
- c. Parameter applies to enhanced Request mode only ($WR7' D4 = 1$).
- d. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain. TsiEI(RDA) for the SCC and TdIEI(IEO) for each device separating them in the daisy chain.
- e. Parameter applies only between transactions involving the Z85C30 SL1480, if $\overline{WR}/\overline{RD}$ falling edge is synchronized to PCLK falling edge, then $Trc = 3TcPc$.
- f. This specification is only applicable when Valid Access Recovery Time is less than 35 PCLK.

Figure 33 is the Z85C30 general timing diagram. Table 6 lists the Z85C30 general timing characteristics. Z85C30 system timing is shown in Figure 34 and described in Table 7. Table 8 provides Z85C30 read/write timing characteristics.

Figures 35 through 37 illustrate Z80C30 read/write timing, interrupt acknowledge timing, and reset timing respectively. Table 9 provides Z80C30 read/write timing characteristics.

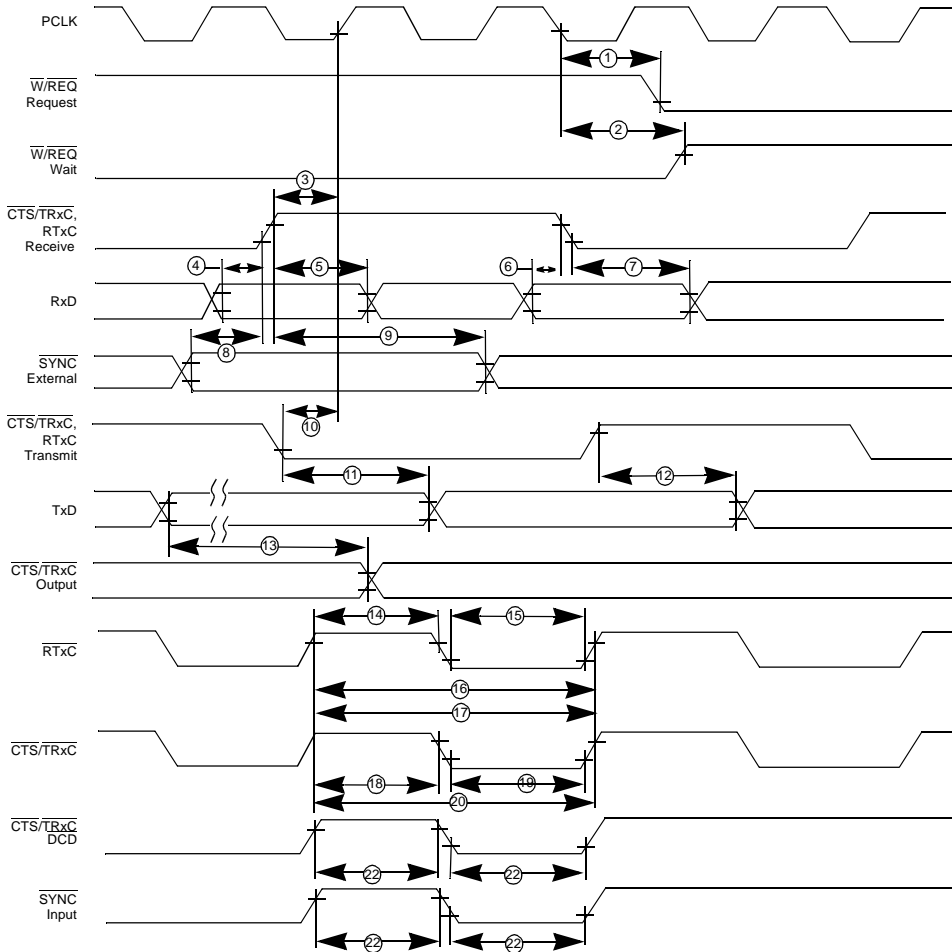


Figure 33. Z85C30 General Timing Diagram

Z80C30/Z85C30 CMOS SCC
Serial Communications Controller



Table 6. Z85C30 General Timing Table

| No | Symbol | Parameter | 8.5 MHz | | 10 MHz | | 16 MHz | |
|-----|-------------|---|---------|-----|--------|-----|--------|-----|
| | | | Min | Max | Min | Max | Min | Max |
| 1 | TdPC(REQ) | $\overline{\text{PCLK}}$ to W/REQ Valid | | 250 | | 150 | | 80 |
| 2 | TdPC(W) | $\overline{\text{PCLK}}$ to Wait Inactive | | 350 | | 250 | | 180 |
| 3 | TsRXC(PC) | $\overline{\text{RxC}}$ to $\overline{\text{PCLK}}$ Setup Time ^{a,b} | N/A | | N/A | | N/A | |
| 4 | TsRXD(RXCr) | RxD to $\overline{\text{RxC}}$ Setup Time ¹ | | 0 | | 0 | | 0 |
| 5 | ThRXD(RxCr) | RxD to /RXC Hold Time ¹ | 150 | | 125 | | 50 | |
| 6 | TsRXD(RXCf) | RxD to /RXC Setup Time ^{1,c} | 0 | | 0 | | 0 | |
| 7 | ThRXD(RXCf) | RxD to /RXC Hold Time ^{1,3} | 150 | 1 | 25 | | 50 | |
| 8 | TsSY(RXC) | $\overline{\text{SYNC}}$ to $\overline{\text{RxC}}$ Setup Time ¹ | -200 | | -150 | | -100 | |
| 9 | ThSY(RXC) | $\overline{\text{SYNC}}$ to $\overline{\text{RxC}}$ Hold Time ¹ | 5TcPc | | 5TcPc | | 5TcPc | |
| 10 | TsTXC(PC) | TxC to $\overline{\text{PCLK}}$ Setup Time ^{d,e} | N/A | | N/A | | N/A | |
| 11 | TdTXCf(TXD) | $\overline{\text{TxC}}$ to TxD Delay ⁴ | | 200 | | 150 | | 80 |
| 12 | TdTxCr(TXD) | $\overline{\text{TxC}}$ to TxD Delay ^{4,3} | | 200 | | 150 | | 80 |
| 13 | TdTXD(TRX) | TxD to TRxC Delay | | 200 | | 140 | | 80 |
| 14a | TwRTXh | RTxC High Width ^f | 150 | | 120 | | 80 | |
| 14b | TwRTXh(E) | $\overline{\text{RTxC}}$ High Width ⁹ | 50 | | 40 | | 15.6 | |
| 15a | TwRTXI | TRxC Low Width ⁶ | 150 | | 120 | | 80 | |
| 15b | TwRTXI(E) | $\overline{\text{RTxC}}$ Low Width ⁷ | 50 | | 40 | | 15.6 | |
| 16a | TcRTX | RTxC Cycle Time ^{6,h} | 488 | | 400 | | 244 | |



Table 6. Z85C30 General Timing Table (continued)

| No | Symbol | Parameter | 8.5 MHz | | 10 MHz | | 16 MHz | |
|-----|----------|--|---------|------|--------|------|--------|------|
| | | | Min | Max | Min | Max | Min | Max |
| 16b | TcRTX(E) | $\overline{\text{RTxC}}$ Cycle Time ⁷ | 125 | | 100 | | 31.25 | |
| 17 | TcRTXX | Crystal Osc. Period ⁱ | 125 | 1000 | 100 | 1000 | 62 | 1000 |
| 18 | TwTRXh | TRxC High Width ⁶ | 150 | | 120 | | 180 | |
| 19 | TwTRXI | TRxC Low Width ⁶ | 150 | | 120 | | 80 | |
| 20 | TcTRX | TRxC Cycle Time ^{6,8} | 488 | | 400 | | 244 | |
| 21 | TwEXT | $\overline{\text{DCD}}$ or $\overline{\text{CTS}}$ Pulse Width | 200 | | 120 | | 70 | |
| 22 | TwSY | $\overline{\text{SYNC}}$ Pulse Width | 200 | | 120 | | 70 | |

- a. RxC is $\overline{\text{RTxC}}$ or $\overline{\text{TRxC}}$, whichever is supplying the receive clock.
- b. Synchronization of RxC to PCLK is eliminated in divide by four operation.
- c. Parameter applies only to FM encoding/decoding.
- d. TxC is $\overline{\text{TRxC}}$ or $\overline{\text{RTxC}}$, whichever is supplying the transmit clock.
- e. External PCLK to RTxC or TxC synchronization requirement eliminated for PCLK divide-by-four operation. $\overline{\text{TRxC}}$ and $\overline{\text{RTxC}}$ rise and fall times are identical to PCLK. Reference timing specs TfPC and TrPC. Tx and Rx input clock slew rates should be kept to a maximum of 30 nsec. All parameters related to input CLK edges should be referenced at the point at which the transition begins or ends, whichever is worst case.
- f. Parameter applies only for transmitter and receiver; DPLL and Baud Rate Generator timing requirements are identical to case PCLK requirements.
- g. ENHANCED FEATURE — $\overline{\text{RTxC}}$ used as input to internal DPLL only.
- h. The maximum receive or transmit data rate is 1/4 PCLK.
- i. Both $\overline{\text{RTxC}}$ and $\overline{\text{SYNC}}$ have 30 pF capacitors to ground connections.

Z80C30 general timing is shown in Figure 38 with parameters provided in Table 10. Z80C30 system timing is shown in Figure 39 with parameters provided in Table 11.

Z80C30/Z85C30 CMOS SCC
Serial Communications Controller

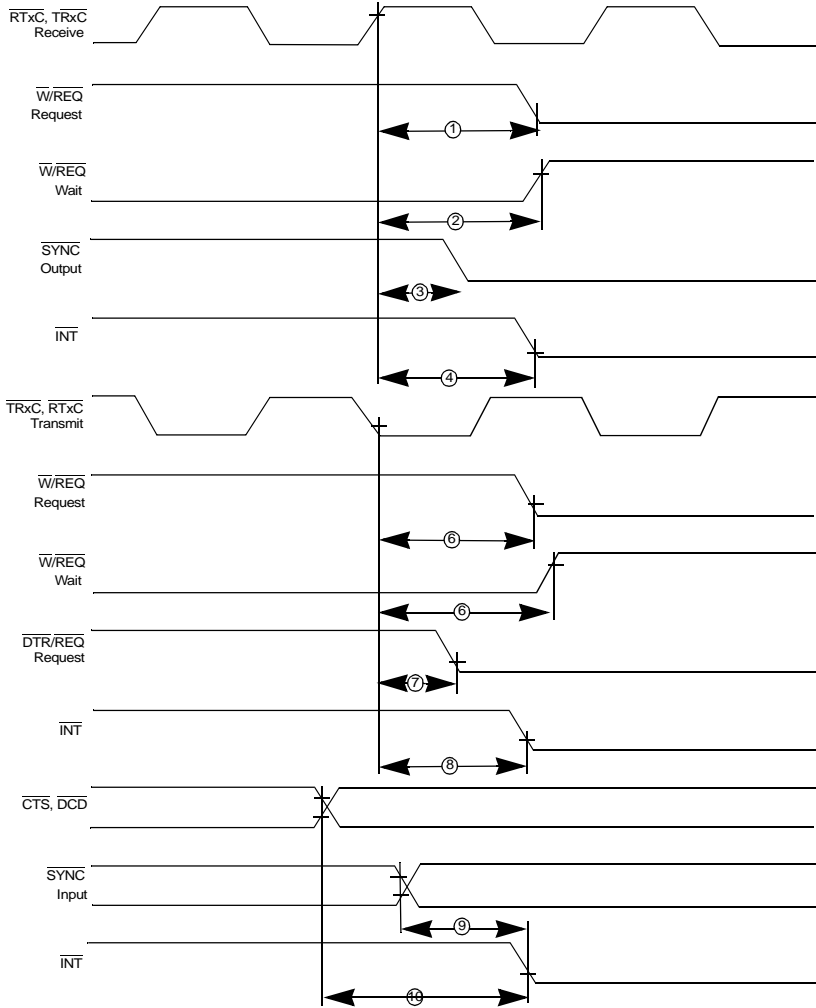


Figure 34. Z85C30 System Timing Diagram



Table 7. Z85C30 System Timing Table

| No | Symbol | Parameter | 8.5 MHz | | 10 MHz | | 16 MHz | |
|----|------------|--|---------|-----|--------|-----|--------|-----|
| | | | Min | Max | Min | Max | Min | Max |
| 1 | TdRXC(REQ) | $\overline{\text{Rx}}\overline{\text{C}}$ High to $\overline{\text{W}}/\overline{\text{REQ}}$ Valid ^{a,b} | 8 | 12 | 8 | 12 | 8 | 12 |
| 2 | TdRXC(W) | $\overline{\text{Rx}}\overline{\text{C}}$ High to Wait Inactive ^{1,2,c} | 8 | 14 | 8 | 14 | 8 | 14 |
| 3 | TdRdXC(SY) | $\overline{\text{Rx}}\overline{\text{C}}$ High to $\overline{\text{SYNC}}$ Valid ^{1,2} | 4 | 7 | 4 | 7 | 4 | 70 |
| 4 | TsRXC(INT) | $\overline{\text{Rx}}\overline{\text{C}}$ High to INT Valid ^{1,2,3} | 10 | 16 | 10 | 16 | 10 | 16 |
| 5 | TdTXC(REQ) | $\overline{\text{Tx}}\overline{\text{C}}$ Low to $\overline{\text{W}}/\overline{\text{REQ}}$ Valid ^{2,d} | 5 | 8 | 5 | 8 | 5 | 8 |
| 6 | TdTXC(W) | $\overline{\text{Tx}}\overline{\text{C}}$ Low to Wait Inactive ^{2,3,4} | 5 | 11 | 5 | 11 | 5 | 11 |
| 7 | TdTXC(DRQ) | $\overline{\text{Tx}}\overline{\text{C}}$ Low to $\overline{\text{DTR}}/\overline{\text{REQ}}$ Valid ^{3,4} | 4 | 7 | 4 | 7 | 4 | 7 |
| 8 | TdTXC(INT) | $\overline{\text{Tx}}\overline{\text{C}}$ Low to $\overline{\text{INT}}$ Valid ^{2,3,4} | 6 | 10 | 6 | 10 | 6 | 10 |
| 9a | TdSY(INT) | SYNC to INT Valid ^{2,3} | 2 | 6 | 2 | 6 | 2 | 6 |
| 9b | TdSY(INT) | SYNC to INT Valid ^{2,3,e} | 2 | 3 | 2 | 3 | 2 | 3 |
| 10 | TdEXT(INT) | $\overline{\text{DCD}}$ or $\overline{\text{CTS}}$ to $\overline{\text{INT}}$ Valid ^{2,3} | 2 | 6 | 2 | 6 | 2 | 6 |

- a. $\overline{\text{Rx}}\overline{\text{C}}$ is $\overline{\text{RTx}}\overline{\text{C}}$ or $\overline{\text{TRx}}\overline{\text{C}}$, whichever is supplying the receive clock.
- b. Units equal to TcPc.
- c. Open-drain output, measured with open-drain test load.
- d. $\overline{\text{Tx}}\overline{\text{C}}$ is $\overline{\text{TRx}}\overline{\text{C}}$ or $\overline{\text{RTx}}\overline{\text{C}}$ whichever is supplying the transmit clock.
- e. Units equal to AS.

Z80C30/Z85C30 CMOS SCC
Serial Communications Controller



Table 8. Z85C30 Read/Write Timing

| No | Symbol | Parameter | 8.5 MHz | | 10 MHz | | 16 MHz | |
|----|----------|---|---------|------|--------|------|--------|------|
| | | | Min | Max | Min | Max | Min | Max |
| 1 | TwPCI | PCLK Low Width | 45 | 2000 | 40 | 2000 | 26 | 2000 |
| 2 | TwPCh | PCLK High Width | 45 | 2000 | 40 | 2000 | 26 | 2000 |
| 3 | TfPC | PCLK Fail Time | | 10 | | 10 | | 5 |
| 4 | TrPC | PCLK Rise Time | | 10 | | 10 | | 5 |
| 5 | TcPC | PCLK Cycle Time | 118 | 4000 | 100 | 4000 | 61 | 4000 |
| 6 | TsA(WR) | Address to \overline{WR} Fail Setup Time | 66 | | 50 | | 35 | |
| 7 | ThA(WR) | Address to \overline{WR} Rise Hold Time | 0 | | 0 | | 0 | |
| 8 | TsA(RD) | Address to \overline{RD} Fall Setup Time | 66 | | 50 | | 35 | |
| 9 | ThA(RD) | Address to \overline{RD} Rise Hold Time | 0 | | 0 | | 0 | |
| 10 | TsiA(PC) | \overline{INTACK} to PCLK Rise Setup Time | 20 | | 20 | | 15 | |

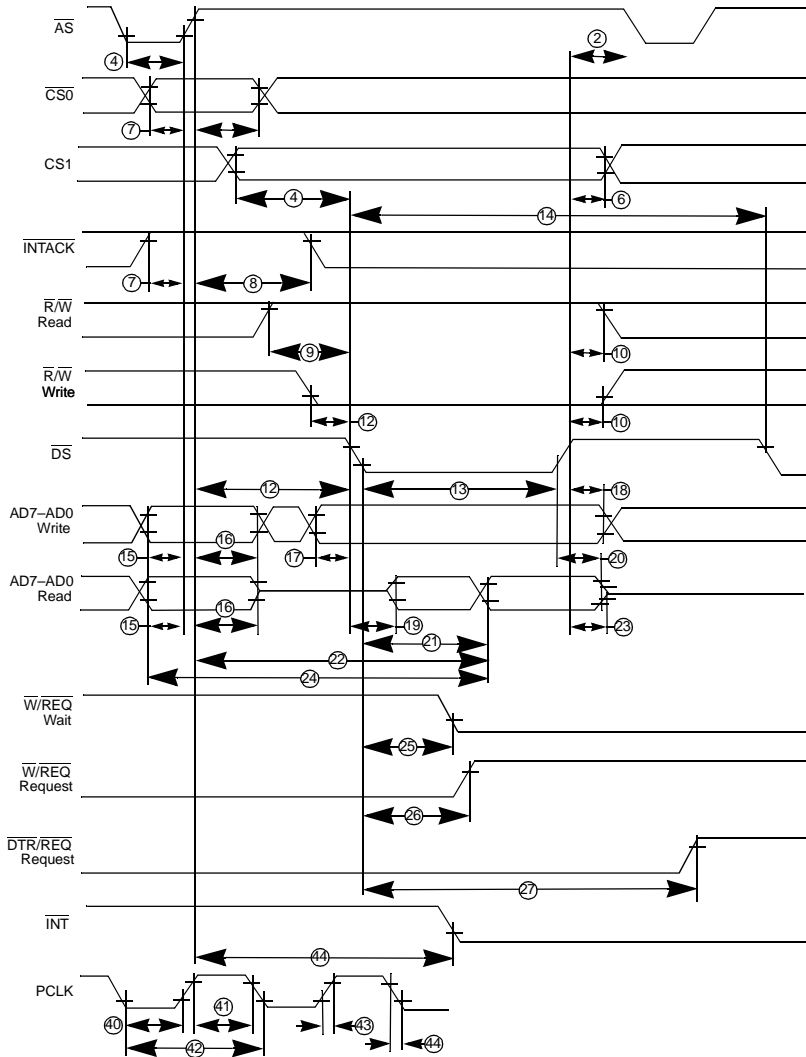


Figure 35. Z80C30 Read/Write Timing Diagram

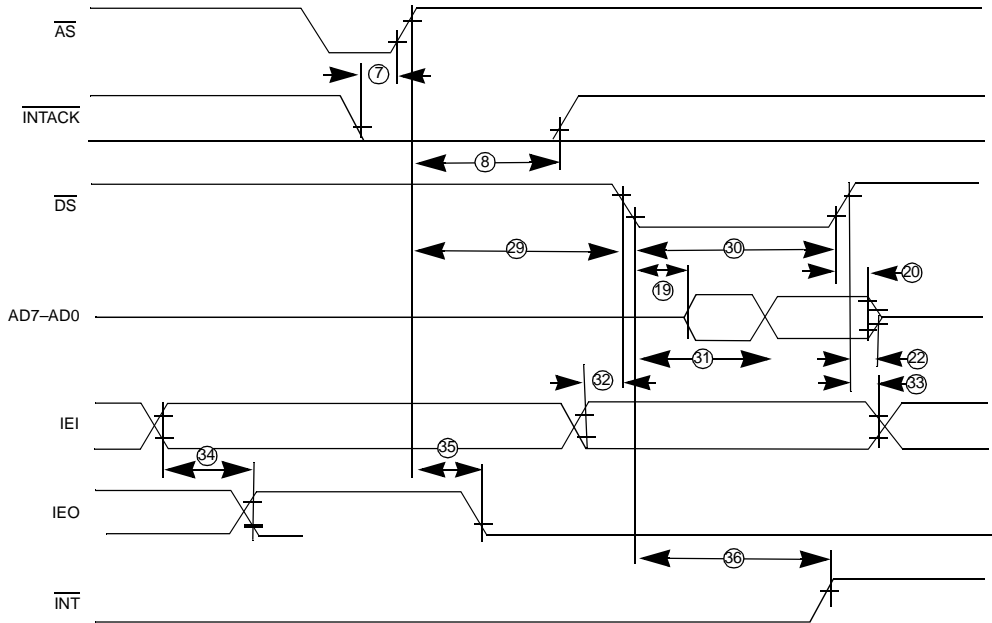


Figure 36. Z80C30 Interrupt Acknowledge Timing Diagram

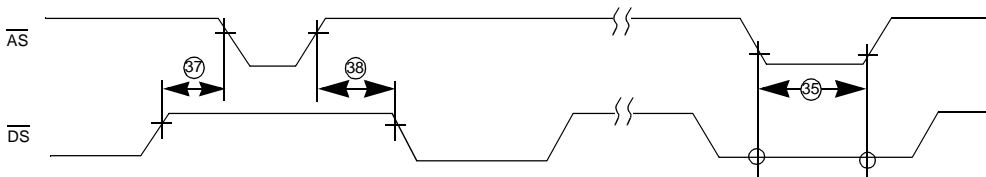


Figure 37. Z80C30 Reset Timing Diagram



Table 9. Z80C30 Read/Write Timing^a

| No | Symbol | Parameter | 8 MHz | | 10 MHz | |
|----|-----------|--|-------|-----|--------|-----|
| | | | Min | Max | Min | Max |
| 1 | TwAS | \overline{AS} Low Width | 35 | | 30 | |
| 2 | TdDS(AS) | \overline{DS} Rise to \overline{AS} Fall Delay ^b | 15 | | 10 | |
| 3 | TsCSO(AS) | $\overline{CS0}$ to \overline{AS} Rise Setup Time ² | 0 | | 0 | |
| 4 | ThCSO(AS) | $\overline{CS0}$ to \overline{AS} Rise Hold Time ² | 30 | | 20 | |
| 5 | TsCS1(DS) | CS1 to \overline{DS} Fall Setup Time ² | 65 | | 50 | |
| 6 | ThCS1(DS) | CS1 to \overline{DS} Rise Hold Time ² | 30 | | 20 | |
| 7 | TsiA(AS) | \overline{INTACK} to \overline{AS} Rise Setup Time | 10 | | 10 | |
| 8 | ThIA(AS) | \overline{INTACK} to \overline{AS} Rise Hold Time | 150 | | 125 | |
| 9 | TsRWR(DS) | R/ \overline{W} (Read) to \overline{DS} Fall Setup Time | 65 | | 50 | |
| 10 | ThRW(DS) | R/ \overline{W} to \overline{DS} Rise Hold Time | 0 | | 0 | |
| 11 | TsRWW(DS) | R/ \overline{W} (Write) to \overline{DS} Fall Setup Time | 0 | | 0 | |
| 12 | TdAS(DS) | \overline{AS} Rise to \overline{DS} Fall Delay | 30 | | 20 | |
| 13 | TwDSI | \overline{DS} Low Width | 150 | | 125 | |
| 14 | TrC | Valid Access Recovery Time ^c | 4TcPC | | 4TcPC | |
| 15 | TsA(AS) | Address to \overline{AS} Rise Setup Time ² | 10 | | 10 | |
| 16 | ThA(AS) | Address to \overline{AS} Rise Hold Time ² | 25 | | 20 | |
| 17 | TsDW(DS) | Write Data to \overline{DS} Fall Setup Time | 15 | | 10 | |
| 18 | ThDW(DS) | Write Data to \overline{DS} Rise Hold Time | 0 | | 0 | |



Table 9. Z80C30 Read/Write Timing^a (continued)

| No | Symbol | Parameter | 8 MHz | | 10 MHz | |
|----|------------|---|-------|-------|--------|-------|
| | | | Min | Max | Min | Max |
| 19 | TdDS(DA) | \overline{DS} Fall to Data Active Delay | 0 | | 0 | |
| 20 | TdDSr(DR) | \overline{DS} Rise to Read Data Not Valid Delay | 0 | | 0 | |
| 21 | TdDSf(DR) | \overline{DS} Fall to Read Data Valid Delay | | 140 | | 120 |
| 22 | TdAS(DR) | \overline{AS} Rise to Read Data Valid Delay | | 250 | | 190 |
| 23 | TdDS(DRz) | \overline{DS} Rise to Read Data Float Delay ^d | | 40 | | 35 |
| 24 | TdA(DR) | Address Required Valid to Read Data Valid Delay | | 260 | | 210 |
| 25 | TdDS(W) | \overline{DS} Fall to Wait Valid Delay ^e | | 170 | | 160 |
| 26 | TdDSf(REQ) | \overline{DS} Fall to $\overline{W/REQ}$ Not Valid Delay | | 170 | | 160 |
| 27 | TdDSr(REQ) | \overline{DS} Fall to $\overline{DTR/REQ}$ Not Valid Delay | | 4TcPC | | 4TcPC |
| 28 | TdAS(INT) | \overline{AS} Rise to \overline{INT} Valid Delay ⁵ | | 500 | | 500 |
| 29 | TdAS(DSA) | \overline{AS} Rise to \overline{DS} Fall (Acknowledge) Delay ^f | 250 | | 225 | |
| 30 | TwDSA | \overline{DS} (Acknowledge) Low Width | 150 | | 125 | |
| 31 | TdDSA(DR) | \overline{DS} Fall (Acknowledge) to Read Data Valid Delay | | 140 | | 120 |
| 32 | TsiEI(DSA) | IEI to \overline{DS} Fall (Acknowledge) Setup Time | 80 | | 80 | |



Table 9. Z80C30 Read/Write Timing^a (continued)

| No | Symbol | Parameter | 8 MHz | | 10 MHz | |
|----|------------|--|-------|------|--------|------|
| | | | Min | Max | Min | Max |
| 33 | ThIEI(DSA) | IEI to \overline{DS} Rise (Acknowledge) Hold Time | 0 | | 0 | |
| 34 | TdIEI(IEO) | IEI to IEO Delay | | 90 | | 90 |
| 35 | TdAS(IEO) | \overline{AS} Rise to IEO Delay ^g | | 200 | | 175 |
| 36 | TdDSA(INT) | \overline{DS} Fall (Acknowledge) to \overline{INT} Inactive Delay ⁵ | | 450 | | 450 |
| 37 | TdDS(ASQ) | \overline{DS} Rise to \overline{AS} Fall Delay for No Reset | 15 | | 15 | |
| 38 | TdASQ(DS) | \overline{AS} Rise to \overline{DS} Fall Delay for No Reset | 20 | | 15 | |
| 39 | TwRES | \overline{AS} and \overline{DS} Coincident Low for Reset ^h | 150 | | 100 | |
| 40 | TwPCI | PCLK Low Width | 50 | 1000 | 40 | 1000 |
| 41 | TwPCh | PCLK High Width | 50 | 1000 | 40 | 1000 |
| 42 | TcPC | PCLK Cycle Time | 125 | 2000 | 100 | 2000 |
| 43 | TrPC | PCLK Rise Time | | 10 | | 10 |
| 44 | TfPC | PCLK Fall Time | | 10 | | 10 |

- a. Units in nanoseconds (ns) unless otherwise noted.
- b. Parameter does not apply to Interrupt Acknowledge transactions.
- c. Parameter applies only between transactions involving the SCC.
- d. Float delay is defined as the time required for a $\pm 0.5V$ change in the output with a maximum DC load and a minimum AC load.
- e. Open-drain output, measured with open-drain test load.
- f. Parameter is system dependent. For any Z-SCC in the daisy chain. TdAS(DSA) must be greater than the sum of TdAS(IEO) for the highest priority device in the daisy chain TsiEI(DSA) for the Z-SCC, and TdIEI(IEO) for each device separating them in the daisy chain.

Z80C30/Z85C30 CMOS SCC Serial Communications Controller



78

- g. Parameter applies only to a Z-SCC pulling INT Low at the beginning of the Interrupt Acknowledge transaction.
- h. Internal circuitry allows for the reset provided by the ZB to be recognized as a reset by the Z-SCC. All timing references assume 20V for a logic "1" and 08V for a logic "0".

Figure 38 displays Z80C30 general timing and Table 10 lists the associated general timing characteristics. Figure 39 displays the Z80C30 system timing with the associated parameters listed in Table 11.

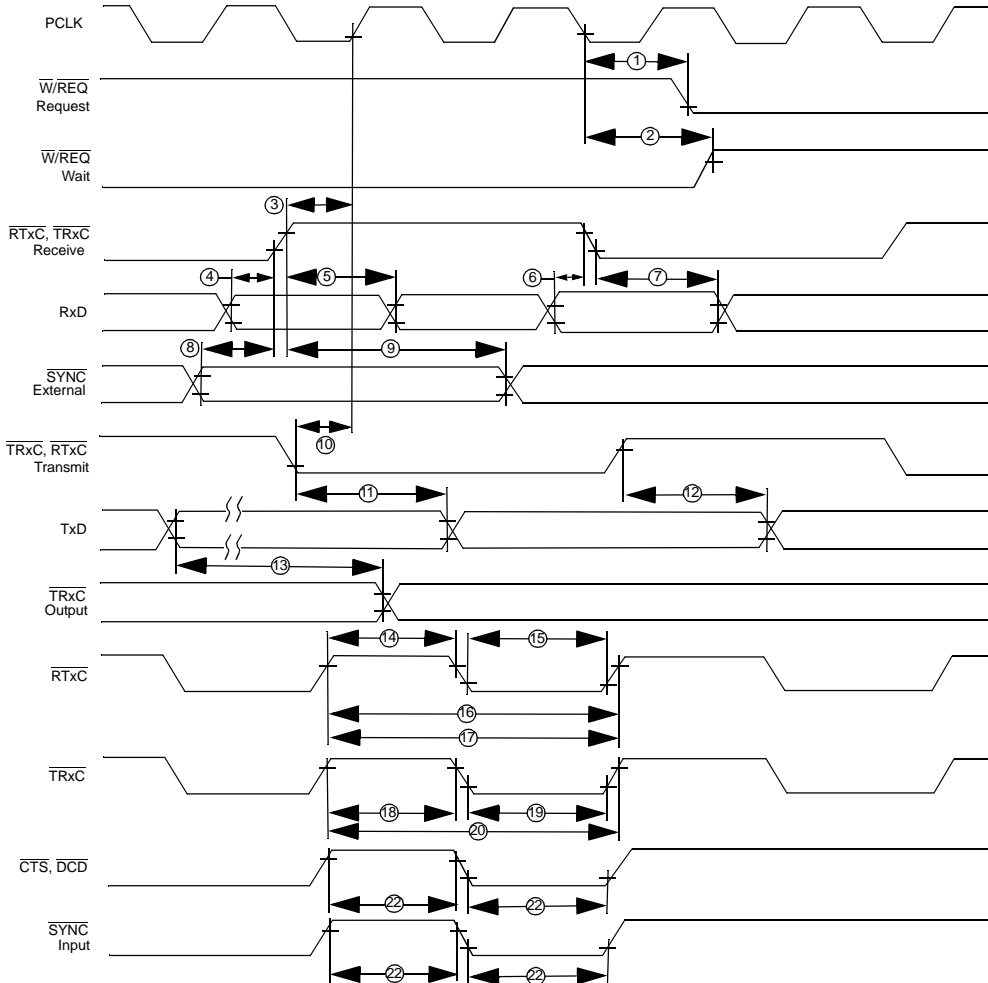


Figure 38. Z80C30 General Timing Diagram

Z80C30/Z85C30 CMOS SCC
Serial Communications Controller



Table 10. Z80C30 General Timing^a

| No | Symbol | Parameter | 8 MHz | | 10 MHz | |
|-----|-------------|---|-------|------|--------|------|
| | | | Min | Max | Min | Max |
| 1 | TdPC(REQ) | $\overline{\text{PCLK}}$ Low to W/REQ Valid | | 250 | | 200 |
| 2 | TsPC(W) | $\overline{\text{PCLK}}$ Low to Wait Inactive | | 350 | | 300 |
| 3 | TsRXC(PC) | $\overline{\text{RxC}}$ High to $\overline{\text{PCLK}}$ High Setup Time ^{b,c} | NA | NA | NA | NA |
| 4 | TsRXD(RXCr) | RxD to $\overline{\text{RxC}}$ High Setup Time | 0 | | 0 | |
| 5 | ThRXD(RxCr) | RxD to $\overline{\text{RxC}}$ High Hold Time ² | 150 | | 125 | |
| 6 | TsRXD(RXCf) | RxD to $\overline{\text{RxC}}$ Low Setup Time ^{2,d} | 0 | | 0 | |
| 7 | ThRXD(RXCf) | RxD to $\overline{\text{RxC}}$ Low Hold Time ^{2,4} | 150 | | 125 | |
| 8 | TsSY(RXC) | SYNC to $\overline{\text{RxC}}$ High Setup Time ² | -200 | | -150 | |
| 9 | ThSY(RXC) | SYNC to $\overline{\text{RxC}}$ High Hold Time ² | 5TcPc | | 5TcPc | |
| 10 | TsTXC(PC) | $\overline{\text{TxC}}$ Low to $\overline{\text{PCLK}}$ High Setup Time ^{e,3} | NA | | NA | |
| 11 | TdTXCf(TXD) | $\overline{\text{TxC}}$ Low to TxD Delay ⁵ | | 190 | | 150 |
| 12 | TdTxCr(TXD) | $\overline{\text{TxC}}$ High to TxD Delay ^{5,4} | | 190 | | 150 |
| 13 | TdTXD(TRX) | TxD to TRxC Delay | | 200 | | 140 |
| 14 | TwRTXh | RTxC High Width ^f | 130 | | 120 | |
| 15 | TwRTXI | TRxC Low Width ⁶ | 130 | | 120 | |
| 16a | TcRTX | RTxC Cycle Time ^{6,g} | 472 | | 400 | |
| 16b | TxRx (DPLL) | DPLL Cycle Time Min ^{7,h} | 59 | | 50 | |
| 17 | TcRTXX | Crystal Osc. Period ⁱ | 118 | 1000 | 100 | 1000 |
| 18 | TwTRXh | TRxC High Width ⁶ | 130 | | 120 | |



Table 10. Z80C30 General Timing^a (continued)

| No | Symbol | Parameter | 8 MHz | | 10 MHz | |
|----|--------|--------------------------------|-------|-----|--------|-----|
| | | | Min | Max | Min | Max |
| 19 | TwTRXI | TRxC Low Width ⁶ | 130 | | 120 | |
| 20 | TcTRX | TRxC Cycle Time ^{6,7} | 472 | | 400 | |
| 21 | TwEXT | DCD or CTS Pulse Width | 200 | | 120 | |
| 22 | TwSY | SYNC Pulse Width | 200 | | 120 | |

- a. Units in nanoseconds (ns) otherwise noted.
- b. RxC is \overline{RTxC} or \overline{TRxC} , whichever is supplying the receive clock.
- c. Synchronization of RxC to PCLK is eliminated in divide by four operation.
- d. Parameter applies only to FM encoding/decoding.
- e. TxC is \overline{TRxC} or \overline{RTxC} , whichever is supplying the transmit clock.
- f. Parameter applies only for transmitter and receiver; DPLL and Baud Rate Generator timing requirements are identical to case PCLK requirements.
- g. The maximum receive or transmit data rate is 1/4 PCLK.
- h. Applies to DPLL clock source only Maximum data rate of 1/4 PCLK still applies DPLL clock should have a 50% duty cycle.
- i. Both \overline{RTxC} and SYNC have 30 pf capacitors to ground connected to them.

Z80C30/Z85C30 CMOS SCC
Serial Communications Controller

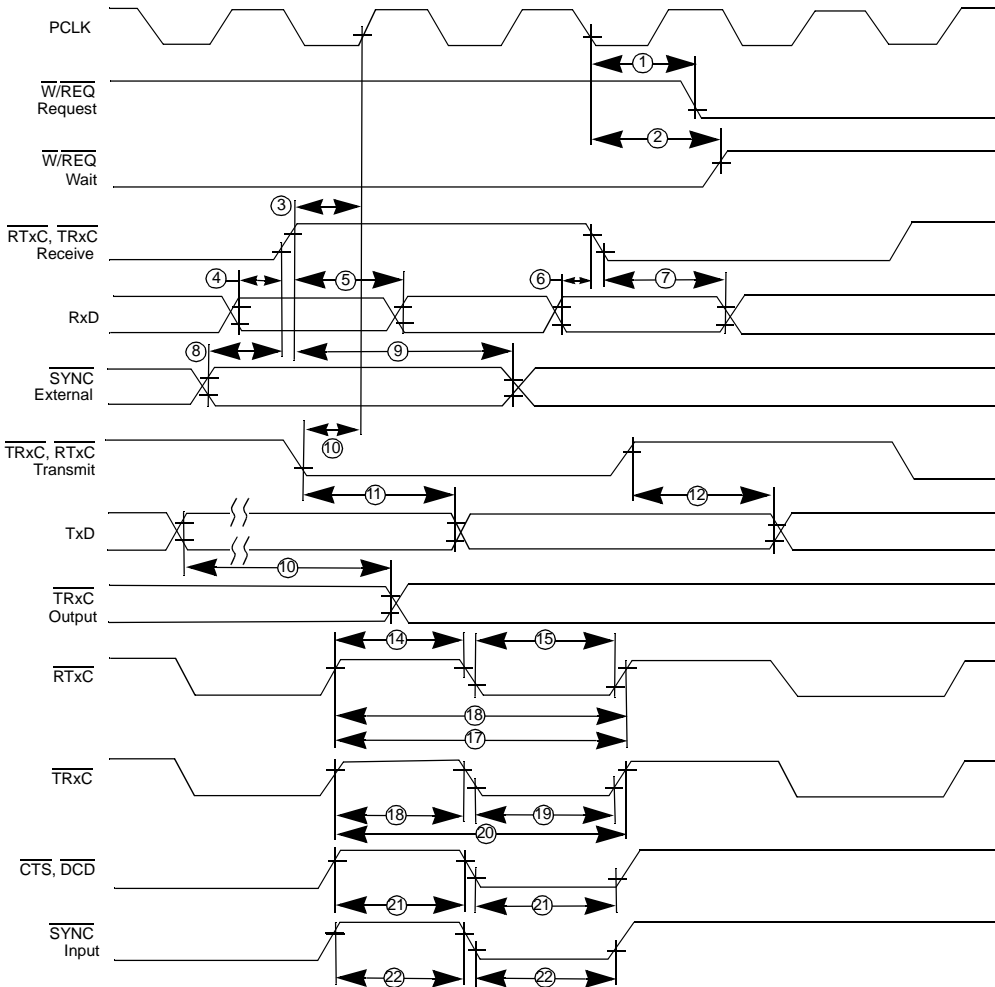


Figure 39. Z80C30 System Timing Diagram



Table 11. Z80C30 System Timing

| No | Symbol | Parameter | 8 MHz | | 10 MHz | |
|----|------------|---|-------|-----|--------|-----|
| | | | Min | Max | Min | Max |
| 1 | TdRXC(REQ) | $\overline{\text{Rx}}\overline{\text{C}}$ High to $\overline{\text{W}}/\overline{\text{REQ}}$ Valid ^{a,b} | 8 | 12 | 8 | 12 |
| 2 | TdRXC(W) | $\overline{\text{Rx}}\overline{\text{C}}$ High to Wait Inactive ^{1,2,c} | 8 | 14 | 8 | 14 |
| 3 | TdRdXC(SY) | $\overline{\text{Rx}}\overline{\text{C}}$ High to $\overline{\text{SY}}\overline{\text{NC}}$ Valid ^{1,2} | 4 | 7 | 4 | 7 |
| 4 | TdRXC(INT) | $\overline{\text{Rx}}\overline{\text{C}}$ High to INT Valid ^{1,2,3} | 8 | 12 | 8 | 12 |
| | | Note ^{d,2} | 2 | 3 | 2 | 3 |
| 5 | TdTXC(REQ) | $\overline{\text{Tx}}\overline{\text{C}}$ Low to $\overline{\text{W}}/\overline{\text{REQ}}$ Valid ^{e,2} | 5 | 8 | 5 | 8 |
| 6 | TdTXC(W) | $\overline{\text{Tx}}\overline{\text{C}}$ Low to Wait Inactive ^{1,2,3} | 5 | 11 | 5 | 11 |
| 7 | TdTXC(DRQ) | $\overline{\text{Tx}}\overline{\text{C}}$ Low to $\overline{\text{DTR}}/\overline{\text{REQ}}$ Valid ^{2,3} | 4 | 7 | 4 | 7 |
| 8 | TdTXC(INT) | $\overline{\text{Tx}}\overline{\text{C}}$ Low to $\overline{\text{INT}}$ Valid ^{1,2} | 4 | 6 | 4 | 6 |
| | | Note ^{2,4} | 2 | 3 | 2 | 3 |
| 9a | TdSY(INT) | SYNC to INT Valid ^{2,3} | 2 | 6 | 2 | 6 |
| 9b | TdSY(INT) | SYNC to INT Valid ^{2,3,4} | 2 | 3 | 2 | 3 |
| 10 | TdEXT(INT) | Note ^{2,3,4} | 2 | 3 | 2 | 3 |

- a. $\overline{\text{Rx}}\overline{\text{C}}$ is $\overline{\text{RTx}}\overline{\text{C}}$ or $\overline{\text{TRx}}\overline{\text{C}}$ whichever is supplying the receive clock.
- b. Units equal to TcPc.
- c. Open-drain output, measured with open-drain test load.
- d. Units equal to AS.
- e. $\overline{\text{Tx}}\overline{\text{C}}$ is $\overline{\text{TRx}}\overline{\text{C}}$ or $\overline{\text{RTx}}\overline{\text{C}}$, whichever is supplying the transmit clock.

Z80C30/Z85C30 CMOS SCC
Serial Communications Controller



84



Packaging and Ordering

Package Information

Figures 40 and 41 illustrate the 40-pin DIP package and 44-pin PLCC package diagrams.

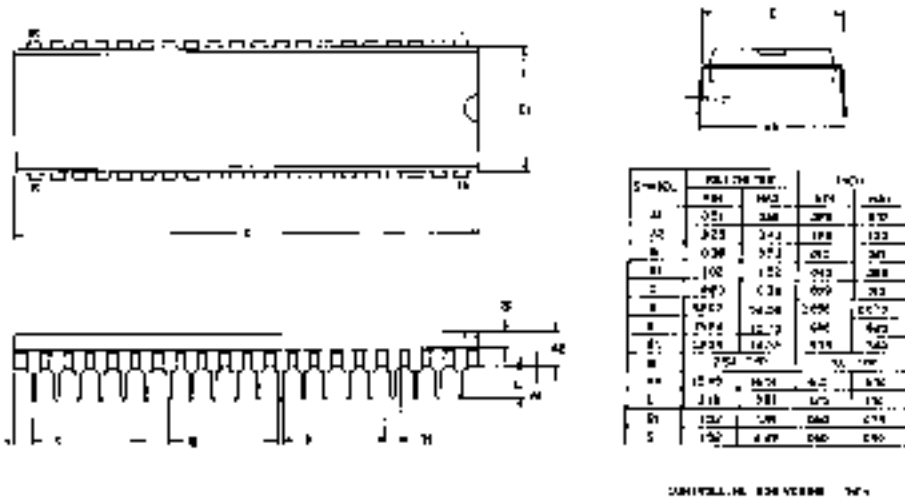


Figure 40. 40-Pin DIP Package Diagram

Z80C30/Z85C30 CMOS SCC
Serial Communications Controller

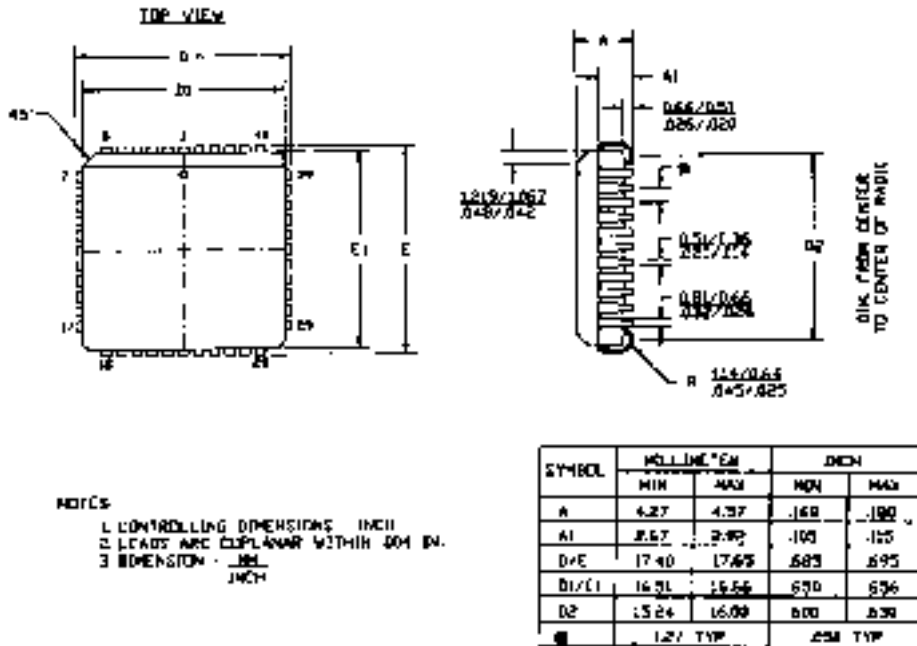


Figure 41. 44-Pin PLCC Package Diagram

Ordering Information

Z80C30/Z85C30

Table 12 provides ordering information for the Z80C30 and the Z85C30 devices.



Table 12. Z80C30/Z85C30 Ordering Information

| 8 MHz | 10 MHz | 16 MHz |
|-----------------|-----------------|---------------|
| Z80C3008PSC | Z80C3010PSC | Z85C3016PSC |
| Z80C3008VSC | Z80C3010VSC | Z85C3016VSC |
| Z85C3008PSC/PEC | Z85C3010PSC/PEC | |
| Z85C3008VSCNEC | Z85C3010VSCNEC | |

Package

P = Plastic DIP

V = Plastic Leaded Chip Carrier

D = Ceramic DIP

Temperature

E = -40°C to + 100°C

S = 0° to +70°C

Speeds

8 = 8 MHz

10 = 10 MHz

16 = 16 MHz

Environmental

C = Plastic Standard

Z80C30/Z85C30 CMOS SCC Serial Communications Controller



86

Example:

Example

