82530/82530-6
SERIAL COMMUNICATIONS CONTROLLER (SCC)

- **Two Independent Full Duplex Serial Channels**
- **On Chip Crystal Oscillator, Baud-Rate Generator and Digital Phase Locked Loop for Each Channel**
- **Programmable for NRZ, NRZI or FM Data Encoding/Decoding**
- **Diagnostic Local Loopback and Automatic Echo for Fault Detection and Isolation**
- **System Clock Rates:**
  - **—4 MHz for 82530**
  - **—6 MHz for 82530-6**
- **Max Bit Rate (6 MHz)**
  - **—Externally Clocked: 1.5 Mbps**
    Self-Clocked:
    375 Kbps FM CODING
    187 Kbps NRZI CODING
    93 Kbps Asynchronous
- **Interfaces with Any INTEL CPU, DMA or I/O Processor**
- **Available in 40 Pin DIP and 44 Lead PLCC**
- **Available in Express Version**
- **Asynchronous Modes**
  - **—5–8 bit Character; Odd, Even or No Parity; 1, 1.5 or 2 Stop Bits**
  - **—Independent Transmit and Receive Clocks. 1X, 16X, 32X or 64X Programmable Sampling Rate**
  - **—Error Detection: Framing, Overrun and Parity**
  - **—Break Detection and Generation**
- **Bit Synchronous Modes**
  - **—SDLC Loop/Non-Loop Operation**
  - **—CRC-16 or CCITT Generation Detection**
  - **—Abort Generation and Detection**
  - **—I-field Residue Handling**
  - **—CCITT X.25 Compatible**
- **Byte Synchronous Modes**
  - **—Internal or External Character Synchronization (1 or 2 Characters)**
  - **—Automatic CRC Generation and Checking (CRC 16 or CCITT)**
  - **—IBM Bisync Compatible**

The INTEL 82530 Serial Communications Controller (SCC) is a dual-channel, multi-protocol data communications peripheral. It is designed to interface high speed communications lines using Asynchronous, Byte synchronous and Bit synchronous protocols to INTEL's microprocessors based systems. It can be interfaced with Intel's MCS51/96, iAPX86/88/186 and 188 in polled, interrupt driven or DMA driven modes of operation.

The SCC is a 40-pin device manufactured using INTEL's high-performance HMOS* II technology.

* HMOS is a patented process of Intel Corporation.

October 1987
Order Number: 230834-003

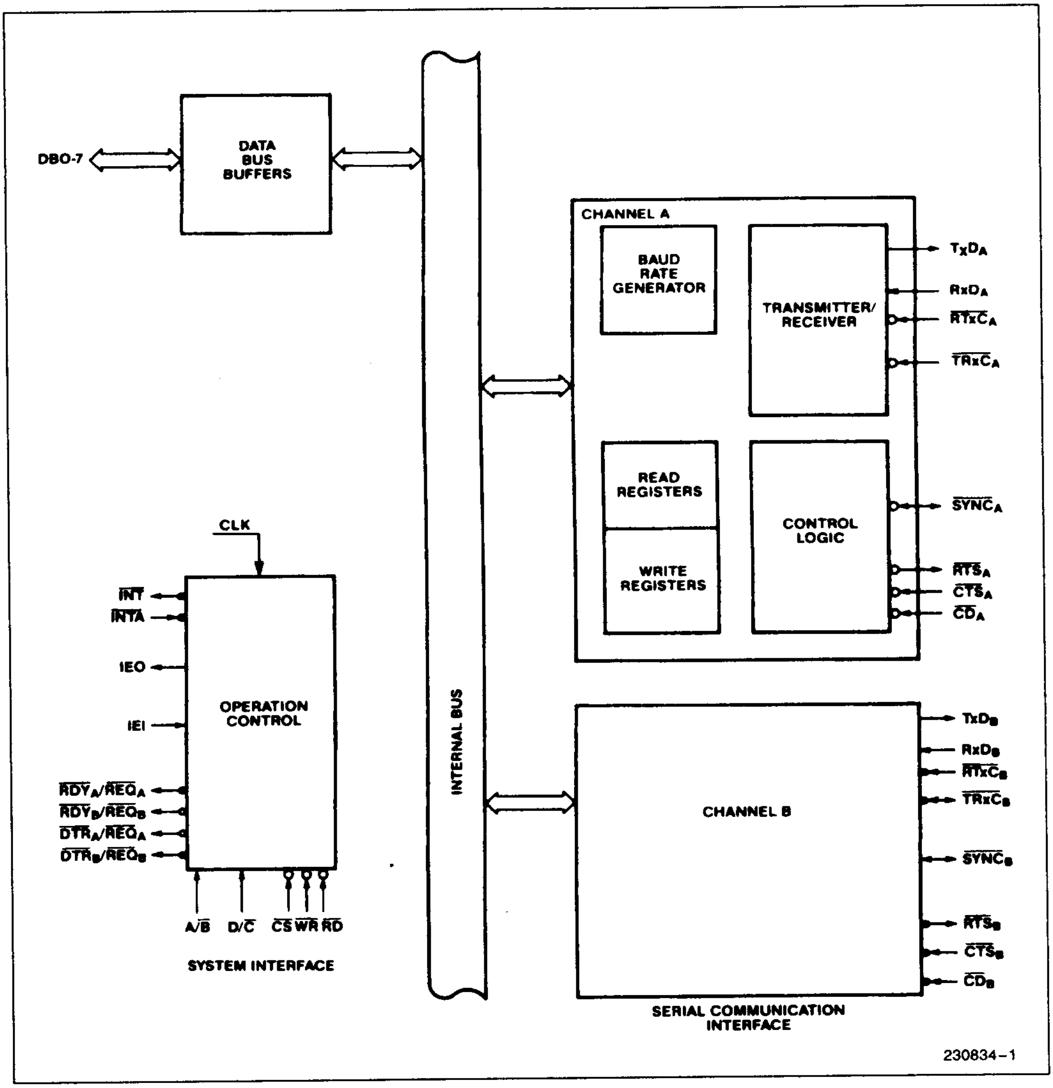


Figure 1. 82530 Internal Block Diagram

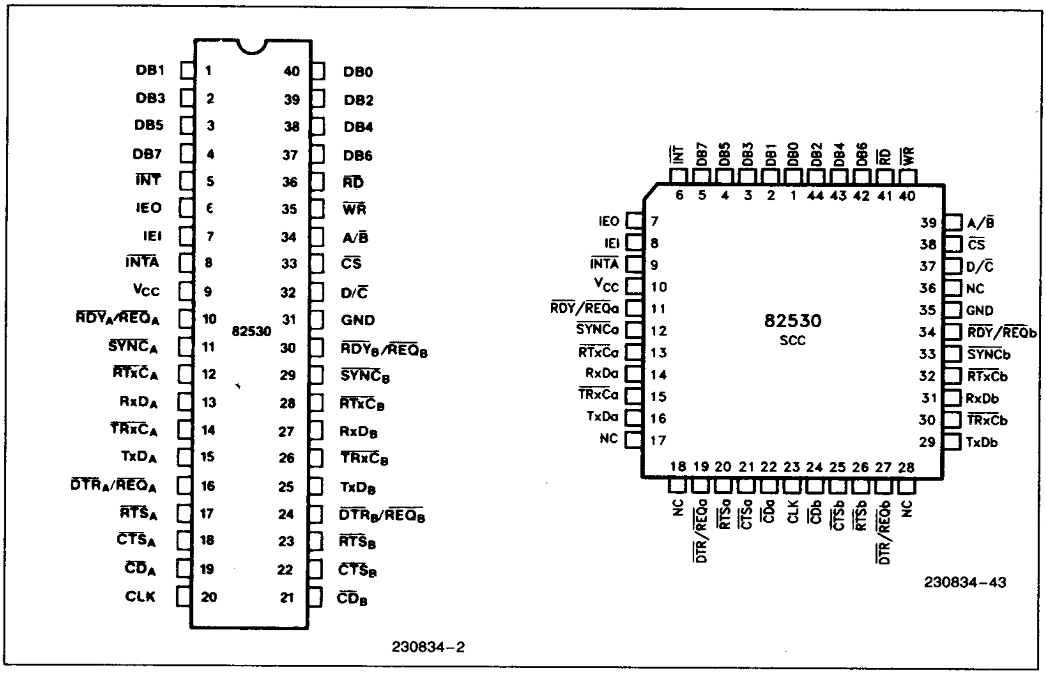


Figure 2. Pin Configurations

The following section describes the pin functions of the SCC. Figure 2 details the pin assignments.

Table 1. Pin Description

Symbol	Pin No.		Type	Name and Function
	DIP	PLCC		
DB$_0$ DB$_1$ DB$_2$ DB$_3$ DB$_4$ DB$_5$ DB$_6$ DB$_7$	40 1 39 2 38 3 37 4	1 2 44 3 43 4 42 5	I/O I/O I/O I/O I/O I/O I/O I/O	**DATA BUS:** The Data Bus lines are bi-directional three-state lines which interface with the system's Data Bus. These lines carry data and commands to and from the SCC.
$\overline{\text{INT}}$	5	6	O	**INTERRUPT REQUEST:** The interrupt signal is activated when the SCC requests an interrupt. It is an open drain output.
IEO	6	7	O	**INTERRUPT ENABLE OUT:** 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.
IEI	7	8	I	**INTERRUPT ENABLE IN:** 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.

Table 1. Pin Description (Continued)

Symbol	Pin No.		Type	Name and Function
	DIP	PLCC		
$\overline{INTA}$	8	9	I	**INTERRUPT ACKNOWLEDGE:** This signal indicates an active Interrupt Acknowledge cycle. During this cycle, the SCC interrupt daisy chain settles. When $\overline{RD}$ becomes active, the SCC places an interrupt vector on the data bus (if IEI is High). $\overline{INTA}$ is latched by the rising edge of CLK.
V_{CC}	9	10		**POWER:** +5V Power supply.
$\overline{RDY}_A/\overline{REQ}_A$ $\overline{RDY}_B/\overline{REQ}_B$	10 30	11 34	O O	**READY/REQUEST:** (output, open-drain when programmed for a Ready 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 Ready lines to synchronize the CPU to the SCC data rate. The reset state is Ready.
$\overline{SYNC}_A$ $\overline{SYNC}_B$	11 29	12 33	I/O I/O	**SYNCHRONIZATION:** 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 $\overline{CD}$. In this mode, transitions on these lines affect the state of the Synchronous/Hunt status bits in Read Register 0 (Figure 9) 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 characters boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.
$\overline{RTxC}_A$ $\overline{RTxC}_B$	12 28	13 32	I I	**RECEIVE/TRANSMIT CLOCKS:** 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 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.
RxD_A RxD_B	13 27	14 31	I I	**RECEIVE DATA:** These lines receive serial data at standard TTL levels.
$\overline{TRxC}_A$ $\overline{TRxC}_B$	14 26	15 30	I/O I/O	**TRANSMIT/RECEIVE CLOCKS:** 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.
TxD_A TxD_B	15 25	16 29	O O	**TRANSMIT DATA:** These output signals transmit serial data at standard TTL levels
$\overline{DTR}_A/\overline{REQ}_A$ $\overline{DTR}_B/\overline{REQ}_B$	16 24	19 27	O O	**DATA TERMINAL READY/REQUEST:** 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.

Table 1. Pin Description (Continued)

Symbol	Pin No.		Type	Name and Function
	DIP	PLCC		
$\overline{\text{RTS}}_A$ $\overline{\text{RTS}}_B$	17 23	20 26	O O	**REQUEST TO SEND:** When the Request to Send (RTS) bit in Write Register 5 is set (Figure 10), the $\overline{\text{RTS}}$ 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 $\overline{\text{RTS}}$ pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.
$\overline{\text{CTS}}_A$ $\overline{\text{CTS}}_B$	18 22	21 25	I I	**CLEAR TO SEND:** 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.
$\overline{\text{CD}}_A$ $\overline{\text{CD}}_B$	19 21	22 24	I I	**CARRIER DETECT:** 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 accommodate slow rise time signals. The SCC detects pulses on these pins and can interrupt the CPU on both logic level transitions.
CLK	20	23	I	**CLOCK:** This is the system SCC clock used to synchronize internal signals.
GND	31	35		**GROUND**
D/$\overline{\text{C}}$	32	37	I	**DATA/COMMAND SELECT:** This signal defines the type of information transferred to or from the SCC. A High means data is transferred; a Low indicates a command.
$\overline{\text{CS}}$	33	38	I	**CHIP SELECT:** This signal selects the SCC for a read or write operation.
A/$\overline{\text{B}}$	34	39	I	**CHANNEL A/CHANNEL B SELECT:** This signal selects the channel in which the read or write operation occurs.
$\overline{\text{WR}}$	35	40	I	**WRITE:** When the SCC is selected this signal indicates a write operation. The coincidence of $\overline{\text{RD}}$ and $\overline{\text{WR}}$ is interpreted as a reset.
$\overline{\text{RD}}$	36	41	I	**READ:** 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.

GENERAL DESCRIPTION

The Intel 82350 Serial Communications Controller (SCC) is a dual-channel, multi-protocol data communications peripheral. The SCC functions as a serial-to-parallel, parallel-to-serial convertor/controller. The SCC can be software-configured to satisfy a wide range of serial communications applications. The device contains sophisticated internal functions including on-chip baud rate generators, digital phase locked loops, various data encoding and decoding schemes, and crystal oscillators that reduce the need for external logic.

In addition, diagnostic capabilities—automatic echo and local loopback—allow the user to detect and isolate a failure in the network. They greatly improve the reliability and fault isolation of the system.

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 (Terminal, Personal Computer, Peripherals, Industrial Controller, Telecommunication sytem, etc.).

The 82530 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 control in both channels. In applications where these controls are not needed, the modem control can be used for general purpose I/O.

The Intel 82530 is designed to support Intel's MCS51/96, iAPX 86/88 and iAPX 186/188 families.

ARCHITECTURE

The 82530 internal structure includes two full-duplex channels, two baud rate generators, internal control and interrupt logic, and a bus interface to a non-multiplexed CPU 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 modems or other external devices.

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 synchronous 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 write-only Master Interrupt Control register and three read registers; one containing the vector with status information (Channel B only), one containing the vector without status (A only), and one containing the Interrupt Pending bits (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 2 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 3 is identical for both channels. The receiver has three 8-bit buffer registers in a 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 sync-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).

Table 2. Read and Write Register Functions

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 (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, shift right/shift left command
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

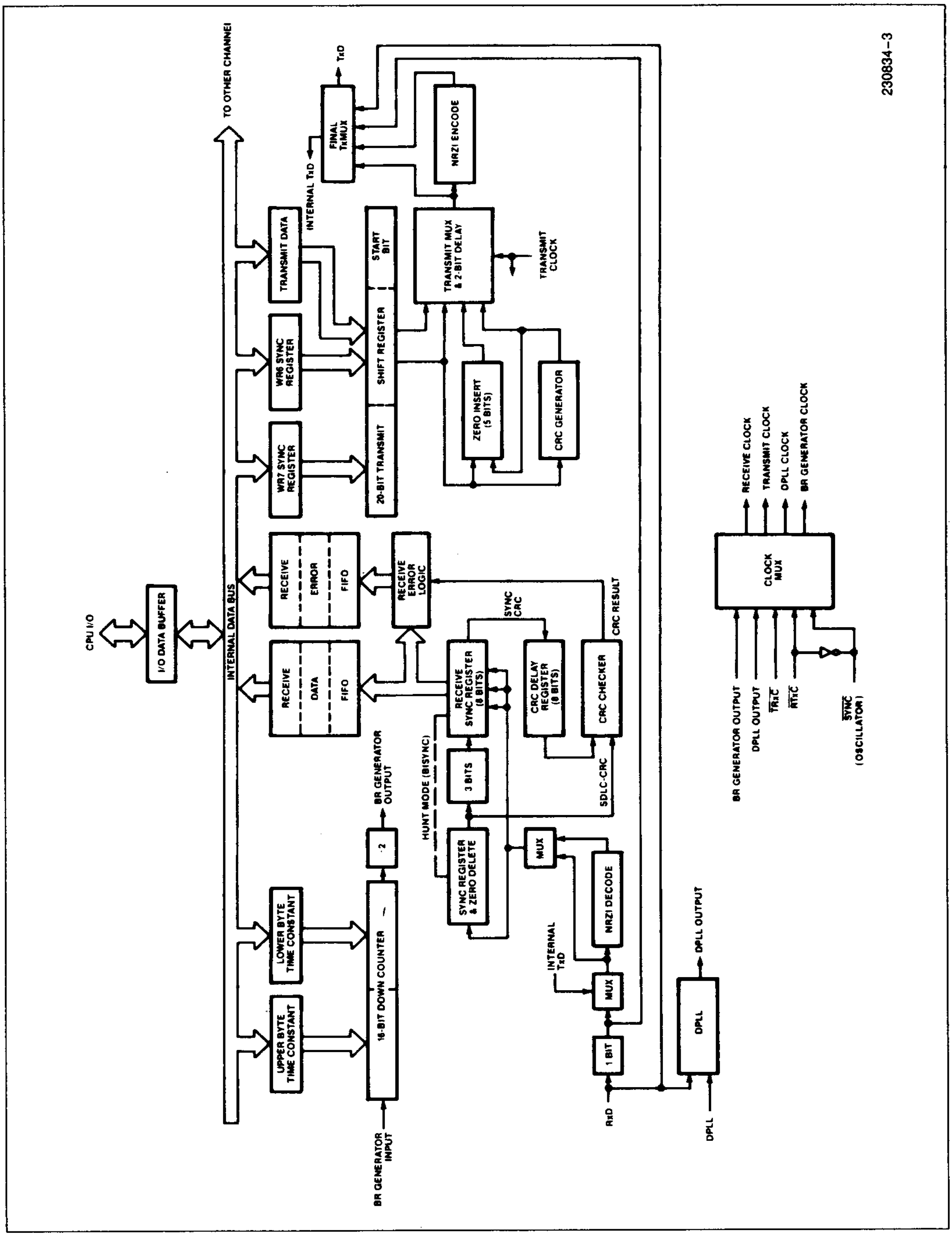


Figure 3. Data Path

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, it interacts with the CPU and provides vectored interrupts and handshaking signals.

DATA COMMUNICATIONS CAPABILITIES

The SCC provides two independent full-duplex channels programmable for use in any common asynchronous or synchronous data-communications protocol. Figure 4 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 transmitter 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). 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 of 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 the asynchronous modes, a data rate equal to the clock rate, 1x mode, requires external synchronization. In asynchronous modes, the $\overline{SYNC}$ pin may be programmed as an input used for functions such as monitoring a ring indicator.

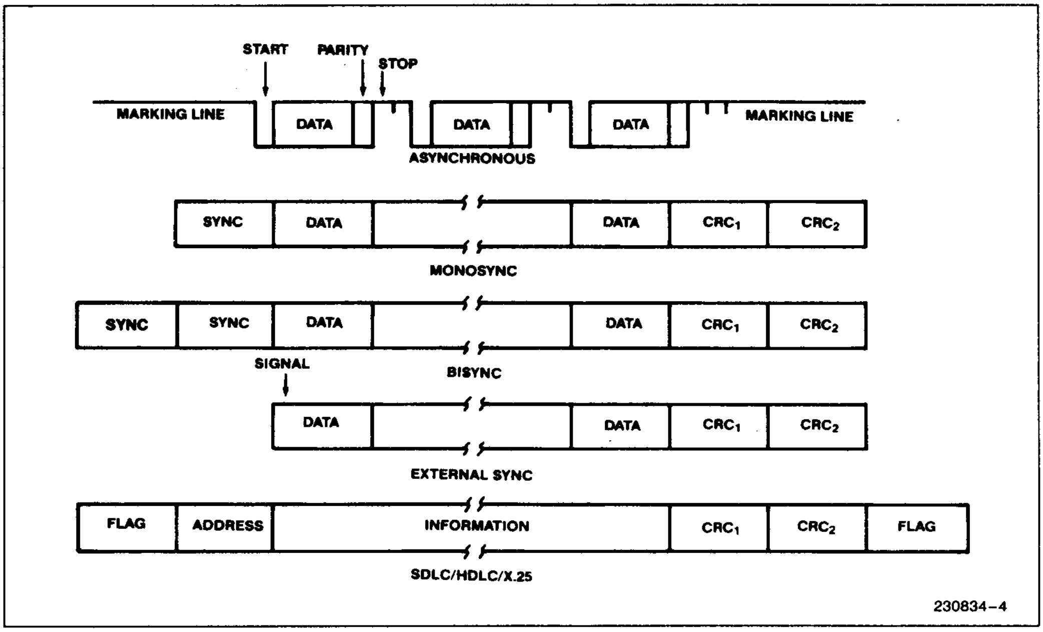


Figure 4. SCC Protocols

Synchronous Modes

The SCC supports both byte-oriented and bit-oriented synchronous communication. Synchronous-byte-oriented protocols can be handled in several modes allowing character synchronization with a 6-bit or 8-bit synchronous character (Monosync), any 12-bit or 16-bit synchronous pattern (Bisync), or with an external synchronous signal. Leading synchronous characters can be 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 shown in Figure 5.

CRC checking for Synchronous byte-oriented mode 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 mode 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.

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.

2

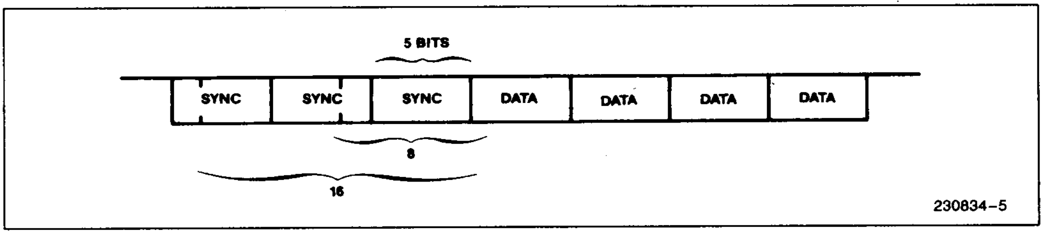


Figure 5. Detecting 5- or 7-Bit Synchronous Characters

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 DMA.

SDLC LOOP MODE

The SCC supports SDLC Loop mode in addition to normal SDLC. In a loop topology, there is a primary controller station that manages the message traffic flow and any number of secondary stations. In 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 6).

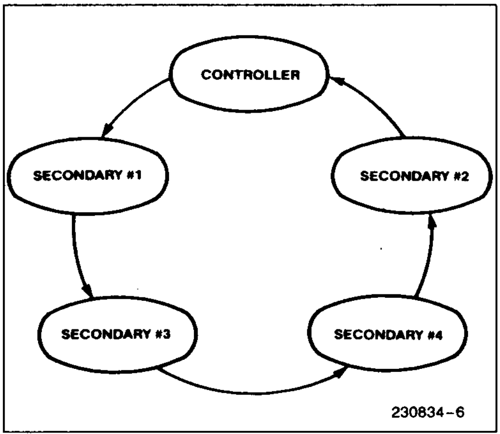


Figure 6. An SDLC Loop

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.

When a secondary station has a message to transmit and recognizes an EOP on the line, it changes the last binary one of the EOP to a zero 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 GENERATORS

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 zero, 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 $\overline{\text{TRxC}}$ pin, the 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, the BR clock frequency is in Hz, and clock mode is 1, 16, 32, or 64.)

$$\begin{array}{c}\text{time}\\\text{constant}\end{array} = \frac{\text{BR clock frequency}}{2 \times \text{baud rate} \times \text{clock mode}} - 2$$

Table 3. Time Constant Values for Standard Baud Rates at BR Clock = 3.9936 MHz

Rate (BAUD)	Time Constant (decimal notation)	Error
19200	102	—
9600	206	—
7200	275	0.12%
4800	414	—
3600	553	0.06%
2400	830	—
2000	996	0.04%
1800	1107	0.03%
1200	1662	—
600	3326	—
300	6654	—
150	13310	—
134.5	14844	0.0007%
110	18151	0.0015%
75	26622	—
50	39934	—

DIGITAL PHASE LOCKED LOOP

The SCC contains a digital phase locked-loop (DPLL) to recover clock information from a datastream 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 datastream, 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 coding, the DPLL counts the 32X clock to create nominal bit times. As the 32X clock is counted, the DPLL is searching the incoming datastream for edges (either 1/0 or 0/1). 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 1 to 31, but with a cycle corresponding to two bit times. When the DPLL is locked, the clock edges in the datastream 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 $^{15}/_{16}$ 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 7). In NRZ encoding, a 1 is represented by a High level and a 0 is represented by a Low level. In NRZI encoding, as 1 is represented by no change in level and a 0 is represented by a change in level. In FM_1 (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 FM_0 (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 0/1 the bit is a 0. If the transition is 1/0 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 datastream 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 $\overline{\text{READY/REQUEST}}$ 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). $\overline{\text{CTS}}$ and $\overline{\text{CD}}$ 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.

SERIAL BIT RATE

To run the 82530 (4 MHz/6 MHz) at 1/1.5 Mbps the receive and transmit clocks must be externally generated and synchronized to the system clock. If the serial clocks ($\overline{\text{RTxC}}$ and $\overline{\text{TRxC}}$) and the system clock (CLK) are asynchronous, the maximum bit rate is 880 Kbps/1.3 Mbps. For self-clocked operation, i.e using the on chip DPLL, the maximum bit rate is 125/187 Kbps if NRZI coding is used and 250/375 Kbps if FM coding is used.

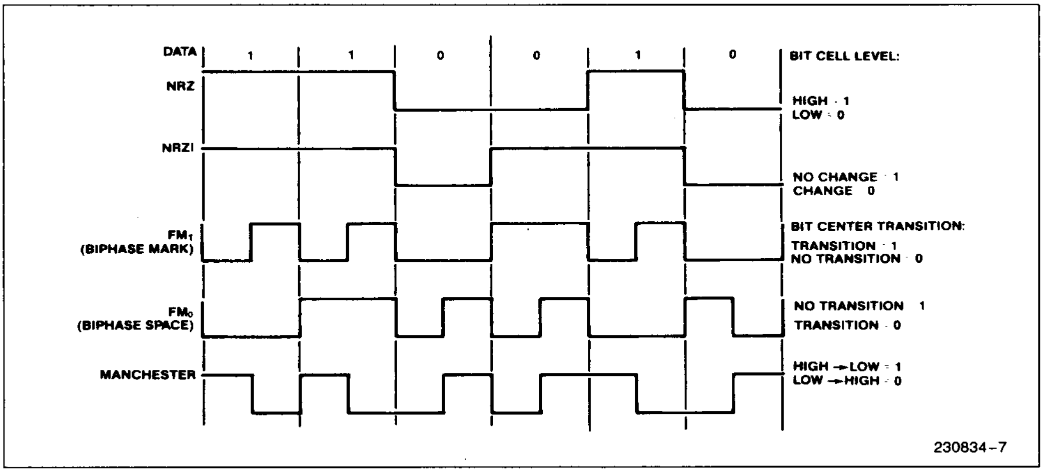


Figure 7. Data Encoding Methods

Table 4. Maximum Bit Rates

Mode	System Clock	System Clock/ Serial Clock	Serial Bit Rate	Conditions
Serial clocks generated externally	4 MHz	4	1 Mbps	Serial clocks synchronized with system clock. Refer to parameter #3 and 10 in general timings.
	6 MHz	4	1.5 Mbps	Serial clocks synchronized with system clock. Refer to parameter #3 and #10 in general timings.
	4 MHz	4.5	880 Kbps	Serial clocks and system clock asynchronous.
	6 MHz	4.5	1.3 Mbps	Serial clocks and system clock asynchronous
Self-clocked operation				
NRZI	4 MHz	32	125 Kbps	
	6 MHz	32	187 Kbps	
FM	4 MHz	16	250 Kbps	
	6 MHz	16	375 Kbps	
ASYNC	4 MHz	16	62.5 Kbps	
	6 MHz	16	93.75 Kbps	

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 a SCC responds to an Interrupt Acknowledge signal (INTA) 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 9 and 10).

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 bits 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 8). As a 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 INT. The CPU then responds with INTA, 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 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.

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 interrupts. 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

2

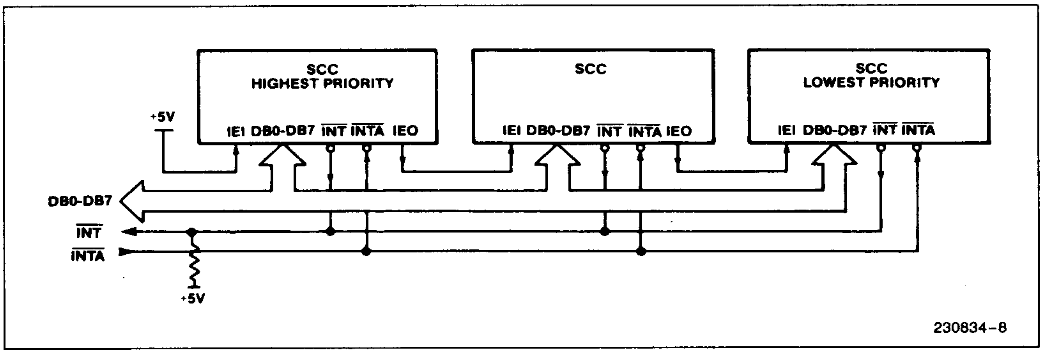


Figure 8. Daisy Chaining SCC's

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 $\overline{\mathrm{CTS}}$, $\overline{\mathrm{CD}}$, and $\overline{\mathrm{SYNC}}$ 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. The Block Transfer mode uses the $\overline{\mathrm{READY}}/\overline{\mathrm{REQUEST}}$ output in conjunction with the $\overline{\mathrm{READY}}/\overline{\mathrm{REQUEST}}$ bits in WR1. The $\overline{\mathrm{READY}}/\overline{\mathrm{REQUEST}}$ output can be defined under software control as a $\overline{\mathrm{READY}}$ line in the CPU Block Transfer mode (WR1; D6 = 0) or as a request line in the DMA Block Transfer mode (WR1; D6 = 1). To a DMA controller, the SCC $\overline{\mathrm{REQUEST}}$ output indicates that the SCC is ready to transfer data to or from memory. To the CPU, The $\overline{\mathrm{READY}}$ line indicates that the SCC is not ready to transfer data, thereby requesting that the CPU extend the I/O cycle. The $\overline{\mathrm{DTR}/\mathrm{REQUEST}}$ line allows full-duplex operation under DMA control.

PROGRAMMING

Each channel has fifteen Write registers that are individually programmed from the system bus to configure the functional personality of each channel. Each channel also has eight Read registers from which the system can read Status, Baud rate, or Interrupt information.

Only the four data registers (Read, Write for channels A and B) are directly selected by a High on the D/$\overline{\mathrm{C}}$ input and the appropriate levels on the $\overline{\mathrm{RD}}$, $\overline{\mathrm{WR}}$ and A/$\overline{\mathrm{B}}$ pins. All other registers are addressed indirectly by the content of Write Register 0 in conjunction with a Low on the D/$\overline{\mathrm{C}}$ input and the appropriate levels on the $\overline{\mathrm{RD}}$, $\overline{\mathrm{WR}}$ and A/$\overline{\mathrm{B}}$ pins. If bit 3 in WW0 is 1 and bits 4 and 5 are 0 then bits 0, 1, 2 address the higher registers 8 through 15. If bits 3, 4, 5 contain a different code, bits 0, 1, 2 address the lower registers 0 through 7 as shown on Table 5.

Writing to or reading from any register except RR0, WR0 and the Data Registers thus involves two operations.

Table 5. Register Addressing

D/C	"Point High" Code in WR0	D2 in WR0	D1 in WR0	D0	Write Register	Read Register
High	Either Way	X	X	X	Data	Data
Low	Not True	0	0	0	0	0
Low	Not True	0	0	1	1	1
Low	Not True	0	1	0	2	2
Low	Not True	0	1	1	3	3
Low	Not True	1	0	0	4	(0)
Low	Not True	1	0	1	5	(1)
Low	Not True	1	1	0	6	(2)
Low	Not True	1	1	1	7	(3)
Low	True	0	0	0	Data	Data
Low	True	0	0	1	9	—
Low	True	0	1	0	10	10
Low	True	0	1	1	11	(15)
Low	True	1	0	0	12	12
Low	True	1	0	1	13	13
Low	True	1	1	0	14	(10)
Low	True	1	1	1	15	15

First write the appropriate code into WR0, then follow this by a write or read operation on the register thus specified. Bits 0 through 4 in WW0 are automatically cleared after this operation, so that WW0 then points to WR0 or RR0 again.

Channel A/Channel B selection is made by the A/$\overline{B}$ input (High = A, Low = B)

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 determine 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 9 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).

WRITE REGISTERS

The SCC contains 15 write registers (16 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 10 shows the format of each write register.

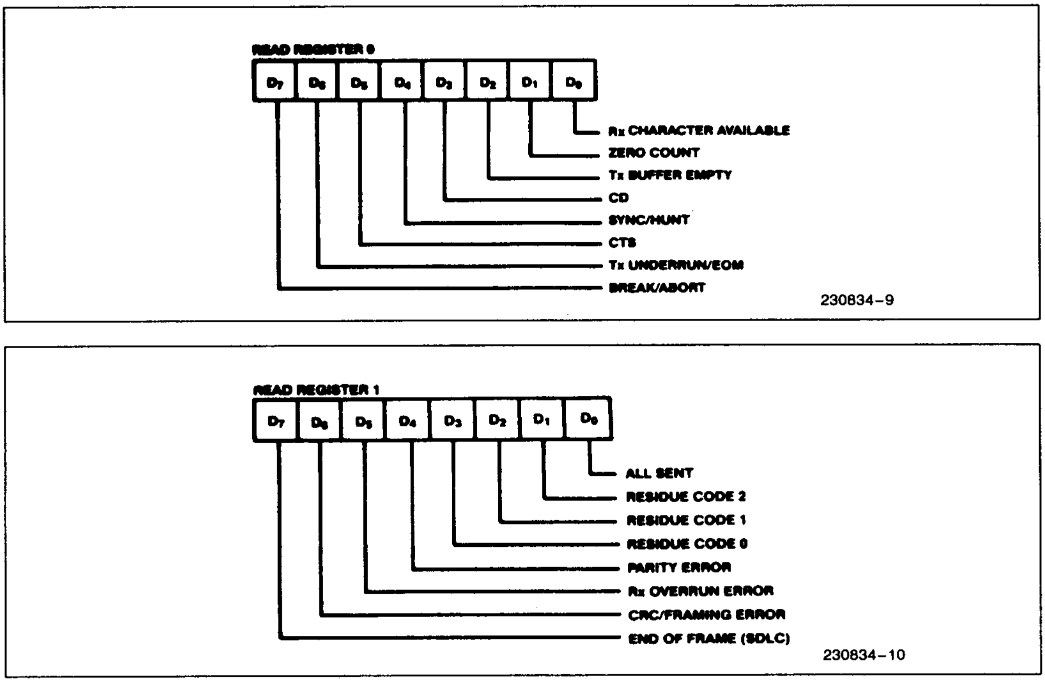
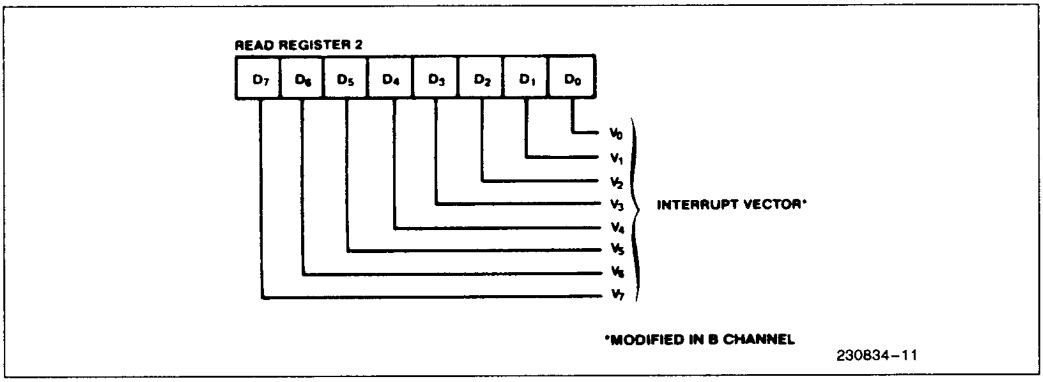
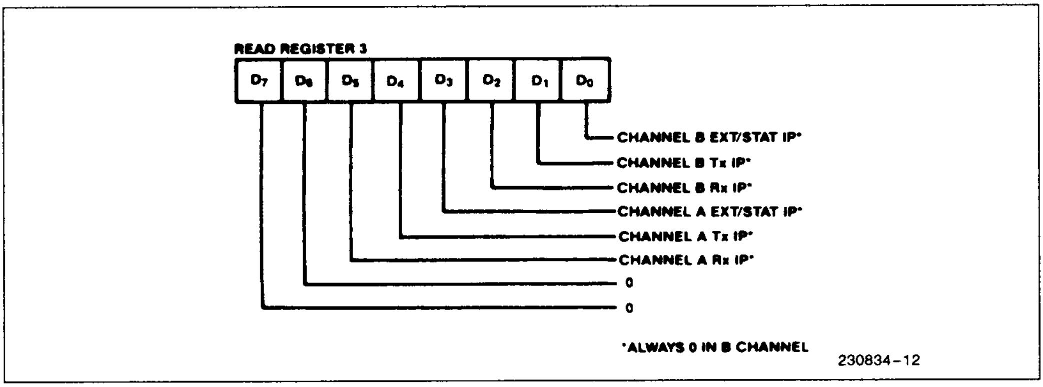


Figure 9. Read Register Bit Functions





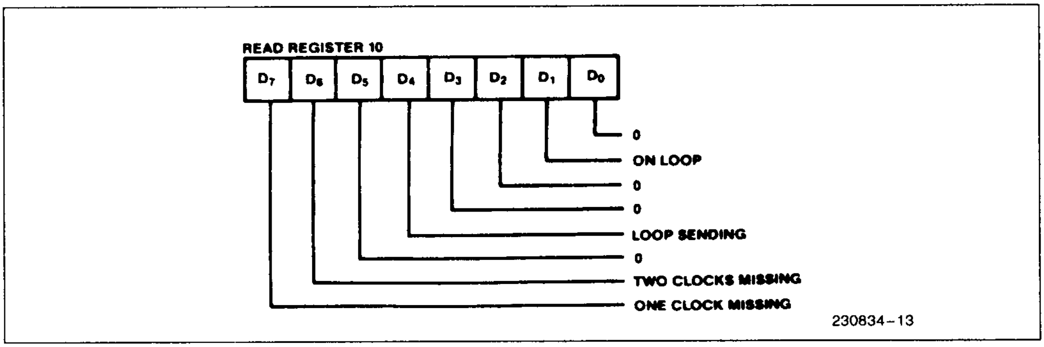
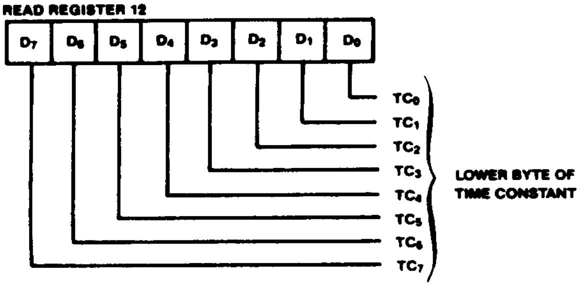
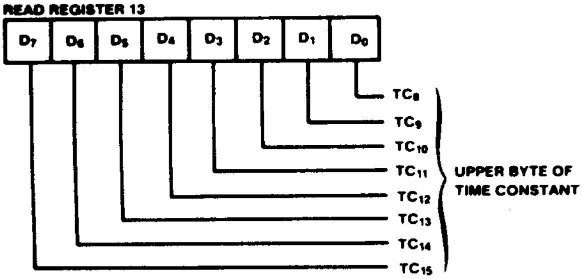


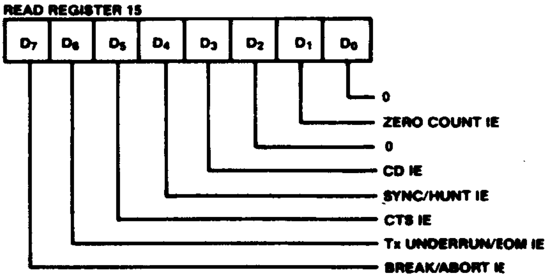
Figure 9. Read Register Bit Functions (Continued)



230834–14

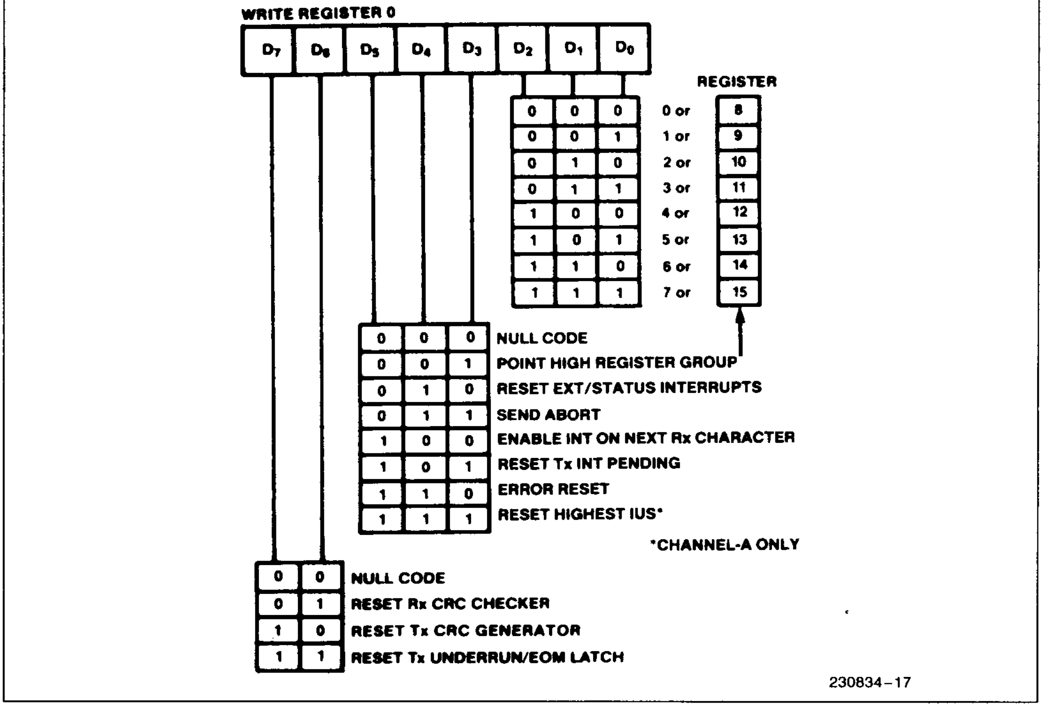


230834–15



230834–16

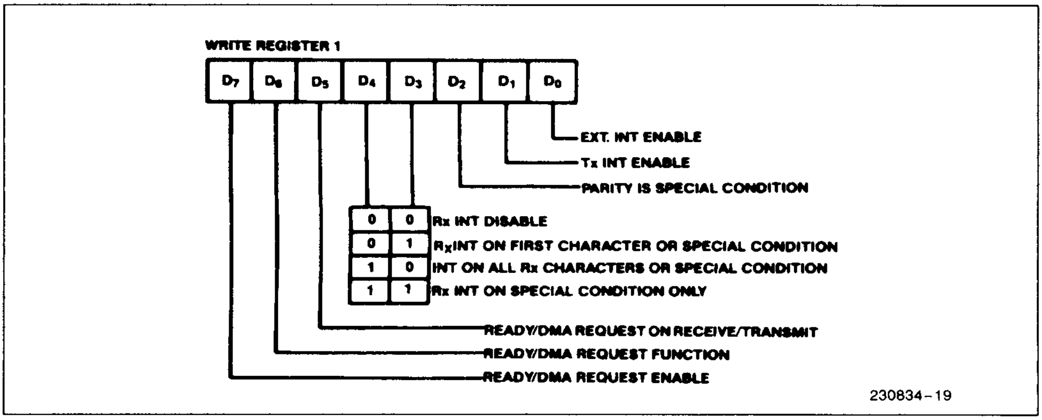
Figure 9. Read Register Bit Functions (Continued)



D2	D1	D0		REGISTER
0	0	0	0 or	8
0	0	1	1 or	9
0	1	0	2 or	10
0	1	1	3 or	11
1	0	0	4 or	12
1	0	1	5 or	13
1	1	0	6 or	14
1	1	1	7 or	15

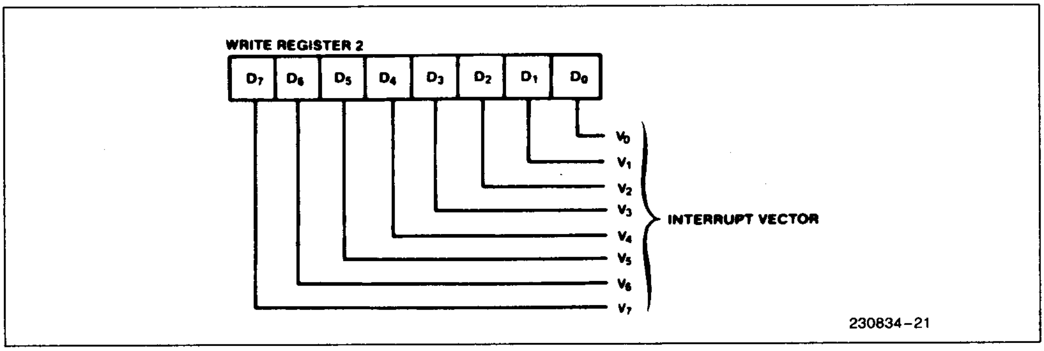
D5	D4	D3	
0	0	0	NULL CODE
0	0	1	POINT HIGH REGISTER GROUP
0	1	0	RESET EXT/STATUS INTERRUPTS
0	1	1	SEND ABORT
1	0	0	ENABLE INT ON NEXT Rx CHARACTER
1	0	1	RESET Tx INT PENDING
1	1	0	ERROR RESET
1	1	1	RESET HIGHEST IUS*

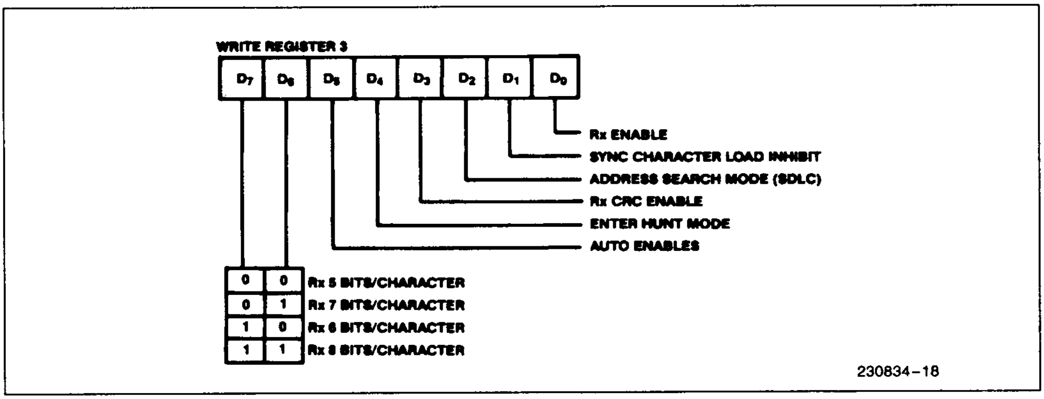
D7	D6	
0	0	NULL CODE
0	1	RESET Rx CRC CHECKER
1	0	RESET Tx CRC GENERATOR
1	1	RESET Tx UNDERRUN/EOM LATCH



D4	D3	
0	0	Rx INT DISABLE
0	1	RxINT ON FIRST CHARACTER OR SPECIAL CONDITION
1	0	INT ON ALL Rx CHARACTERS OR SPECIAL CONDITION
1	1	Rx INT ON SPECIAL CONDITION ONLY

Figure 10. Write Register Bit Functions





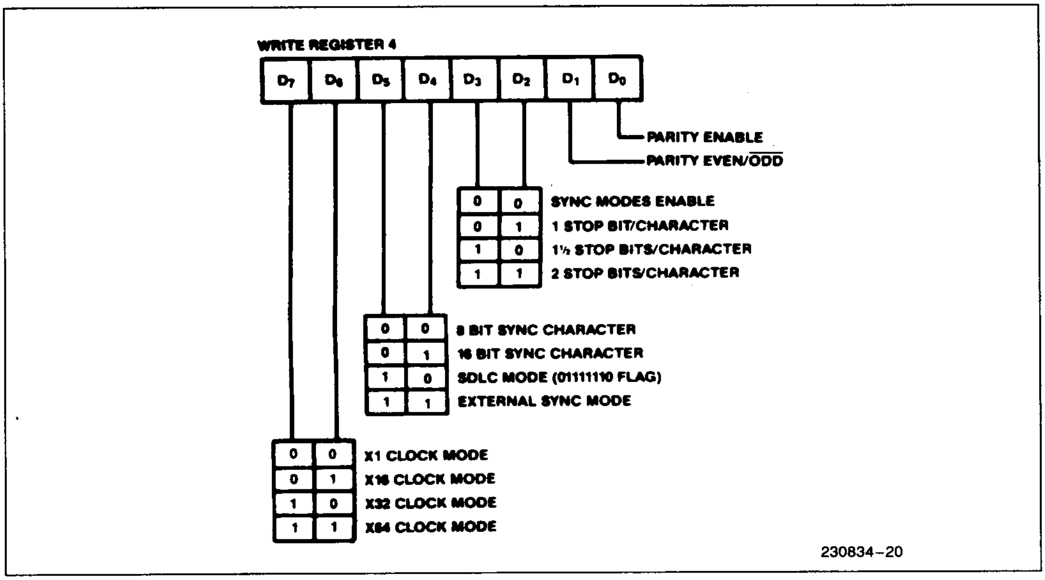
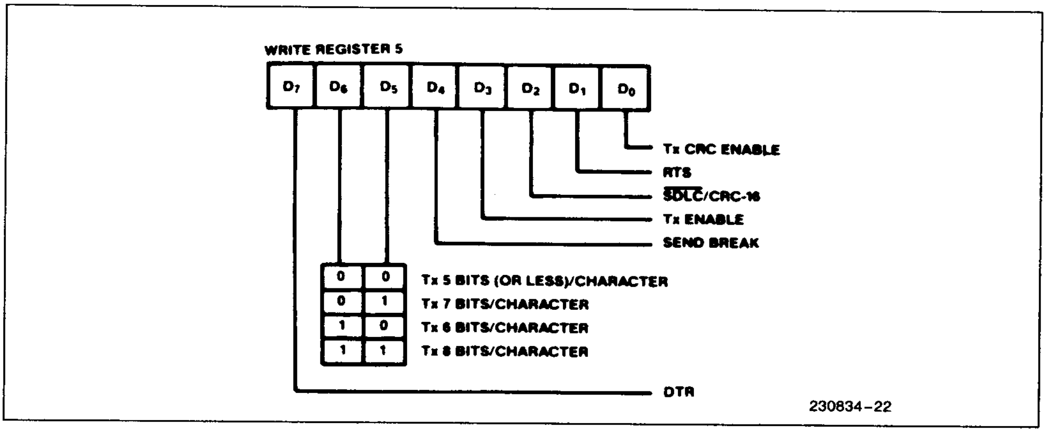
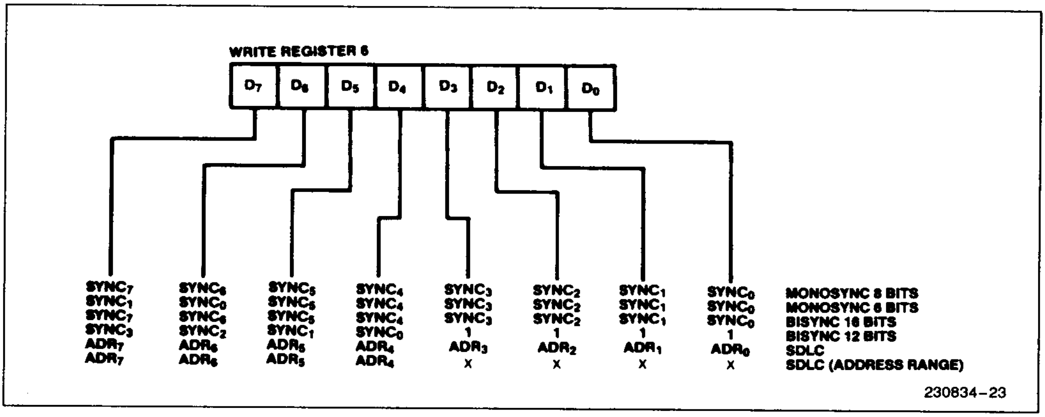


Figure 10. Write Register Bit Functions (Continued)




WRITE REGISTER 6
D7 D6 D5 D4 D3 D2 D1 D0
SYNC7 SYNC1 SYNC7 SYNC3 ADR7 ADR7
SYNC6 SYNC0 SYNC6 SYNC2 ADR6 ADR6
SYNC5 SYNC5 SYNC5 SYNC1 ADR5 ADR5
SYNC4 SYNC4 SYNC4 SYNC0 ADR4 ADR4
SYNC3 SYNC3 SYNC3 1 ADR3 X
SYNC2 SYNC2 SYNC2 1 ADR2 X
SYNC1 SYNC1 SYNC1 1 ADR1 X
SYNC0 SYNC0 SYNC0 1 ADR0 X
MONOSYNC 8 BITS
MONOSYNC 6 BITS
BISYNC 16 BITS
BISYNC 12 BITS
SDLC
SDLC (ADDRESS RANGE)
230834-23

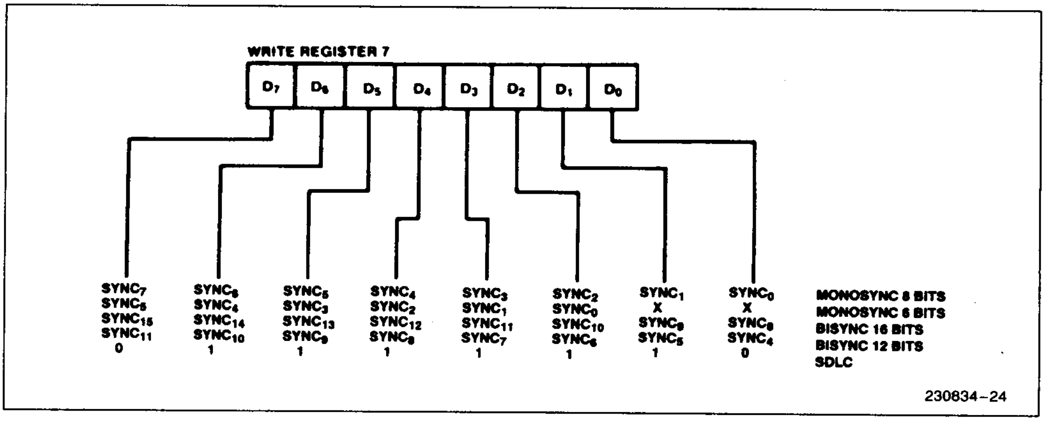
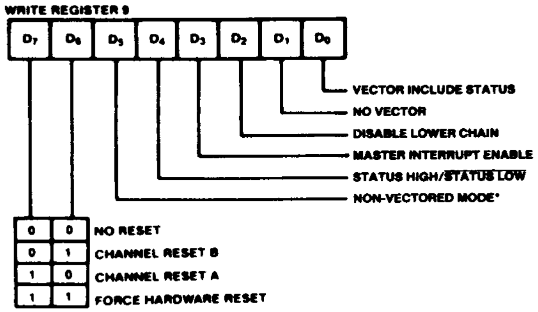
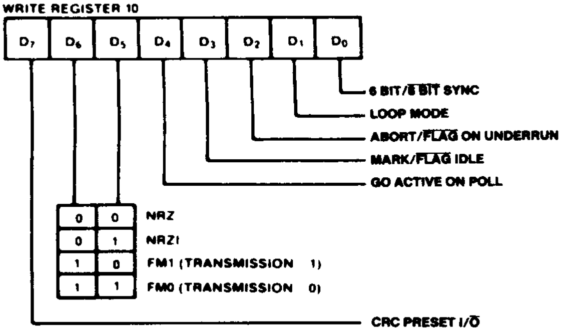

WRITE REGISTER 7
D7 D6 D5 D4 D3 D2 D1 D0
SYNC7 SYNC5 SYNC15 SYNC11 0
SYNC6 SYNC4 SYNC14 SYNC10 1
SYNC5 SYNC3 SYNC13 SYNC9 1
SYNC4 SYNC2 SYNC12 SYNC8 1
SYNC3 SYNC1 SYNC11 SYNC7 1
SYNC2 SYNC0 SYNC10 SYNC6 1
SYNC1 X SYNC9 SYNC5 1
SYNC0 X SYNC8 SYNC4 0
MONOSYNC 8 BITS
MONOSYNC 6 BITS
BISYNC 16 BITS
BISYNC 12 BITS
SDLC
230834-24

Figure 10. Write Register Bit Functions (Continued)

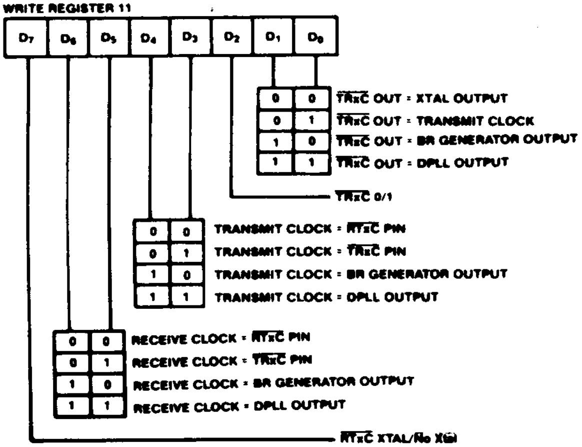


230834-25

*See 82530 Technical User's Manual for details of this mode. (Document # 230925-002 or -003)

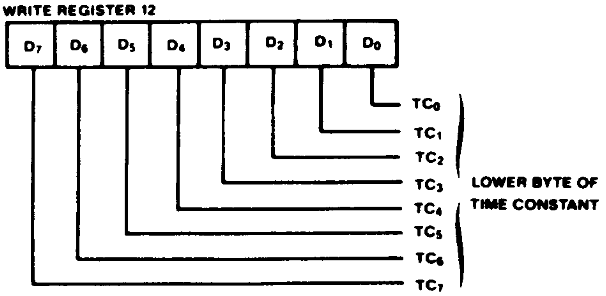


230834-27

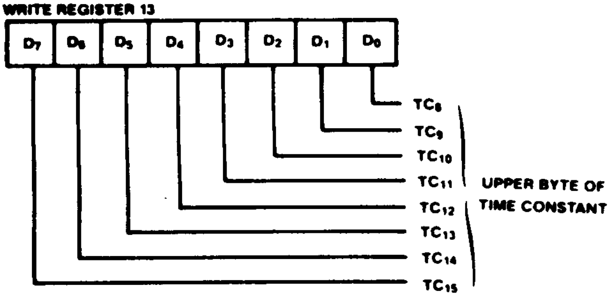

WRITE REGISTER 11
D7
D6
D5
D4
D3
D2
D1
D0
0 0 TRxC OUT = XTAL OUTPUT
0 1 TRxC OUT = TRANSMIT CLOCK
1 0 TRxC OUT = BR GENERATOR OUTPUT
1 1 TRxC OUT = DPLL OUTPUT
TRxC 0/I
0 0 TRANSMIT CLOCK = RTxC PIN
0 1 TRANSMIT CLOCK = TRxC PIN
1 0 TRANSMIT CLOCK = BR GENERATOR OUTPUT
1 1 TRANSMIT CLOCK = DPLL OUTPUT
0 0 RECEIVE CLOCK = RTxC PIN
0 1 RECEIVE CLOCK = TRxC PIN
1 0 RECEIVE CLOCK = BR GENERATOR OUTPUT
1 1 RECEIVE CLOCK = DPLL OUTPUT
RTxC XTAL/No XTAL

230834-29

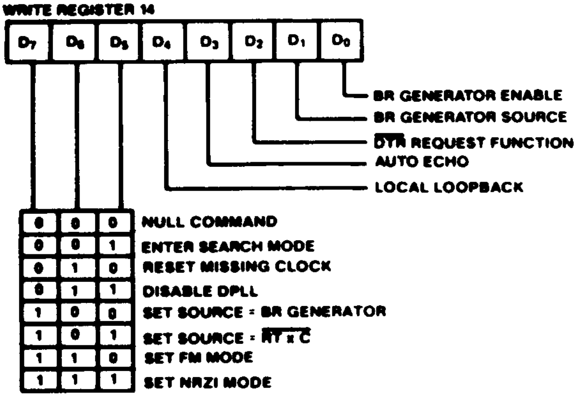
Figure 10. Write Register Bit Functions (Continued)



230834-26



230834-28



0	0	0	NULL COMMAND
0	0	1	ENTER SEARCH MODE
0	1	0	RESET MISSING CLOCK
0	1	1	DISABLE DPLL
1	0	0	SET SOURCE = BR GENERATOR
1	0	1	SET SOURCE = $\overline{\text{RTxC}}$
1	1	0	SET FM MODE
1	1	1	SET NRZI MODE

230834-30

Figure 10. Write Register Bit Functions (Continued)

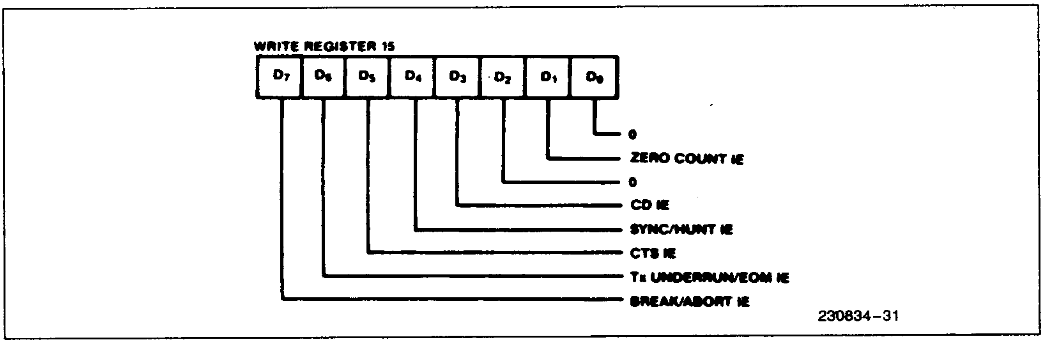


Figure 10. Write Register Bit Functions (Continued)

82530 TIMING

The SCC generates internal control signals from $\overline{WR}$ and $\overline{RD}$ that are related to CLK. Since CLK 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 CLK. 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, T_{REC} must be at least 6 CLK cycles plus 130 ns, for the 82530-6.

Read Cycle Timing

Figure 11 illustrates Read cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTA}$ must remain stable throughout the cycle. If $\overline{CS}$ falls after $\overline{RD}$ falls or if it rises before $\overline{RD}$ rises, the effective $\overline{RD}$ is shortened.

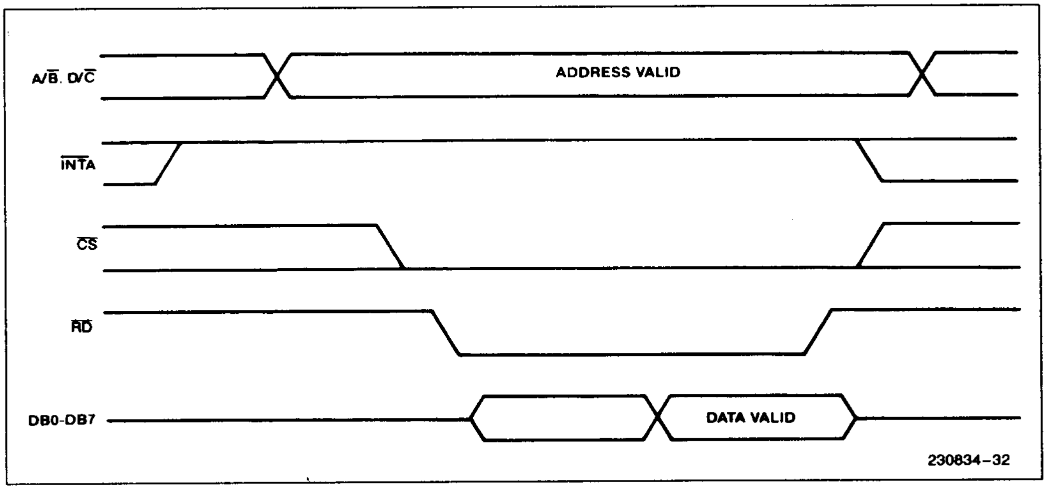


Figure 11. Read Cycle Timing

Write Cycle Timing

Figure 12 illustrates Write cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTA}$ must remain stable throughout the cycle. If $\overline{CS}$ falls after $\overline{WR}$ falls or if it rises before $\overline{WR}$ rises, the effective $\overline{WR}$ is shortened.

Interrupt Acknowledge Cycle Timing

Figure 13 illustrates Interrupt Acknowledge cycle timing. Between the time $\overline{INTA}$ 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 D_0–D_7 and it then sets the appropriate Interrupt-Under-Service internally.

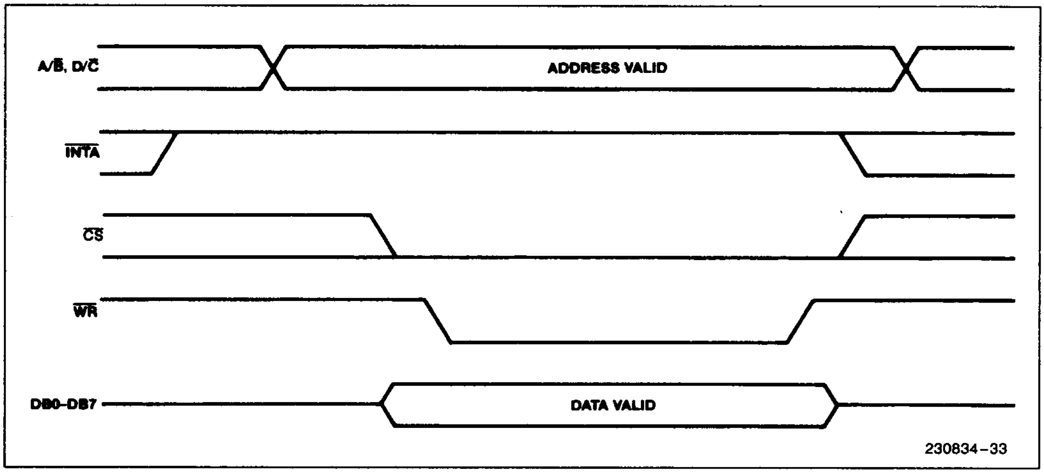


Figure 12. Write Cycle Timing

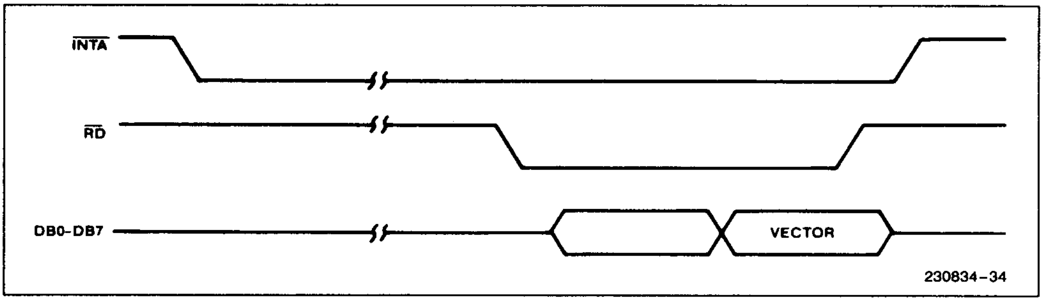


Figure 13. Interrupt Acknowledge Cycle Timing

ABSOLUTE MAXIMUM RATINGS*

Case Temperature
Under Bias ....0°C to +70°C

Storage Temperature
Ceramic Package .... −65°C to +150°C
Plastic Package .... −40°C to +125°C

Voltage on Any Pin with
Respect to Ground .... −0.5V to +7.0V

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

D.C. CHARACTERISTICS T_C = 0°C to 70°C; V_{CC} = +5V ±5%

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	−0.3	+0.8	V	
V_{IH}	Input High Voltage	+2.4	V_{CC} +0.3	V	
V_{OL}	Output Low Voltage		+0.45	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	+2.4		V	I_{OH} = −250 μA
I_{IL}	Input Leakage Current		±10	μA	0.4V to 2.4V
I_{OL}	Output Leakage Current		±10	μA	0.4V to 2.4V
I_{CC}	V_{CC} Supply Current		250	mA	

CAPACITANCE T_C = 25°C; V_{CC} = GND = 0V

Symbol	Parameter	Min	Max	Units	Test Conditions
C_{IN}	Input Capacitance		10	pF	f_c = 1 MHz
C_{OUT}	Output Capacitance		15	pF	Unmeasured pins returned to GND
$C_{I/O}$	Input/Output Capacitance		20	pF	

A.C CHARACTERISTICS $T_C = 0°C$ to $+70°C$; $V_{CC} = +5V \pm 5\%$

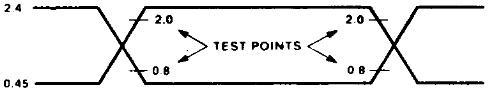
READ AND WRITE TIMING

Number	Symbol	Parameter	82530 (4 MHz)		82530-6 (6 MHz)		Units
			Min	Max	Min	Max	
1	tCL	CLK Low Time	105	2000	70	1000	ns
2	tCH	CLK High Time	105	2000	70	1000	ns
3	tf	CLK Fall Time		20		10	ns
4	tr	CLK Rise Time		20		15	ns
5	tCY	CLK Cycle Time	250	4000	165	2000	ns
6	tAW	Address to $\overline{\text{WR}}$ ↓ Setup Time	80		0		ns
7	tWA	Address to $\overline{\text{WR}}$ ↑ Hold Time	0		0		ns
8	tAR	Address to $\overline{\text{RD}}$ ↓ Setup Time	80		0		ns
9	tRA	Address to $\overline{\text{RD}}$ ↑ Hold Time	0		0		ns
10	tIC	$\overline{\text{INTA}}$ to CLK ↑ Setup Time	5		5		ns
11	tIW	$\overline{\text{INTA}}$ to $\overline{\text{WR}}$ ↓ Setup Time (Note 1)	200		55		ns
12	tWI	$\overline{\text{INTA}}$ to $\overline{\text{WR}}$ ↑ Hold Time	0		0		ns
13	tIR	$\overline{\text{INTA}}$ to $\overline{\text{RD}}$ ↓ Setup Time (Note 1)	200		55		ns
14	tRI	$\overline{\text{INTA}}$ to $\overline{\text{RD}}$ ↑ Hold Time	0		0		ns
15	tCI	$\overline{\text{INTA}}$ to CLK ↑ Hold Time	100		100		ns
16	tCLW	$\overline{\text{CS}}$ Low to $\overline{\text{WR}}$ ↓ Setup Time	0		0		ns
17	tWCS	$\overline{\text{CS}}$ to $\overline{\text{WR}}$ ↑ Hold Time	0		0		ns
18	tCHW	$\overline{\text{CS}}$ High to $\overline{\text{WR}}$ ↓ Setup Time	100		5		ns
19	tCLR	$\overline{\text{CS}}$ Low to $\overline{\text{RD}}$ ↓ Setup Time (Note 1)	0		0		ns
20	tRCS	$\overline{\text{CS}}$ to $\overline{\text{RD}}$ ↑ Hold Time (Note 1)	0		0		ns
21	tCHR	$\overline{\text{CS}}$ High to $\overline{\text{RD}}$ ↓ Setup Time (Note 1)	100		5		ns
22	tRR	$\overline{\text{RD}}$ Low Time (Note 1)	390		150		ns
23	Null	Parameter Deleted					
24	tRDI	$\overline{\text{RD}}$ ↑ to Data Not Valid Delay	0		0		ns
25	tRDV	$\overline{\text{RD}}$ ↓ to Data Valid Delay		250		105	ns
26	tDF	$\overline{\text{RD}}$ ↑ to Output Float Delay (Note 2)		70		45	ns

NOTES:

1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time required for a +0.5V change in the output with a maximum D.C. load and minimum A.C. load.

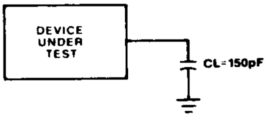
A.C. TESTING INPUT, OUTPUT WAVEFORM



230834-35

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.45V for a Logic "0". Timing measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0".

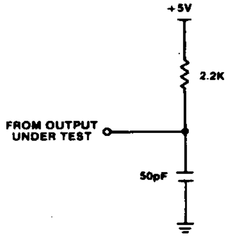
A.C. TESTING LOAD CIRCUIT



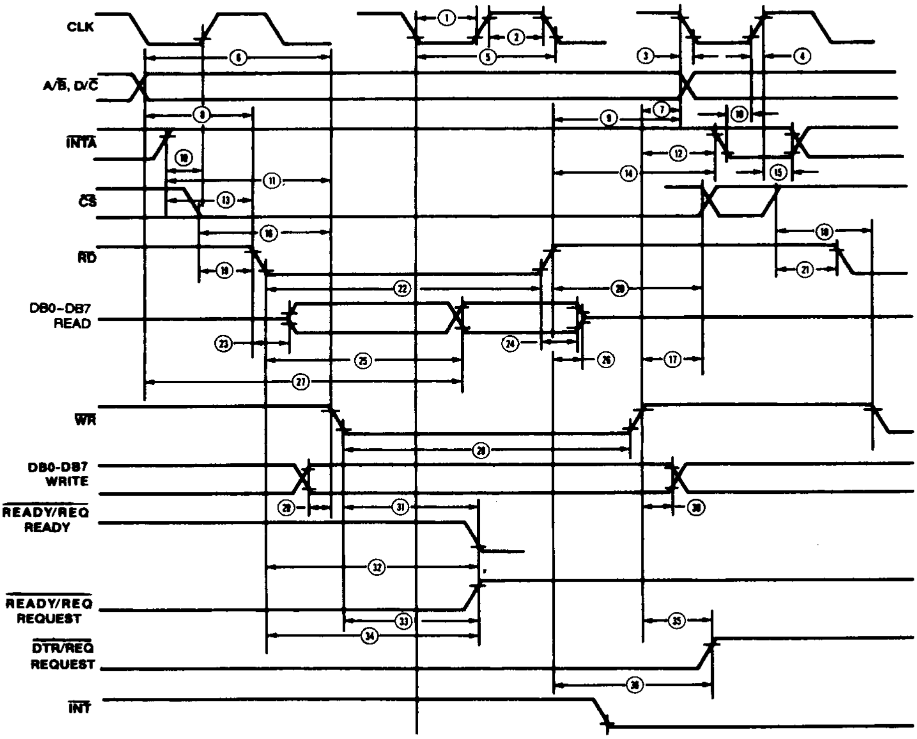
230834-41

C_L = 150 pF
C_L Includes Jig Capacitance

OPEN DRAIN TEST LOAD



230834-42



230834-36

Figure 14. Read and Write Timing

INTERRUPT ACKNOWLEDGE TIMING, RESET TIMING, CYCLE TIMING

Number	Symbol	Parameter	82530 (4 MHz)		82530-6 (6 MHz)		Units
			Min	Max	Min	Max	
27	tAD	Address Required Valid to Read Data Valid Delay		590		325	ns
28	TWW	$\overline{\text{WR}}$ Low Time	390		60		ns
29	tDW	Data to $\overline{\text{WR}}$ ↓ Setup Time	0		0		ns
30	tWD	Data to $\overline{\text{WR}}$ ↑ Hold Time	0		0		ns
31	tWRV	$\overline{\text{WR}}$ ↓ to Ready Valid Delay (Note 4)		240		200	ns
32	tRRV	$\overline{\text{RD}}$ ↓ to Ready Valid Delay (Note 4)		240		200	ns
33	tWRI	$\overline{\text{WR}}$ ↓ to $\overline{\text{READY}}$/$\overline{\text{REQ}}$ Not Valid Delay		240		200	ns
34	tRRI	$\overline{\text{RD}}$ ↓ to $\overline{\text{READY}}$/$\overline{\text{REQ}}$ Not Valid Delay		240		200	ns
35	tDWR	$\overline{\text{WR}}$ ↑ to $\overline{\text{DTR}}$/$\overline{\text{REQ}}$ Not Valid Delay		5 tCY +300		5 tCY +250	ns
36	tDRD	$\overline{\text{RD}}$ ↑ to $\overline{\text{DTR}}$/$\overline{\text{REQ}}$ Not Valid Delay		5 tCY +300		5 tCY +250	ns
37	tIID	$\overline{\text{INTA}}$ to $\overline{\text{RD}}$ ↓ (Acknowledge) Delay (Note 5)	250		250		ns
38	tII	$\overline{\text{RD}}$ (Acknowledge) Low Time	285		125		ns
39	tIDV	$\overline{\text{RD}}$ ↓ (Acknowledge) to Read Data Valid Delay		190		100	ns
40	tEI	IEI to $\overline{\text{RD}}$ ↓ (Acknowledge) Setup Time	120		100		ns
41	tIE	IEI to $\overline{\text{RD}}$ ↑ (Acknowledge) Hold Time	0		0		ns
42	tEIEO	IEI to IEO Delay Time		120		100	ns
43	tCEQ	CLK ↑ to IEO Delay		250		250	ns
44	tRII	$\overline{\text{RD}\downarrow}$ to $\overline{\text{INT}}$ Inactive Delay (Note 4)		500		500	ns
45	tRW	$\overline{\text{RD}}$ ↑ to $\overline{\text{WR}}$ ↓ Delay for No Reset	30		15		ns
46	tWR	$\overline{\text{WR}}$ ↑ to $\overline{\text{RD}}$ ↓ Delay for No Reset	30		30		ns
47	tRES	$\overline{\text{WR}}$ and $\overline{\text{RD}}$ Coincident Low for Reset	250		250		ns
48	tREC	Valid Access Recovery Time (Note 3)	6 tCY +200		6 tCY +130		ns

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, tIID must be greater than the sum of tCEQ for the highest priority device in the daisy chain, tEI for the SCC and tEIEO for each device separating them in the daisy chain.

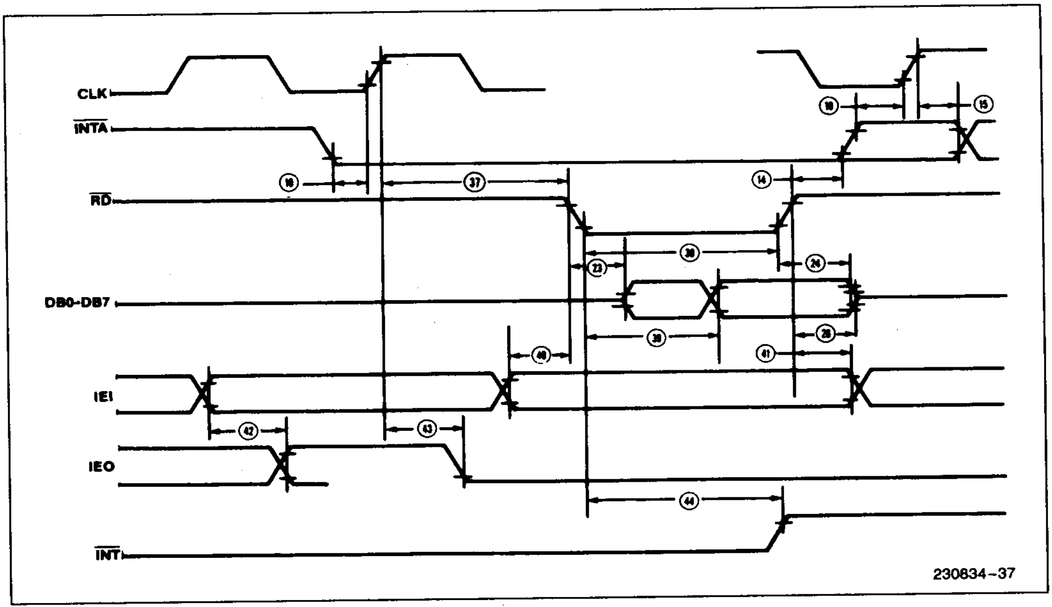


Figure 15. Interrupt Acknowledge Timing

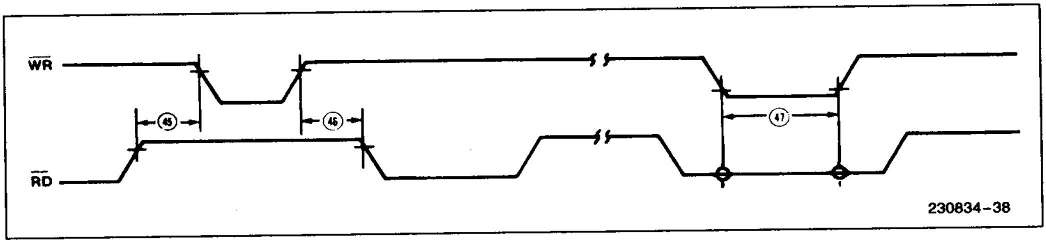


Figure 16. Reset Timing

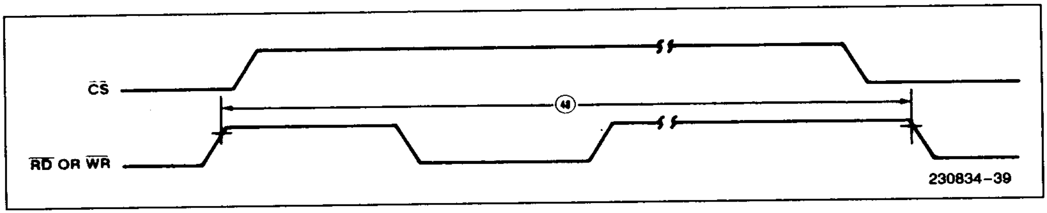


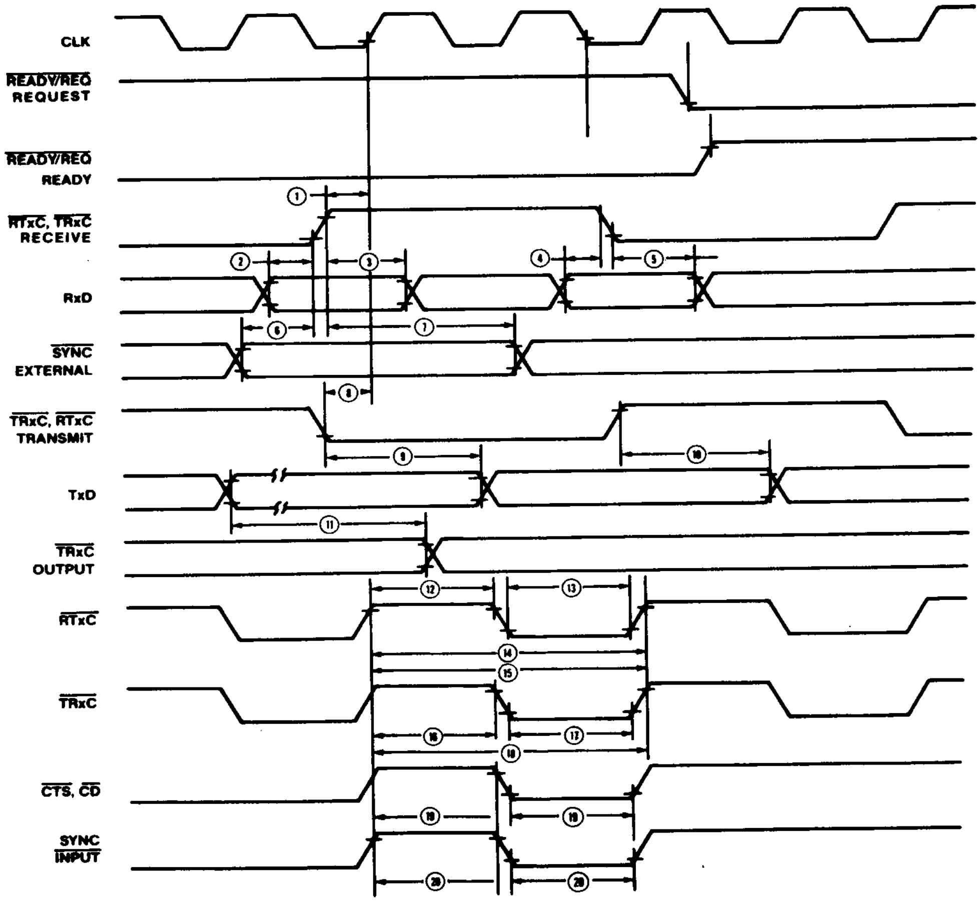
Figure 17. Cycle Timing

GENERAL TIMING

Number	Symbol	Parameter	82530 (4 MHz)		82530-6 (6 MHz)		Units
			Min	Max	Min	Max	
1	tRCC	$\overline{\text{RxC}}$ ↑ to CLK ↑ Setup Time (Notes 1, 4)	100		100		ns
2	tRRC	RxD to $\overline{\text{RxC}}$ ↑ Hold Time (X1 Mode) (Note 1)	0		0		ns
3	tRCR	RxD to $\overline{\text{RxC}}$ ↑ Hold Time (X1 Mode) (Note 1)	150		150		ns
4	tDRC	RxD to $\overline{\text{RxC}}$ ↓ Setup Time (X1 Mode) (Notes 1, 5)	0		0		ns
5	tRCD	RxD to $\overline{\text{RxC}}$ ↓ Hold Time (X1 Mode) (Notes 1, 5)	150		150		ns
6	tSRC	$\overline{\text{SYNC}}$ to $\overline{\text{RxC}}$ ↑ Setup Time (Note 1)	−200		−200		ns
7	tRCS	$\overline{\text{SYNC}}$ to$\overline{\text{RxC}}$ ↑ Hold Time (Note 1)	3 tCY +200		3 tCY +200		ns
8	tTCC	$\overline{\text{TxC}}$ ↓ to CLK ↑ Setup Time (Notes 2, 4)	100		100		ns
9	tTCT	$\overline{\text{TxC}}$ ↓ to TxD Delay (X1 Mode) (Note 2)		300		300	ns
10	tTCD	$\overline{\text{TxC}}$ ↑ to TxD Delay (X1 Mode) (Notes 2, 5)		300		300	ns
11	tTDT	TxD to $\overline{\text{TRxC}}$ Delay (Send Clock Echo)		200		200	ns
12	tDCH	$\overline{\text{RTxC}}$ High Time	180		150		ns
13	tDCL	$\overline{\text{RTxC}}$ Low Time	180		150		ns
14	tDCY	$\overline{\text{RTxC}}$ Cycle Time	4tCY		4tCY		ns
15	tCLCL	Crystal Oscillator Period (Note 3)	250	1000	165	1000	ns
16	tRCH	$\overline{\text{TRxC}}$ High Time	180		150		ns
17	tRCL	$\overline{\text{TRxC}}$ Low Time	180		150		ns
18	tRCY	$\overline{\text{TRxC}}$ Cycle Time (Note 6)	4tCY		4tCY		ns
19	tCC	$\overline{\text{CD}}$ or $\overline{\text{CTS}}$ Pulse Width	200		200		ns
20	tSS	$\overline{\text{SYNC}}$ Pulse Width	200		200		ns
21	tWRT	$\overline{\text{WR}}$ to $\overline{\text{RTS}}$ Valid Delay		6tCY		6tCY	ns
22	tWDT	$\overline{\text{WR}}$ to $\overline{\text{DTR}}$ Valid Delay		5tCY		5tCY	ns

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 system clock (CLK) rate. In all other cases, no phase relationship between $\overline{\text{RxC}}$ and CLK or $\overline{\text{TxC}}$ and CLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Only applies to transmitter and receiver. For DPLL and Baud Rate Generator Timings, the requirements are identical to system clock, CLK, specifications.



230834-40

Figure 18. General Timing