Serial Communications Controller

Z8530 SCC

Features

- Two independent, 0 to 1M bit/second, full-duplex channels, each with a separate crystal oscillator, baud rate generator, and Digital Phase-Locked Loop 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 mode operation.
- Local Loopback and Auto Echo modes.

General Description

The Z8530 SCC Serial Communications Controller is a dual-channel, multi-protocol data communications peripheral designed for use with conventional non-multiplexed buses. The SCC functions as a serial-to-parallel, parallel-to-serial converter/controller. The SCC can be software-configured to satisfy a wide variety of serial communications applications. The device contains a variety of new, sophisticated internal functions including on-chip baud rate generators, Digital Phase-Locked Loops, and crystal oscillators that dramatically reduce the need for external logic.

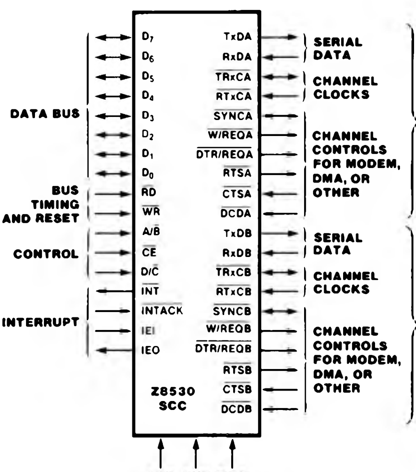


Figure 1. Logic Functions

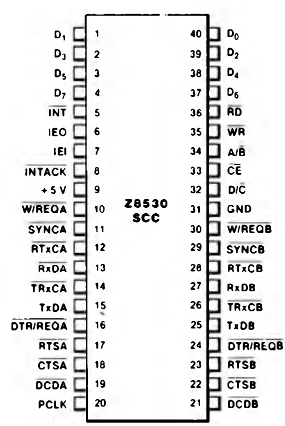


Figure 2. Pin Configuration

General Description (Continued)

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 can generate and check CRC codes in any Synchronous mode and can be programmed to check data integrity in various modes. The SCC also has 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 Z-BUS daisy-chain interrupt hierarchy is also supported — as is standard for SGS peripheral components.

The Z8530 SCC is packaged in a 40-pin ceramic DIP and uses a single +5 V power supply.

Pin Description

The following section describes the pin functions of the SCC. Figures 1 and 2 detail the respective pin functions and pin assignments.

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

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

CTSA, CTSB. *Clear To Send* (inputs, active Low). If these pins are programmed as Auto Enables, a Low on the inputs enables the respective transmitters. If not programmed as Auto Enables, 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.

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

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

D_0-D_7. *Data Bus* (bidirectional, 3-state). These lines carry data and commands to and from the SCC.

DTR/REQA, 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 an 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.

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

INTACK. *Interrupt Acknowledge* (input, active Low). This signal indicates an active Interrupt Acknowledge cycle. During this cycle, the

Pin Description (Continued)

SCC interrupt daisy chain settles. When $\overline{RD}$ becomes active, the SCC places an interrupt vector on the data bus (if IEI is High). $\overline{INTACK}$ 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.

$\overline{RD}$. *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.

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

$\overline{RTxCA}$, $\overline{RTxCB}$. *Receive/Transmit Clocks* (inputs, active Low). These pins can be programmed in several different modes of operation. 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.

$\overline{RTSA}$, $\overline{RTSB}$. *Request To Send* (outputs, active Low). When the Request To Send (RTS) bit in Write Register 5 (Figure 11) is set, the $\overline{RTS}$ 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 or in Asynchronous mode with Auto Enable off, the $\overline{RTS}$ pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.

$\overline{SYNCA}$, $\overline{SYNCB}$. *Synchronization* (inputs or outputs, active Low). These pins can act either 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 10) but have no other function.

In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, $\overline{SYNC}$ must be driven Low 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 $\overline{SYNC}$.

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. The synchronous condition is not latched, so 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.

$\overline{TRxCA}$, $\overline{TRxCB}$. *Transmit/Receive Clocks* (inputs or outputs, active Low). These pins can be programmed in several different modes of operation. $\overline{TRxC}$ 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.

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

$\overline{W/REQA}$, $\overline{W/REQB}$. *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.

Functional Description

The functional capabilities of the SCC can be described from two different points of view: as a data communications device, it transmits and receives data in a wide variety of data communications protocols; as a microprocessor peripheral, the SCC offers valuable features such as vectored interrupts, polling, and simple handshake capability.

Data Communications Capabilities. The SCC provides two independent full-duplex channels programmable for use in any common Asynchronous or Synchronous data-communication protocol. Figure 3 and the following description briefly detail these protocols.

Asynchronous Modes. Transmission and reception can be 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 (RxDA or RxDB in Figure 1). If the Low does not persist (as in the case of 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 or error conditions using dedicated routines. Furthermore, 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 can handle data at a rate of 1, 1/16, 1/32, or 1/64 of the clock rate supplied to the receive and transmit clock inputs. In Asynchronous modes, the $\overline{SYNC}$ pin may be programmed as an input used for functions such as monitoring a ring indicator.

Synchronous Modes. The SCC supports both byte-oriented and bit-oriented synchronous communication. Synchronous byte-oriented protocols can be handled in several modes,

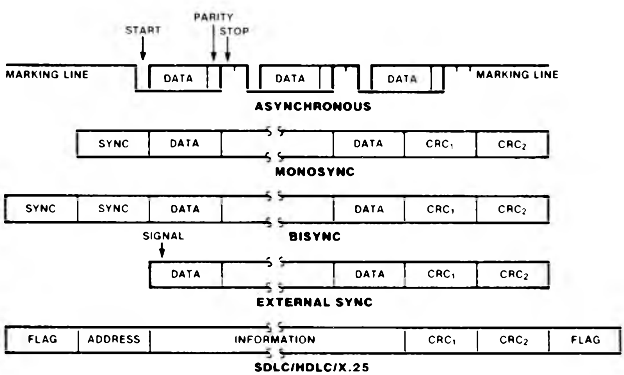


Figure 3. Some SCC Protocols

Functional Description (Continued)

allowing character synchronization with a 6-bit or 8-bit synchronous character (Monosync), any 12-bit synchronization pattern (Bisync), or with an external synchronous signal. Leading sync characters can be removed without interrupting the CPU.

Five- 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 shown in Figure 4.

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 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 1s or all 0s. The SCC also provides a feature that automatically transmits CRC data when no other data is available for transmission. This 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 synchronous characters, regardless of the programmed character length.

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 can be 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 so that an abort may be issued. 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 can be 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 0s 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 1s or all 0s. The CRC is inverted before transmission and the receiver checks against the bit pattern 0001110100001111.

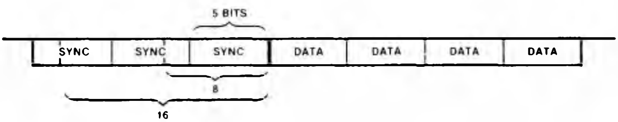


Figure 4. Detecting 5 - or 7 - Bit Synchronous Characters

Functional Description (Continued)

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

The SCC can be conveniently used under DMA control to provide high speed reception or transmission. In reception, for example, the SCC can interrupt the CPU when the first character of a message is received. The CPU then enables the DMA to transfer the message to memory. The SCC then issues an end-of-frame interrupt and the CPU can check the status of the received message. Thus, the CPU is freed for other service while the message is being received. The CPU may also enable the DMA first and have the SCC interrupt only on end-of-frame. This procedure allows all data to be transferred via the DMA.

SDLC Loop Mode. The SCC supports SDLC Loop mode in addition to normal SDLC. In an SDLC Loop, there is a primary controller station that 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 can act as a controller (Figure 5).

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop, and in fact must pass these messages to the rest of the loop by retransmitting them with a one-bit-time delay. The secondary station can place its own message on the loop only at specific times. The controller signals that secondary stations may 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.

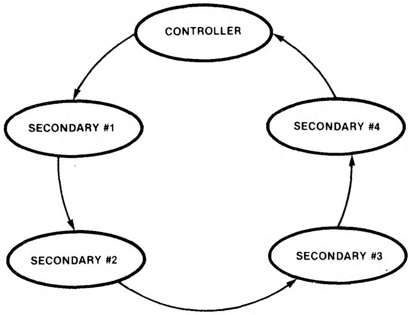


Figure 5. An SDLC Loop

When a secondary station has 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 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 can then append their messages to the message of the first secondary station by the same process. Any secondary stations without messages to send merely echo the incoming messages and are prohibited from placing messages on the loop (except upon recognizing an EOP).

SDLC Loop mode is a programmable option in the SCC. NRZ, NRZI, and FM coding may all be used in SDLC Loop mode.

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 startup, the flip-flop on the output 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 upon 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 $\overline{TRxC}$ pin, the

Functional Description (Continued)

output of the baud rate generator may be echoed out via the $\overline{\text{TRxC}}$ pin.

The following formula relates the time constant to the baud rate (the baud rate is in bits/second and the BR clock period is in seconds):

$$\text{baud rate} = \frac{1}{2\,(\text{time constant} + 2) \times (\text{BR clock period})}$$

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 may then be used as the SCC receive clock, the transmit clock, or both.

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 still 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 should 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 $\overline{\text{RTxC}}$ input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the SCC via the $\overline{\text{TRxC}}$ 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 ways (Figure 6). 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

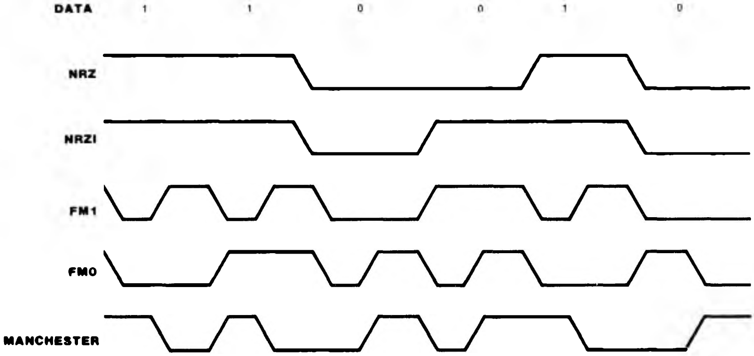


Figure 6. Data Encoding Methods

Functional Description (Continued)

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.

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. In Auto Echo mode, TxD is RxD. Auto Echo mode can be used with NRZI or FM encoding with no additional delay, because the data stream is not decoded before retransmission. In Auto Echo mode, the $\overline{CTS}$ 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 $\overline{WAIT}/\overline{REQUEST}$ on transmit.

The SCC is also capable of local loopback. In this mode TxD is RxD, just as in Auto Echo mode. However, in Local Loopback mode, the internal transmit data is tied to the internal receive data and RxD is ignored (except to be echoed out via TxD). The $\overline{CTS}$ and $\overline{DCD}$ inputs are also ignored as transmit and receive enables. However, transitions on these inputs can still cause interrupts. Local Loopback works in Asynchronous, Synchronous and SDLC modes with NRZ, NRZI or FM coding of the data stream.

I/O Interface Capabilities. The SCC offers the choice of Polling, Interrupt (vectored or nonvectored), and Block Transfer modes to transfer data, status, and control information to and from the CPU. The Block Transfer mode can be implemented under CPU or DMA control.

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 idea behind 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 needs to be 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. When an SCC responds to an Interrupt Acknowledge signal ($\overline{INTACK}$) from the CPU, an interrupt vector may be placed on the data bus. This vector is written in WR2 and may be read in RR2A or RR2B (Figures 10 and 11).

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 it 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 straightforward. 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 may be requested. The IE bits are write only.

The other two bits are related to the interrupt priority chain (Figure 7). As a microprocessor peripheral, the SCC may request an interrupt only when no higher priority device is requesting one, e.g., when IEI is High. If the device in question requests an interrupt, it pulls down $\overline{INT}$. The CPU then responds with $\overline{INTACK}$, and the interrupting device places the vector on the data bus.

In the SCC, the IP bit signals a need for interrupt servicing. When an IP bit is 1 and the IEI input is High, the $\overline{INT}$ 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 can never be set. The IP bits are readable in RR3A.

Functional Description (Continued)

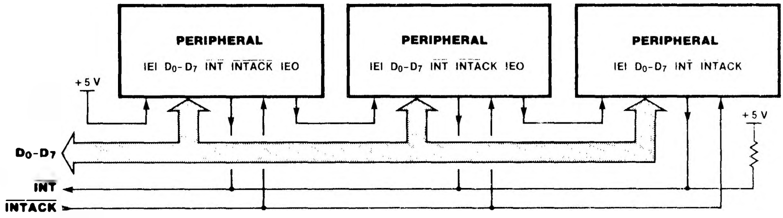


Figure 7. Interrupt Schedule

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 the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. (This implies that the transmitter must have had a data character written into it so that it can become empty.) When enabled, the receiver can interrupt the CPU in one of three ways:

- Interrupt on First Receive Character or Special Receive Condition.
- Interrupt on All Receive Characters or Special Receive Condition.
- Interrupt on Special Receive Condition 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 in the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt can occur from Special Receive Conditions any time after the first receive character interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the $\overline{CTS}$, $\overline{DCD}$, and $\overline{SYNC}$ pins; however, an External/Status interrupt is also caused by a Transmit Underrun condition, or a zero count in the baud rate generator, or 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 wishes of the primary station to regain control of the loop during a poll sequence.

CPU/DMA Block Transfer. The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers.

Functional Description (Continued)

The Block Transfer mode uses the $\overline{WAIT}$/$\overline{REQUEST}$ output in conjunction with the Wait/Request bits in WR1. The $\overline{WAIT}$/$\overline{REQUEST}$ output can be defined under software control as a $\overline{WAIT}$ line in the CPU Block Transfer mode or as a $\overline{REQUEST}$ line in the DMA Block Transfer mode.

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

Architecture

The SCC internal structure includes two full-duplex channels, two baud rate generators, internal control and interrupt logic, and a bus interface to a nonmultiplexed bus. Associated with each channel are a number of read and write registers for mode control and status information, as well as logic necessary to interface to modems or other external devices (Figure 8).

The logic for both channels provides formats, synchronization, and validation for data transferred to and from the channel interface. The modem control inputs are monitored by the control logic under program control. All of the modem control signals are general-purpose in nature and can optionally be used for functions other than modem control.

The register set for each channel includes ten control (write) registers, two sync-character (write) registers, and four status (read) registers. In addition, each baud rate generator has two (read/write) registers for holding the time constant that determines the baud rate. Finally, associated with the interrupt logic is a write register for the interrupt vector accessible through either channel, a

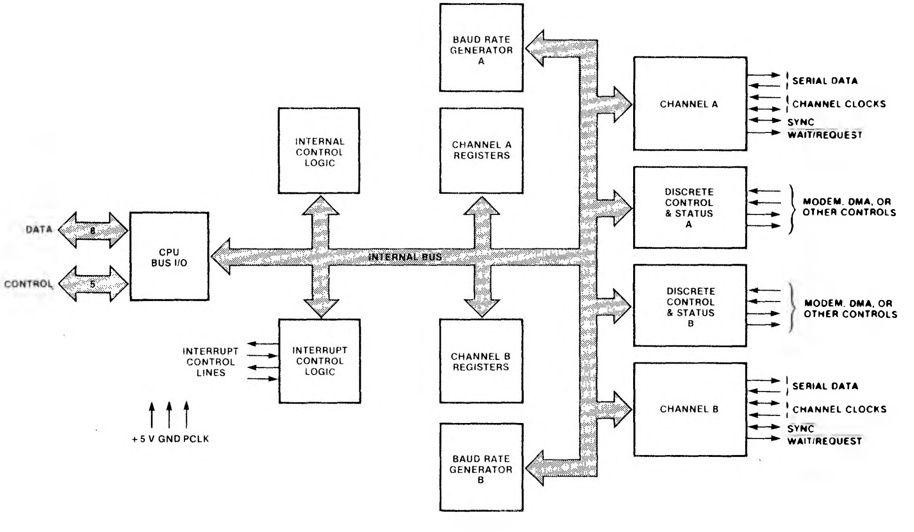


Figure 8. Block Diagram of SCC Architecture

Architecture (Continued)

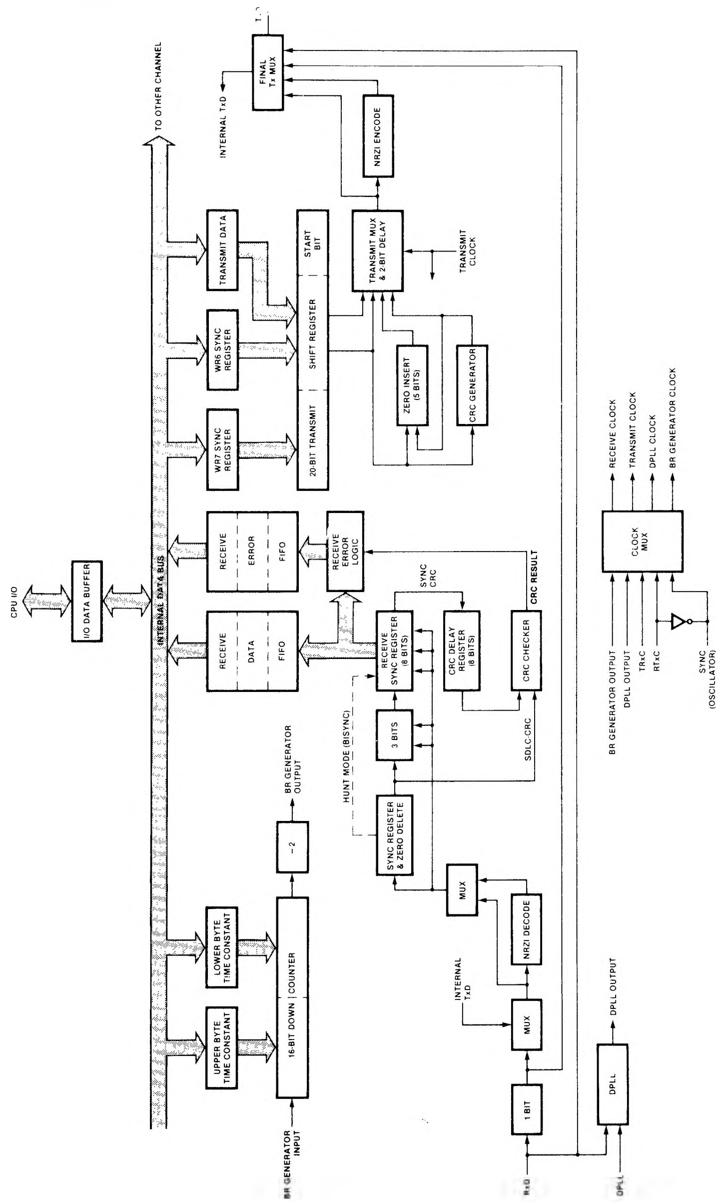


Figure 9. Data Path

Architecture (Continued)

write only Master Interrupt Control register and three read registers: one containing the vector with status infomation (Channel B only), one containing the vector without status (Channel A only), and one containing the Interrupt Pending bits (Channel A only).

The registers for each channel are designated as follows:

WR0-WR15 — Write Registers 0 through 15.

RR0-RR3, RR10, RR12, RR13, RR15 — Read Registers 0 through 3, 10, 12, 13, 15.

Table 1 lists the functions assigned to each read or write register. The SCC contains only one WR2 and WR9, but they can be accessed by either channel. All other registers are paired (one for each channel).

Data Path. The transmit and receive data path illustrated in Figure 9 is identical for both channels. The receiver has three 8-bit buffer registers in an FIFO arrangement, in addition to the 8-bit receive shift register. This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high speed data. Incoming data is routed through one of several paths (data or CRC) depending on the selected mode (the character length in Asynchronous modes also determines the data path).

The transmitter has an 8-bit Transmit Data buffer register loaded from the internal data bus and a 20-bit Transmit Shift register that can be loaded either from the synchronous character registers or from the Transmit Data register. Depending on the operational mode, outgoing data is routed through one of four main paths before it is transmitted from the Transmit Data output (TxD)

Read Register Functions	
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

Write Register Functions	
WR0	CRC initialize, initialization commands for the various modes, Register Pointers
WR1	Transmit/Receive interrupt and data transfer mode definition
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
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

Table 1. Read and Write Register Functions

Programming

The SCC contains 13 write registers in each channel that are programmed by the system separately to configure the functional personality of the channels.

In the SCC, register addressing is direct for the data registers only, which are selected by a High on the D/$\overline{C}$ pin. In all other cases (with the exception of 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

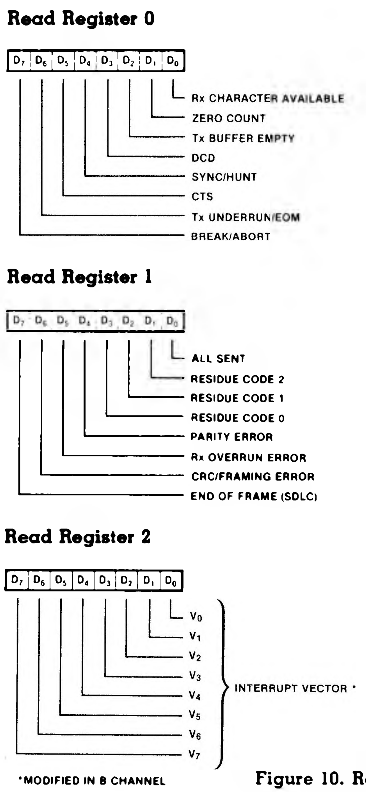
Programming (Continued)

register, and if the second operation is read, the selected read register is accessed. All of the registers in the SCC, 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.

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, the Asynchronous mode, character length, clock rate, number of stop bits, even or odd parity might be set first. Then the interrupt mode would be set, and finally, receiver or transmitter enable.

Read Registers. The SCC contains eight read registers (actually nine, 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) may be 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). Figure 10 shows the formats for each read register.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring; e.g., when the interrupt vector indicates a Special Receive Condition interrupt, all the appropriate error bits can be read from a single register (RR1).



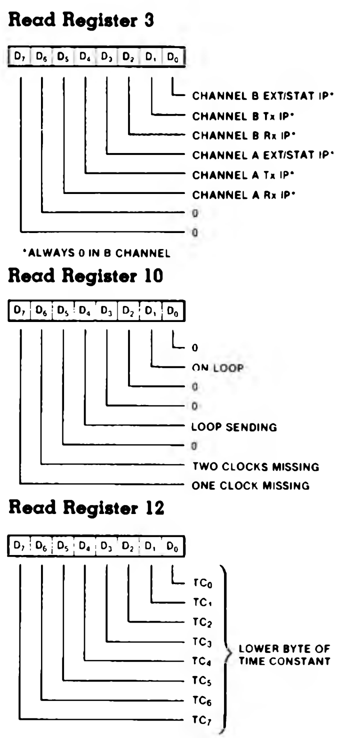
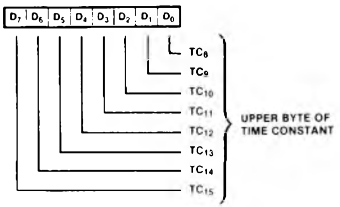


Figure 10. Read Register Bit Functions

Programming (Continued)

Read Register 13



Read Register 15

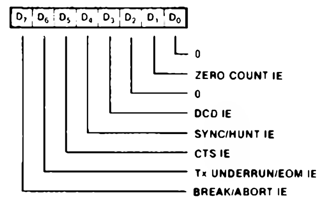
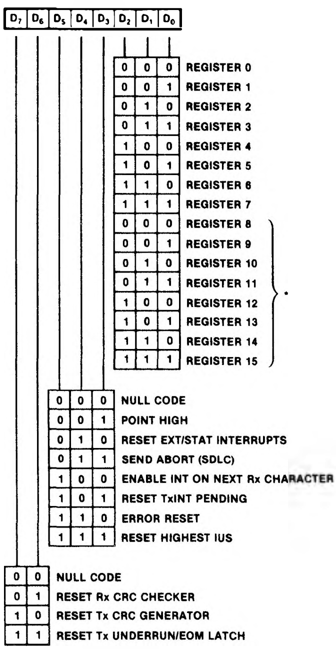


Figure 10. Read Register Bit Functions (Continued)

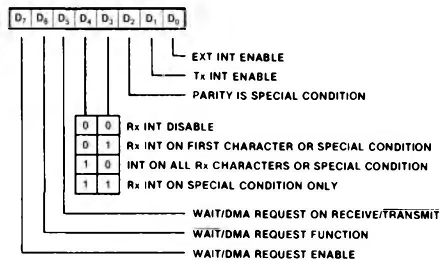
Write Registers. The SCC contains 13 write registers (14 counting WR8, the transmit buffer) in each channel. These write registers are programmed separately to configure the functional "personality" of the channels. In addition, there are two registers (WR2 and WR9) shared by the two channels that may be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. Figure 11 shows the format of each write register.

Write Register 0



*WITH POINT HIGH COMMAND

Write Register 1



Write Register 2

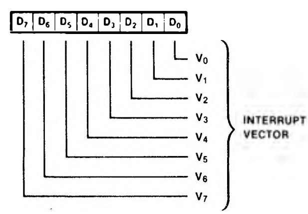
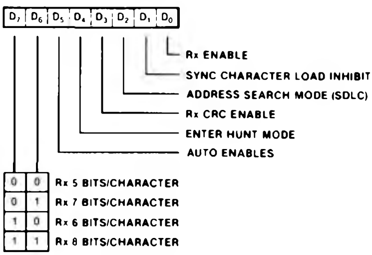


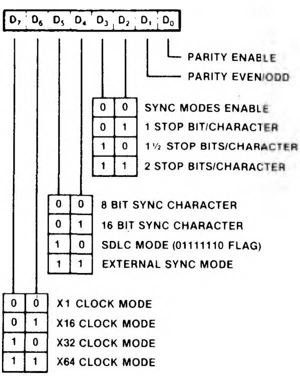
Figure 11. Write Register Bit Functions

Programming (Continued)

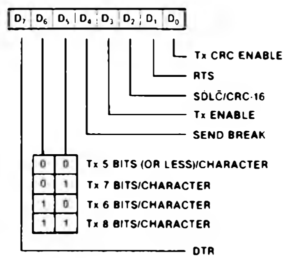
Write Register 3



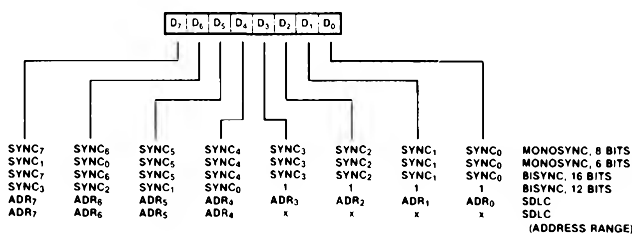
Write Register 4



Write Register 5



Write Register 6



Write Register 7

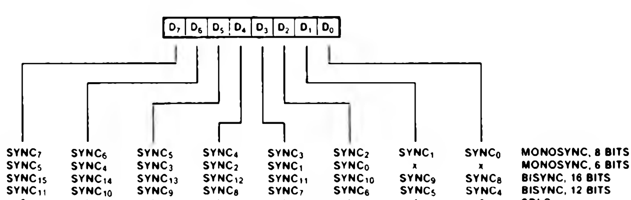
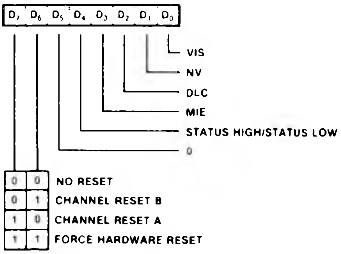


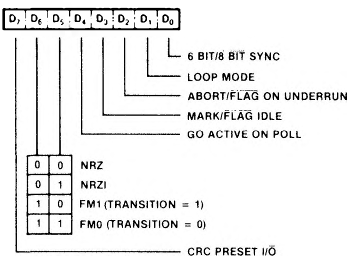
Figure 11. Write Register Bit Functions (Continued)

Programming (Continued)

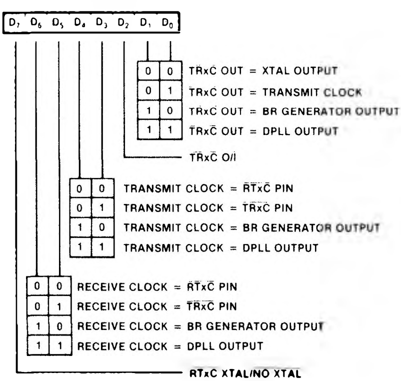
Write Register 9



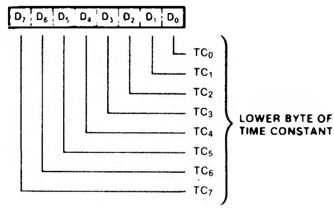
Write Register 10



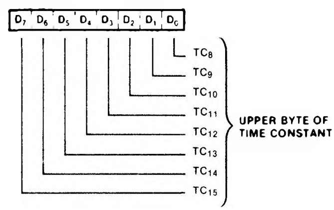
Write Register 11



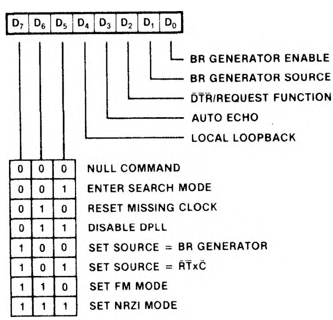
Write Register 12



Write Register 13



Write Register 14



Write Register 15

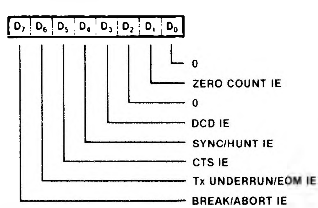


Figure 11. Write Register Bit Functions (Continued)

Timing

The SCC generates internal control signals from $\overline{WR}$ and $\overline{RD}$ that are related to PCLK. Since PCLK has no phase relationship with $\overline{WR}$ and $\overline{RD}$, 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 rising 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 6 PCLK cycles plus 200 ns.

Read Cycle Timing. Figure 12 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 it rises before $\overline{RD}$ rises, the effective $\overline{RD}$ is shortened.

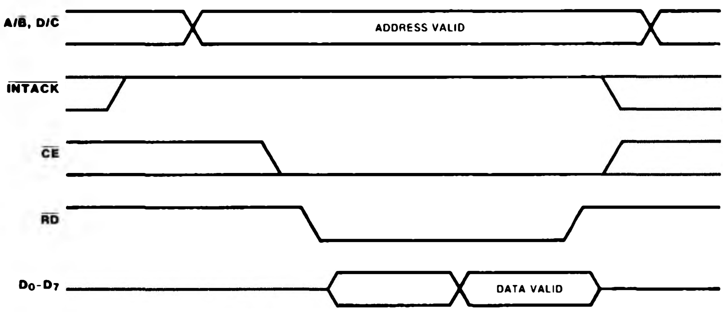


Figure 12. Read Cycle Timing

Timing (Continued)

Write Cycle Timing. Figure 13 illustrates Write cycle timing. Addresses on $A/\overline{B}$ and $D/\overline{C}$ and the status on $\overline{\text{INTACK}}$ must remain stable throughout the cycle. If $\overline{\text{CE}}$ falls after $\overline{\text{WR}}$ falls or if it rises before $\overline{\text{WR}}$ rises, the effective $\overline{\text{WR}}$ is shortened.

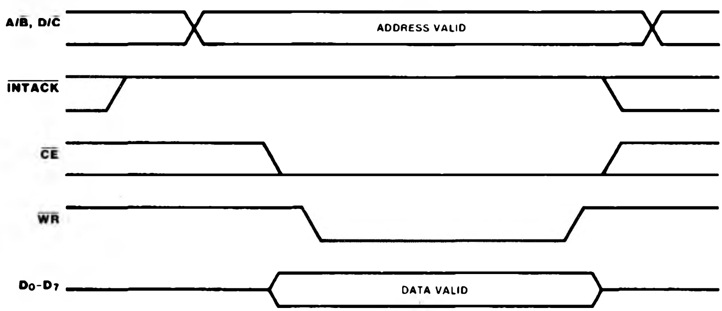


Figure 13. Write Cycle Timing

Interrupt Acknowledge Cycle Timing. Figure 14 illustrates Interrupt Acknowledge cycle timing. Between the time $\overline{\text{INTACK}}$ goes Low and the falling edge of $\overline{\text{RD}}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the SCC and IEI is High when $\overline{\text{RD}}$ falls, the Acknowledge cycle is intended for the SCC. In this case, the SCC may be programmed to respond to $\overline{\text{RD}}$ Low by placing its interrupt vector on D_0-D_7 and it then sets the appropriate Interrupt-Under-Service latch internally.

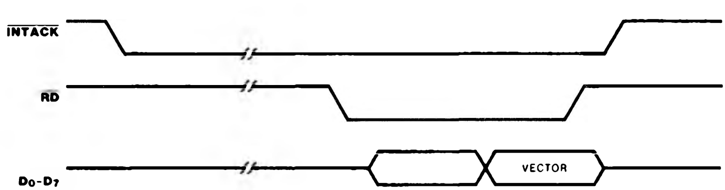


Figure 14. Interrupt Acknowledge Cycle Timing

Absolute Maximum Ratings

Voltages on all inputs and outputs with respect to GND . . . . -0.3 V to +7.0 V

Operating Ambient Temperature . . . . As Specified in 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 characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

- $+4.75\ V \leq V_{CC} \leq +5.25\ V$
- GND = 0 V
- T_A as specified in Ordering Information

All ac parameters assume a load capacitance of 50 pF max.

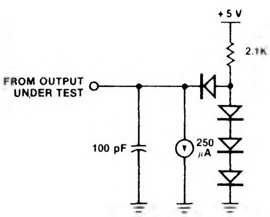


Figure 15. Standard Test Load

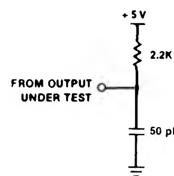


Figure 16. Open-Drain Test Load

DC Characteristics

Symbol	Parameter	Min	Max	Unit	Condition
V_{IH}	Input High Voltage	2.0	$V_{CC}+0.3$	V	
V_{IL}	Input Low Voltage	-0.3	0.8	V	
V_{OH}	Output High Voltage	2.4		V	$I_{OH} = -250\ \mu A$
V_{OL}	Output Low Voltage		0.4	V	$I_{OL} = +2.0$ mA
I_{IL}	Input Leakage		±10.0	μA	$0.4 \leq V_{IN} \leq +2.4V$
I_{OL}	Output Leakage		±10.0	μA	$0.4 \leq V_{OUT} \leq +2.4V$
I_{CC}	V_{CC} Supply Current		250	mA	

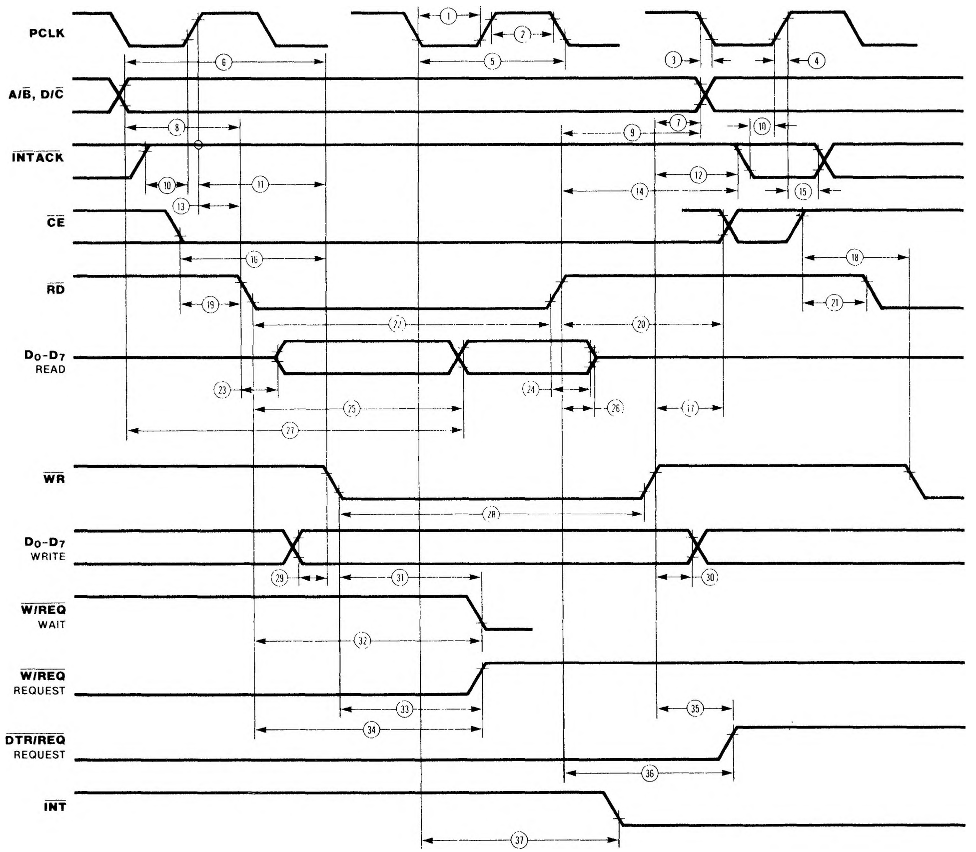
V_{CC} = 5 V ± 5% unless otherwise specified, over specified temperature range.

Capacitance

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

f = 1 MHz, over specified temperature range.

Read and Write Timing

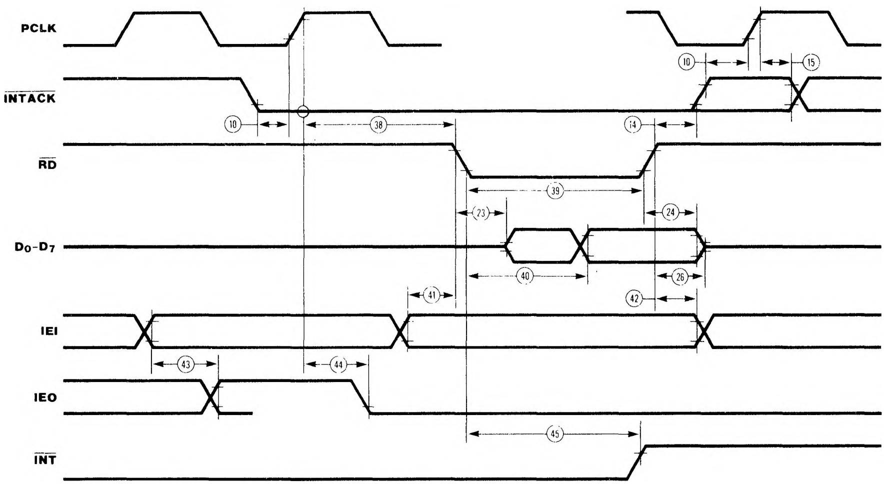


Read and Write Timing (Continued)

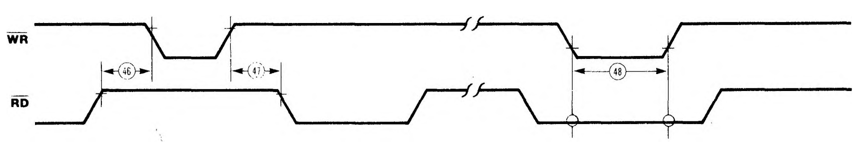
Number	Symbol	Parameters	Min(ns)	Max(ns)	Notes*
1	TwPCl	PCLK Low Width	105	2000	
2	TwPCh	PCLK High Width	105	2000	
3	TfPC	PCLK Fall Time		20	
4	TrPC	PCLK Rise Time		20	
5	TcPC	PCLK Cycle Time	250	4000	
6	TsA(WR)	Address to $\overline{\text{WR}}$ ↓ Setup Time	80		
7	ThA(WR)	Address to $\overline{\text{WR}}$ ↑ Hold Time	0		
8	TsA(RD)	Address to $\overline{\text{RD}}$ ↓ Setup Time	80		
9	ThA(RD)	Address to $\overline{\text{RD}}$ ↑ Hold Time	0		
10	TsIA(PC)	$\overline{\text{INTACK}}$ to PCLK ↑ Setup Time	0		
11	TsIAi(WR)	$\overline{\text{INTACK}}$ to $\overline{\text{WR}}$ ↓ Setup Time	200		1
12	ThIA(WR)	$\overline{\text{INTACK}}$ to $\overline{\text{WR}}$ ↑ Hold Time	0		
13	TsIAi(RD)	$\overline{\text{INTACK}}$ to $\overline{\text{RD}}$ ↓ Setup Time	200		1
14	ThIA(RD)	$\overline{\text{INTACK}}$ to $\overline{\text{RD}}$ ↑ Hold Time	0		
15	ThIA(PC)	$\overline{\text{INTACK}}$ to PCLK ↑ Hold Time	100		
16	TsCEl(WR)	$\overline{\text{CE}}$ Low to $\overline{\text{WR}}$ ↓ Setup Time	0		
17	ThCE(WR)	$\overline{\text{CE}}$ to $\overline{\text{WR}}$ ↑ Hold Time	0		
18	TsCEh(WR)	$\overline{\text{CE}}$ High to $\overline{\text{WR}}$ ↓ Setup Time	100		
19	TsCEl(RD)	$\overline{\text{CE}}$ Low to $\overline{\text{RD}}$ ↓ Setup Time	0		1
20	ThCE(RD)	$\overline{\text{CE}}$ to $\overline{\text{RD}}$ ↑ Hold Time	0		1
21	TsCEh(RD)	$\overline{\text{CE}}$ High to $\overline{\text{RD}}$ ↓ Setup Time	100		1
22	TwRDl	$\overline{\text{RD}}$ Low Width	390		1
23	TdRD(DRA)	$\overline{\text{RD}}$ ↓ to Read Data Active Delay	0		
24	TdRDr(DR)	$\overline{\text{RD}}$ ↑ to Read Data Not Valid Delay	0		
25	TdRDf(DR)	$\overline{\text{RD}}$ ↓ to Read Data Valid Delay		255	
26	TdRD(DRz)	$\overline{\text{RD}}$ ↑ to Read Data Float Delay		70	2

NOTES:
1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time required for a ± 0.5 V change in the output with a maximum dc load and minimum ac load.

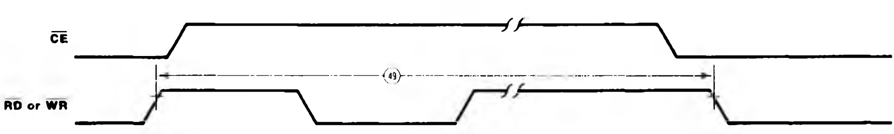
Interrupt Acknowledge Timing



Reset Timing



Cycle Timing



Number	Symbol	Parameters	Min(ns)	Max(ns)	Notes*
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		590	
28	TwWRl	$\overline{WR}$ Low Width	390		
29	TsDW(WR)	Write Data to $\overline{WR}$ ↓ Setup Time	0		
30	ThDW(WR)	Write Data to $\overline{WR}$ ↑ Hold Time	0		
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay		240	4
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay		240	4
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $W/\overline{REQ}$ Not Valid Delay		240	
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $W/\overline{REQ}$ Not Valid Delay		240	
35	TdWRr(REQ)	$\overline{WR}$ ↑ to $\overline{DTR}/\overline{REQ}$ Not Valid Delay		5TcPC + 300	
36	TdRDr(REQ)	$\overline{RD}$ ↑ to $\overline{DTR}/\overline{REQ}$ Not Valid Delay		5TcPC + 300	
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay		500	4
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay			5
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	285		
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		190	
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	120		
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		
43	TdIEI(IEO)	IEI to IEO Delay Time		120	
44	TdPC(IEO)	PCLK ↑ to IEO Delay		250	
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay		500	4
46	TdRD(WRQ)	$\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset	30		
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	30		
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident Low for Reset	250		
49	Trc	Valid Access Recovery Time	6TcPC + 200		3

NOTES:

3. Parameter applies only between transactions involving the SCC.
4. Open-drain output, measured with open-drain test load.
5. 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 TdIEIf(IEO) for each device separating them in the daisy chain.

*Timings are preliminary and subject to change.

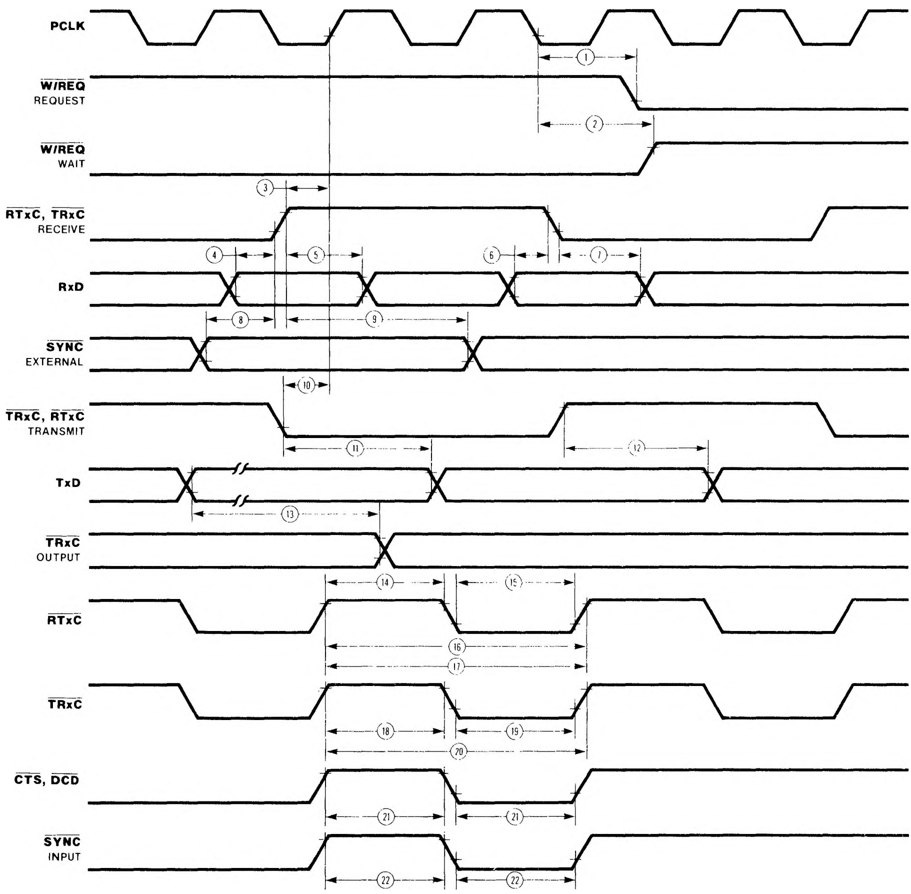
General Timing

Number	Symbol	Parameters	Min(ns)	Max(ns)	Notes*
1	TdPC(REQ)	PCLK ↓ to $\overline{\text{W/REQ}}$ Valid Delay		250	
2	TdPC(W)	PCLK ↓ to Wait Inactive Delay		350	
3	TsRXC(PC)	$\overline{\text{RxC}}$ ↑ to PCLK ↑ Setup Time	50		1,4
4	TsRXD(RXCr)	RxD to $\overline{\text{RxC}}$ ↑ Setup Time (X1 Mode)	0		1
5	ThRXD(RXCr)	RxD to $\overline{\text{RxC}}$ ↑ Hold Time (X1 Mode)	150		1
6	TsRXD(RXCf)	RxD to $\overline{\text{RxC}}$ ↓ Setup Time (X1 Mode)	0		1,5
7	ThRXD(RXCf)	RxD to $\overline{\text{RxC}}$ ↓ Hold Time (X1 Mode)	150		1,5
8	TsSY(RXC)	$\overline{\text{SYNC}}$ to $\overline{\text{RxC}}$ ↑ Setup Time	-200		1
9	ThSY(RXC)	$\overline{\text{SYNC}}$ to $\overline{\text{RXC}}$↑ Hold Time	3TcPC + 200		1
10	TsTXC(PC)	$\overline{\text{TxC}}$ ↓ to PCLK ↑ Setup Time	0		2,4
11	TdTXCf(TXD)	$\overline{\text{TxC}}$ ↓ to TxD Delay (X1 Mode)		300	2
12	TdTXCr(TXD)	$\overline{\text{TxC}}$ ↑ to TxD Delay (X1 Mode)		300	2,5
13	TdTXD(TRX)	TxD to $\overline{\text{TRxC}}$ Delay (Send Clock Echo)			
14	TwRTXh	$\overline{\text{RTxC}}$ High Width	180		
15	TwRTXl	$\overline{\text{RTxC}}$ Low Width	180		
16	TcRTX	$\overline{\text{RTxC}}$ Cycle Time	400		
17	TcRTXX	Crystal Oscillator Period	250	1000	3
18	TwTRXh	$\overline{\text{TRxC}}$ High Width	180		
19	TwTRXl	$\overline{\text{TRxC}}$ Low Width	180		
20	TcTRX	$\overline{\text{TRxC}}$ Cycle Time	400		
21	TwEXT	$\overline{\text{DCD}}$ or $\overline{\text{CTS}}$ Pulse Width	200		
22	TwSY	$\overline{\text{SYNC}}$ Pulse Width	200		

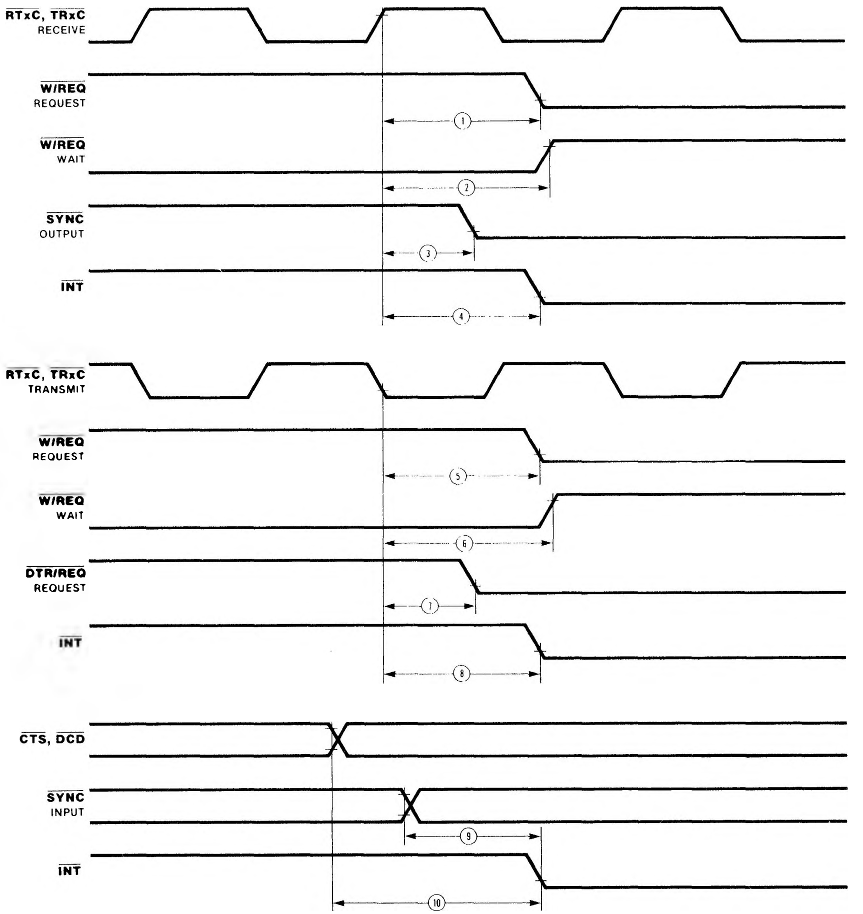
NOTES:
1. $\overline{\text{RxC}}$ is $\overline{\text{RTxC}}$ or $\overline{\text{TRxC}}$, whichever is supplying the receive clock.
2. $\overline{\text{TxC}}$ is $\overline{\text{TRxC}}$ or $\overline{\text{RTxC}}$, whichever is supplying the transmit clock.
3. Both $\overline{\text{RTxC}}$ and $\overline{\text{SYNC}}$ have 30 pF capacitors to ground connected to them.
4. Parameter applies *only* if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between $\overline{\text{RxC}}$ and PCLK or $\overline{\text{TxC}}$ and PCLK is required.
5. Parameter applies only to FM encoding/decoding.

*Timings are preliminary and subject to change.

General Timing (Continued)



System Timing



System Timing (Continued)

Number	Symbol	Parameter	Min	Max	Units	Notes*
1	TdRXC (REQ)	$\overline{\text{RxC}}$ ↑ to $\overline{\text{W/REQ}}$ Valid Delay	8	12	TcPC	2
2	TdRXC(W)	$\overline{\text{RxC}}$ ↑ to Wait Inactive Delay	8	12	TcPC	1,2
3	TdRXC(SY)	$\overline{\text{RxC}}$ ↑ to $\overline{\text{SYNC}}$ Valid Delay	4	7	TcPC	2
4	TdRXC(INT)	$\overline{\text{RxC}}$ ↑ to $\overline{\text{INT}}$ Valid Delay	10	16	TcPC	1,2
5	TdTXC(REQ)	$\overline{\text{TxC}}$ ↓ to $\overline{\text{W/REQ}}$ Valid Delay	5	8	TcPC	3
6	TdTXC(W)	$\overline{\text{TxC}}$ ↓ to Wait Inactive Delay	5	8	TcPC	1,3
7	TdTXC(DRQ)	$\overline{\text{TxC}}$ ↓ to $\overline{\text{DTR/REQ}}$ Valid Delay	4	7	TcPC	3
8	TdTXC(INT)	$\overline{\text{TxC}}$ ↓ to $\overline{\text{INT}}$ Valid Delay	6	10	TcPC	1,3
9	TdSY(INT)	$\overline{\text{SYNC}}$ Transition to $\overline{\text{INT}}$ Valid Delay	2	6	TcPC	1
10	TdEXT(INT)	$\overline{\text{DCD}}$ or $\overline{\text{CTS}}$ Transition to $\overline{\text{INT}}$ Valid Delay	2	6	TcPC	1

NOTES:
1. Open-drain output, measured with open-drain test load.
2. $\overline{\text{RxC}}$ is $\overline{\text{RTxC}}$ or $\overline{\text{TRxC}}$, whichever is supplying the receive clock.
3. $\overline{\text{TxC}}$ is $\overline{\text{TRxC}}$ or $\overline{\text{RTxC}}$, whichever is supplying the transmit clock.

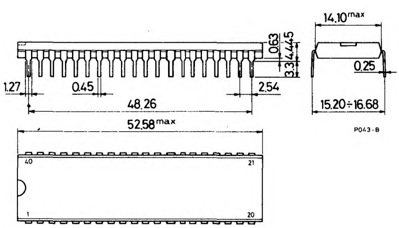
*Timings are preliminary and subject to change.

Ordering Information

Type		Package	Temp	Clock	Description
Z8530	B1	Plastic 40 pin	0/+70°C	4MHz	Z8530 Serial Communications Controller
	B6	Plastic 40 pin	-40/+85°C		
	D1	Ceramic 40 pin	0/+70°C		
	D2	Ceramic 40 pin	-5S/+125°C		
	D6	Ceramic 40 pin	-40/+85°C		
Z8530A	B1	Plastic 40 pin	0/+70°C	6MHz	
	B6	Plastic 40 pin	-40/+85°C		
	D1	Ceramic 40 pin	0/+70°C		
	D6	Ceramic 40 pin	-40/+85°C		

Packages

Plastic



Ceramic