Final

# Z85C30

## Enhanced Serial Communications Controller

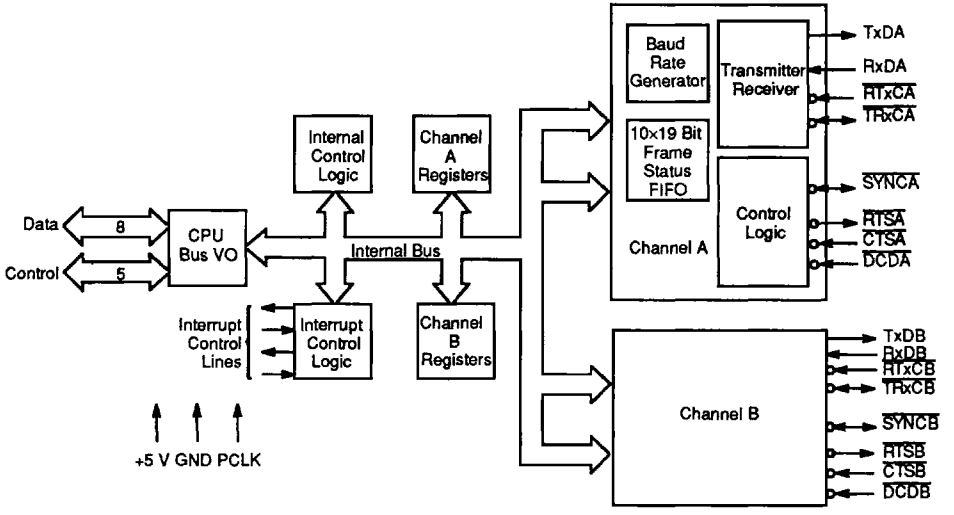
Advanced Micro Devices

## DISTINCTIVE CHARACTERISTICS

- **Fastest Data Rate of any Z8530**
  - 8.192 MHz / 2.048 Mb/s
  - 10 MHz / 2.5 Mb/s
  - 12.5 MHz / 3 Mb/s
  - 16.384 MHz / 4.096 Mb/s
  - 20 MHz / 5 Mb/s (prelim)
- **Low Power CMOS Technology**
- **Pin and Function Compatible with other NMOS and CMOS Z8530s**
- **Easily Interfaced with most CPUs**
  Compatible with non-multiplexed bus
- **Many Enhancements over NMOS Z8530H**
  - Allows 85C30 to be used more effectively in high-speed applications
  - Improves interface capabilities
- **Two Independent Full-duplex Serial Channels**
- **Asynchronous Mode Features**
  - Programmable stop bits, clock factor, character length and parity
  - Break detection/generation
  - Error detection for framing, overrun and parity
- **Synchronous Mode Features**
  - Supports IBM BISYNC, SDLC, SDLC Loop, HDLC and ADCCP Protocols
  - Programmable CRC generators and checkers
  - SDLC/HDLC support includes frame control, zero insertion and deletion, abort, and residue handling

1

## BLOCK DIAGRAM




10216A-001A

BD008260

Publication # 10216 Rev. B Amendment /0
Issue Date: May 1989

## DISTINCTIVE CHARACTERISTICS (continued)

- **Enhanced SCC functions support high-speed frame reception using DMA**
  - 14-bit byte counter
  - 10 × 19 SDLC/HDLC Frame Status FIFO
  - Independent Control on both channels
  - Enhanced operation does not allow special receive conditions to lock the three-byte DATA FIFO when the 10 × 19 FIFO is enabled
- **Local Loopback and Auto Echo Modes**
- **Internal or External Character Synchronization**
- **2 Mb/s FM Encoding Transmit and Receive capability using Internal DPLL for 16.384-MHz product**
- **Internal Synchronization between RxC to PCLK and TxC to PCLK**
  This allows the user to eliminate external sychronization hardware required by the NMOS device when transmitting or receiving data at the maximum rate of 1/4 PCLK frequency.

## GENERAL DESCRIPTION

AMD's Z85C30 is an enhanced pin-compatible version of the popular Z8530/Z85C30 Serial Communications Controller. The Enhanced Serial Communications Controller (ESCC) is a high-speed, low-power, multi-protocol communications peripheral designed for use with 8- and 16-bit microprocessors. It has two independent, full duplex channels and functions as a serial-to-parallel, parallel-to-serial converter/controller. AMD's proprietary enhancements make the Z85C30 easier to interface and more effective in high-speed applications due to a reduction in software burden and the elimination of the need for some external glue logic.

The Z85C30 is easy to use due to a variety of sophisticated internal functions, including on-chip baud rate generators, digital phase-locked loops and crystal oscillators which dramatically reduce the need for external logic. The device can generate and check CRC codes in any SYNC mode, and can be programmed to check data integrity in various modes. The ESCC 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.

This versatile device supports virtually any serial data transfer application such as networks, modems, cassettes and tape drivers. The ESCC is designed for non-multiplexed buses and is easily interfaced with most CPUs, such as 80188, 80186, 80286, 8080, Z80, 6800, 68000 and MULTIBUS.

Enhancements which allow the Z85C30 to be used more effectively in high-speed applications include:

- a 10 × 19 bit SDLC/HDLC frame status FIFO array
- a 14-bit SDLC/HDLC frame byte counter
- automatic SDLC/HDLC opening frame flag transmission
- TxD pin forced HIGH in SDLC NRZI mode after closing flag
- automatic SDLC/HDLC Tx underrun/EOM flag reset
- automatic SDLC/HDLC Tx CRC generator reset/preset
- $\overline{RTS}$ synchronization to closing SDLC/HDLC flag
- $\overline{DTR/REQ}$ deactivation delay significantly reduced
- external PCLK to $\overline{RxC}$ or $\overline{TxC}$ synchronization requirement eliminated for PCLK divide-by-four operation

Other enhancements to improve the Z85C30 interface capabilities include:

- write data valid setup time to falling edge of $\overline{WR}$ requirement eliminated
- reduced $\overline{INT}$ response time
- reduced access recovery time ($t_{RC}$) to 3 PCLK best case (3 1/2 PCLK worst case)
- improved $\overline{Wait}$ timing
- write registers WR3, WR4, WR5, and WR10 made readable
- lower priority interrupt masking without $\overline{INTACK}$
- complete SDLC/HDLC CRC character reception

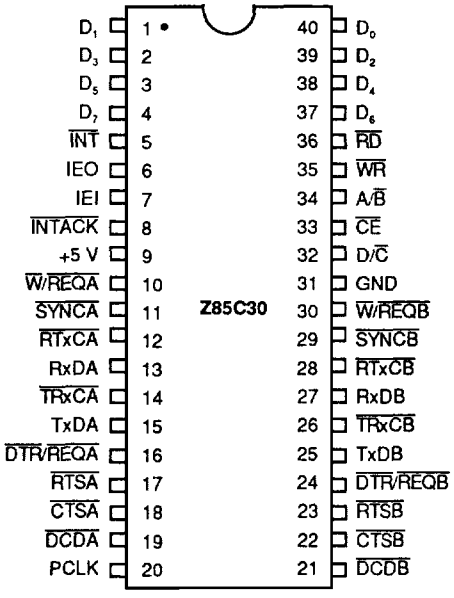
## RELATED AMD PRODUCTS

| Part No. | Description | Part No. | Description |
|---|---|---|---|
| Am7960 | Coded Data Transceiver | Am9517A | DMA Controller |
| 80186 | Highly Integrated 16-Bit Microprocessor | 5380, 53C80 | SCSI Bus Controller |
| | | 80188 | Highly Integrated 8-Bit Microprocessor |
| 80286, 80C286 | High-Performance 16-Bit Microprocessor | | |

## CONNECTION DIAGRAMS

**Top View**

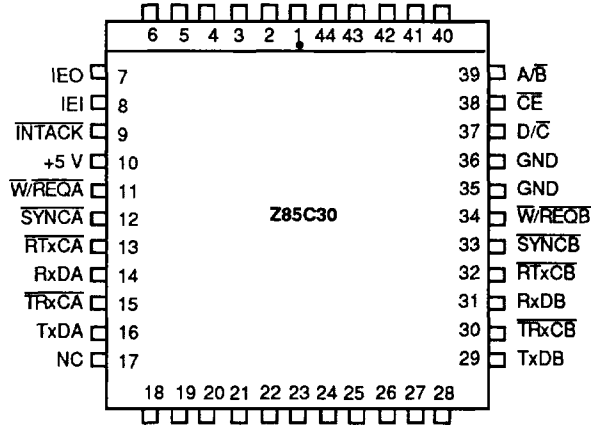
DIPs




10216A-002A

CD011530

PLCC/LCC



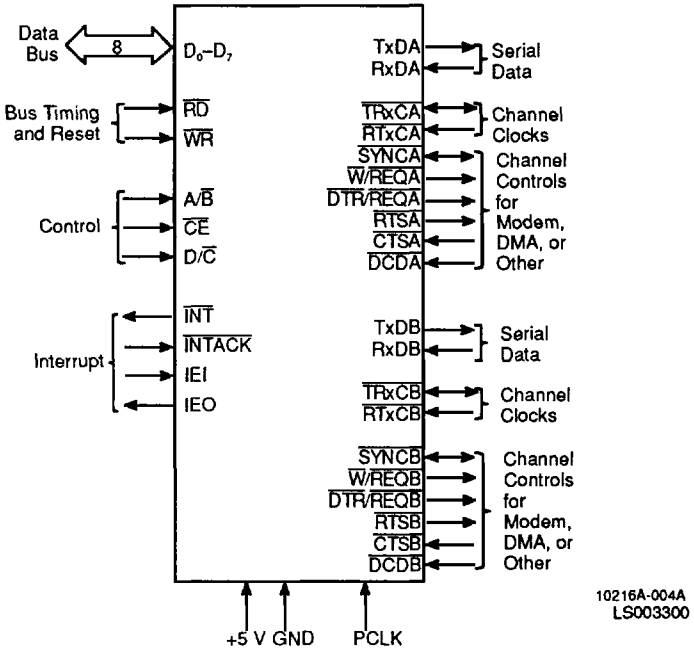

10216A-003A

CD011540

Note: Pin 1 is marked for orientation.

1

## LOGIC SYMBOL



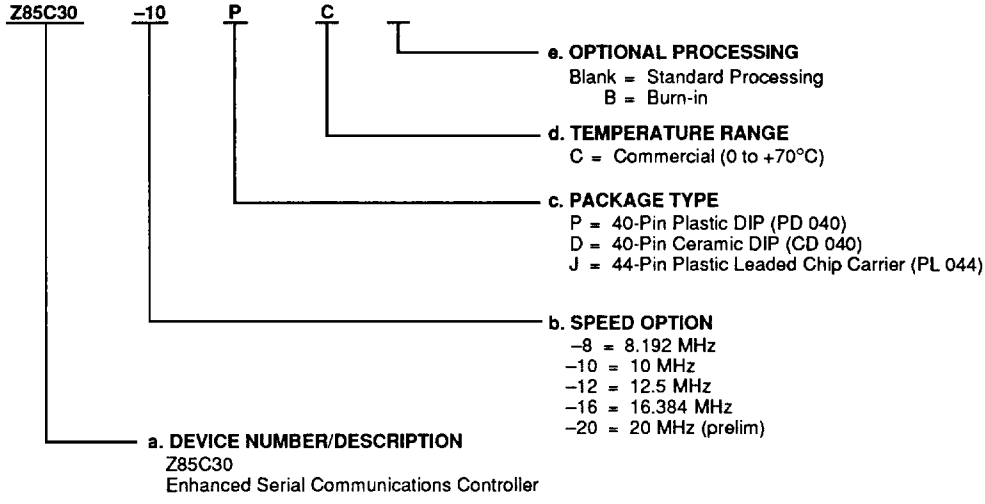

10216A-004A
LS003300

## ORDERING INFORMATION

### Commodity Products

AMD commodity products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

**a. Device Number**
**b. Speed Option (if applicable)**
**c. Package Type**
**d. Temperature Range**
**e. Optional Processing**




| Valid Combinations | |
|---|---|
| Z85C30-8<br>Z85C30-10<br>Z85C30-12<br>Z85C30-16<br>Z85C30-20 (prelim) | PC, DC, JC<br>DCB |

**Valid Combinations**

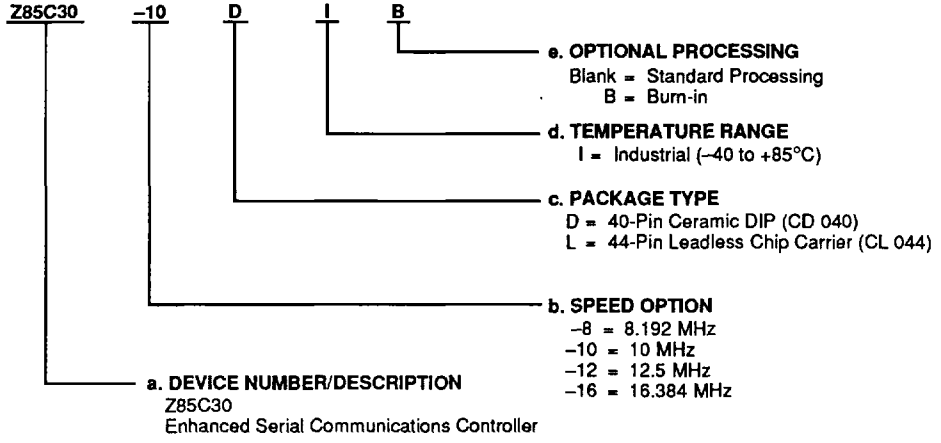
Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

## ORDERING INFORMATION

**Industrial Products**

AMD industrial products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

**a. Device Number**
**b. Speed Option (if applicable)**
**c. Package Type**
**d. Temperature Range**
**e. Optional Processing**




| Valid Combinations | |
|---|---|
| Z85C30-8<br>Z85C30-10<br>Z85C30-12<br>Z85C30-16 | DIB, LIB |

**Valid Combinations**

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

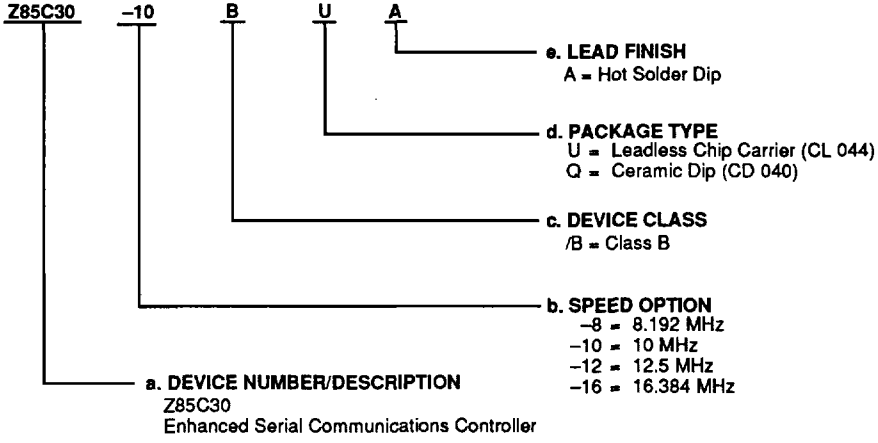
1

## MILITARY ORDERING INFORMATION

### APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The ordering number (Valid Combination) is formed by a combination of:

**a. Device Number**
**b. Speed Option (if applicable)**
**c. Device Class**
**d. Package Type**
**e. Lead Finish**




| Valid Combinations | |
|---|---|
| Z85C30-8<br>Z85C30-10<br>Z85C30-12<br>Z85C30-16 | BQA, BUA |

**Valid Combinations**

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

## PIN DESCRIPTION

### Bus Timing and Reset

**$\overline{RD}$**
**Read (Input; Active LOW)**

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

**$\overline{WR}$**
**Write (Input; Active LOW)**

When the SCC is selected, this signal indicates a write operation. The coincidence of $\overline{RD}$ and $\overline{WR}$ is interpreted as a reset.

### Channel Clocks

**$\overline{RTxCA}$, $\overline{RTxCB}$**
**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 of 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.

**$\overline{TRxCA}$, $\overline{TRxCB}$**
**Transmit/Receive Clocks**
**(Inputs/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.

### Channel Controls for Modem, DMA, or Other

**$\overline{CTSA}$, $\overline{CTSB}$**
**Clear to Send (Inputs; Active LOW)**

If these pins are programmed as Auto Enables, a LOW on these inputs enables their respective transmitter. 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 may interrupt the CPU on both logic level transitions.

**$\overline{DCDA}$, $\overline{DCDB}$**
**Data Carrier Detect (Inputs; Active LOW)**

These pins function as receiver enables if they are programmed as Auto Enables; otherwise, they may be used as general-purpose input pins. Both are Schmitt-trigger buffered to accommodate slow rise-time signals. The SCC detects pulses on these pins and may interrupt the CPU on both logic level transitions.

**$\overline{DTR/REQA}$**
**Data Terminal Ready/Request**
**(Outputs; Active LOW)**

These outputs follow the inverted state programmed into the DTR bit in WR5. They can also be used as general-purpose outputs or as Request Lines for a DMA controller.

**$\overline{RTSA}$, $\overline{RTSB}$**
**Request to Send (Outputs; Active LOW)**

When the Request to Send RTS bit in Write Register 5 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 SYNC mode or in asynchronous mode with Auto Enable off, the $\overline{RTS}$ pins strictly follow the inverted state of the RTS bit. Both pins can be used as general-purpose outputs.

**$\overline{SYNCA}$, $\overline{SYNCB}$**
**Synchronization (Inputs/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 Sync/Hunt status bits in Read Register 0 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 SYNC 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 SYNC characters are recognized. The SYNC condition is not latched, so these outputs are active each time a SYNC pattern is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.

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

## PIN DESCRIPTION (continued)

### Control

**A/$\overline{B}$**
**Channel A/$\overline{\text{Channel B}}$ Select (Input)**

This signal selects the channel in which the read or write operation occurs.

**$\overline{CE}$**
**$\overline{\text{Chip Enable}}$ (Input; Active LOW)**

This signal selects the SCC for a read or write operation.

**D/$\overline{C}$**
**Data/$\overline{\text{Control}}$ 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 is transferred.

### Data Bus

**$D_0$–$D_7$**
**Data Bus (Input/Output; Three State)**

These lines carry data and commands to and from the SCC.

### Interrupt

**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.

**$\overline{INT}$**
**$\overline{\text{Interrupt Request}}$ (Output; Active LOW, Open Drain)**

This signal is activated when the SCC requests an interrupt.

**$\overline{INTACK}$**
**$\overline{\text{Interrupt Acknowledge}}$ (Input; Active LOW)**

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). $\overline{INTACK}$ is latched by the rising edge of PCLK.

### Serial Data

**RxDa, RxDB**
**Receive Data (Inputs; Active HIGH)**

These input signals receive serial data at standard TTL levels.

**TxDA, TxDB**
**Transmit Data (Outputs; Active HIGH)**

These output signals transmit serial data at standard TTL levels.

### Miscellaneous

**GND**
**Ground**

**PCLK**
**Clock (Input)**

This is the master SCC clock used to synchronize internal signals. PCLK is not required to have any phase relationship with the master system clock. PCLK is a TTL-level signal. Maximum transmit rate is 1/4 PCLK.

**$V_{cc}$**
**+ 5 V Power Supply**

## ARCHITECTURE

The ESCC internal structure includes two full-duplex channels, two 10 × 19 bit SDLC/HDLC frame status FIFOs, two baud rate generators, internal control and interrupt logic, and a bus interface to a non-multiplexed 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 with modems or other external devices (see Logic Symbol).

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 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. An additional write register, WR7 Prime (WR7'), is available for enabling or disabling additional SDLC/HDLC enhancements if bit D0 of WR15 is set.

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

If bit D2 of WR15 is set, then two additional Read Registers, RR6 and RR7, are available. These registers are used with the 10 × 19 bit Frame Status FIFO.

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

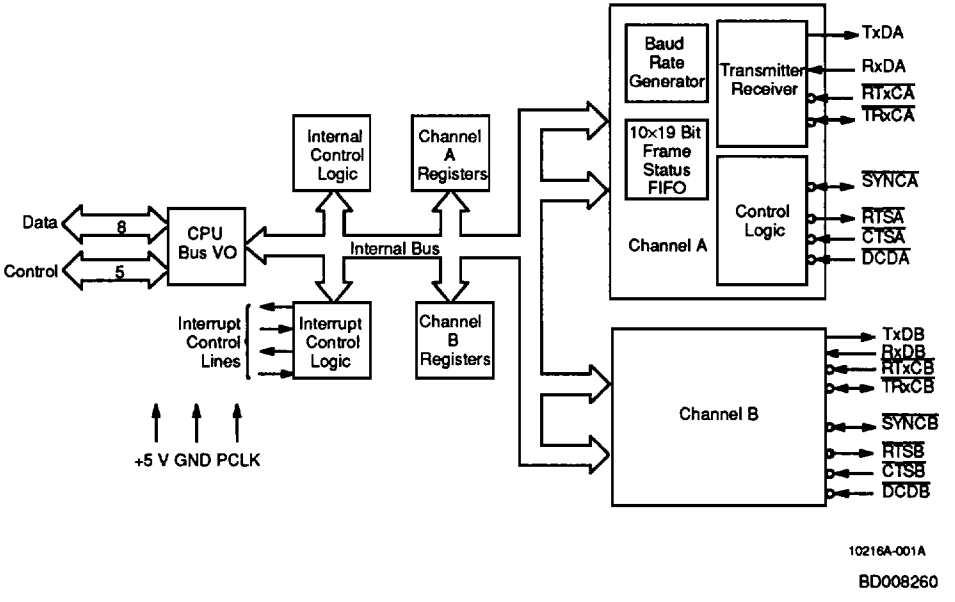



**Figure 1. Block Diagram of ESCC Architecture**

### Data Path

The transmit and receive data path illustrated in Figure 2 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 are 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 are routed through one of four main paths before they are transmitted from the Transmit Data output (TxD).

**Table 1. Read and Write Register Functions**

| Read Register Functions | | Write Register Functions | |
|---|---|---|---|
| **RR0** | Transmit/Receive buffer status and External status | **WR0** | Command Register, Register Pointers CRC initialize, initialization commands for the various modes, shift right/shift left command |
| **RR1** | Special Receive Condition status (also 10 x 19 bit FIFO Frame Reception Status if WR15 bit D2 is set) | **WR1** | Interrupt conditions and data transfer mode definition |
| **RR2** | Modified interrupt vector (Channel B only) Unmodified interrupt vector (Channel A only) | **WR2** | Interrupt vector (accessed through either channel) |
| | | **WR3** | Receive parameters and control |
| | | **WR4** | Transmit/Receive miscellaneous parameters and modes |
| **RR3** | Interrupt Pending bits (Channel A only) | **WR5** | Transmit parameters and controls |
| **RR6** | LSB Byte Count (14-bit counter) (if WR15 bit D2 set) | **WR6** | Sync character or SDLC address field |
| | | **WR7** | Sync character or SDLC flag |
| **RR7** | MSB Byte Count (14-bit counter) and 10 x 19 bit FIFO Status (if WR15 bit D2 is set) | **WR7′** | SDLC/HDLC enhancements (if bit D0 of WR15 set) |
| | | **WR8** | Transmit buffer |
| **RR8** | Receive buffer | **WR9** | Master interrupt control and reset (accessed through either channel) |
| **RR10** | Miscellaneous XMTR, RCVR status | **WR10** | Miscellaneous transmitter/receiver control bits, data encoding |
| **RR12** | Lower byte of baud rate generator time constant | **WR11** | Clock mode control, Rx and Tx clock source |
| **RR13** | Upper byte of baud rate generator time constant | **WR12** | Lower byte of baud rate generator time constant |
| | | **WR13** | Upper byte of baud rate generator time constant |
| | | **WR14** | Miscellaneous control bits, DPLL control |
| **RR15** | External/Status interrupt information | **WR15** | External/Status interrupt control |

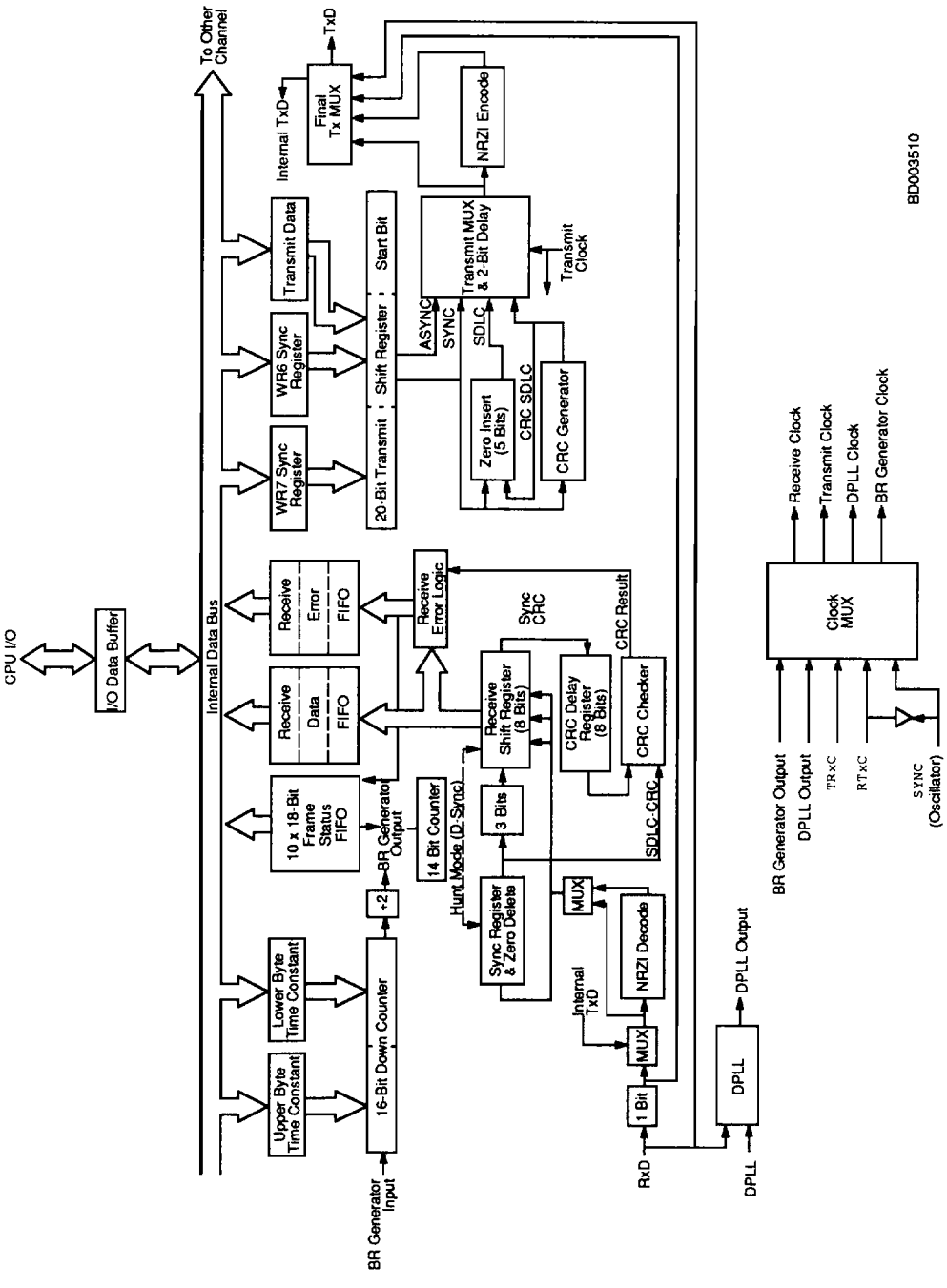



**Figure 2. Data Path**

## DETAILED DESCRIPTION

The functional capabilities of the ESCC 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 ESCC provides two independent full-duplex channels programmable for use in any common asynchronous or SYNC 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. 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 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 ESCC 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 ESCC supports both byte-oriented and bit-oriented synchronous communication. SYNC byte-oriented protocols can be handled in several modes, allowing character synchronization with a 6-bit or 8-bit SYNC character (Monosync), any 12-bit or 16-bit SYNC pattern (Bisync), or with an external SYNC signal. Leading SYNC characters can be removed without interrupting the CPU.

Five- or 7-bit SYNC characters are detected with 8- or 16-bit patterns in the ESCC by overlapping the larger pattern across multiple incoming SYNC characters as shown in Figure 4.

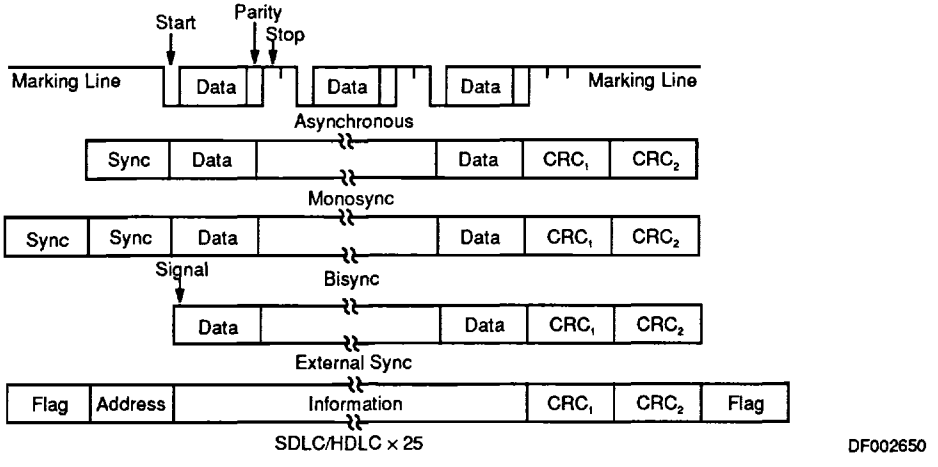



**Figure 3. SCC Protocols**

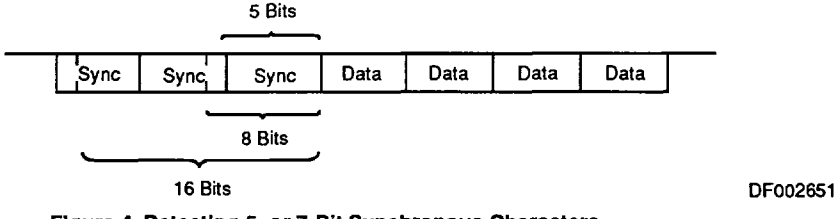



**Figure 4. Detecting 5- or 7-Bit Synchronous Characters**

CRC checking for Synchronous byte-oriented modes is delayed by one character time so that the CPU may disable CRC checking on specific characters. This 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 BISYNC and MONOSYNC modes. Users may preset the CRC generator and checker to all "1"s or all "0"s. The ESCC also provides a feature that automatically transmits CRC data when no other data are 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 are no data or CRC to send in SYNC modes, the transmitter inserts 6-, 8-, or 16-bit SYNC characters, regardless of the programmed character length.

The ESCC supports SYNC bit-oriented protocols, such as SDLC and HDLC, by performing automatic flag sending, zero bit insertion, and CRC generation. A special command can be used to abort a frame in transmission. At the end of a message, the ESCC 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 ESCC 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 $\overline{SYNC}$ 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 "0"s inserted by the transmitter during character assembly. CRC is also calculated and is automatically checked to validate frame transmission. At the end of transmission, the status of a received frame is available in the status registers. In SDLC mode, the ESCC must be programmed to use the SDLC CRC polynomial, but the generator and checker may be preset to all "1"s or all "0"s. The CRC 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 ESCC can be conveniently used under DMA control to provide high-speed reception or transmission. In reception, for example, the ESCC 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 ESCC 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 ESCC interrupt only on end-of-frame. This procedure allows all data to be transferred via the DMA.

**SDLC Loop Mode**

The ESCC 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 and any number of secondary stations. In SDLC Loop mode, the ESCC performs the functions of a secondary station while an ESCC operating in regular SDLC mode can act as a controller (Figure 5).

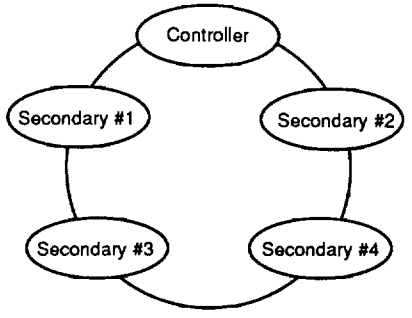



**Figure 5. An SDLC Loop**

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 1-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 ESCC. NRZ, NRZI, and FM coding may all be used in SDLC Loop mode.

**Baud Rate Generator**

Each channel in the ESCC contains a programmable baud rate generator. Each generator consists of two 8-bit time constant registers that form a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output producing a square wave. On start-up, the 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 $\overline{TRxC}$ pin, the output of the baud rate generator may be echoed out via the $\overline{TRxC}$ pin.

The following formula relates the time constant to the baud rate where PCLK or $\overline{RTxC}$ is the baud rate generator input frequency in Hz. The clock mode is X1, X16, X32, or X64 as selected in Write Register 4, bits D6 and D7. Synchronous operation modes should select X1 and Asynchronous should select X16, X32, or X64.

$$\text{Time Constant} = \left[\frac{\text{PCLK or RTxC Frequency}}{2\,(\text{Baud Rate})(\text{Clock Mode})}\right] - 2$$

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 given by Clock Mode/Clock Frequency.)

$$\text{baud rate} = \frac{1}{2\,(\text{Time Constant} + 2) \times (\text{BR Clock Period})}$$

| Time Constant Values for Standard Baud Rates at BR Clock = 3.9936 MHz | | | |
|---|---|---|---|
| Rate (Baud) | Time Constant (decimal/Hex notation) | | Error |
| 19200 | 102 | (0066) | 0 |
| 9600 | 206 | (00CE) | 0 |
| 7200 | 275 | (0113) | 0.12% |
| 4800 | 414 | (019E) | 0 |
| 3600 | 553 | (0229) | 0.06% |
| 2400 | 830 | (033E) | 0 |
| 2000 | 996 | (03E4) | 0.04% |
| 1800 | 1107 | (0453) | 0.03% |
| 1200 | 1662 | (067E) | 0 |
| 600 | 3326 | (0CFE) | 0 |
| 300 | 6654 | (19FE) | 0 |
| 150 | 13310 | (33FE) | 0 |
| 134.5 | 14844 | (39FC) | 0.0007% |
| 110 | 18151 | (46E7) | 0.0015% |
| 75 | 26622 | (67FE) | 0 |
| 50 | 39934 | (98FE) | 0 |

**Digital Phase-Locked Loop**

The ESCC 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 1/0 or 0/1). As long as no transitions are detected, the DPLL output will be free running and its input clock source will be divided by 32, producing an output clock without any phase jitter. Upon detecting a transition the DPLL will adjust its clock output (during the next counting cycle) by adding or subtracting a count of 1, thus producing a terminal count closer to the center of the bit cell. The adding or subtracting of a count of 1 will produce a phase jitter of ±5.63° on the output of the DPLL. Because the SCC's DPLL uses both edges of the incoming signal to compare with its clock source, the mark-space ratio (50%) of the incoming signal should not deviate by more than ±1.5% if proper locking is to occur.

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 15/16 counting transition.

The 32X clock for the DPLL can be programmed to come from either the $\overline{RTxC}$ input or the output of the baud rate generator. The DPLL output may be programmed to be echoed out of the SCC via the $\overline{TRxC}$ pin (if this pin is not being used as an input).

### Crystal Oscillator

When using a crystal oscillator to supply the receive or transmit clocks to a channel of the SCC, the user should:

1. Select a crystal oscillator which satisfies the following specifications:
   - 30 ppm @ 25°C
   - 50 ppm over temperature of –20° to 70°C
   - 5 ppm/yr aging
   - 5 mW drive level
2. Place crystal across $\overline{RTxC}$ and $\overline{SYNC}$ pins
3. Place 30 pF capacitors to ground from both $\overline{RTxC}$ and $\overline{SYNC}$ pins
4. Set bit $D_7$ of WR11 to "1."

### Data Encoding

The ESCC 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 $FM_1$ (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 ESCC 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."

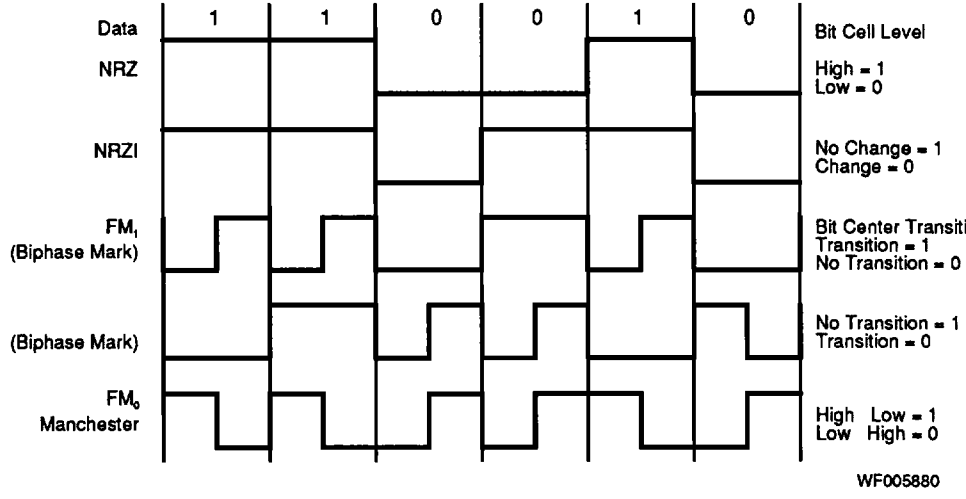



**Figure 6. Data Encoding Methods**

### Auto Echo and Local Loopback

The ESCC is capable of automatically echoing everything it receives. This feature is useful mainly in asynchronous modes but works in SYNC 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 $\overline{WAIT/REQUEST}$ on transmit.

The ESCC 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, SYNC and SDLC modes with NRZ, NRZI or FM coding of the data stream.

### I/O Interface Capabilities

The ESCC 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 ESCC 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 ESCC responds to an Interrupt Acknowledge signal (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 8 and 9).

To speed interrupt response time, the ESCC 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 ESCC (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.

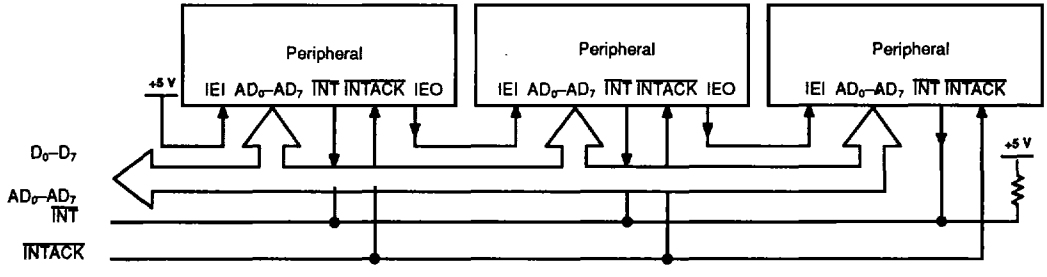
The other two bits are related to the Z-Bus interrupt priority chain (Figure 7). As a Z-Bus peripheral, the ESCC may request an interrupt only when no higher priority device is requesting one, for example, when IEI is HIGH. If the device in question requests an interrupt, it pulls down $\overline{INT}$. The CPU then responds with $\overline{INTACK}$, and the interrupting device places the vector on the A/D bus.

In the SCC, the IP bit signals a need for interrupt servicing. When an IP bit is set to "1" and the IEI input is HIGH, the $\overline{INT}$ output is pulled LOW, requesting an interrupt. In the ESCC, if the IE bit is set for an interrupt, 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 ESCC and external to the ESCC 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 ESCC 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 Receive, 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 Receive 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




AF002770

**Figure 7. Z-Bus Interrupt Schedule**

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, a zero count in the baud rate generator, 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 ESCC 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 $\overline{WAIT/REQUEST}$ output in conjunction with the Wait/Request bits in WR1. The $\overline{WAIT/REQUEST}$ 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 ESCC $\overline{REQUEST}$ output indicates that the ESCC 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 $\overline{DTR/REQUEST}$ can be used as the transmit request line, thus allowing full-duplex operation under DMA control.

## PROGRAMMING INFORMATION

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.

On the Z85C30, only four data registers (Read and Write for Channels A and B) are directly selected by a HIGH on the D/$\overline{C}$ input and the appropriate levels on the $\overline{RD}$, $\overline{WR}$ and A/$\overline{B}$ pins. All other registers are addressed indirectly by the content of Write Register 0 in conjunction with a LOW on the D/$\overline{C}$ input and the appropriate levels on the $\overline{RD}$, $\overline{WR}$ and A/$\overline{B}$ pins. If bit D3 in WR0 is 1 and bits 5 and 6 are 0, then bits 0, 1, 2 address the higher registers 8 through 15. If bits 4, 5, 6 contain a different code, bits 0, 1, 2 address the lower registers 0 through 7 as shown in Table 2.

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

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 WR0 are automatically cleared after this operation, so that WR0 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.

**Table 2. Register Addressing**

| D/$\overline{C}$ | "Point High" Code in WR0: | D2, D1, D0 in WR0: | | | 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 |

### Read Registers

The ESCC 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). In addition, if bit D2 of WR15 is set, RR6 and RR7 are available for providing frame status from the 10 × 19 bit Frame Status FIFO. Figure 8 shows the formats for each read register.

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

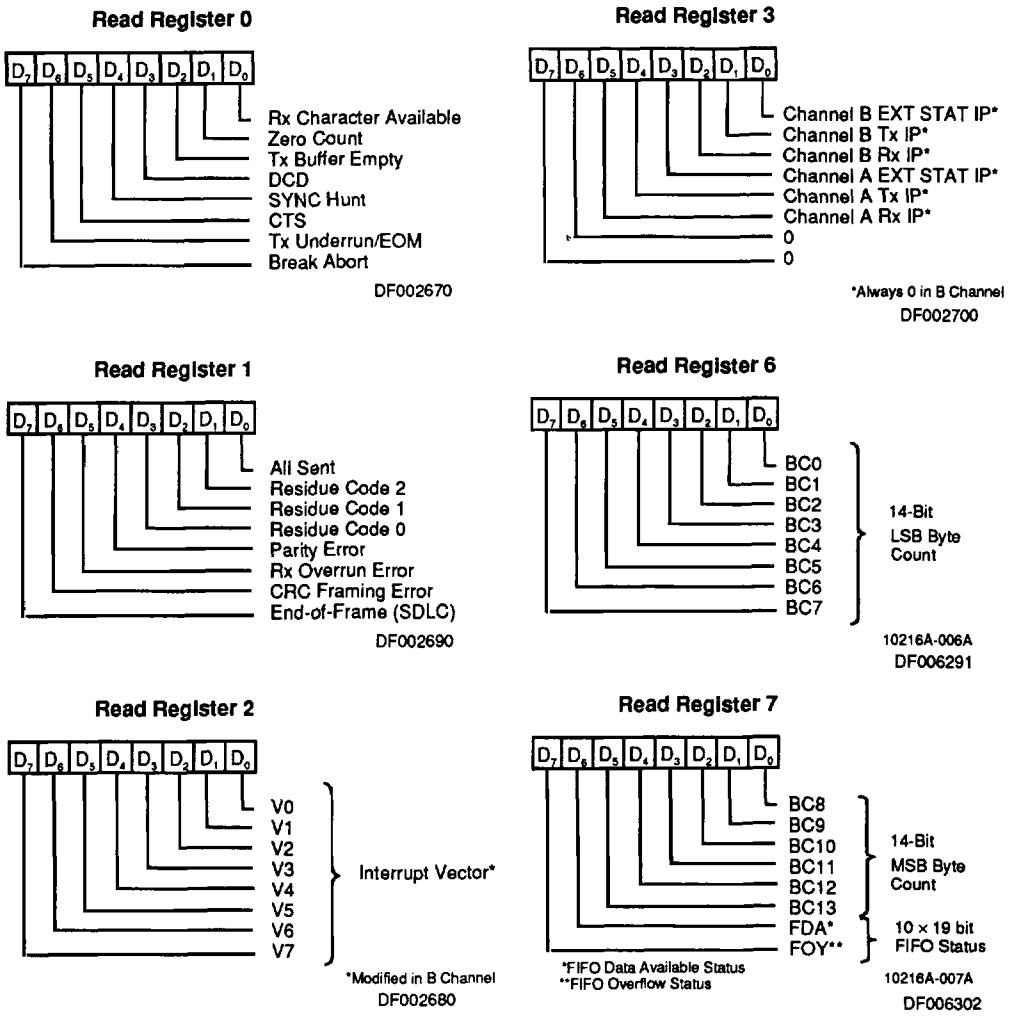



**Figure 8. Read Register Bit Functions**

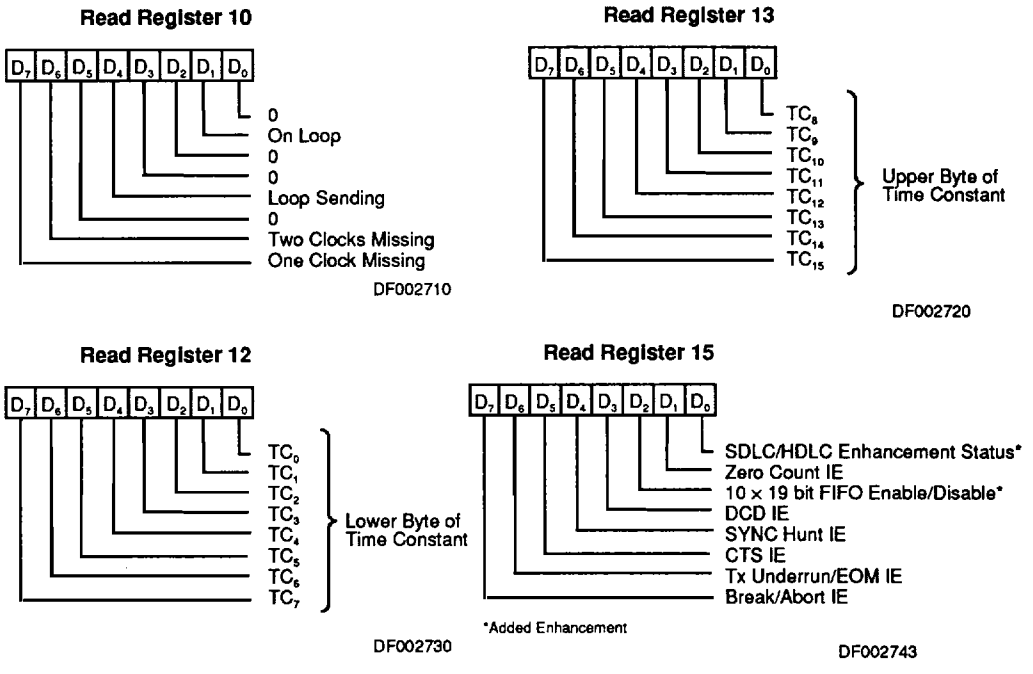



**Figure 8. Read Register Bit Functions (continued)**

### Write Registers

The ESCC 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. Two registers (WR2 and WR9) are shared by the two channels that can be accessed through either of them. WR2 contains the interrupt vector for both channels, while WR9 contains the interrupt control bits. In addition, if bit D0 of WR15 is set, write register seven prime (WR7') is available for programming additional SDLC/HDLC enhancements. When bit D0 of WR15 is set, executing a write to WR7 actually writes to WR7' to further enhance the functional "personality" of each channel. Figure 9 shows the format of each write register.

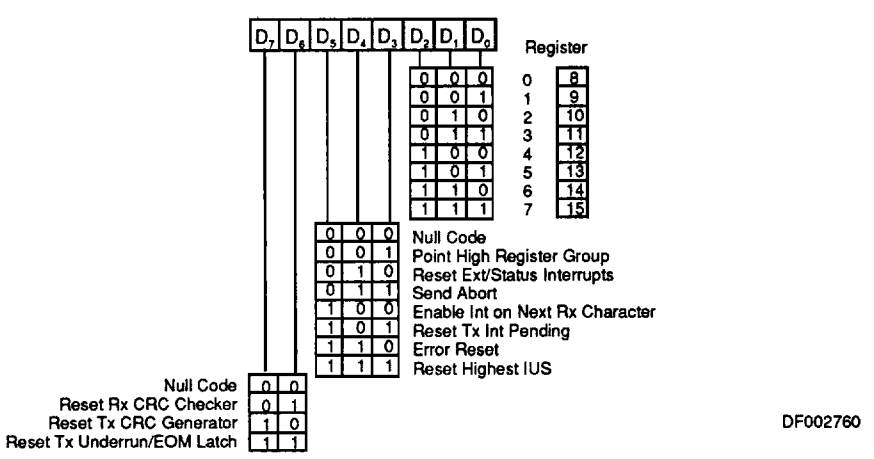



**Figure 9. Write Register Bit Functions**

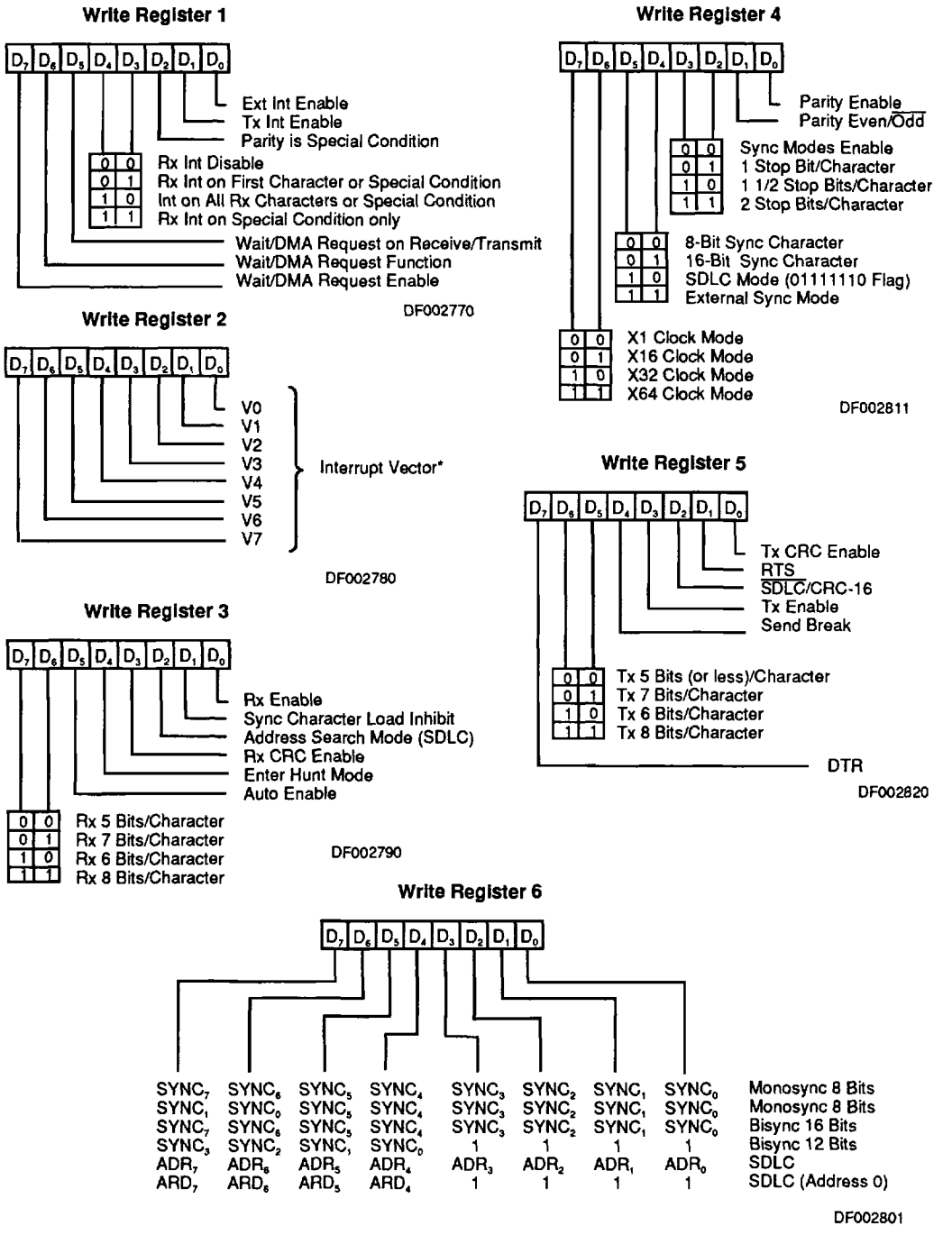
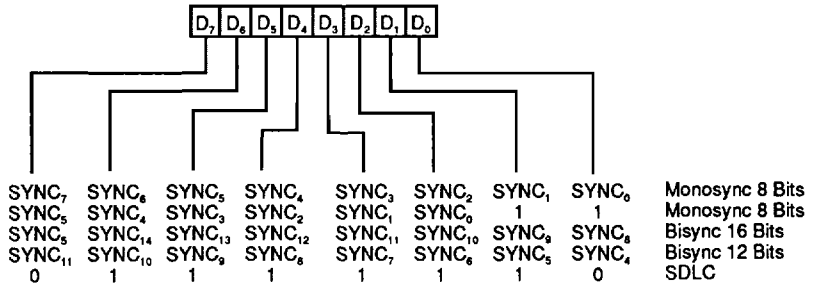


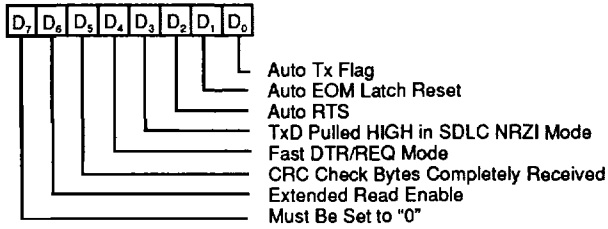

**Figure 9. Write Register Bit Functions (continued)**

### Write Register 7




DF002831

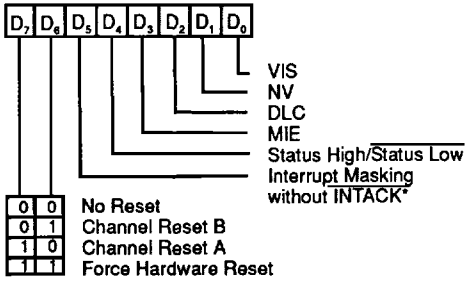
### Write Register 7′




10216A-008A
DF006331

1

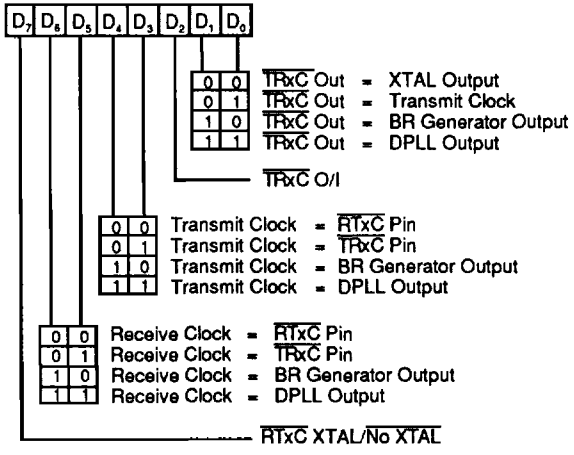
### Write Register 9




*Added Enhancement

DF002842

### Write Register 11




DF002850

**Figure 9. Write Register Bit Functions (continued)**

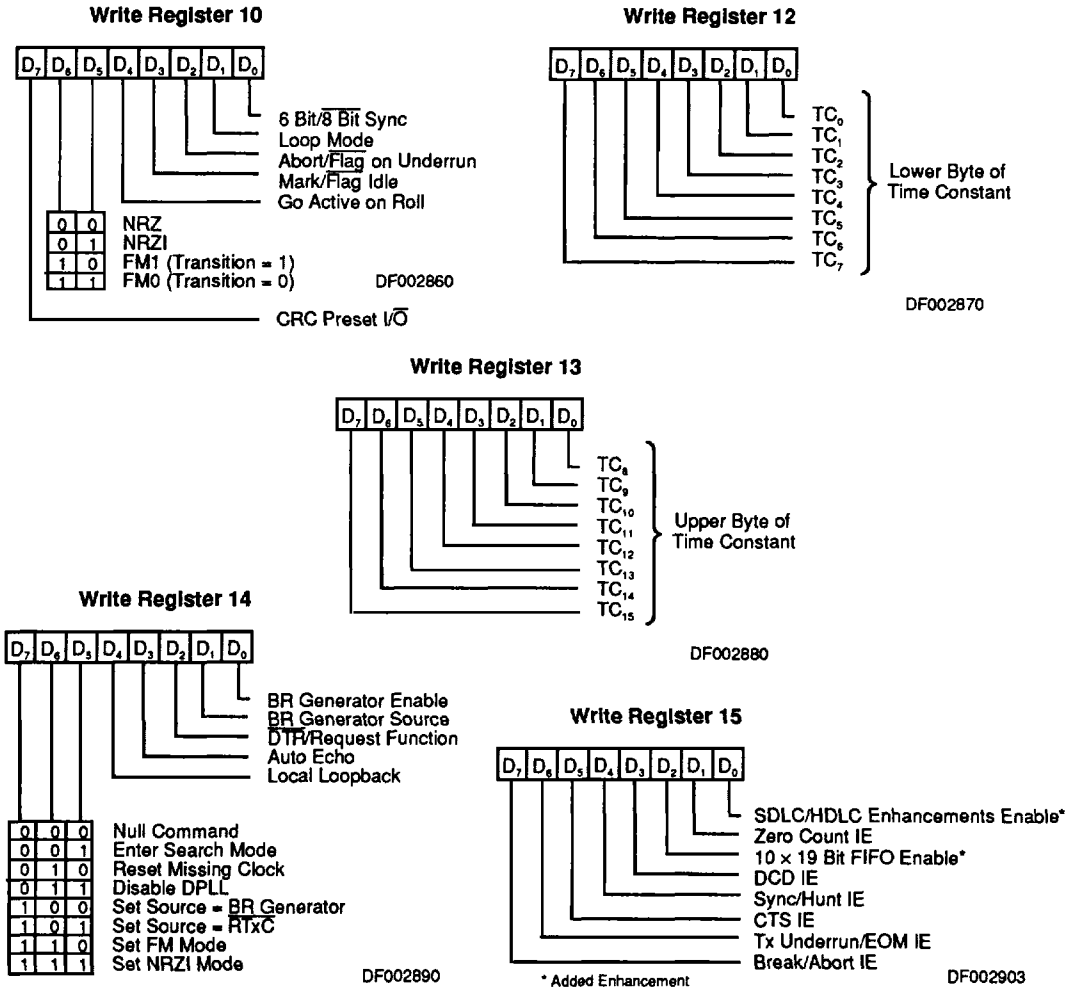



**Figure 9. Write Register Bit Functions (continued)**

### Z85C30 Timing

The ESCC 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 ESCC. The recovery time required for proper operation is specified from the falling edge of $\overline{WR}$ or $\overline{RD}$ in the first transaction involving the ESCC, to the falling edge of $\overline{WR}$ or $\overline{RD}$ in the second transaction involving the ESCC. This time must be at least 3 1/2 PCLK regardless of which register or channel is being accessed.

### Read Cycle Timing

Figure 10 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.

### Write Cycle Timing

Figure 11 illustrates Write Cycle timing. Addresses on A/$\overline{B}$ and D/$\overline{C}$ and the status on $\overline{INTACK}$ must remain stable throughout the cycle. If $\overline{CE}$ falls after $\overline{WR}$ falls or if it rises before $\overline{WR}$ rises, the effective $\overline{WR}$ is shortened. Data must be valid before the rising edge of $\overline{WR}$.

**Interrupt Acknowledge Cycle Timing**

Figure 12 illustrates Interrupt Acknowledge cycle timing. Between the time $\overline{INTACK}$ goes LOW and the falling edge of $\overline{RD}$, the internal and external IEI/IEO daisy chains settle. If there is an interrupt pending in the ESCC and IEI is HIGH when $\overline{RD}$ falls, the Acknowledge cycle is intended for the SCC. In this case, the ESCC may be programmed to respond to $\overline{RD}$ LOW by placing its interrupt vector on D0–D7 and it then sets the appropriate Interrupt-Under-Service latch internally.

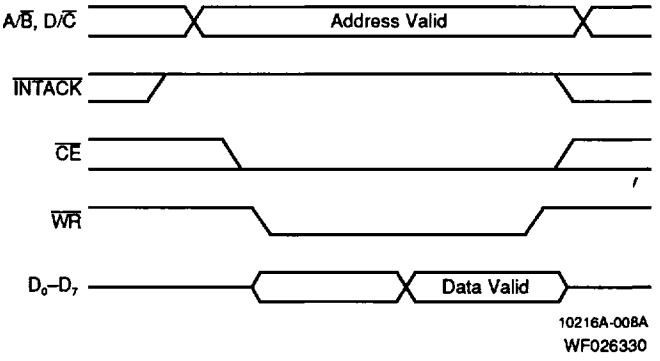



**Figure 10. Read Cycle Timing**

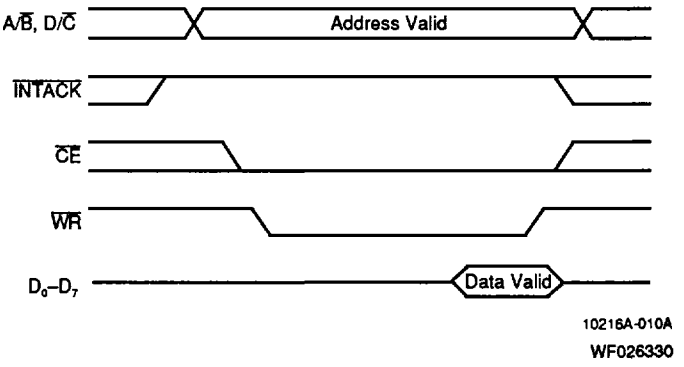



**Figure 11. Write Cycle Timing**

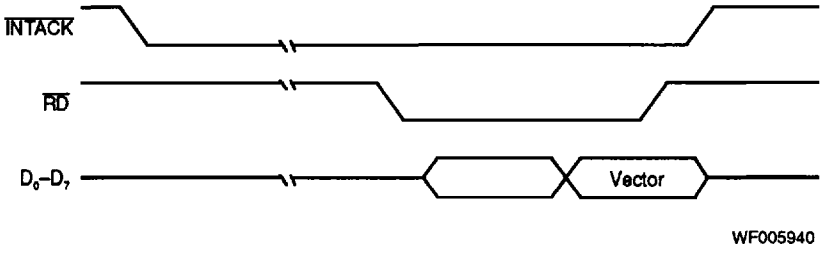



**Figure 12. Interrupt Acknowledge Cycle Timing**

### FIFO Enhancements

When used with a DMA controller, the Z85C30 Frame Status FIFO enhancement maximizes the ESCC's ability to receive high-speed back-to-back SDLC messages while minimizing frame overruns due to CPU latencies in responding to interrupts.

Additional logic was added to the industry-standard NMOS SCC consisting of a 10-deep by 19-bit status FIFO, a 14-bit receive byte counter, and control logic as shown in Figure 13. The 10 x 19 bit status FIFO is separate from the existing three-byte receive data and Error FIFOs.

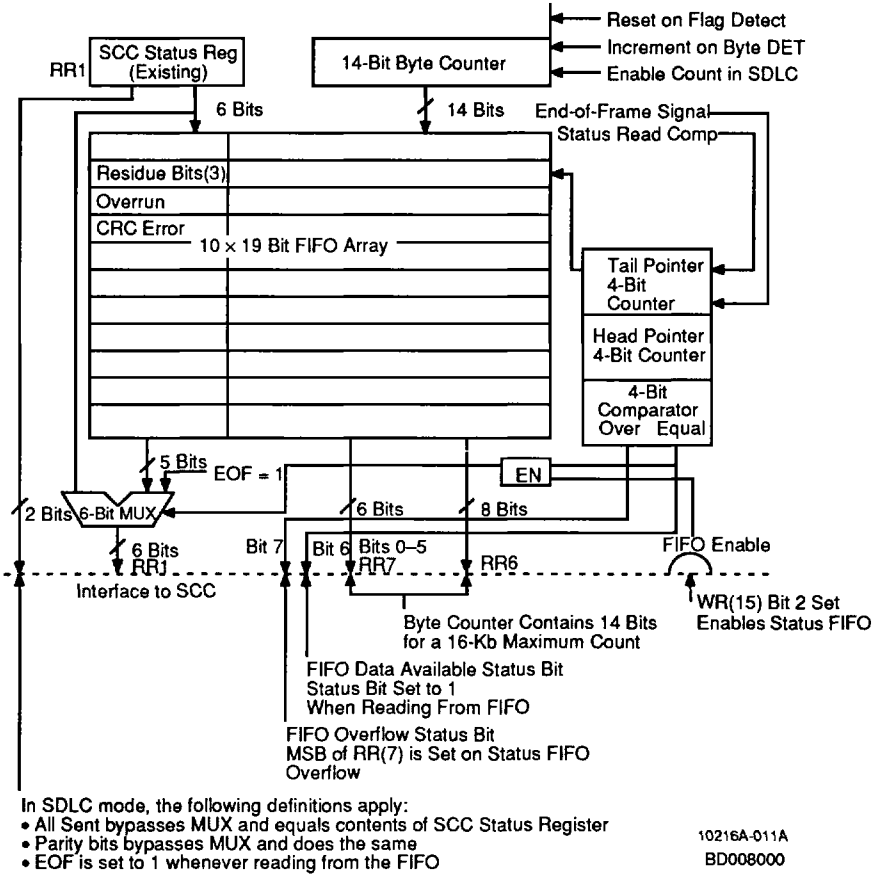



**Figure 13. SCC Status Register Modifications**

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

Summarizing the operation, data is received, assembled, and loaded into the three-byte receive FIFO before being transferred to memory by the DMA controller. When a flag is received at the end of an SDLC frame, the frame byte count from the 14-bit counter and five status bits are loaded into the status FIFO for verification by the CPU. The CRC checker is automatically reset in preparation for the next frame which can begin immediately.

Since the byte count and status are saved for each frame, the message integrity can be verified at a later time. Status information for up to 10 frames can be stored before a status FIFO overrun could occur.

If receive interrupts are enabled while the 10 x 19 FIFO is enabled, an SDLC end-of-frame special condition will not lock the three-byte Receive data FIFO. An SDLC end-of-frame still locks the three-byte Receive data FIFO in "Interrupt on first Receive Character or Special Condition" and "Interrupt on Special Condition Only" modes when the 10 x 19 FIFO is disabled. This feature allows the 10 x 19 SDLC FIFO to accept multiple SDLC frames without CPU intervention at the end of each frame.

### FIFO Detail

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

### Enable/Disable

This FIFO is implemented so that it is enabled when WR15 bit 2 is set and the ESCC is in the SDLC/HDLC mode, otherwise the status register contents bypass the FIFO and go directly to the bus interface (the FIFO pointer logic is reset either when disabled or via a channel or power-on reset). When the FIFO mode is disabled, the ESCC is completely downward-compatible with the NMOS Z8530. The FIFO mode is disabled on power-up (WR15 bit 2 is set to "0" on reset). The effects of backward compatibility on the register set are that RR4 is an image of RR0, RR5 is an image of RR1, RR6 is an image of RR2, and RR7 is an image of RR3. For the details of the added registers, refer to Figure 15. The status of the FIFO Enable signal can be obtained by reading RR15 bit 2. If the FIFO is enabled, the bit will be set to "1"; otherwise, it will be reset.

### Read Operation

When WR15 bit 2 is set and the FIFO is not empty, the next read to status register RR1 or the additional registers RR7 and RR6 will actually be from the FIFO. Reading status register RR1 causes one location of the FIFO to be emptied, so status should be read after reading the byte count, otherwise the count will be incorrect. Before the FIFO underflows, it is disabled. In this case, the multiplexer is switched to allow status to be read directly from the status register, and reads from RR7 and RR6 will contain bits that are undefined. Bit 6 of RR7 (FIFO Data Available) can be used to determine if status data is coming from the FIFO or directly from the status register, since it is set to "1" whenever the FIFO is not empty.

Because not all status bits are stored in the FIFO, the All Sent, Parity, and EOF bits will bypass the FIFO. The status bits sent through the FIFO will be Residue Bits (3), Overrun, and CRC Error.

The sequence for proper operation of the byte count and FIFO logic is to read the registers in the following order, RR7, RR6, and RR1 (reading RR6 is optional). Additional logic prevents the FIFO from being emptied by multiple reads from RR1. The read from RR7 latches the FIFO empty/full status bit (bit 6) and steers the status multiplexer to read from the SCC megacell instead of the status FIFO (since the status FIFO is empty). The read from RR1 allows an entry to be read from the FIFO (if the FIFO was empty, logic is added to prevent a FIFO underflow condition).

### Write Operation

When the end of an SDLC frame (EOF) has been received and the FIFO is enabled, the contents of the status and byte-count registers are loaded into the FIFO. The EOF signal is used to increment the FIFO. If the FIFO overflows, the MSB of RR7 (FIFO Overflow) is set to indicate the overflow. This bit and the FIFO control logic is reset by disabling and re-enabling the FIFO control bit (WR15 bit 2). For details of FIFO control timing during an SDLC frame, refer to Figure 14.



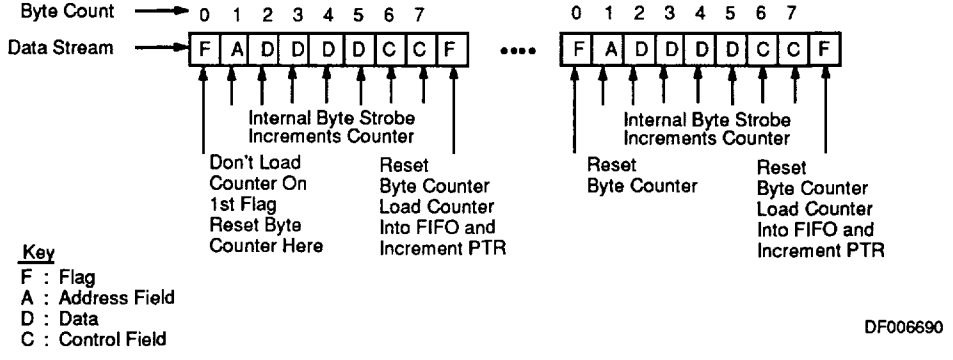



**Figure 14. SDLC Byte Counting Detail**

### Byte Counter Detail

The 14-bit byte counter allows for packets up to 16K bytes to be received. For a better understanding of its operation, refer to Figures 13 and 14.

### Enable

The byte counter is enabled when the SCC is in the SDLC/HDLC mode and WR15 bit 2 is set to "1."

### Reset

The byte counter is reset whenever an SDLC flag character is received. The reset is timed so that the contents of the byte counter are successfully written into the FIFO.

### Increment

The byte counter is incremented by writes to the data FIFO. The counter represents the number of bytes received by the SCC, rather than the number of bytes transferred from the SCC. (These counts may differ by up to the number of bytes in the receive data FIFO contained in the SCC.)

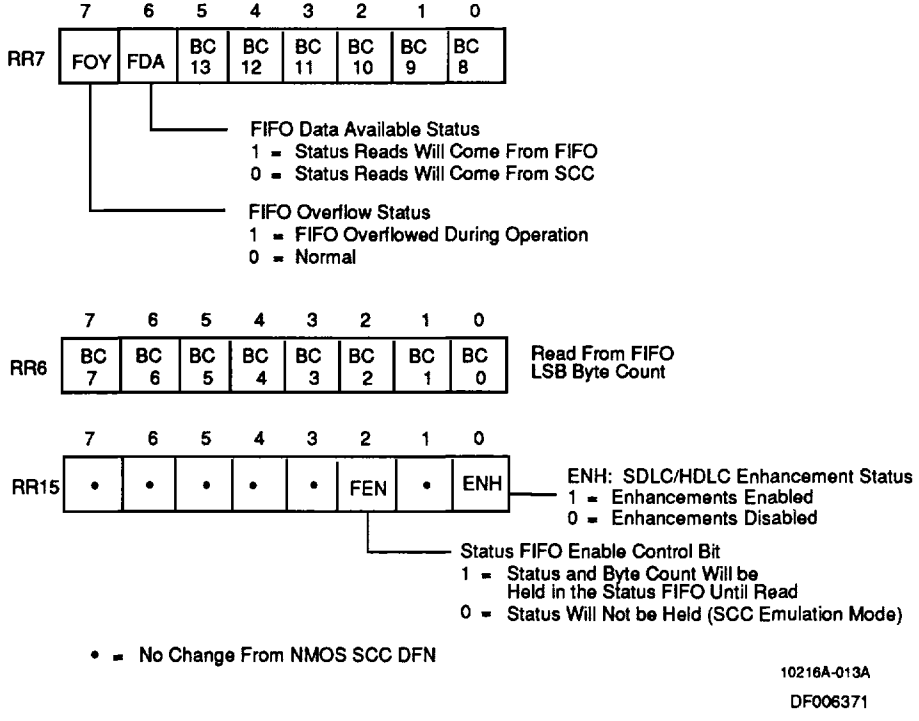



**Figure 15. SCC Additional Registers**

### Z85C30 SDLC/HDLC Enhancement Register Access

SDLC/HDLC enhancements on the Z85C30 are enabled or disabled via bits D2 or D0 in WR15. Bit D2 determines whether or not the 10 × 19 bit SDLC/HDLC frame status FIFO is enabled while bit D0 determines whether or not other enhancements are enabled via WR7'. Table 3 shows what functions on the Z85C30 are enabled when these bits are set.

**Table 3. Enhancement Options**

| WR15 Bit D2 10 × 19 Bit FIFO Enabled | WR15 Bit D0 SDLC/HDLC Enhancement Enabled | WR7' Bit D6 Extended Read Enabled | Functions Enabled |
|---|---|---|---|
| 1 | 0 | x | 10 × 19 bit FIFO enhancement enabled only |
| 0 | 1 | 0 | SDLC/HDLC enhancements enabled only |
| 0 | 1 | 1 | SDLC/HDLC enhancements enabled with extended read enabled |
| 1 | 1 | 0 | 10 × 19 bit FIFO and SDLC/HDLC enhancements enabled |
| 1 | 1 | 1 | 10 × 19 bit FIFO and SDLC/HDLC enhancements with extended read enabled |

When bit D2 of WR15 is set to "1," two additional registers (RR6 and RR7) per channel specific to the 10 × 19 bit Frame Status FIFO are made available. The Z85C30 register map when this function is enabled is shown in Table 4.

**Table 4. 10 × 19 Bit FIFO Enabled**

| A/$\overline{B}$ | $PNT_2$ | $PNT_1$ | $PNT_0$ | Write | Read |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | WR0B | RR0B |
| 0 | 0 | 0 | 1 | WR1B | RR1B |
| 0 | 0 | 1 | 0 | WR2 | RR2B |
| 0 | 0 | 1 | 1 | WR3B | RR3B |
| 0 | 1 | 0 | 0 | WR4B | (RR0B) |
| 0 | 1 | 0 | 1 | WR5B | (RR1B) |
| 0 | 1 | 1 | 0 | WR6B | RR6B |
| 0 | 1 | 1 | 1 | WR7B | RR7B |
| 1 | 0 | 0 | 0 | WR0A | RR0A |
| 1 | 0 | 0 | 1 | WR1A | RR1A |
| 1 | 0 | 1 | 0 | WR2 | RR2A |
| 1 | 0 | 1 | 1 | WR3A | RR3A |
| 1 | 1 | 0 | 0 | WR4A | (RR0A) |
| 1 | 1 | 0 | 1 | WR5A | (RR1A) |
| 1 | 1 | 1 | 0 | WR6A | RR6A |
| 1 | 1 | 1 | 1 | WR7A | RR7A |
| **With the Point High command:** | | | | | |
| 0 | 0 | 0 | 0 | WR8B | RR8B |
| 0 | 0 | 0 | 1 | WR9 | RR13B |
| 0 | 0 | 1 | 0 | WR10B | RR10B |
| 0 | 0 | 1 | 1 | WR11B | (RR15B) |
| 0 | 1 | 0 | 0 | WR12B | RR12B |
| 0 | 1 | 0 | 1 | WR13B | RR13B |
| 0 | 1 | 1 | 0 | WR14B | (RR10B) |
| 0 | 1 | 1 | 1 | WR15B | RR15B |
| 1 | 0 | 0 | 0 | WR8A | RR8A |
| 1 | 0 | 0 | 1 | WR9 | (RR13A) |
| 1 | 0 | 1 | 0 | WR10A | RR10A |
| 1 | 0 | 1 | 1 | WR11A | (RR15A) |
| 1 | 1 | 0 | 0 | WR12A | RR12A |
| 1 | 1 | 0 | 1 | WR13A | RR13A |
| 1 | 1 | 1 | 0 | WR14A | (RR10A) |
| 1 | 1 | 1 | 1 | WR15A | RR15A |

Bit D0 of WR15 determines whether or not other enhancements pertinent only to SDLC/HDLC Mode operation are available for programming via WR7′ as shown below. Write Register 7 prime (WR7′) can be written to when bit D0 of WR15 is set to "1." When this bit is set, writing to WR7 (flag register) actually writes to WR7′. If bit D6 of this register is set to "1," previously unreadable registers WR3, WR4, WR5, and WR10 are readable by the processor. In addition, WR7′ is also readable by having this bit set. WR3 is read when a bogus RR9 register is accessed during a read cycle. WR10 is read by accessing RR11, and WR7′ is accessed by executing a read to RR14. The Z85C30 register map with bit D0 of WR15 and bit D6 of WR7′ set is shown in Table 5.

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| Must Be Set to 0 | Ext. Read Enable | Rx comp. CRC | $\overline{DTR}/\overline{REQ}$ Fast Mode | Force TxD High | SDLC/HDLC Auto $\overline{RTS}$ Turnoff | SDLC/HDLC Auto EOM Reset | SDLC/HDLC Auto Tx Flag |

WR7′—SDLC/HDLC Programmable Enhancements*

*Note: Options 3, 4, 5, and 6 may be used regardless of whether SDLC/HDLC mode is selected.

Table 5. SDLC/HDLC Enhancements Enabled

| A/$\overline{B}$ | $PNT_2$ | $PNT_1$ | $PNT_0$ | Write | Read |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | WR0B | RR0B |
| 0 | 0 | 0 | 1 | WR1B | RR1B |
| 0 | 0 | 1 | 0 | WR2 | RR2B |
| 0 | 0 | 1 | 1 | WR3B | RR3B |
| 0 | 1 | 0 | 0 | WR4B | RR4B (WR4B) |
| 0 | 1 | 0 | 1 | WR5B | RR5B (WR5B) |
| 0 | 1 | 1 | 0 | WR6B | (RR2B) |
| 0 | 1 | 1 | 1 | WR7B | (RR3B) |
| 1 | 0 | 0 | 0 | WR0A | RR0A |
| 1 | 0 | 0 | 1 | WR1A | RR1A |
| 1 | 0 | 1 | 0 | WR2 | RR2A |
| 1 | 0 | 1 | 1 | WR3A | RR3A |
| 1 | 1 | 0 | 0 | WR4A | RR4A (WR4A) |
| 1 | 1 | 0 | 1 | WR5A | RR5A (WR5A) |
| 1 | 1 | 1 | 0 | WR6A | (RR2A) |
| 1 | 1 | 1 | 1 | WR7A | (RR3A) |
| | | | | **With the Point High command:** | |
| 0 | 0 | 0 | 0 | WR8B | RR8B |
| 0 | 0 | 0 | 1 | WR9 | RR9 (WR3B) |
| 0 | 0 | 1 | 0 | WR10B | RR10B |
| 0 | 0 | 1 | 1 | WR11B | RR11B (WR10B) |
| 0 | 1 | 0 | 0 | WR12B | RR12B |
| 0 | 1 | 0 | 1 | WR13B | RR13B |
| 0 | 1 | 1 | 0 | WR14B | RR14B (WR7'B) |
| 0 | 1 | 1 | 1 | WR15B | RR15B |
| 1 | 0 | 0 | 0 | WR8A | RR8A |
| 1 | 0 | 0 | 1 | WR9 | RR9A (WR3A) |
| 1 | 0 | 1 | 0 | WR10A | RR10A |
| 1 | 0 | 1 | 1 | WR11A | RR11A (WR10A) |
| 1 | 1 | 0 | 0 | WR12A | RR12A |
| 1 | 1 | 0 | 1 | WR13A | RR13A |
| 1 | 1 | 1 | 0 | WR14A | RR14A (WR7A) |
| 1 | 1 | 1 | 1 | WR15A | RR15A |

If both bits D0 and D2 of WR15 are set to "1" and D6 of WR7' is set to "1," then the Z85C30 register map is as shown in Table 6.

Table 6. SDLC/HDLC Enhancements and 10 × 19 Bit FIFO Enabled

| A/$\overline{B}$ | $PNT_2$ | $PNT_1$ | $PNT_0$ | Write | Read |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | WR0B | RR0B |
| 0 | 0 | 0 | 1 | WR1B | RR1B |
| 0 | 0 | 1 | 0 | WR2 | RR2B |
| 0 | 0 | 1 | 1 | WR3B | RR3B |
| 0 | 1 | 0 | 0 | WR4B | RR4B (WR4B) |
| 0 | 1 | 0 | 1 | WR5B | RR5B (WR5B) |
| 0 | 1 | 1 | 0 | WR6B | RR6B |
| 0 | 1 | 1 | 1 | WR7B | RR7B |
| 1 | 0 | 0 | 0 | WR0A | RR0A |
| 1 | 0 | 0 | 1 | WR1A | RR1A |
| 1 | 0 | 1 | 0 | WR2 | RR2A |
| 1 | 0 | 1 | 1 | WR3A | RR3A |
| 1 | 1 | 0 | 0 | WR4A | RR4A (WR4A) |
| 1 | 1 | 0 | 1 | WR5A | RR5A (WR5A) |
| 1 | 1 | 1 | 0 | WR6A | RR6A |
| 1 | 1 | 1 | 1 | WR7A | RR7A |
| **With the Point High command:** | | | | | |
| 0 | 0 | 0 | 0 | WR8B | RR8B |
| 0 | 0 | 0 | 1 | WR9 | RR9 (WR3B) |
| 0 | 0 | 1 | 0 | WR10B | RR10B |
| 0 | 0 | 1 | 1 | WR11B | RR11B (WR10B) |
| 0 | 1 | 0 | 0 | WR12B | RR12B |
| 0 | 1 | 0 | 1 | WR13B | RR13B |
| 0 | 1 | 1 | 0 | WR14B | RR14B (WR7'B) |
| 0 | 1 | 1 | 1 | WR15B | RR15B |
| 1 | 0 | 0 | 0 | WR8A | RR8A |
| 1 | 0 | 0 | 1 | WR9 | RR9A (WR3A) |
| 1 | 0 | 1 | 0 | WR10A | RR10A |
| 1 | 0 | 1 | 1 | WR11A | RR11A (WR10A) |
| 1 | 1 | 0 | 0 | WR12A | RR12A |
| 1 | 1 | 0 | 1 | WR13A | RR13A |
| 1 | 1 | 1 | 0 | WR14A | RR14A (WR7'A) |
| 1 | 1 | 1 | 1 | WR15A | RR15A |

**Auto RTS Reset**

On the CMOS ESCC, if bit D0 of WR15 and bit D2 of WR7' are set to "1" and the channel is in SDLC Mode, the $\overline{RTS}$ pin may be reset early in the Tx Underrun routine and the $\overline{RTS}$ pin will remain active until the last zero bit of the closing flag leaves the TxD pin as shown in Figure 16. Note that in order for this to function properly, bits D3 and D2 of WR10 must be set to "1" and "0" respectively.

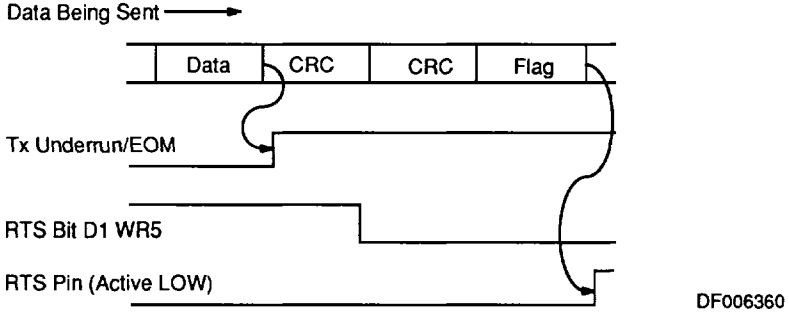



Figure 16. Auto $\overline{RTS}$ Reset Mode

### CRC Character Reception

#### NMOS Z8530H

On the NMOS 8530H , when the end-of-frame flag is detected, the contents of the Receive Shift Register are transferred to the Receive Data FIFO regardless of the number of bits accumulated. Because of the 3-bit delay between the Receive SYNC Register and Receive Shift Register, the last two bits of the CRC check character received are never transferred to the Receive Data FIFO. Thus, the received CRC characters are unavailable for use.

#### CMOS Z85C30

On the Z85C30, the option of being able to receive the complete CRC characters generated by the Transmitter is provided when both bit D0 of WR15 and bit D5 of WR7' are set to "1." When these two bits are set and an end-of-frame flag is detected, the last two bits of the CRC will be clocked into the Receive Shift Register before its contents are transferred to the Receive Data FIFO.The data-CRC boundary and CRC character bit formatsfor each Residue Code provided is shown in Figures17A through 17D for each character length selected.

Residue Code 012 001

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | $C_0$ | $C_1$ | $C_2$ |
| $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ |
| $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012 101

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | D | $C_0$ | $C_1$ |
| D | $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ |
| $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012 100

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | D | D | $C_0$ |
| D | D | $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ |
| $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012 010

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | D | D | D |
| D | D | D | $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
| $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ |
| $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |

Residue Code 012 110

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | D | D | D |
| D | D | D | D | $C_0$ | $C_1$ | $C_2$ | $C_3$ |
| $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ |
| $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |

10216A-015A
TB001170

**Figure 17A. 5 Bits/Character**

Residue Code 012/010

| D | D | D | D | D | D | $C_0$ | $C_1$ |
|---|---|---|---|---|---|---|---|
| $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ |
| $C_8$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012/110

| D | D | D | D | D | D | D | $C_0$ |
|---|---|---|---|---|---|---|---|
| D | $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ |
| $C_5$ | $6_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012/001

| D | D | D | D | D | D | D | D |
|---|---|---|---|---|---|---|---|
| D | D | $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ |
| $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012/101

| D | D | D | D | D | D | D | D |
|---|---|---|---|---|---|---|---|
| D | D | D | $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
| $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012/011

| D | D | D | D | D | D | D | D |
|---|---|---|---|---|---|---|---|
| D | D | D | D | $C_0$ | $C_1$ | $C_2$ | $C_3$ |
| $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012/100

| D | D | D | D | D | D | D | D |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | $C_0$ | $C_1$ | $C_2$ |
| $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ |
| $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |

10216A-016A
TB001180

**Figure 17B. 6 Bits/Character**

Residue Code 012/111

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | D | D | $C_0$ |
| $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ |
| $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012/100

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | D | D | D |
| D | $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ |
| $C_8$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012/010

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | D | D | D |
| D | D | $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ |
| $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012/110

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | D | D | D |
| D | D | D | $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
| $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012/001

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | D | D | D |
| D | D | D | D | $C_0$ | $C_1$ | $C_2$ | $C_3$ |
| $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012/101

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | D | D | D |
| D | D | D | D | D | $C_0$ | $C_1$ | $C_2$ |
| $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012/011

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | D | D | D |
| D | D | D | D | D | D | $C_0$ | $C_1$ |
| $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

10216A-017A
TB001190

**Figure 17C. 7 Bits/Character**

Residue Code 012 = 011 (No Residue)

| D | D | D | D | D | D | D | D |
|---|---|---|---|---|---|---|---|
| $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |
| | | | | | | | |

Residue Code 012 = 111 (1 Residue Bit)

| D | D | D | D | D | D | D | D |
|---|---|---|---|---|---|---|---|
| D | $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ |
| $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012 = 000 (2 Residue Bits)

| D | D | D | D | D | D | D | D |
|---|---|---|---|---|---|---|---|
| D | D | $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ |
| $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012 = 100 (3 Residue Bits)

| D | D | D | D | D | D | D | D |
|---|---|---|---|---|---|---|---|
| D | D | D | $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
| $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012 = 010 (4 Residue Bits)

| D | D | D | D | D | D | D | D |
|---|---|---|---|---|---|---|---|
| D | D | D | D | $C_0$ | $C_1$ | $C_2$ | $C_3$ |
| $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012 = 110 (5 Residue Bits)

| D | D | D | D | D | D | D | D |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | $C_0$ | $C_1$ | $C_2$ |
| $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ | $C_{10}$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012 = 001 (6 Residue Bits)

| D | D | D | D | D | D | D | D |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | D | $C_0$ | $C_1$ |
| $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ | $C_9$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

Residue Code 012 = 101 (7 Residue Bits)

| D | D | D | D | D | D | D | D |
|---|---|---|---|---|---|---|---|
| D | D | D | D | D | D | D | $C_0$ |
| $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ | $C_8$ |
| $C_8$ | $C_9$ | $C_{10}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ | $C_{14}$ | $C_{15}$ |
| | | | | | | | |

10216A-018A
TB001-200

**Figure 17D. 8 Bits/Character**

### Auto Flag Mode

On the NMOS Z8530H, if the transmitter is actively mark idling and a frame of data is ready to be transmitted, the MARK/FLAG idle bit must be set to "0" before data is written to WR8, otherwise the opening flag will not be sent properly. However, care must be exercised in doing this because the mark idle pattern (eight "1" bits) is transmitted eight bits at a time, and all eight bits must have transferred out the Transmit Shift Register before a flag may be loaded and sent. If data is written into the Transmit Buffer (WR8) before the flag is loaded into the Transmit Shift Register, the data character written to WR8 will supersede flag transmission and the opening flag will not be transmitted.

On the CMOS Z85C30, if bit D0 of WR15 is set to "1," and the ESCC is programmed for SDLC operation, an option is provided via bit D0 of WR7′ that eliminates this requirement. If bit D0 of WR7′ is set to "1" and a character is written to the Transmit Buffer while the Transmitter is mark idling, the Mark/Flag Idle bit in WR10 need not be reset to "0" in order to have the opening flag sent because the Transmitter will automatically send it before commencing to send data.

In addition, as long as bit D0 of WR15 and bit D1 of WR7′ are set to "1," the CRC transmit generator will be automatically preset to the initial state programmed by bit D7 of WR10 (so the Reset Tx CRC Generator command is also not necessary), and the Tx Underrun/EOM latch will be reset automatically on every new frame sent. This ensures that an opening flag and proper CRC generation and transmission will always be sent without processor intervention under varying bus latency conditions.

### Auto Transmit CRC Generator Preset

The NMOS Z8530H does not automatically preset the CRC generator prior to frame transmission. This must be done in software, usually during the initialization routine. This is accomplished by issuing the Reset Tx CRC Generator Command via WR0. For proper results, this command must be issued while the transmitter is enabled and idling and before any data are written to the Transmit Buffer.

In addition, if CRC is to be used, the transmit CRC generator must be enabled by setting bit D0 of WR5 to "1." CRC is normally calculated on all characters between opening and closing flags, so this bit should be set to "1" at initialization and never changed.

On the CMOS Z85C30, setting bit D0 of WR15 to "1" will cause the transmit CRC generator to be preset automatically every time an opening flag is sent, so the Reset Tx CRC Generator Command is not necessary.

### Auto Tx Underrun/EOM Latch Reset

On the ESCC, the transmission of the CRC check characters is controlled by the Transmit CRC Enable bit in WR5 (D0) and the Tx Underrun/EOM bit in RR0 (D6). However, if the Transmit Enable bit is set to "0" when a transmit underrun (i.e., both the Transmit Buffer and Transmit Shift Register go empty) occurs, the CRC check characters will not be sent regardless of the state of the Tx Underrun/EOM bit.

If the Transmit Enable bit is set to "1" when an underrun occurs, then the state of the Tx Underrun/EOM bit and the Abort/Flag on Underrun bit in WR10 (D2) determine the action taken by the Transmitter. The Abort/Flag on Underrun bit may be set or reset by the processor, whereas, the Tx Underrun/EOM bit is set by the Transmitter and can only be reset by the processor via the Reset Tx Underrun/EOM Command in WR0.

If the Tx Underrun/EOM bit is set to "1" when an underrun occurs, the Transmitter will close the frame by sending a flag; however, if this bit is set to "0," the frame data will be appended with either the accumulated CRC characters followed by a flag or an abort pattern followed by a flag, depending on the state of the Abort/Flag on Underrun bit in the WR10 (D2). In either case, after the closing flag is sent, the Transmitter will idle the transmission line as specified by the Mark/Flag Idle bit D3 in WR10.

Hence, if the CRC check characters are to be properly appended to a frame, the Abort/Flag on Underrun bit must be set to "0," and the Reset Tx Underrun/EOM Command must be issued after the first but before the last character is written to the Transmit Buffer. This will ensure that either an abort or the CRC will be transmitted if an underrun occurs. Normally, the Abort/Flag on Underrun bit in WR10 should be set to "1" around the same time that the Tx Underrun/EOM bit is reset so that an abort will be sent if the transmitter accidentally underruns, and then set to "0" near the end of the frame to allow the correct transmission of CRC.

On the Z85C30, if bit D0 of WR15 is set to "1," the option of having the Tx Underrun/EOM bit reset automatically at the start of every frame is provided via bit D1 of WR7′. This helps alleviate the software burden of having to respond within one character time when high-speed data are being sent.

### SDLC/HDLC NRZI Transmitter Disabling

On the NMOS Z8530H, if NRZI encoding is being used and the Transmitter is disabled, the state of the TxD pin will depend on the last bit sent. That is, the TxD pin may either idle in a Low or High state as shown in Figure 18.

On the CMOS Z85C30, an option is provided that allows setting the TxD pin High when operating in SDLC Mode with NRZI encoding enabled. If bit D0 of WR15 is set to "1," then bit D3 of WR7′ can be used to set the TxD pin High. Note that the operation of this bit is independent of the Tx Enable bit in WR5. The Tx Enable bit in WR5 is used to disable and enable the transmitter, whereas bit D3 of WR7′ acts as a pseudo transmitter disable and enable by just forcing the TxD pin High when set even though the transmitter may actually be mark or flag idling. Care must be used when setting this bit because any character being transmitted at the time this bit is set will be "chopped off," and data written to the Transmit Buffer while this bit is set will be lost.

When the transmit underrun occurs and the CRC and closing flag have been sent, bit D3 can be set to pull TxD High. When ready to start sending data again this bit must be reset to "0" before the first character is written to the Transmit Buffer. Note that resetting this bit causes the TxD pin to take whatever state the NRZI encoder is in at the time so synchronization at the Receiver may take longer because the first transition seen on the TxD pin may not coincide with a bit boundary. Note that in order for this to function properly, bits D3 and D2 of WR10 must be set to "1" and "0" respectively.

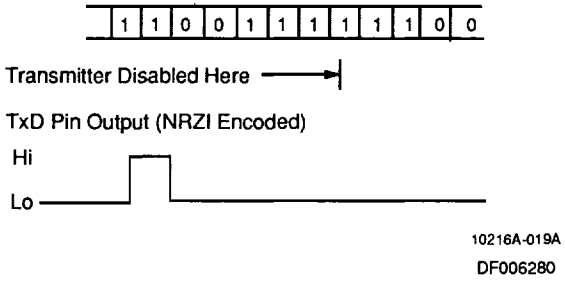



**Figure 18. Transmitter Disabling with NRZI Encoding**

### Interrupt Masking Without $\overline{\text{INTACK}}$

The NMOS Z8530H's ability to mask lower priority interrupts is done via the IUS bit. This bit is internal to the SCC and is not observable by the processor. Being able to automatically mask lower priority interrupts allows a modular approach to coding interrupt routines. However, using the masking capabilities of the NMOS SCC requires that the $\overline{\text{INTACK}}$ cycle be generated. In standalone applications, having to generate $\overline{\text{INTACK}}$ through external hardware in order to use this capability is an unnecessary expense.

On the CMOS Z85C30, if bit D5 in WR9 is set to "1," the $\overline{\text{INTACK}}$ cycle does not need to be generated in order to have the IUS bit set. This allows the user to respond to ESCC interrupt requests with a software acknowledgment through RR2. When bit D5 in WR9 is set and an interrupt occurs, a read to RR2 emulates a hardware Interrupt Acknowledge cycle as it functions in Vectored Mode. In this case the CPU must first read RR2 to determine the internal interrupt source and then jump to the appropriate interrupt routine. Reading RR2 sets the IUS bit for the highest priority IP. After the interrupting condition is cleared, the routine can then read RR3 to determine if any other IPs are set and clear them. At the end of the interrupt routine, a Reset IUS Command must be issued to unlock the internal daisy chain.

Since the CPU can acknowledge the ESCC of highest priority with a read of its RR2 interrupt vector, there is no need for an external daisy chain. IEI for all ESCC devices should be tied active HIGH. When acknowledging an ESCC interrupt request, the CPU must issue one read to RR2 per interrupt request. The modified interrupt vector can be read from Channel B, or the original vector stored in WR2 can be read from Channel A. Either action will produce the same internal actions on the IUS logic. Note that the No Vector and Vector Includes Status bits in WR9 are ignored when bit D5 in WR9 is set to "1."

### 2 Mb/s FM Data Transmission and Reception

The 16-MHz version of the CMOS Z85C30 (Z85C30-16) is capable of transmitting and receiving FM-encoded data at the rate of 2 Mb/s. This is accomplished by applying a 32-MHz clock to the $\overline{\text{RTxC}}$ pin and assigning this waveform to drive the Internal Digital Phase Locked Loop (DPLL) clock. This feature allows the user to send both clock and data information over the same line at 2 Mb/s and can eliminate external DPLLs required for high-speed NRZ data clock generation.

## ABSOLUTE MAXIMUM RATINGS

| | |
|---|---|
| Storage Temperature | –65 to +150°C |
| Voltage at any Pin Relative to $V_{SS}$ | –0.5 to +7.0 V |

*Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.*

## OPERATING RANGES

| | |
|---|---|
| **Commercial (C) Devices** | |
| Ambient Temperature ($T_A$) | 0 to +70°C |
| Supply Voltage ($V_{CC}$) | +5 V ± 10% |
| **Industrial (I) Devices** | |
| Ambient Temperature ($T_A$) | –40 to +85°C |
| Supply Voltage ($V_{CC}$) | 5 V ±10% |
| **Military (M) Devices** | |
| Case Temperature ($T_C$) | –55° to 125° |
| Supply Voltage ($V_{CC}$) | 5 V ±10% |

*Operating ranges define those limits between which the functionality of the device is guaranteed.*

## DC CHARACTERISTICS over operating range

| Parameter Symbol | Parameter Description | Test Conditions | Min. | Max. | Unit |
|---|---|---|---|---|---|
| $V_{IH}$ | Input HIGH Voltage | Commercial | 2.2 | $V_{CC}$ +0.3* | V |
| $V_{IL}$ | Input LOW Voltage | | –0.3* | 0.8 | V |
| $V_{OH1}$ | Output HIGH Voltage | $I_{OH}$ = –1.6 mA | 2.4 | | V |
| $V_{OH2}$ | Output HIGH Voltage | $I_{OH}$ = –250 µA | $V_{CC}$–0.8 | | V |
| $V_{OL}$ | Output LOW Voltage | $I_{OL}$ = +2.0 mA | | 0.4 | V |
| $I_{IL}$ | Input Leakage | 0.4 V ≤ $V_{IN}$ ≤ 2.4 V | | ±10.0 | µA |
| $I_{OL}$ | Output Leakage | 0.4 V ≤ $V_{OUT}$ ≤ 2.4 V | | ±10.0 | µA |
| $I_{CC1}$ | $V_{CC}$ Supply Current | 8.192 MHz | | 18 | mA |
| | | 10 MHz Inputs at | | 18 | mA |
| | | 12 MHz voltage rails, | | 22 | mA |
| | | 16.384 MHz output unloaded | | 22 | mA |
| | | 20 MHz | | 30 | mA |
| $C_{IN}$ | Input Capacitance | Unmeasured pins returned | | 10 | pF |
| $C_{OUT}$ | Output Capacitance | to ground ! = 1 MHz over | | 15 | pF |
| $C_{IO}$ | Bidirectional Capacitance | specified temperature range | | 20 | pF |

* $V_{IH}$ Max. and $V_{IL}$ Min. not tested. Guaranteed by design.

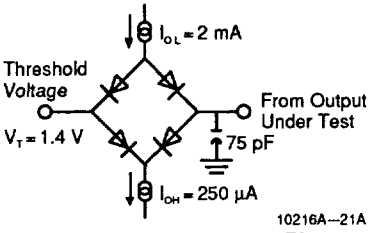
### Standard Test Conditions

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:

+4.5 V ≤ $V_{CC}$ ≤ +5.5 V
GND = 0 V
0°C ≤ $T_A$ ≤ 70°C

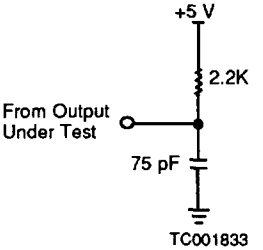
## SWITCHING TEST CIRCUITS

**Standard Test Dynamic Load Circuit**




10216A–21A
TC004800

**Open-Drain Test Load**




TC001833

## SWITCHING CHARACTERISTICS over COMMERCIAL operating range

### General Timing (see Figure 19)

| No. | Parameter Symbol | Parameter Description | 8.192 MHz Min. | 8.192 MHz Max. | 10 MHz Min. | 10 MHz Max. | 12.5 MHz Min. | 12.5 MHz Max. | Unit |
|---|---|---|---|---|---|---|---|---|---|
| 1 | TdPC(REQ) | PCLK ↓ to $\overline{W/REQ}$ Valid Delay | | 250 | | 150 | | 120 | nsec |
| 2 | TdPC(W) | PCLK ↓ to Wait Inactive Delay | | 350 | | 250 | | 220 | nsec |
| 3 | TsRXC(PC) | $\overline{RxC}$ ↑ to PCLK ↑ Setup Time (Notes 1, 4 & 8) | NA | NA | NA | NA | NA | NA | |
| 4 | TsRXD(RXCr) | RxD to $\overline{RxC}$ ↑ Setup Time (XI Mode) (Note 1) | 0 | | 0 | | 0 | | nsec |
| 5 | ThRXD(RXCr) | RxD to $\overline{RxC}$ ↑ Hold Time (XI Mode) (Note 1) | 150 | | 125 | | 100 | | nsec |
| 6 | TsRXD(RXCf) | RxD to $\overline{RxC}$ ↓ Setup Time (XI Mode) (Notes 1, 5) | 0 | | 0 | | 0 | | nsec |
| 7 | ThRXD(RXCf) | RxD to $\overline{RxC}$ ↓ Hold Time (XI Mode) (Notes 1, 5) | 150 | | 125 | | 100 | | nsec |
| 8 | TsSY(RXC) | $\overline{SYNC}$ to $\overline{RxC}$ ↑ Setup Time (Note 1) | –200 | | –150 | | –125 | | nsec |
| 9 | ThSY(RXC) | $\overline{SYNC}$ to $\overline{RxC}$ ↑ Hold Time (Note 1) | 5TcPC | | 5TcPC | | 5TcPC | | nsec |
| 10 | TsTXC(PC) | $\overline{TxC}$ ↓ to PCLK ↑ Setup Time (Notes 2, 4 & 8) | NA | | NA | | NA | | |
| 11 | TdTXCf(TXD) | $\overline{TxC}$ ↓ to TxD Delay (XI Mode) (Note 2) | | 200 | | 150 | | 130 | nsec |
| 12 | TdTXCr(TXD) | $\overline{TxC}$ ↑ to TxD Delay (XI Mode) (Notes 2, 5) | | 200 | | 150 | | 130 | nsec |
| 13 | TdTXD(TRX) | TxD to $\overline{TRxC}$ Delay (Send Clock Echo) | | 200 | | 140 | | 120 | nsec |
| 14a | TwRTXh | $\overline{RTxC}$ HIGH Width (Note 6) | 150 | | 120 | | 100 | | nsec |
| 14b | TwRTxh(E) | $\overline{RTxC}$ HIGH Width (Note 9) | 50 | | 40 | | 34 | | nsec |
| 15a | TwRTXl | $\overline{RTxC}$ LOW Width (Note 6) | 150 | | 120 | | 100 | | nsec |
| 15b | TwRTXl(E) | $\overline{RTxC}$ LOW Width (Note 9) | 50 | | 40 | | 34 | | nsec |
| 16a | TcRTX | $\overline{RTxC}$ Cycle Time (Notes 6, 7) | 488 | | 400 | | 320 | | nsec |
| 16b | TcRTx(E) | $\overline{RTxC}$ Cycle Time (Note 9) | 125 | | 100 | | 80 | | nsec |
| 17 | TcRTXX | Crystal Oscillator Period (Note 3) | 125 | 1000 | 100 | 1000 | 80 | 1000 | nsec |
| 18 | TwTRXh | $\overline{TRxC}$ HIGH Width (Note 6) | 150 | | 120 | | 100 | | nsec |
| 19 | TwTRXl | $\overline{TRxC}$ LOW Width (Note 6) | 150 | | 120 | | 100 | | nsec |
| 20 | TcTRX | $\overline{TRxC}$ Cycle Time (Notes 6, 7) | 488 | | 400 | | 320 | | nsec |
| 21 | TwEXT | $\overline{DCD}$ or $\overline{CTS}$ Pulse Width | 200 | | 120 | | 100 | | nsec |
| 22 | TwSY | $\overline{SYNC}$ Pulse Width | 200 | | 120 | | 100 | | nsec |

Notes:
1. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
2. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
3. Both $\overline{RTxC}$ and $\overline{SYNC}$ 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 $\overline{RxC}$ and PCLK or $\overline{TxC}$ and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver, DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.
8. External PCLK to $\overline{RxC}$ or $\overline{TxC}$ synchronization requirement eliminated for PCLK divide-by-four operation.

   $\overline{TRxC}$ and $\overline{RTxC}$ rise and fall times are identical to PCLK. Reference timing specs Tfpc and Trpc.

   Tx and Rx input clock slow rates should be kept to a maximum of 30 nsec. All parameters related to input CLK edges should be referenced at the point at which the transition begins or ends, whichever is worst case.
9. ENHANCED FEATURE— $\overline{RTxC}$ used as input to internal DPLL only.

## SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

**General Timing (see Figure 19)**

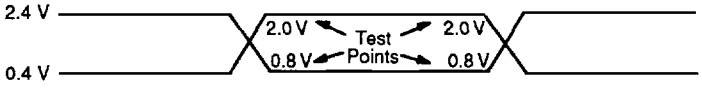
| No. | Parameter Symbol | Parameter Description | 16.384 MHz Min. | 16.384 MHz Max. | 20 MHz Min. | 20 MHz Max. | Unit |
|---|---|---|---|---|---|---|---|
| 1 | TdPC(REQ) | PCLK ↓ to W/REQ Valid Delay | | 80 | | 70 | nsec |
| 2 | TdPC(W) | PCLK ↓ to Wait Inactive Delay | | 180 | | 170 | nsec |
| 3 | TsRXC(PC) | RxC ↑ to PCLK ↑ Setup Time (Notes 1, 4 & 8) | NA | NA | NA | NA | |
| 4 | TsRXD(RXCr) | RxD to RxC ↑ Setup Time (XI Mode) (Note 1) | 0 | | 0 | | nsec |
| 5 | ThRXD(RXCr) | RxD to RxC ↑ Hold Time (XI Mode) (Note 1) | 50 | | 45 | | nsec |
| 6 | TsRXD(RXCf) | RxD to RxC ↓ Setup Time (XI Mode) (Notes 1, 5) | 0 | | 0 | | nsec |
| 7 | ThRXD(RXCf) | RxD to RxC ↓ Hold Time (XI Mode) (Notes 1, 5) | 50 | | 45 | | nsec |
| 8 | TsSY(RXC) | SYNC to RxC ↑ Setup Time (Note 1) | −100 | | −90 | | nsec |
| 9 | ThSY(RXC) | SYNC to RxC ↑ Hold Time (Note 1) | 5TcPc | | 5TcPc | | nsec |
| 10 | TsTXC(PC) | TxC ↓ to PCLK ↑ Setup Time (Notes 2, 4 & 8) | NA | | NA | | |
| 11 | TdTXCf(TXD) | TxC ↓ to TxD Delay (XI Mode) (Note 2) | | 80 | | 70 | nsec |
| 12 | TdTXCr(TXD) | TxC ↑ to TxD Delay (XI Mode) (Notes 2, 5) | | 80 | | 70 | nsec |
| 13 | TdTXD(TRX) | TxD to TRxC Delay (Send Clock Echo) | | 80 | | 70 | nsec |
| 14a | TwRTXh | RTxC HIGH Width (Note 6) | 80 | | 70 | | nsec |
| 14b | TwRTxh(E) | RTxC HIGH Width (Note 9) | 15.6 | | 15.6 | | nsec |
| 15a | TwRTXl | RTxC LOW Width (Note 6) | 80 | | 70 | | nsec |
| 15b | TwRTXl(E) | RTxC LOW Width (Note 9) | 15.6 | | 15.6 | | nsec |
| 16a | TcRTX | RTxC Cycle Time (Notes 6, 7) | 244 | | 200 | | nsec |
| 16b | TcRTx(E) | RTxC Cycle Time (Note 9) | 31.25 | | 31.25 | | nsec |
| 17 | TcRTXX | Crystal Oscillator Period (Note 3) | 62 | 1000 | 50 | 1000 | nsec |
| 18 | TwTRXh | TRxC HIGH Width (Note 6) | 80 | | 70 | | nsec |
| 19 | TwTRXl | TRxC LOW Width (Note 6) | 80 | | 70 | | nsec |
| 20 | TcTRX | TRxC Cycle Time (Notes 6, 7) | 244 | | 200 | | nsec |
| 21 | TwEXT | DCD or CTS Pulse Width | 70 | | 65 | | nsec |
| 22 | TwSY | SYNC Pulse Width | 70 | | 65 | | nsec |

Notes:
1. RxC is RTxC or TRxC, whichever is supplying the receive clock.
2. TxC is TRxC or RTxC, whichever is supplying the transmit clock.
3. Both RTxC and 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 RxC and PCLK or TxC and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver, DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.
8. External PCLK to RxC or TxC synchronization requirement eliminated for PCLK divide-by-four operation.

   TRxC and RTxC rise and fall times are identical to PCLK. Reference timing specs Tfpc and Trpc.

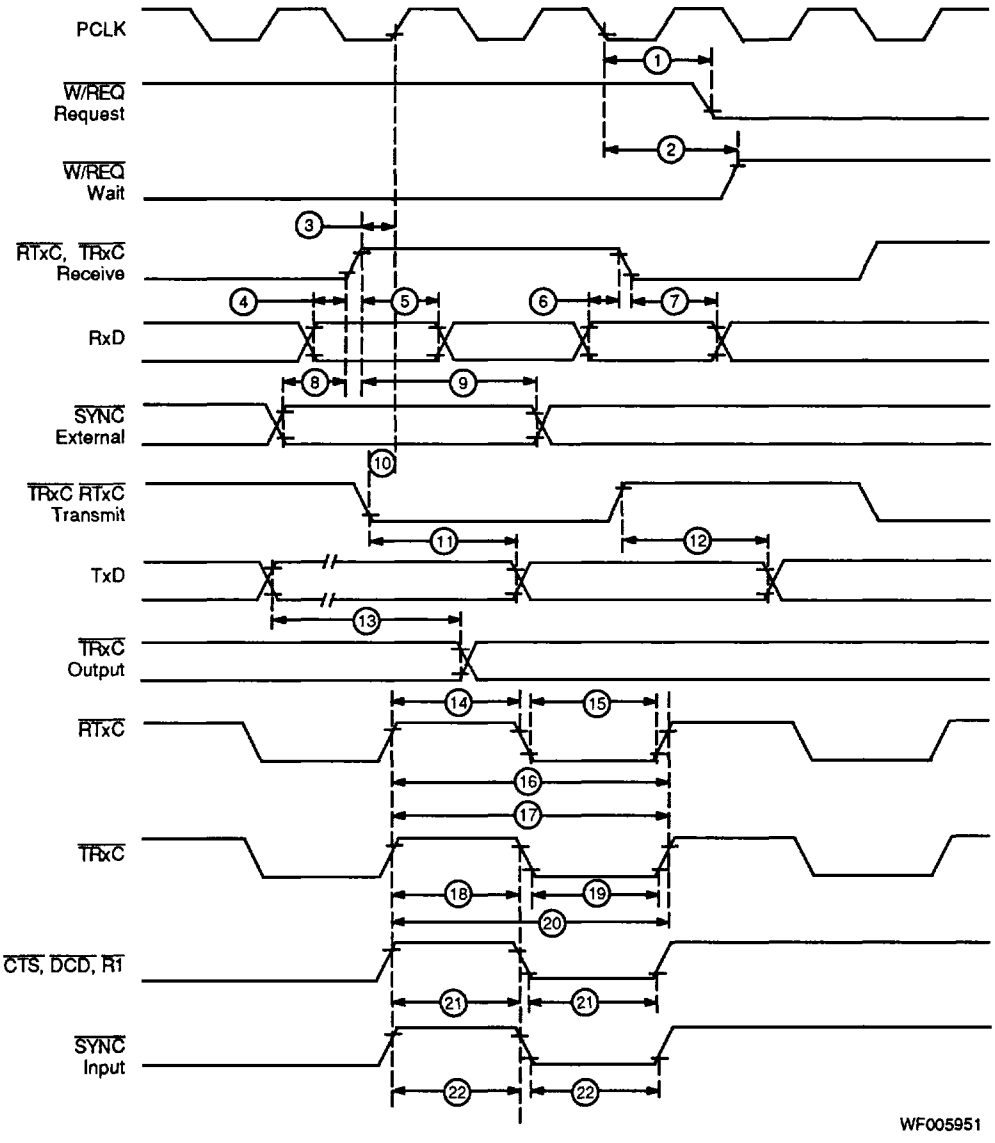
   Tx and Rx input clock slow rates should be kept to a maximum of 30 nsec. All parameters related to input CLK edges should be referenced at the point at which the transition begins or ends, whichever is worst case.
9. ENHANCED FEATURE— RTxC used as input to internal DPLL only.

## SWITCHING TEST INPUT/OUTPUT WAVEFORM




WR006354

AC testing: Inputs are driven at 2.4 V for a logic "1" and 0.4 V for a logic "0."
Timing measurements are made at 2.0 V for a logic "1" and 0.8 V for logic "0."




WF005951

**Figure 19. General Timing**

## SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

**System Timing (see Figure 20)**

| No. | Parameter Symbol | Parameter Description | 8.192 MHz Min. | 8.192 MHz Max. | 10 MHz Min. | 10 MHz Max. | 12.5 MHz Min. | 12.5 MHz Max. | Unit |
|---|---|---|---|---|---|---|---|---|---|
| 1 | TdRXC(REQ) | $\overline{RXC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2) | 8 | 12 | 8 | 12 | 8 | 12 | TcPc |
| 2 | TdRXC(W) | $\overline{RXC}$ ↑ to Wait Inactive Delay (Notes 1, 2) | 8 | 14 | 8 | 14 | 8 | 14 | TcPc |
| 3 | TdRXC(SY) | $\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2) | 4 | 7 | 4 | 7 | 4 | 7 | TcPc |
| 4 | TdRXC(INT) | $\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2) | 10 | 16 | 10 | 16 | 10 | 16 | TcPc |
| 5 | TdTXC(REQ) | $\overline{TxC}$ ↑ to $\overline{W/REQ}$ Valid Delay (Note 3) | 5 | 8 | 5 | 8 | 5 | 8 | TcPc |
| 6 | TdTXC(W) | $\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3) | 5 | 11 | 5 | 11 | 5 | 11 | TcPc |
| 7a | TdTXC(DRQ) | $\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3) | 4 | 7 | 4 | 7 | 4 | 7 | TcPc |
| 7b | TdTXC(EDRQ) | $\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Notes 3, 4) | 5 | 8 | 5 | 8 | 5 | 8 | TcPc |
| 8 | TdTXC(INT) | $\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3) | 6 | 10 | 6 | 10 | 6 | 10 | TcPc |
| 9 | TdSY(INT) | $\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1) | 2 | 6 | 2 | 6 | 2 | 6 | TcPc |
| 10 | TdEXT(INT) | $\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1) | 2 | 6 | 2 | 6 | 2 | 6 | TcPc |

| No. | Parameter Symbol | Parameter Description | 16.384 MHz Min. | 16.384 MHz Max. | 20 MHz Min. | 20 MHz Max. | Unit |
|---|---|---|---|---|---|---|---|
| 1 | TdRXC(REQ) | $\overline{RXC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2) | 8 | 12 | 8 | 12 | TcPc |
| 2 | TdRXC(W) | $\overline{RXC}$ ↑ to Wait Inactive Delay (Notes 1, 2) | 8 | 14 | 8 | 14 | TcPc |
| 3 | TdRXC(SY) | $\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2) | 4 | 7 | 4 | 7 | TcPc |
| 4 | TdRXC(INT) | $\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2) | 10 | 16 | 10 | 16 | TcPc |
| 5 | TdTXC(REQ) | $\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3) | 5 | 8 | 5 | 8 | TcPc |
| 6 | TdTXC(W) | $\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3) | 5 | 11 | 5 | 11 | TcPc |
| 7a | TdTXC(DRQ) | $\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3) | 4 | 7 | 4 | 7 | TcPc |
| 7b | TdTXC(EDRQ) | $\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Notes 3, 4) | 5 | 8 | 5 | 8 | TcPc |
| 8 | TdTXC(INT) | $\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3) | 6 | 10 | 6 | 10 | TcPc |
| 9 | TdSY(INT) | $\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1) | 2 | 6 | 2 | 6 | TcPc |
| 10 | TdEXT(INT) | $\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1) | 2 | 6 | 2 | 6 | TcPc |

Notes: 1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
4. Parameter applies to Enhanced Request mode only.

## SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

**Read and Write Timing (see Figure 21)**

| No. | Parameter Symbol | Parameter Description | 8.192 MHz Min. | 8.192 MHz Max. | 10 MHz Min. | 10 MHz Max. | 12.5 MHz Min. | 12.5 MHz Max. | Unit |
|---|---|---|---|---|---|---|---|---|---|
| 1 | TwPCl | PCLK LOW Width | 50 | 2000 | 40 | 2000 | 34 | 2000 | nsec |
| 2 | TwPCh | PCLK HIGH Width | 50 | 2000 | 40 | 2000 | 34 | 2000 | nsec |
| 3 | TfPC | PCLK Fall Time | | 15 | | 12 | | 10 | nsec |
| 4 | TrPC | PCLK Rise Time | | 15 | | 12 | | 10 | nsec |
| 5 | TcPC | PCLK Cycle Time | 122 | 4000 | 100 | 4000 | 80 | 4000 | nsec |
| 6 | TsA(WR) | Address to $\overline{WR}\downarrow$ Setup Time | 70 | | 50 | | 45 | | nsec |
| 7 | ThA(WR) | Address to $\overline{WR}\uparrow$ Hold Time | 0 | | 0 | | 0 | | nsec |
| 8 | TsA(RD) | Address to $\overline{RD}\downarrow$ Setup Time | 70 | | 50 | | 45 | | nsec |
| 9 | ThA(RD) | Address to $\overline{RD}\uparrow$ Hold Time | 0 | | 0 | | 0 | | nsec |
| 10 | TsIA(PC) | $\overline{INTACK}$ to PCLK $\uparrow$ Setup Time | 20 | | 20 | | 15 | | nsec |
| 11 | TsIA(WR) | $\overline{INTACK}$ to $\overline{WR}\downarrow$ Setup Time (Note 1) | 145 | | 120 | | 95 | | nsec |
| 12 | ThIA(WR) | $\overline{INTACK}$ to $\overline{WR}\uparrow$ Hold Time | 0 | | 0 | | 0 | | nsec |
| 13 | TsIA(RD) | $\overline{INTACK}$ to $\overline{RD}\downarrow$ Setup Time (Note 1) | 145 | | 120 | | 95 | | nsec |
| 14 | ThIAi(RD) | $\overline{INTACK}$ to $\overline{RD}\uparrow$ Hold Time | 0 | | 0 | | 0 | | nsec |
| 15 | ThIA(PC) | $\overline{INTACK}$ to PCLK $\uparrow$ Hold Time | 40 | | 30 | | 20 | | nsec |
| 16 | TsCEl(WR) | $\overline{CE}$ LOW to $\overline{WR}\downarrow$ Setup Time | 0 | | 0 | | 0 | | nsec |
| 17 | ThCE(WR) | $\overline{CE}$ to $\overline{WR}\uparrow$ Hold Time | 0 | | 0 | | 0 | | nsec |
| 18 | TsCEh(WR) | $\overline{CE}$ HIGH to $\overline{WR}\downarrow$ Setup Time | 60 | | 50 | | 40 | | nsec |
| 19 | TsCEl(RD) | $\overline{CE}$ LOW to $\overline{RD}\downarrow$ Setup Time (Note 1) | 0 | | 0 | | 0 | | nsec |
| 20 | ThCE(RD) | $\overline{CE}$ to $\overline{RD}\uparrow$ Hold Time (Note1) | 0 | | 0 | | 0 | | nsec |
| 21 | TsCEh(RD) | $\overline{CE}$ HIGH to $\overline{RD}\downarrow$ Setup Time (Note 1) | 60 | | 50 | | 40 | | nsec |
| 22 | TwRDl | $\overline{RD}$ LOW Width (Note 1) | 150 | | 125 | | 90 | | nsec |
| 23 | TdRD(DRA) | $\overline{RD}\downarrow$ to Read Data Active Delay | 0 | | 0 | | 0 | | nsec |
| 24 | TdRDr(DR) | $\overline{RD}\uparrow$ to Read Data Not Valid Delay | 0 | | 0 | | 0 | | nsec |
| 25 | TdRDf(DR) | $\overline{RD}\downarrow$ to Read Data Valid Delay | | 140 | | 120 | | 85 | nsec |
| 26 | TdRD(DRz) | $\overline{RD}\uparrow$ to Read Data Float Delay (Note 2) | | 40 | | 35 | | 25 | nsec |

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.

2. Float delay is defined as the time at which the data bus is released from its drive state with a maximum DC Load and minimum AC load.

## SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

**Read and Write Timing (see Figure 21)**

| No. | Parameter Symbol | Parameter Description | 16.384 MHz Min. | 16.384 MHz Max. | 20 MHz Min. | 20 MHz Max. | Unit |
|---|---|---|---|---|---|---|---|
| 1 | TwPCl | PCLK LOW Width | 26 | 2000 | 22 | 1000 | nsec |
| 2 | TwPCh | PCLK HIGH Width | 26 | 2000 | 22 | 1000 | nsec |
| 3 | TfPC | PCLK Fall Time | | 8 | | 5 | nsec |
| 4 | TrPC | PCLK Rise Time | | 8 | | 5 | nsec |
| 5 | TcPC | PCLK Cycle Time | 61 | 4000 | 50 | 2000 | nsec |
| 6 | TsA(WR) | Address to $\overline{WR}$ ↓ Setup Time | 35 | | 30 | | nsec |
| 7 | ThA(WR) | Address to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 8 | TsA(RD) | Address to $\overline{RD}$ ↓ Setup Time | 35 | | 30 | | nsec |
| 9 | ThA(RD) | Address to RD ↑ Hold Time | 0 | | 0 | | nsec |
| 10 | TslA(PC) | $\overline{INTACK}$ to PCLK ↑ Setup Time | 15 | | 15 | | nsec |
| 11 | TslAi(WR) | $\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time (Note 1) | 70 | | 65 | | nsec |
| 12 | ThlA(WR) | $\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 13 | TslAi(RD) | $\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time (Note 1) | 70 | | 65 | | nsec |
| 14 | ThlA(RD) | $\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 15 | ThlA(PC) | $\overline{INTACK}$ to PCLK ↑ Hold Time | 15 | | 15 | | nsec |
| 16 | TsCEl(WR) | $\overline{CE}$ LOW to $\overline{WR}$ ↓ Setup Time | 0 | | 0 | | nsec |
| 17 | ThCE(WR) | $\overline{CE}$ to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 18 | TsCEh(WR) | $\overline{CE}$ HIGH to $\overline{WR}$ ↓ Setup Time | 30 | | 25 | | nsec |
| 19 | TsCEl(RD) | $\overline{CE}$ LOW to $\overline{RD}$ ↓ Setup Time (Note 1) | 0 | | 0 | | nsec |
| 20 | ThCE(RD) | $\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note1) | 0 | | 0 | | nsec |
| 21 | TsCEh(RD) | $\overline{CE}$ HIGH to $\overline{RD}$ ↓ Setup Time (Note 1) | 30 | | 25 | | nsec |
| 22 | TwRDl | $\overline{RD}$ LOW Width (Note 1) | 75 | | 65 | | nsec |
| 23 | TdRD(DRA) | $\overline{RD}$ ↓ to Read Data Active Delay | 0 | | 0 | | nsec |
| 24 | TdRDr(DR) | $\overline{RD}$ ↑ to Read Data Not Valid Delay | 0 | | 0 | | nsec |
| 25 | TdRDf(DR) | $\overline{RD}$ ↓ to Read Data Valid Delay | | 70 | | 60 | nsec |
| 26 | TdRD(DRz) | $\overline{RD}$ ↑ to Read Data Float Delay (Note 2) | | 20 | | 20 | nsec |

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.

2. Float delay is defined as the time at which the data bus is released from its drive state with a maximum DC load and minimum AC load.

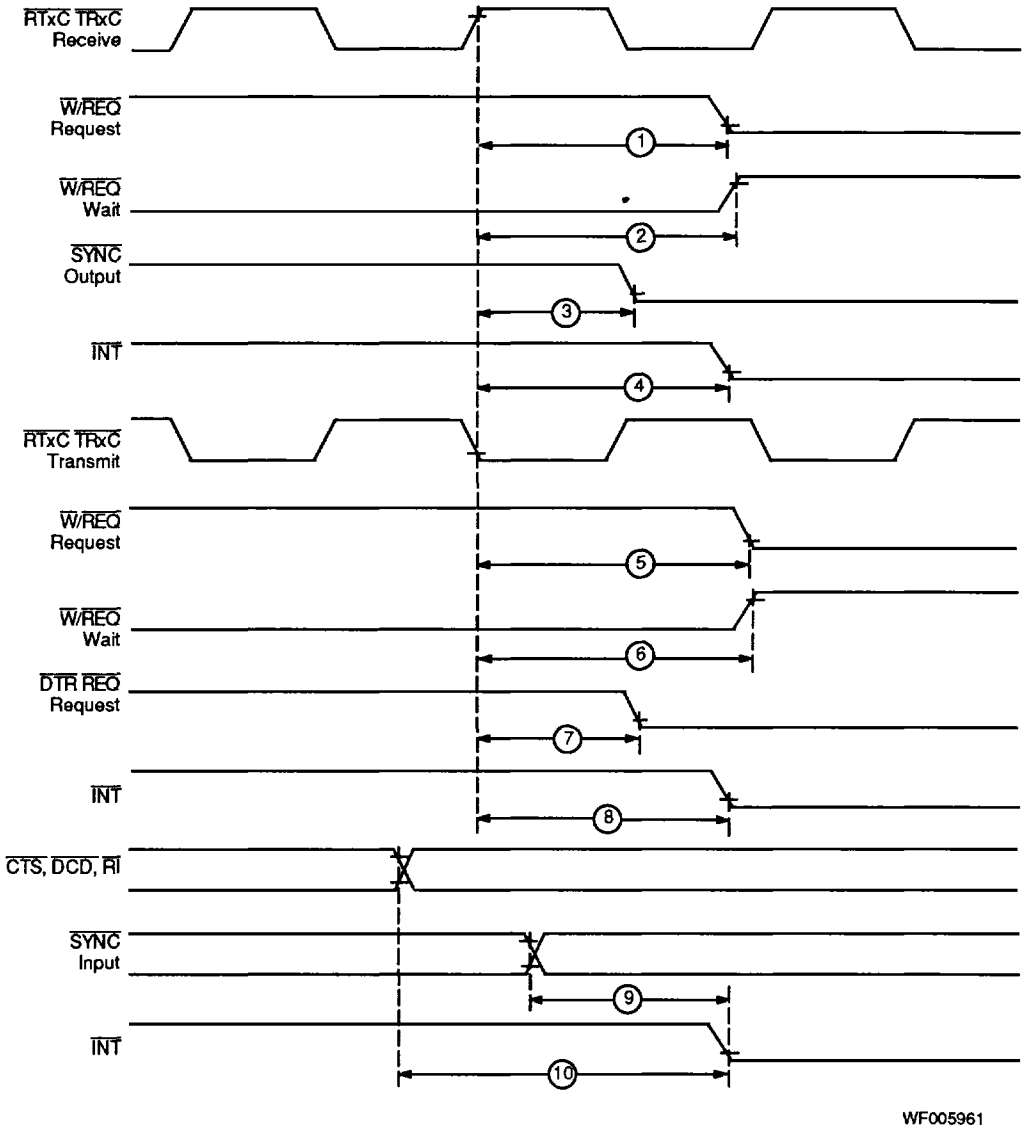



Figure 20. System Timing

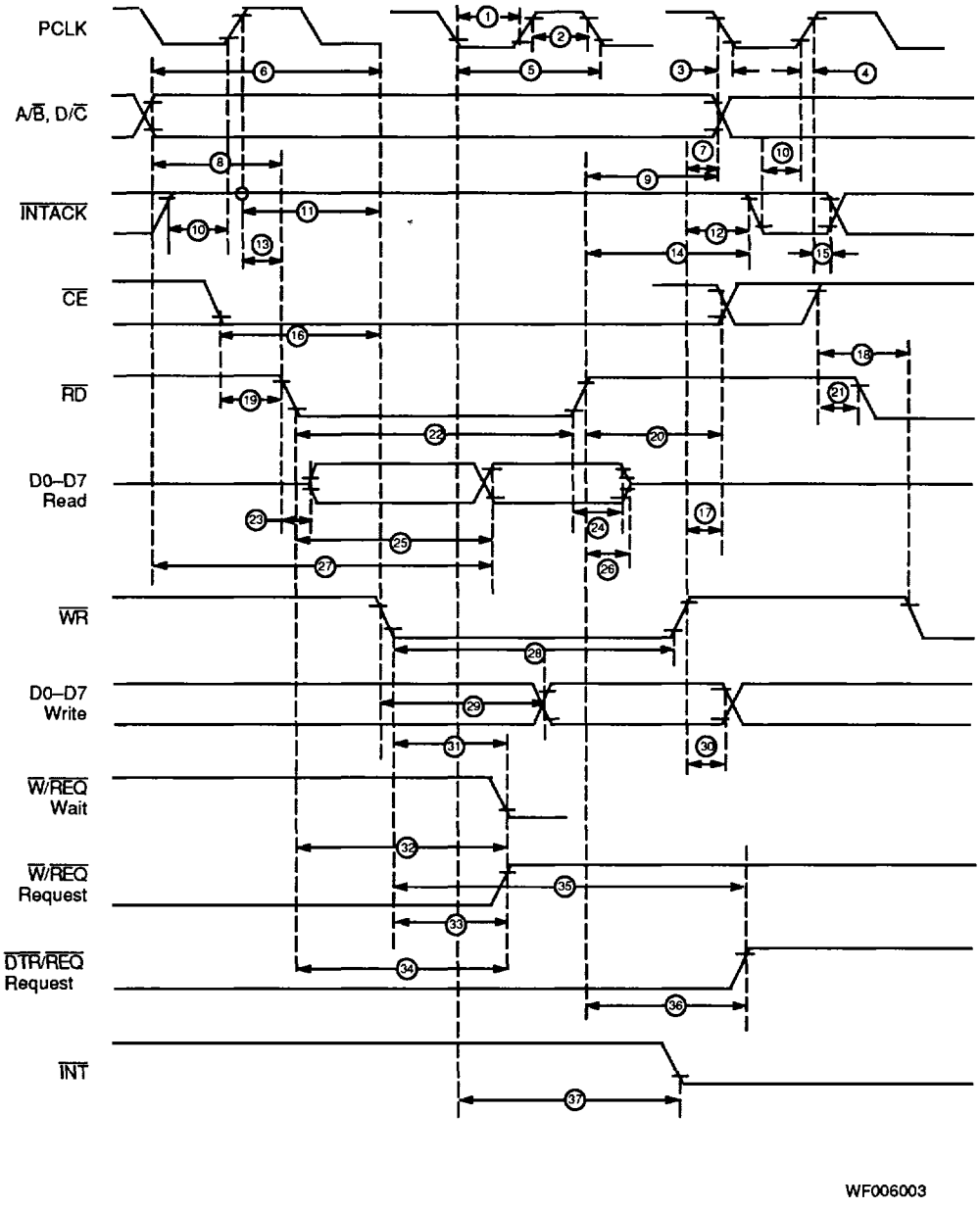



**Figure 21. Read and Write Timing**

## SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

**Interrupt Acknowledge Timing, Reset Timing, Cycle Timing (see Figures 22–24)**

| No. | Parameter Symbol | Parameter Description | 8.192 MHz Min. | 8.192 MHz Max. | 10 MHz Min. | 10 MHz Max. | 12.5 MHz Min. | 12.5 MHz Max. | Unit |
|---|---|---|---|---|---|---|---|---|---|
| 27 | TdA(DR) | Address Required Valid to Read Data Valid Delay | | 220 | | 160 | | 120 | nsec |
| 28 | TwWRI | $\overline{WR}$ LOW Width | 150 | | 125 | | 90 | | nsec |
| 29 | TdWRf(DW) | $\overline{WR}$ ↓ to Write Data Valid | | 35 | | 35 | | 25 | nsec |
| 30 | ThDW(WR) | Write Data to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | 0 | | nsec |
| 31 | TdWR(W) | $\overline{WR}$ ↓ to Wait Valid Delay (Note 2) | | 170 | | 100 | | 70 | nsec |
| 32 | TdRD(W) | $\overline{RD}$ ↓ to Wait Valid Delay (Note 2) | | 170 | | 100 | | 70 | nsec |
| 33 | TdWRf(REQ) | $\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay | | 170 | | 120 | | 100 | nsec |
| 34 | TdRDf(REQ) | $\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay | | 170 | | 120 | | 100 | nsec |
| 35a | TdWRr(REQ) | $\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay | | 4.0TcPc | | 4.0TcPc | | 4.0TcPc | nsec |
| 35b | TdWRr(EREQ) | $\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay | | 120 | | 120 | | 100 | nsec |
| 36 | TdRDr(REQ) | $\overline{RD}$ ↑ $\overline{DTR/REQ}$ Not Valid Delay | | NA | | NA | | NA | nsec |
| 37 | TdPC(INT) | PCLK ↓ to $\overline{INT}$ Valid Delay (Note 2) | | 500 | | 400 | | 350 | nsec |
| 38 | TdIAi(RD) | $\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 3) | 150 | | 125 | | 95 | | nsec |
| 39 | TwRDA | $\overline{RD}$ (Acknowledge) Width | 150 | | 125 | | 95 | | nsec |
| 40 | TdRDA(DR) | $\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay | | 140 | | 120 | | 90 | nsec |
| 41 | TsIEI(RDA) | IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time | 95 | | 80 | | 65 | | nsec |
| 42 | ThIEI(RDA) | IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time | 0 | | 0 | | 0 | | nsec |
| 43 | TdIEI(IEO) | IEI to IEO Delay Time | | 95 | | 80 | | 65 | nsec |
| 44 | TdPC(IEO) | PCLK ↑ to IEO Delay | | 200 | | 175 | | 130 | nsec |
| 45 | TdRDA(INT) | $\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 2) | | 450 | | 320 | | 260 | nsec |
| 46 | TdRD(WRQ) | $\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset | 15 | | 15 | | 10 | | nsec |
| 47 | TdWRQ(RD) | $\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset | 15 | | 15 | | 10 | | nsec |
| 48 | TwRES | $\overline{WR}$ and $\overline{RD}$ Coincident LOW for Reset | 150 | | 100 | | 85 | | nsec |
| 49 | Trc | Valid Access Recovery Time (Note 1) | 3.5 | | 3.5 | | 3.5 | | TcPc |

Notes:
1. Parameter applies only between transactions involving the ESCC, if WR/RD falling edge is synchronized to PCLK falling edge, then TrC = 3TcPc.
2. Open-drain output, measured with open-drain test load.
3. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of DdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.
4. Parameter applies to Enhanced Request mode only.

## SWITCHING CHARACTERISTICS over COMMERCIAL operating range (continued)

**Interrupt Acknowledge Timing, Reset Timing, Cycle Timing (see Figures 22–24)**

| No. | Parameter Symbol | Parameter Description | 16.384 MHz Min. | 16.384 MHz Max. | 20 MHz Min. | 20 MHz Max. | Unit |
|---|---|---|---|---|---|---|---|
| 27 | TdA(DR) | Address Required Valid to Read Data Valid Delay | | 100 | | 90 | nsec |
| 28 | TwWRl | $\overline{WR}$ LOW Width | 75 | | 65 | | nsec |
| 29 | TdWRf(DW) | $\overline{WR}$ ↓ to Write Data Valid | | 20 | | 20 | nsec |
| 30 | ThDW(WR) | Write Data to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 31 | TdWR(W) | $\overline{WR}$ ↓ to Wait Valid Delay (Note 2) | | 50 | | 45 | nsec |
| 32 | TdRD(W) | $\overline{RD}$ ↓ to Wait Valid Delay (Note 2) | | 50 | | 45 | nsec |
| 33 | TdWRf(REQ) | $\overline{WR}$ ↓ to W/$\overline{REQ}$ Not Valid Delay | | 70 | | 65 | nsec |
| 34 | TdRDf(REQ) | $\overline{RD}$ ↓ to W/$\overline{REQ}$ Not Valid Delay | | 70 | | 65 | nsec |
| 35a | TdWRr(REQ) | $\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay | | 4.0TcPc | | 4.0TcPc | nsec |
| 35b | TdWRr(EREQ) | $\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay (Note 4) | | 70 | | 65 | nsec |
| 36 | TdRDr(REQ) | $\overline{RD}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay | | NA | | NA | nsec |
| 37 | TdPC(INT) | PCLK ↓ to $\overline{INT}$ Valid Delay (Note 2) | | 175 | | 160 | nsec |
| 38 | TdIAi(RD) | $\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 3) | 50 | | 45 | | nsec |
| 39 | TwRDA | $\overline{RD}$ (Acknowledge) Width | 75 | | 65 | | nsec |
| 40 | TdRDA(DR) | $\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay | | 70 | | 60 | nsec |
| 41 | TsIEI(RDA) | IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time | 50 | | 45 | | nsec |
| 42 | ThIEI(RDA) | IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time | 0 | | 0 | | nsec |
| 43 | TdIEI(IEO) | IEI to IEO Delay Time | | 45 | | 40 | nsec |
| 44 | TdPC(IEO) | PCLK ↑ to IEO Delay | | 80 | | 70 | nsec |
| 45 | TdRDA(INT) | $\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 2) | | 200 | | 180 | nsec |
| 46 | TdRD(WRQ) | $\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset | 10 | | 10 | | nsec |
| 47 | TdWRQ(RD) | $\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset | 10 | | 10 | | nsec |
| 48 | TwRES | $\overline{WR}$ and $\overline{RD}$ Coincident LOW for Reset | 75 | | 65 | | nsec |
| 49 | Trc | Valid Access Recovery Time (Note 1) | 3.5 | | 3.5 | | TcPc |

Notes:
1. Parameter applies only between transactions involving the ESCC. If $\overline{WR}$/$\overline{RD}$ falling edge is synchronized to PCLK falling edge, then TrC = 3TcPc.
2. Open-drain output, measured with open-drain test load.
3. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.
4. Parameter applies to Enhanced Request mode only.

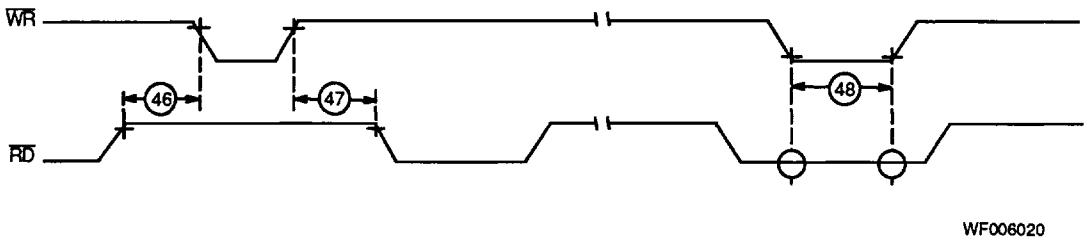



Figure 22. Reset Timing

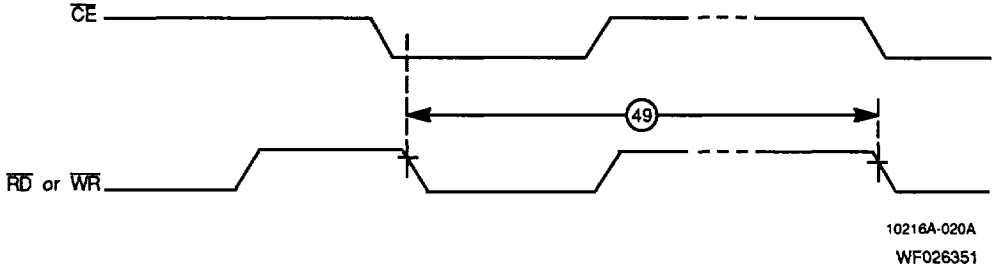



Figure 23. Cycle Timing



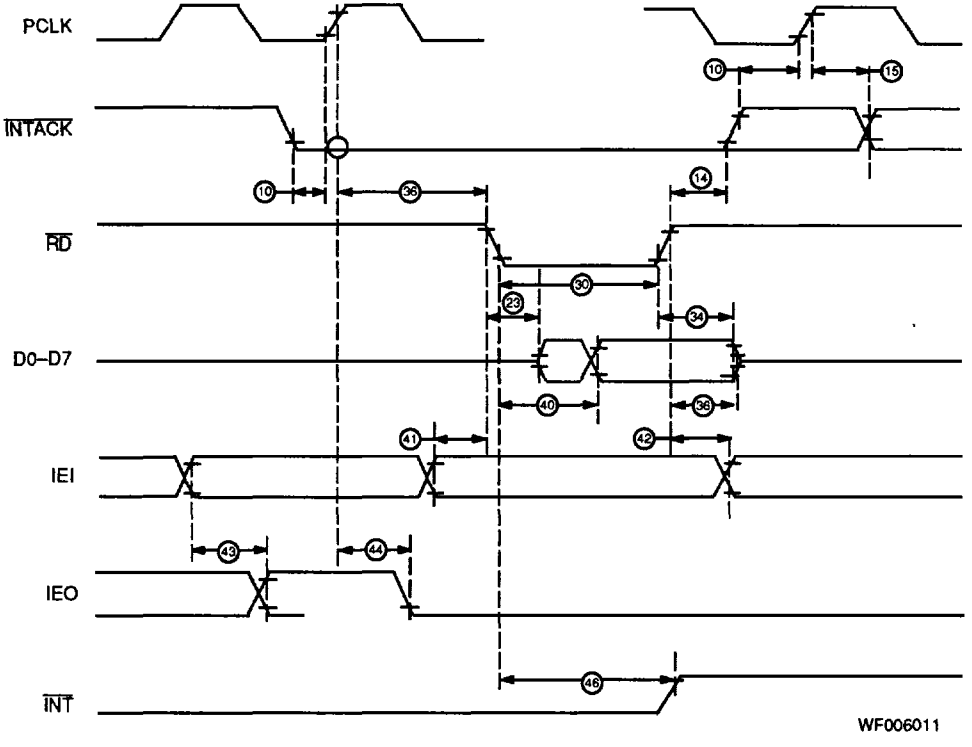



Figure 24. Interrupt Acknowledge Timing

## SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range

**General Timing (see Figure 19)**

| No. | Parameter Symbol | Parameter Description | 8.192 MHz Min. | 8.192 MHz Max. | 10 MHz Min. | 10 MHz Max. | Unit |
|---|---|---|---|---|---|---|---|
| 1 | TdPC(REQ) | PCLK ↓ to $\overline{W/REQ}$ Valid Delay | | 250 | | 150 | nsec |
| 2 | TdPC(W) | PLCK ↓ to Wait Inactive Delay | | 350 | | 250 | nsec |
| 3 | TsRXC(PC) | $\overline{RxC}$ ↑ to PCLK ↑ Setup Time (Notes 1, 4 & 8) | NA | NA | NA | NA | |
| 4 | TsRXD(RXCr) | RxD to $\overline{RxC}$ ↑ Setup Time (XI Mode) (Note 1) | 0 | | 0 | | nsec |
| 5 | ThRXD(RXCr) | RxD to $\overline{RxC}$ ↑ Hold Time (XI Mode) (Note 1) | 150 | | 125 | | nsec |
| 6 | TsRXD(RXCf) | RxD to $\overline{RxC}$ ↓ Setup Time (XI Mode) (Notes 1, 5) | 0 | | 0 | | nsec |
| 7 | ThRXD(RXCf) | RxD to $\overline{RxC}$ ↓ Hold Time (XI Mode) (Notes 1, 5) | 150 | | 125 | | nsec |
| 8 | TsSY(RXC) | $\overline{SYNC}$ to $\overline{RxC}$ ↑ Setup Time (Note 1) | −200 | | −150 | | nsec |
| 9 | ThSY(RXC) | $\overline{SYNC}$ to $\overline{RxC}$ ↑ Hold Time (Note 1) | 5TcPc | | 5TcPc | | nsec |
| 10 | TsTXC(PC) | $\overline{TxC}$ ↓ to PCLK ↑ Setup Time (Notes 2, 4 & 8) | NA | | NA | | |
| 11 | TdTXCf(TXD) | $\overline{TxC}$ ↓ to TxD Delay (XI Mode) (Note 2) | | 200 | | 150 | nsec |
| 12 | TdTXCr(TXD) | $\overline{TxC}$ ↑ to TxD Delay (XI Mode) (Notes 2, 5) | | 200 | | 150 | nsec |
| 13 | TdTXD(TRX) | TxD to $\overline{TRxC}$ Delay (Send Clock Echo) | | 200 | | 140 | nsec |
| 14a | TwRTXh | $\overline{RTxC}$ HIGH Width (Note 6) | 150 | | 120 | | nsec |
| 14b | TwRTXh(E) | $\overline{RTxC}$ HIGH Width (Note 9) | 50 | | 40 | | nsec |
| 15a | TwRTXl | $\overline{RTxC}$ LOW Width (Note 6) | 150 | | 120 | | nsec |
| 15b | TwRTXl(E) | $\overline{RTxC}$ LOW Width (Note 9) | 50 | | 40 | | nsec |
| 16a | TcRTX | $\overline{RTxC}$ Cycle Time (Notes 6, 7) | 488 | | 400 | | nsec |
| 16b | TcRTx(E) | $\overline{RTxC}$ Cycle Time (Note 9) | 125 | | 100 | | nsec |
| 17 | TXRTXX | Crystal Oscillator Period (Note 3) | 125 | 1000 | 100 | 1000 | nsec |
| 18 | TwTRXh | $\overline{TRxC}$ HIGH Width (Note 6) | 150 | | 120 | | nsec |
| 19 | TwTRXl | $\overline{TRxC}$ LOW Width (Note 6) | 150 | | 120 | | nsec |
| 20 | TcTRX | $\overline{TRxC}$ Cycle Time (Notes 6, 7) | 488 | | 400 | | nsec |
| 21 | TwEXT | $\overline{DCD}$ or $\overline{CTS}$ Pulse Width | 200 | | 120 | | nsec |
| 22 | TwSY | $\overline{SYNC}$ Pulse Width | 200 | | 120 | | nsec |

Notes:
1. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
2. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
3. Both $\overline{RTxC}$ and $\overline{SYNC}$ 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 $\overline{RxC}$ and PCLK or $\overline{TxC}$ and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver, DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.
8. External PCLK to $\overline{RxC}$ or $\overline{TxC}$ synchronization requirement eliminated for PCLK divide-by-four operation.

   $\overline{TRxC}$ and $\overline{RTxC}$ rise and fall times are identical to PCLK. Reference timing specs Tfpc and Trpc.

   Tx and Rx input clock slow rates should be kept to a maximum of 30 nsec. All parameters related to input CLK edges should be referenced at the point at which the transition begins or ends, whichever is worst case.
9. ENHANCED FEATURE— $\overline{RTxC}$ used as input to internal DPLL only.

## SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

**General Timing (see Figure 19)**

| No. | Parameter Symbol | Parameter Description | 12.5 MHz Min. | 12.5 MHz Max. | 16.384 MHz Min. | 16.384 MHz Max. | Unit |
|---|---|---|---|---|---|---|---|
| 1 | TdPC(REQ) | PCLK ↓ to $\overline{W/REQ}$ Valid Delay | | 120 | | 80 | nsec |
| 2 | TdPC(W) | PLCK ↓ to Wait Inactive Delay | | 220 | | 180 | nsec |
| 3 | TsRXC(PC) | $\overline{RxC}$ ↑ to PCLK ↑ Setup Time (Notes 1, 4 & 8) | NA | NA | NA | NA | |
| 4 | TsRXD(RXCr) | RxD to $\overline{RxC}$ ↑ Setup Time (XI Mode) (Note 1) | 0 | | 0 | | nsec |
| 5 | ThRXD(RXCr) | RxD to $\overline{RxC}$ ↑ Hold Time (XI Mode) (Note 1) | 100 | | 50 | | nsec |
| 6 | TsRXD(RXCf) | RxD to $\overline{RxC}$ ↓ Setup Time (XI Mode) (Notes 1, 5) | 0 | | 0 | | nsec |
| 7 | ThRXD(RXCf) | RxD to $\overline{RxC}$ ↓ Hold Time (XI Mode) (Notes 1, 5) | 100 | | 50 | | nsec |
| 8 | TsSY(RXC) | $\overline{SYNC}$ to $\overline{RxC}$ ↑ Setup Time (Note 1) | −125 | | −100 | | nsec |
| 9 | ThSY(RXC) | $\overline{SYNC}$ to $\overline{RxC}$ ↑ Hold Time (Note 1) | 5TcPc | | 5TcPc | | nsec |
| 10 | TsTXC(PC) | $\overline{TxC}$ ↓ to PCLK ↑ Setup Time (Notes 2, 4 & 8) | NA | | NA | | |
| 11 | TdTXCf(TXD) | $\overline{TxC}$ ↓ to TxD Delay (XI Mode) (Note 2) | | 130 | | 80 | nsec |
| 12 | TdTXCr(TXD) | $\overline{TxC}$ ↑ to TxD Delay (XI Mode) (Notes 2, 5) | | 130 | | 80 | nsec |
| 13 | TdTXD(TRX) | TxD to $\overline{TRxC}$ Delay (Send Clock Echo) | | 120 | | 80 | nsec |
| 14a | TwRTXh | $\overline{RTxC}$ HIGH Width (Note 6) | 100 | | 80 | | nsec |
| 14b | TwRTXh(E) | $\overline{RTxC}$ HIGH Width (Note 9) | 34 | | 15.6 | | nsec |
| 15a | TwRTXl | $\overline{RTxC}$ LOW Width (Note 6) | 100 | | 80 | | nsec |
| 15b | TwRTXl(E) | $\overline{RTxC}$ LOW Width (Note 9) | 34 | | 15.6 | | nsec |
| 16a | TcRTX | $\overline{RTxC}$ Cycle Time (Notes 6, 7) | 320 | | 244 | | nsec |
| 16b | TcRTx(E) | $\overline{RTxC}$ Cycle Time (Note 9) | 80 | | 31.25 | | nsec |
| 17 | TXRTXX | Crystal Oscillator Period (Note 3) | 80 | 1000 | 62 | 1000 | nsec |
| 18 | TwTRxh | $\overline{TRxC}$ HIGH Width (Note 6) | 100 | | 80 | | nsec |
| 19 | TwTRXl | $\overline{TRxC}$ LOW Width (Note 6) | 100 | | 80 | | nsec |
| 20 | TcTRX | $\overline{TRxC}$ Cycle Time (Notes 6, 7) | 320 | | 244 | | nsec |
| 21 | TwEXT | $\overline{DCD}$ or $\overline{CTS}$ Pulse Width | 100 | | 70 | | nsec |
| 22 | TwSY | $\overline{SYNC}$ Pulse Width | 100 | | 70 | | nsec |

Notes:
1. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
2. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
3. Both $\overline{RTxC}$ and $\overline{SYNC}$ 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 $\overline{RxC}$ and PCLK or $\overline{TxC}$ and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver, DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive or transmit data is 1/4 PCLK.
8. External PCLK to $\overline{RxC}$ or $\overline{TxC}$ synchronization requirement eliminated for PCLK divide-by-four operation.

   $\overline{TRxC}$ and $\overline{RTxC}$ rise and fall times are identical to PCLK. Reference timing specs Tfpc and Trpc.

   Tx and Rx input clock slow rates should be kept to a maximum of 30 nsec. All parameters related to input CLK edges should be referenced at the point at which the transition begins or ends, whichever is worst case.
9. ENHANCED FEATURE— $\overline{RTxC}$ used as input to internal DPLL only.

## SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

**System Timing (see Figure 20)**

| No. | Parameter Symbol | Parameter Description | 8.192 MHz Min. | 8.192 MHz Max. | 10 MHz Min. | 10 MHz Max. | Unit |
|---|---|---|---|---|---|---|---|
| 1 | TdRXC(REQ) | $\overline{RXC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2) | 8 | 12 | 8 | 12 | TcPc |
| 2 | TdRXC(W) | $\overline{RXC}$ ↑ to Wait Inactive Delay (Notes 1, 2) | 8 | 14 | 8 | 14 | TcPc |
| 3 | TdRXC(SY) | $\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2) | 4 | 7 | 4 | 7 | TcPc |
| 4 | TdRXC(INT) | $\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2) | 10 | 16 | 10 | 16 | TcPc |
| 5 | TdTXC(REQ) | $\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3) | 5 | 8 | 5 | 8 | TcPc |
| 6 | TdTXC(W) | $\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3) | 5 | 11 | 5 | 11 | TcPc |
| 7a | TdTXC(DRQ) | $\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3) | 4 | 7 | 4 | 7 | TcPc |
| 7b | TdTXC(EDRQ) | $\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Notes 3, 4) | 5 | 8 | 5 | 8 | TcPc |
| 8 | TdTXC(INT) | $\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3) | 6 | 10 | 6 | 10 | TcPc |
| 9 | TdSY(INT) | $\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1) | 2 | 6 | 2 | 6 | TcPc |
| 10 | TdEXT(INT) | $\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1) | 2 | 6 | 2 | 6 | TcPc |

| No. | Parameter Symbol | Parameter Description | 12.5 MHz Min. | 12.5 MHz Max. | 16.384 MHz Min. | 16.384 MHz Max. | Unit |
|---|---|---|---|---|---|---|---|
| 1 | TdRXC(REQ) | $\overline{RXC}$ ↑ $\overline{W/REQ}$ Valid Delay (Note 2) | 8 | 12 | 8 | 12 | TcPc |
| 2 | TdRXC(W) | $\overline{RXC}$ ↑ to Wait Inactive Delay (Notes 1, 2) | 8 | 14 | 8 | 14 | TcPc |
| 3 | TdRXC(SY) | $\overline{RxC}$ ↑ to $\overline{SYNC}$ Valid Delay (Note 2) | 4 | 7 | 4 | 7 | TcPc |
| 4 | TdRXC(INT) | $\overline{RxC}$ ↑ to $\overline{INT}$ Valid Delay (Notes 1, 2) | 10 | 16 | 10 | 16 | TcPc |
| 5 | TdTXC(REQ) | $\overline{TxC}$ ↓ to $\overline{W/REQ}$ Valid Delay (Note 3) | 5 | 8 | 5 | 8 | TcPc |
| 6 | TdTXC(W) | $\overline{TxC}$ ↓ to Wait Inactive Delay (Notes 1, 3) | 5 | 11 | 5 | 11 | TcPc |
| 7a | TdTXC(DRQ) | $\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Note 3) | 4 | 7 | 4 | 7 | TcPc |
| 7b | TdTXC(EDRQ) | $\overline{TxC}$ ↓ to $\overline{DTR/REQ}$ Valid Delay (Notes 3, 4) | 5 | 8 | 5 | 8 | TcPc |
| 8 | TdTXC(INT) | $\overline{TxC}$ ↓ to $\overline{INT}$ Valid Delay (Notes 1, 3) | 6 | 10 | 6 | 10 | TcPc |
| 9 | TdSY(INT) | $\overline{SYNC}$ Transition to $\overline{INT}$ Valid Delay (Note 1) | 2 | 6 | 2 | 6 | TcPc |
| 10 | TdEXT(INT) | $\overline{DCD}$ or $\overline{CTS}$ Transition to $\overline{INT}$ Valid Delay (Note 1) | 2 | 6 | 2 | 6 | TcPc |

Notes: 1. Open-drain output, measured with open-drain test load.
2. $\overline{RxC}$ is $\overline{RTxC}$ or $\overline{TRxC}$, whichever is supplying the receive clock.
3. $\overline{TxC}$ is $\overline{TRxC}$ or $\overline{RTxC}$, whichever is supplying the transmit clock.
4. Parameter applies to Enhanced Request mode only.

## SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

**Read and Write Timing (see Figure 21)**

| No. | Parameter Symbol | Parameter Description | 8.192 MHz | | 10 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | Min. | Max. | Min. | Max. | |
| 1 | TwPCl | PCLK LOW Width | 50 | 1000 | 40 | 1000 | nsec |
| 2 | TwPCh | PCLK HIGH Width | 50 | 1000 | 40 | 1000 | nsec |
| 3 | TfPC | PCLK Fall Time | | 15 | | 12 | nsec |
| 4 | TrPC | PCLK Rise Time | | 15 | | 12 | nsec |
| 5 | TcPC | PCLK Cycle Time | 122 | 2000 | 100 | 2000 | nsec |
| 6 | TsA(WR) | Address to $\overline{WR}$ ↓ Setup Time | 70 | | 50 | | nsec |
| 7 | ThA(WR) | Address to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 8 | TsA(RD) | Address to $\overline{RD}$ ↓ Setup Time | 70 | | 50 | | nsec |
| 9 | ThA(RD) | Address to $\overline{RD}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 10 | TsIA(PC) | $\overline{INTACK}$ to PCLK ↑ Setup Time | 20 | | 20 | | nsec |
| 11 | TsIAi(WR) | $\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time (Note 1) | 145 | | 120 | | nsec |
| 12 | ThIA(WR) | $\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 13 | TsIAi(RD) | $\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time (Note 1) | 145 | | 120 | | nsec |
| 14 | ThIA(RD) | $\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 15 | ThIA(PC) | $\overline{INTACK}$ to PCLK ↑ Hold Time | 40 | | 30 | | nsec |
| 16 | TsCEl(WR) | $\overline{CE}$ LOW to $\overline{WR}$ ↓ Setup Time | 0 | | 0 | | nsec |
| 17 | ThCE(WR) | $\overline{CE}$ to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 18 | TsCEh(WR) | $\overline{CE}$ HIGH to $\overline{WR}$ ↓ Setup Time | 60 | | 50 | | nsec |
| 19 | TsCEl(RD) | $\overline{CE}$ LOW to $\overline{RD}$ ↓ Setup Time (Note 1) | 0 | | 0 | | nsec |
| 20 | ThCE(RD) | $\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note1) | 0 | | 0 | | nsec |
| 21 | TsCEh(RD) | $\overline{CE}$ HIGH to $\overline{RD}$ ↓ Setup Time (Note 1) | 60 | | 50 | | nsec |
| 22 | TwRDl | $\overline{RD}$ LOW Width (Note 1) | 150 | | 125 | | nsec |
| 23 | TdRD(DRA) | $\overline{RD}$ ↓ to Read Data Active Delay | 0 | | 0 | | nsec |
| 24 | TdRDr(DR) | $\overline{RD}$ ↑ to Read Data Not Valid Delay | 0 | | 0 | | nsec |
| 25 | TdRDf(DR) | $\overline{RD}$ ↓ to Read Data Valid Delay | | 140 | | 125 | nsec |
| 26 | TdRD(DRz) | $\overline{RD}$ ↑ to Read Data Float Delay (Note 2) | | 40 | | 35 | nsec |

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time at which the data bus is released from its drive state with a maximum DC Load and minimum AC load.

## SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

**Read and Write Timing (see Figure 21)**

| No. | Parameter Symbol | Parameter Description | 12.5 MHz | | 16.384 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | Min. | Max. | Min. | Max. | |
| 1 | TwPCl | PCLK LOW Width | 34 | 1000 | 26 | 1000 | nsec |
| 2 | TwPCh | PCLK HIGH Width | 34 | 1000 | 26 | 1000 | nsec |
| 3 | TfPC | PCLK Fall Time | | 10 | | 8 | nsec |
| 4 | TrPC | PCLK Rise Time | | 10 | | 8 | nsec |
| 5 | TcPC | PCLK Cycle Time | 80 | 2000 | 61 | 2000 | nsec |
| 6 | TsA(WR) | Address to $\overline{WR}$ ↓ Setup Time | 45 | | 35 | | nsec |
| 7 | ThA(WR) | Address to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 8 | TsA(RD) | Address to $\overline{RD}$ ↓ Setup Time | 45 | | 35 | | nsec |
| 9 | ThA(RD) | Address to $\overline{RD}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 10 | TsIA(PC) | $\overline{INTACK}$ to PCLK ↑ Setup Time | 15 | | 15 | | nsec |
| 11 | TsIAi(WR) | $\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time (Note 1) | 95 | | 70 | | nsec |
| 12 | ThIA(WR) | $\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 13 | TsIAi(RD) | $\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time | 95 | | 70 | | nsec |
| 14 | ThIA(RD) | $\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 15 | ThIA(PC) | $\overline{INTACK}$ to PCLK ↑ Hold Time | 20 | | 15 | | nsec |
| 16 | TsCEl(WR) | $\overline{CE}$ LOW to $\overline{WR}$ ↓ Setup Time | 0 | | 0 | | nsec |
| 17 | ThCE(WR) | $\overline{CE}$ to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 18 | TsCEh(WR) | $\overline{CE}$ HIGH to $\overline{WR}$ ↓ Setup Time | 40 | | 30 | | nsec |
| 19 | TsCEl(RD) | $\overline{CE}$ LOW to $\overline{RD}$ ↓ Setup Time (Note 1) | 0 | | 0 | | nsec |
| 20 | ThCE(RD) | $\overline{CE}$ to $\overline{RD}$ ↑ Hold Time (Note1) | 0 | | 0 | | nsec |
| 21 | TsCEh(RD) | $\overline{CE}$ HIGH to $\overline{RD}$ ↓ Setup Time | 40 | | 30 | | nsec |
| 22 | TwRDl | $\overline{RD}$ LOW Width (Note 1) | 90 | | 75 | | nsec |
| 23 | TdRD(DRA) | $\overline{RD}$ ↓ to Read Data Active Delay | 0 | | 0 | | nsec |
| 24 | TdRDr(DR) | $\overline{RD}$ ↑ to Read Data Not Valid Delay | 0 | | 0 | | nsec |
| 25 | TdRDf(DR) | $\overline{RD}$ ↓ to Read Data Valid Delay | | 90 | | 70 | nsec |
| 26 | TdRD(DRz) | $\overline{RD}$ ↑ to Read Data Float Delay (Note 2) | | 25 | | 20 | nsec |

Notes: 1. Parameter does not apply to Interrupt Acknowledge transactions.

2. Float delay is defined as the time at which the data bus is released from its drive state with a maximum DC load and minimum AC load.

## SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

**Interrupt Acknowledge Timing, Reset Timing, Cycle Timing (see Figures 22–24)**

| No. | Parameter Symbol | Parameter Description | 8.192 MHz Min. | 8.192 MHz Max. | 10 MHz Min. | 10 MHz Max. | Unit |
|---|---|---|---|---|---|---|---|
| 27 | TdA(DR) | Address Required Valid to Read Data Valid Delay | | 220 | | 160 | nsec |
| 28 | TwWRl | $\overline{WR}$ LOW Width | 150 | | 125 | | nsec |
| 29 | TdWRf(DW) | $\overline{WR}$ ↓ to Write Data Valid | | 35 | | 35 | nsec |
| 30 | ThDW(WR) | Write Data to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 31 | TdWR(W) | $\overline{WR}$ ↓ to Wait Valid Delay (Note 2) | | 170 | | 100 | nsec |
| 32 | TdRD(W) | $\overline{RD}$ ↓ to Wait Valid Delay (Note 2) | | 170 | | 100 | nsec |
| 33 | TdWRf(REQ) | $\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay | | 170 | | 120 | nsec |
| 34 | TdRDf(REQ) | $\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay | | 170 | | 120 | nsec |
| 35a | TdWRr(REQ) | $\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay | | 4.0TcPc | | 4.0TcPc | nsec |
| 35b | TdWRr(EREQ) | $\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay (Note 4) | | 120 | | 120 | nsec |
| 36 | TdRDr(REQ) | $\overline{RD}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay | | NA | | NA | nsec |
| 37 | TdPC(INT) | PCLK ↓ to $\overline{INT}$ Valid Delay (Note 2) | | 500 | | 400 | nsec |
| 38 | TdIAi(RD) | $\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 3) | 150 | | 125 | | nsec |
| 39 | TwRDA | $\overline{RD}$ (Acknowledge) Width | 150 | | 125 | | nsec |
| 40 | TdRDA(DR) | $\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay | | 140 | | 120 | nsec |
| 41 | TsIEI(RDA) | IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time | 95 | | 80 | | nsec |
| 42 | ThIEI(RDA) | IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time | 0 | | 0 | | nsec |
| 43 | TdIEI(IEO) | IEI to IEO Delay Time | | 95 | | 80 | nsec |
| 44 | TdPC(IEO) | PCLK ↑ to IEO Delay | | 200 | | 175 | nsec |
| 45 | TdRDA(INT) | $\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 2) | | 450 | | 320 | nsec |
| 46 | TdRD(WRQ) | $\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset | 15 | | 15 | | nsec |
| 47 | TdWRQ(RD) | $\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset | 15 | | 15 | | nsec |
| 48 | TwRES | $\overline{WR}$ and $\overline{RD}$ Coincident LOW for Reset | 150 | | 100 | | nsec |
| 49 | Trc | Valid Access Recovery Time (Note 1) | 3.5 | | 3.5 | | nsec |

Notes: 1. Parameter applies only between transactions involving the ESCC. If $\overline{WR/RD}$ falling edge is synchronized to PCLK falling edge, then TrC = 3TcPc.

2. Open-drain output, measured with open-drain test load.

3. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.

4. Parameter applies to Enhanced Request mode only.

## SWITCHING CHARACTERISTICS over MILITARY/INDUSTRIAL operating range (continued)

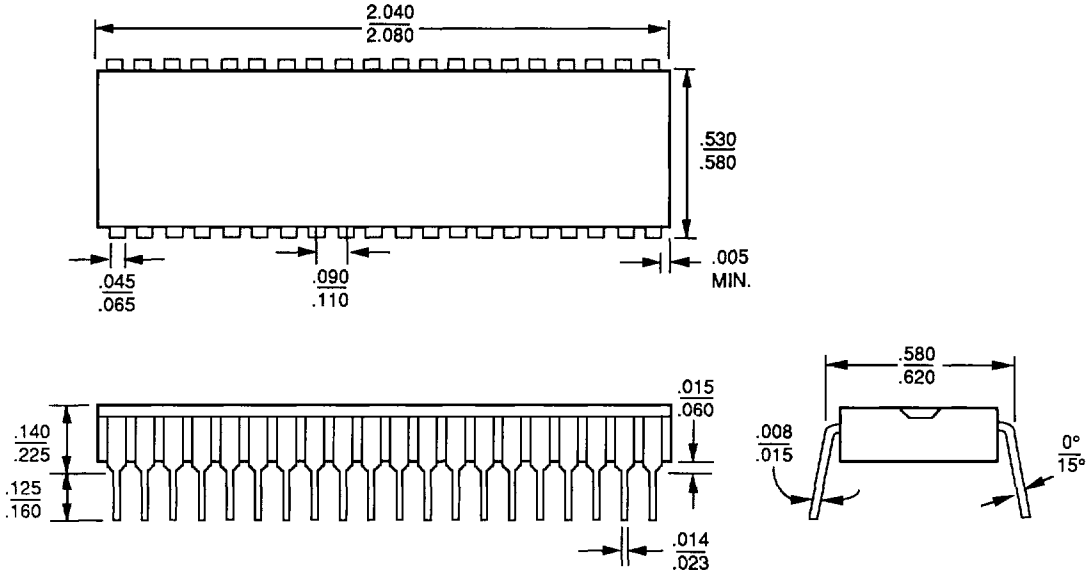
**Interrupt Acknowledge Timing, Reset Timing, Cycle Timing (see Figures 22–24)**

| No. | Parameter Symbol | Parameter Description | 12.5 MHz Min. | 12.5 MHz Max. | 16.384 MHz Min. | 16.384 MHz Max. | Unit |
|---|---|---|---|---|---|---|---|
| 27 | TdA(DR) | Address Required Valid to Read Data Valid Delay | | 120 | | 100 | nsec |
| 28 | TwWRI | $\overline{WR}$ LOW Width | 90 | | 75 | | nsec |
| 29 | TdWRf(DW) | $\overline{WR}$ ↓ to Write Data Valid | | 25 | | 20 | nsec |
| 30 | ThDW(WR) | Write Data to $\overline{WR}$ ↑ Hold Time | 0 | | 0 | | nsec |
| 31 | TdWR(W) | $\overline{WR}$ ↓ to Wait Valid Delay (Note 2) | | 70 | | 50 | nsec |
| 32 | TdRD(W) | $\overline{RD}$ ↓ to Wait Valid Delay (Note 2) | | 70 | | 50 | nsec |
| 33 | TdWRf(REQ) | $\overline{WR}$ ↓ to $\overline{W/REQ}$ Not Valid Delay | | 100 | | 70 | nsec |
| 34 | TdRDf(REQ) | $\overline{RD}$ ↓ to $\overline{W/REQ}$ Not Valid Delay | | 100 | | 70 | nsec |
| 35a | TdWRr(REQ) | $\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay | | 4.0TcPc | | 4.0TcPc | nsec |
| 35b | TdWRr(EREQ) | $\overline{WR}$ ↓ to $\overline{DTR/REQ}$ Not Valid Delay (Note 4) | | 100 | | 70 | nsec |
| 36 | TdRDr(REQ) | $\overline{RD}$ ↑ to $\overline{DTR/REQ}$ Not Valid Delay | | NA | | NA | nsec |
| 37 | TdPC(INT) | PCLK ↓ to $\overline{INT}$ Valid Delay (Note 2) | | 350 | | 175 | nsec |
| 38 | TdIAi(RD) | $\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay (Note 3) | 95 | | 50 | | nsec |
| 39 | TwRDA | $\overline{RD}$ (Acknowledge) Width | 95 | | 75 | | nsec |
| 40 | TdRDA(DR) | $\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay | | 90 | | 70 | nsec |
| 41 | TsIEI(RDA) | IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time | 65 | | 50 | | nsec |
| 42 | ThIEI(RDA) | IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time | 0 | | 0 | | nsec |
| 43 | TdIEI(IEO) | IEI to IEO Delay Time | | 65 | | 45 | nsec |
| 44 | TdPC(IEO) | PCLK ↑ to IEO Delay | | 130 | | 80 | nsec |
| 45 | TdRDA(INT) | $\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay (Note 2) | | 260 | | 200 | nsec |
| 46 | TdRD(WRQ) | $\overline{RD}$ ↑ to $\overline{WR}$ ↓ Delay for No Reset | 10 | | 10 | | nsec |
| 47 | TdWRQ(RD) | $\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset | 10 | | 10 | | nsec |
| 48 | TwRES | $\overline{WR}$ and $\overline{RD}$ Coincident LOW for Reset | 85 | | 75 | | nsec |
| 49 | Trc | Valid Access Recovery Time (Note 1) | 3.5 | | 3.5 | | TcPc |

Notes:
1. Parameter applies only between transactions involving the ESCC. If $\overline{WR}/\overline{RD}$ falling edge is synchronized to PCLK falling edge, then TrC = 3TcPc.
2. Open-drain output, measured with open-drain test load.
3. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the sum of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEI(IEO) for each device separating them in the daisy chain.
4. Parameter applies to Enhanced Request mode only.

## PHYSICAL DIMENSIONS

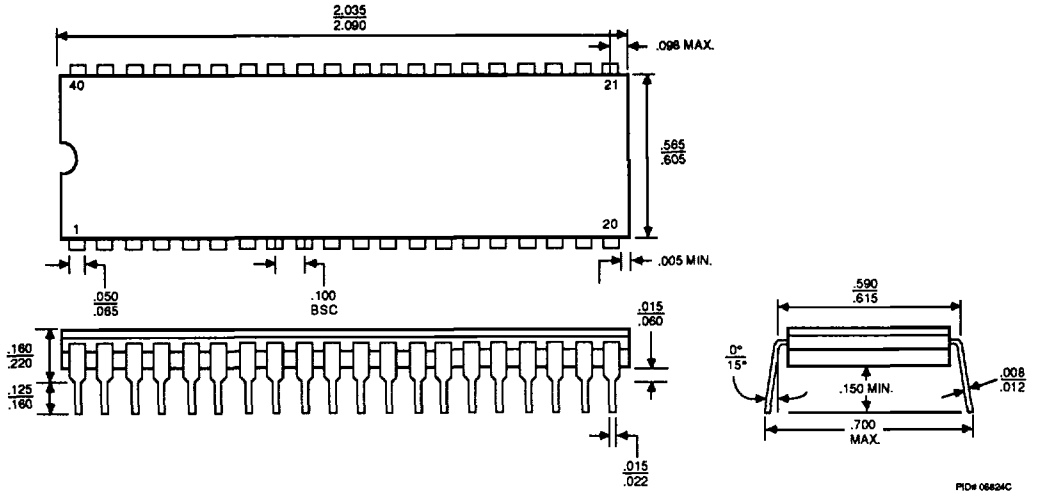
**PD 040**




PID# 06823B

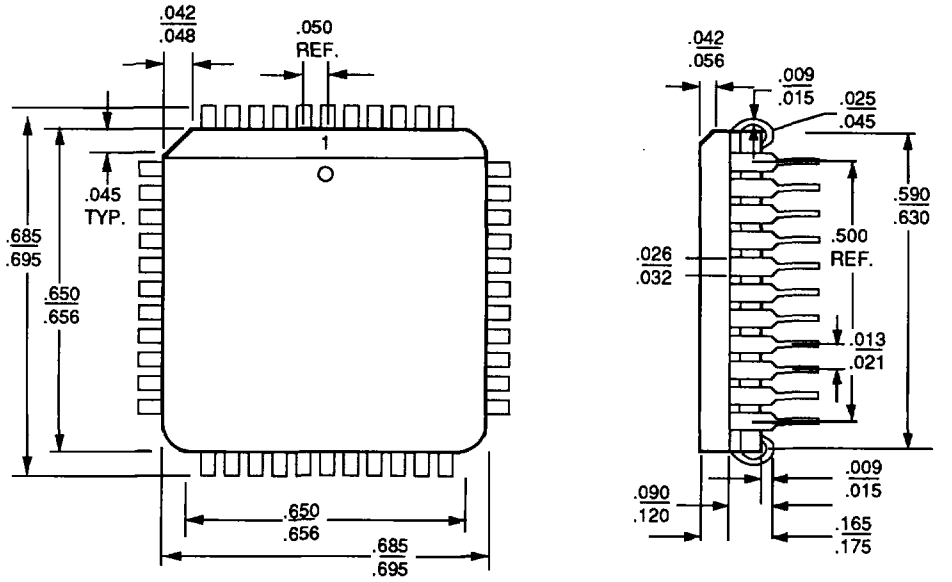
## PHYSICAL DIMENSIONS

**CD 040**



## PHYSICAL DIMENSIONS

PL 044




06752D

1

## PHYSICAL DIMENSIONS

CL 044