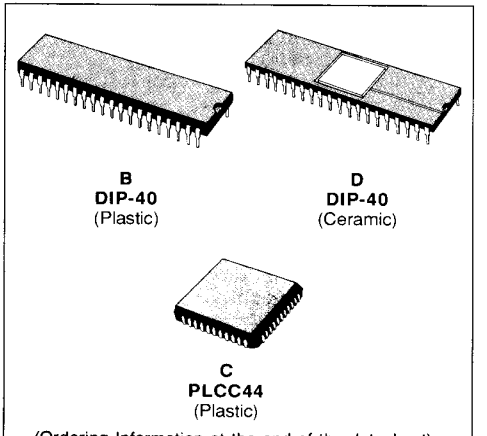
Z80C SIO CMOS VERSION

- TWO INDEPENDENT FULL-DUPLEX CHANNELS, WITH SEPARATE CONTROL AND STATUS LINES FOR MODEMS OR OTHER DEVICES
- DATA TRANSFER RATE UP TO 800K BIT/SECOND
- ASYNCHRONOUS PROTOCOLS : EVERYTHING NECESSARY FOR COMPLETE MESSAGES IN 5, 6, 7 OR 8 BITS/CHARACTER. INCLUDES VARIABLE STOP BITS AND SEVERAL CLOCK-RATE MULTIPLIERS ; BREAK GENERATION AND DETECTION ; PARITY ; OVERRUN AND FRAMING ERROR DETECTION
- SYNCHRONOUS PROTOCOLS : EVERYTHING NECESSARY FOR COMPLETE BIT- OR BYTE-ORIENTED MESSAGES IN 5, 6, 7 OR 8 BITS/CHARACTER, INCLUDING IBM BISYNC, SDLC, HDLC, CCITT-X.25 AND OTHERS. AUTOMATIC CRC GENERATION/CHECKING SYNC CHARACTER AND ZERO INSERTION/DELETION, ABORT GENERATION/DETECTION AND FLAG INSERTION
- RECEIVER DATA REGISTERS QUADRUPLY BUFFERED, TRANSMITTER REGISTERS DOUBLY BUFFERED
- HIGHLY SOPHISTICATED AND FLEXIBLE DAISY-CHAIN INTERRUPT VECTORING FOR INTERRUPTS WITHOUT EXTERNAL LOGIC
- SINGLE 5V ± 10% POWER SUPPLY
- LOW POWER CONSUPTION :
  - 2.5mA TYP. AT 4MHz
  - 4mA TYP. AT 6MHz
  - LESS THAN 10µA IN POWER DOWN MODE
- EXTENDED OPERATING TEMPERATURE – 40°C TO + 85°C



B
DIP-40
(Plastic)

D
DIP-40
(Ceramic)

C
PLCC44
(Plastic)

(Ordering Information at the end of the datasheet)

DESCRIPTION

The Z80C SIO Serial Input/Output Controller is a dual-channel data communication interface with extraordinary versatility and capability. Its basic functions as a serial-to-parallel, parallel-to-serial converter/controller can be programmed by a CPU for a broad range of serial communication applications.

The device supports all common asynchronous and synchronous protocols, byte- or bit-oriented and performs all of the functions traditionally done by UARTs, USARTs and synchronous communication controllers combined, plus additional functions traditionally performed by the CPU, Moreover, it does this on two fully-independent channels, with an exceptionally sophisticated interrupt structure that allows very fast transfers.

Full interfacing is provided for CPU or DMA control. In addition to data communication, the circuit can handle virtually all types of serial I/O with fast (or slow) peripheral devices.

While designed primarily as a member of Z80 family, its versatility makes it well suited to many other CPUs.

PIN DESCRIPTIONS

Figures 1 through 6 illustrate the three pin configurations (bonding options) available in the SIO. The constraints of a 40-pin package make it impossible to bring out the Receive Clock (RxC), Transmitt Clock (TxC), Data Terminal Ready (DTR) and Sync (SYNC) signals for both channels. Therefore, either Channel B lacks a signal or two signals are bonded together in the three bonding options offered :

- Z80C SIO-2 lacks $\overline{SYNCB}$
- Z80C SIO-1 lacks $\overline{DTRB}$
- Z80C SIO-0 as a four signal, but $\overline{TxCB}$ and $\overline{RxCB}$ are bonded together

The first bonding option above (SIO-2) is the preferred version for most applications. The Chip-Carrier package version, having a 44-pin facility, resume the three bonding option configurations. It is named Z84C44 (figure 7). The pin description are as follows :

B/$\overline{A}$. *Channel A Or B Select* (Input, High Selects Channel B). This input defines which channel is accessed during a data transfer between the CPU and the SIO. Address bit A_0 from the CPU is often used for the selection function.

C/$\overline{D}$. *Control Or Data Select* (Input, High Selects Control). This input defines the type of information transfer performed between the CPU and the SIO. A High at this input during a CPU write to the SIO causes the information on the data bus to be interpreted as a command for the channel selected by B/$\overline{A}$. A Low at C/$\overline{D}$ means that the information on the data bus is data. Address bit A_1 is often used for this function.

$\overline{CE}$. *Chip Enable* (Input, Active Low). A Low level at this input enables the SIO to accept command or data input from the CPU during a write cycle or to transmit data to the CPU during a read cycle.

CLK. *System Clock* (Input). The SIO uses the standard Z80 System Clock to synchronize internal signals. This is a single-phase clock.

$\overline{CTSA}$, **$\overline{CTSB}$**. *Clear To Send* (Inputs, Active Low). When programmed as Auto Enables, a Low on these inputs enables the respective transmitter. If not programmed as Auto Enables, these inputs may be programmed as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow-risetime signals. The SIO detects pulses on these inputs and interrupts the CPU on both logic level transitions. The Schmitt-trigger buffering does not guarantee a specified noise-level margin.

D_0-D_7. *System Data Bus* (Bidirectional, 3-state). The system data bus transfers data and commands between the CPU and the Z80C SIO. D_0 is the least significant bit.

$\overline{DCDA}$, **$\overline{DCDB}$**. *Data Carrier Detect* (Inputs, Active Low). These pins function as receiver enables if the SIO is programmed for Auto Enables ; otherwise they may be used as general-purpose input pins.

Figure 1 : Z80C SIO-2 Logic Functions.

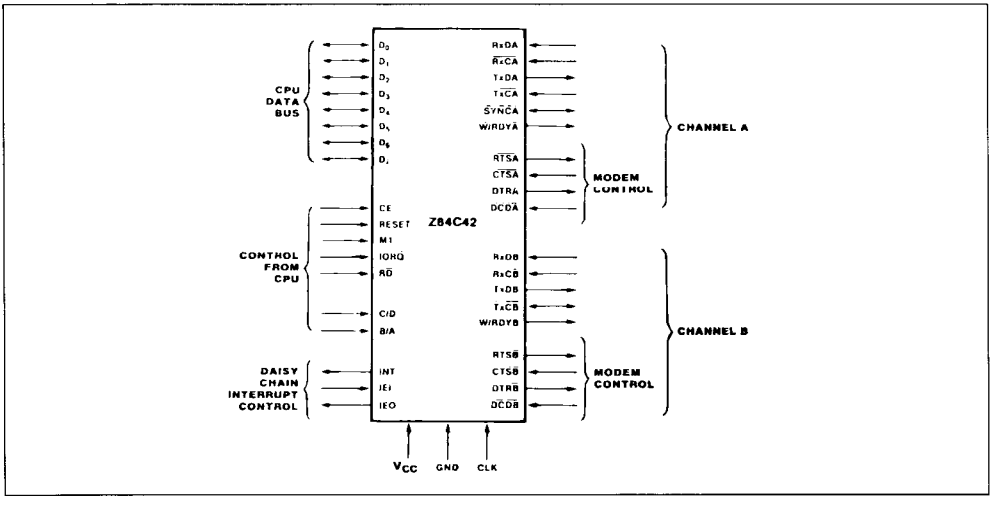
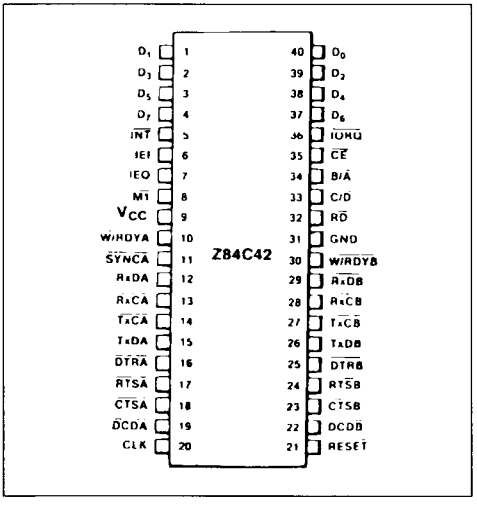


Figure 2 : Z80C SIO-2 Dual in Line Pin Configuration.



Both pins are Schmitt-trigger buffered to accommodate slow-risetime signals. The SIO detects pulses on these pins and interrupts the CPU on both logic level transitions. Schmitt-trigger buffering does not guarantee a specific noise-level margin.

DTRA, **DTRB**. *Data Terminal Ready* (Outputs, Active Low). These outputs follow the state programmed into Z80C SIO. They can also be programmed as general-purpose outputs.

In the Z80C SIO-1 bonding option, DTRB is omitted.

IEI. *Interrupt Enable In* (Input, Active High). This signal is used with IEO to form a priority daisy-chain when there is more than one interrupt-driven device. A High on this line indicates that no other device of higher priority is being serviced by a CPU interrupt service routine.

IEO. *Interrupt Enable Out* (Output, Active High). IEO is High only if IEI is High and the CPU is not servicing an interrupt from this SIO. Thus, this signal blocks lower priority devices form interrupting while a higher priority device is being serviced by its CPU interrupt service routine.

INT. *Interrupt Request* (Output, Open Drain, Active Low). When the SIO is requesting an interrupt, it pulls INT Low.

IORQ. *Input/Output Request* (Input from CPU, Active Low). IORQ is used in conjunction with B/A, C/D, CE and RD to transfer commands and data between the CPU and the SIO. When CE, RD and IORQ are all active, the channel selected by B/A transfers data to the CPU (a read operation). When CE and IORQ are active but RD is inactive, the channel selected by B/A is written to by the CPU with either data or

Figure 3 : Z80C SIO-1 Logic Functions.

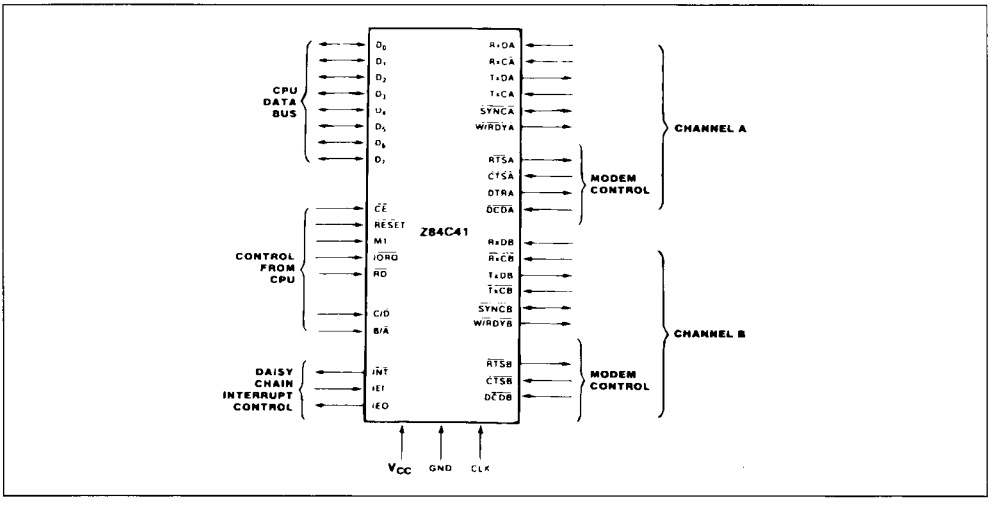
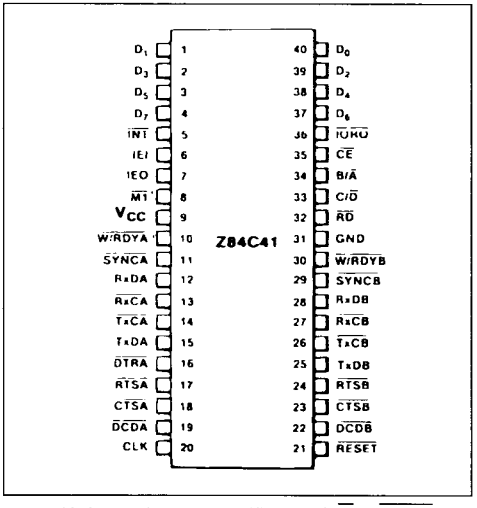


Figure 4 : Z80C SIO-1 Dual in Line Pin Configuration.



control information as specified by C/D. If IORQ and M1 are active simultaneously, the CPU is acknowledging an interrupt and the SIO automatically places its interrupt vector on the CPU data bus if it is the highest priority device requesting an interrupt.

M1. *Machine Cycle* (Input from Z80C CPU, Active Low). When M1 is active and RD is also active, the Z80C CPU is fetching an instruction from memory ; when M1 is active while IORQ is active, the SIO accepts M1 and IORQ as an interrupt acknowledge if the SIO is the highest priority device that has interrupted the Z80C CPU.

RxCA, **RxCB**. *Receiver Clocks* (Inputs). Receive data is sampled on the rising edge of RxC. The Receive Clocks may be 1, 16, 32 or 64 times the data rate in asynchronous modes. These clocks may be driven by the Z80C CTC Counter Timer Circuit for programmable baud rate generation. Both inputs are Schmitt-trigger buffered (no noise level margin is specified).

In the Z80C SIO-0 bonding option, RxCB is bonded together with TxCB.

RD. *Read Cycle Status* (Input from CPU, Active Low). If RD is active, a memory or I/O read operation is in progress. RD is used with B/A, CE and IORQ to transfer data from the SIO to the CPU.

RxDA, **RxDB**. *Receive Data* (Inputs, Active High). Serial data at TTL levels.

RESET. *Reset* (Input, Active Low). A Low RESET disables both receivers and transmitters, forces TxDA and TxDB marking, forces the modem con-

Figure 5 : ZCC SIO-0 Logic Functions.

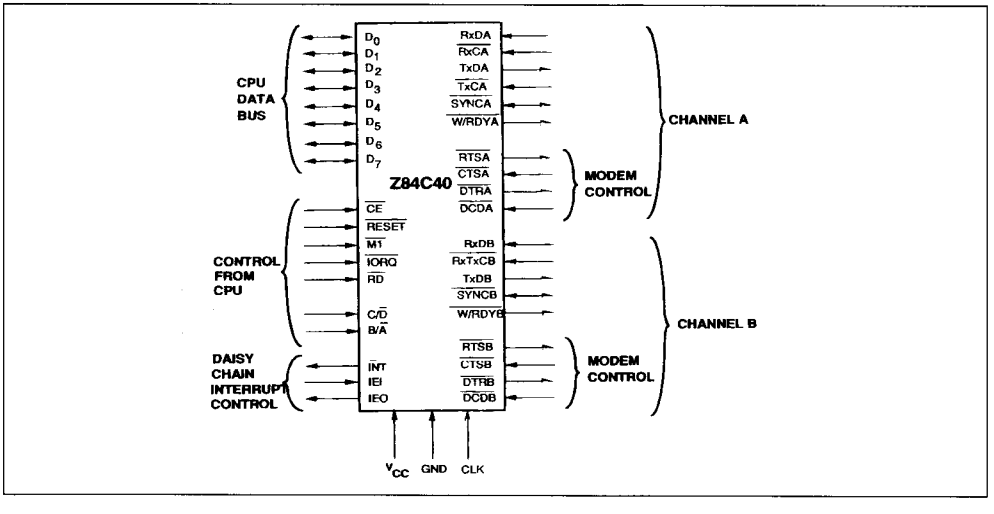


Figure 6 : Z80C SIO-0 Dual in Line Pin Configuration.

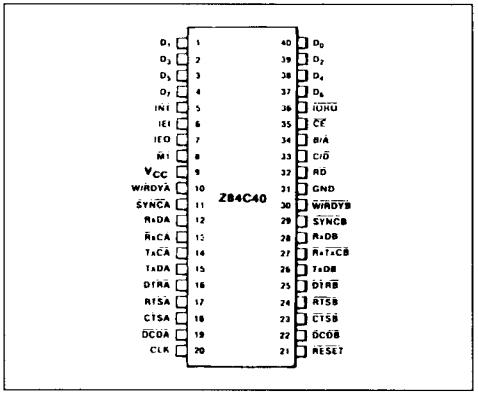
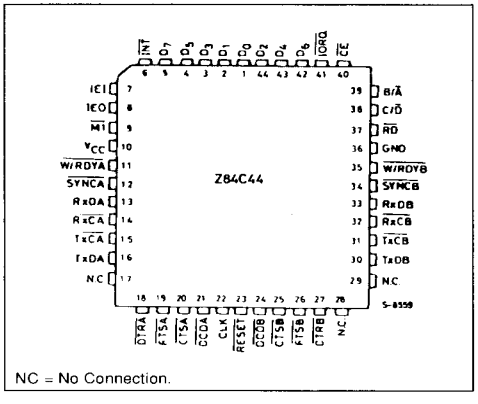


Figure 7 : Chip Carrier Pin Configuration.



NC = No Connection.

trols High and disables all interrupts. The control registers must be rewritten after the SIO is reset and before data is transmitted or received.

RTSA, **RTSB**. *Request To Send* (Outputs, Active Low). When the RTS bit in Write register 5 (figure 14) is set, the RTS output goes Low. When the RTS bit is reset in the Asynchronous mode, the output goes High after the transmitter is empty. In Synchronous modes, the RTS pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.

SYNCA, **SYNCB**. *Synchronization* (Inputs/Outputs, Active Low). These pins can act either as inputs or outputs. In the asynchronous receive mode, they are inputs similar to CTS and DCD. In this mode, the transitions on these lines affect the state of the Sync/Hunt status bits in Read Register 0 (figure 14), but have no other function. In the External Sync mode, these lines also act as inputs. When external synchronization is achieved, SYNC must be driven Low on the second rising edge of RxC after that rising edge of RxC on which the last bit of the sync character was received. In other words, after the sync pattern is detected, the external logic must wait for two full Receive Clock cycles to activate the SYNC input. Once SYNC is forced Low, it should be kept Low until the CPU informs the external synchronization detect logic that synchronization has been lost or a new message is about to start. Character assembly begins on the rising edge of RxC that immediately precedes the falling edge of SYNC in the External Sync mode.

In the internal synchronization mode (Monosync and Bisync) these pins act as outputs that are active during the part of the receive clock (RxC) cycle in which sync characters are recognized. The sync condition is not latched so these outputs are active each time a sync pattern in recognized, regardless of character boundaries.

In the Z80C SIO-2 bonding option, SYNCB is omitted.

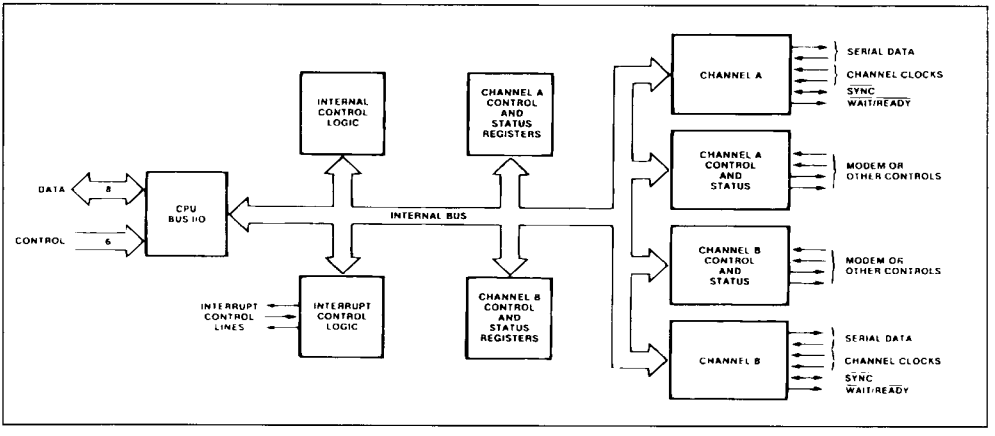
TxCA, **TxCB**. *Transmitter Clocks* (Inputs). In asynchronous modes, the Transmitter Clocks may be 1, 16, 32 or 64 times the data rate ; however, the clock multiplier for the transmitted and the receiver must be the same. The transmit Clock inputs are Schmitt-trigger buffered for relaxed rise- and fall-time requirements (no noise level margin is specified).

Transmitter Clocks may be driven by the Z80C CTC Counter Timer Circuit for programmable baud rate generation. In the Z80C SIO-0 bonding option, TxCB is bonded together with RxCB.

TxDA, **TxDB**. *Transmitt Data* (Outputs, Active High). Serial data at TTL levels. TxD changes from the falling edge of TxC.

W/RDYA, **W/RDYB**. *Wait/Ready A, Wait/Ready B* (Outputs, Open Drain when Programmed for Wait Function, Driven High and Low when Programmed for Ready Function). These dual-purpose outputs may be programmed as Ready lines for a DMA controller or as Wait lines that synchronize the CPU to the SIO data rate. The reset state is open drain.

Figure 8 : Block Diagram.



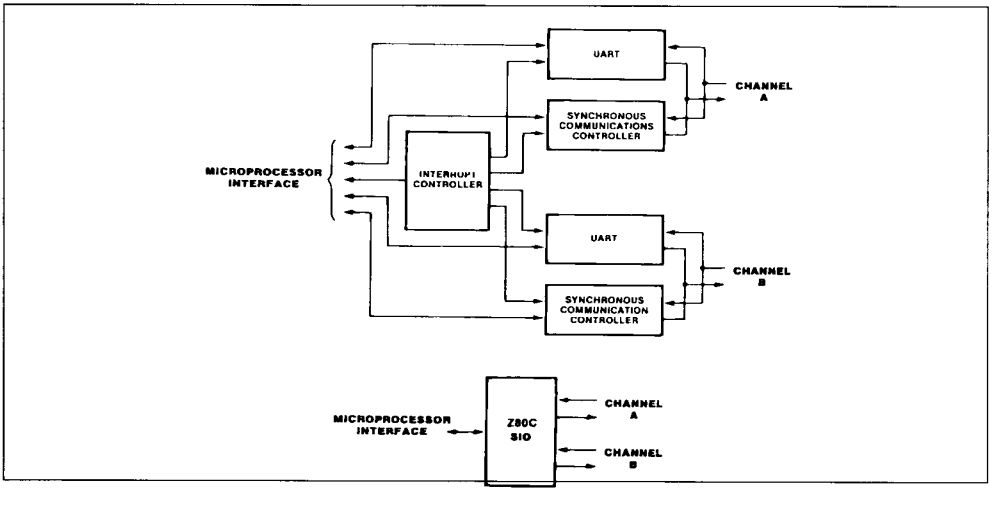
FUNCTIONAL DESCRIPTION

The functional capabilities of the Z80C SIO can be described from two different points of view : as a data communication device, it transmits and receives serial data in a wide variety of data-communication protocols ; as a Z80C family peripheral, it interacts with the Z80C CPU and other peripheral circuits, sharing the data, address and control buses, as well as being a part of the Z80C interrupt structure. As a peripheral to other microprocessors, the SIO offers valuable features such as non-vectored interrupts, polling and simple handshake capability.

Figure 9 illustrates the conventional devices that the SIO replaces.

The first part of the following discussion covers SIO data-communication capabilities ; the second part describes interactions between the CPU and the SIO.

Figure 9 : Conventional Devices Replaced by Z80C SIO.



DATA COMMUNICATION CAPABILITIES

The SIO provides two independent full-duplex channels that can be programmed for use in any common asynchronous or synchronous data-communication protocol. Figure 10 illustrates some of these protocols. The following is a short description of them. A more detailed explanation of these modes can be found in the *Z80 Family Technical Manual.*

ASYNCHRONOUS MODES

Transmission and reception can be done 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 end of a received break. Reception is protected from spikes by a transient spikerejection mechanism that checks the signal one-half a bit time after a Low level is detected on the receive data input (RxDA or RxDB in figure 6). If the Low does not persist – as in the case of a transient – the character assembly process is not started.

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

The SIO does not require symmetric transmit and receive clock signals – a feature that allows it to be used with a Z80C CTC or many other clock sources. The transitter 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 $\overline{\text{SYNC}}$ pin may be programmed as an input that can be used for functions such as monitoring a ring indicator.

SYNCHRONOUS MODES

The SIO supports both byte-oriented and bit oriented synchronous communication.

Synchronous byte-oriented protocols can be handled in several modes that allow character synchronization with an 8-bit sync character (Monosync), any 16-bit sync pattern (Bysinc), or with an external sync signal. Leading sync characters can be removed without interrupting the CPU.

Five-, six- or seven-bit sync characters are detected with 8- or 16-bit patterns in the SIO by overlapping the larger pattern across multiple in-coming sync characters, as shown in figure 11.

CRC checking for synchronous byte-oriented modes is delayed by one character time so the CPU may disable CRC checking on specific characters. This permits 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. In all non-SDLC modes, the CRC generator is initialized to 0's ; in SDLC modes, it is initialized to 1's. The SIO can be used for interfacing to peripherals such as hard-sectored floppy disk, but it cannot generate or check CRC for IBM-compatible soft-sectored disks. The SIO also provides a feature that automatically transmits CRC data when no other data is available for transmissions. This allows very high-speed transmissions under DMA control with no need for CPU intervention at the end of a message. When there is no data or CRC to send in synchronous modes, the transmitter inserts 8- or 16-bit sync characters regardless of the programmed character length.

The SIO supports synchronous bit-oriented protocols such as SDLC and HDLC by performing automatic flag seding, zero insertion and CRC generation. A special command can be used to abort a frame in transmission. At the end of a message the SIO automatically transmits the CRC and trailing flag when the transmit buffer becomes empty. 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. One to eight bits per character can be sent, which allows reception of a message with no prior information about the character structure in the information field of a frame.

The receiver automatically synchronizes on the leading flag of a frame in SDLC or HDLC, and provides a synchronization signal on the $\overline{\text{SYNC}}$ pin ; an interrupt can also be programmed. The receiver can be programmed to search for frames addressed by a single byte to only a specified user-selected address or to a global broadcast address. In this mode, frames that do not match either the user-selected or broadcast address are ignored. The number of address bytes can be extended under software control. For transmitting data, an interrupt on the first received character or on every character can be selected. The receiver automatically deletes all zeroes inserted by the transmitter during character assembly. It also calculates and automatically checks the CRC to validate frame transmission. At the end of

transmission, the status of a received frame is available in the status registers.

The SIO can be conveniently used under DMA control to provide high-speed reception or transmission. In reception, for example, the SIO 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 SIO 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.

Figure 10 : Some Z80C SIO Protocols.

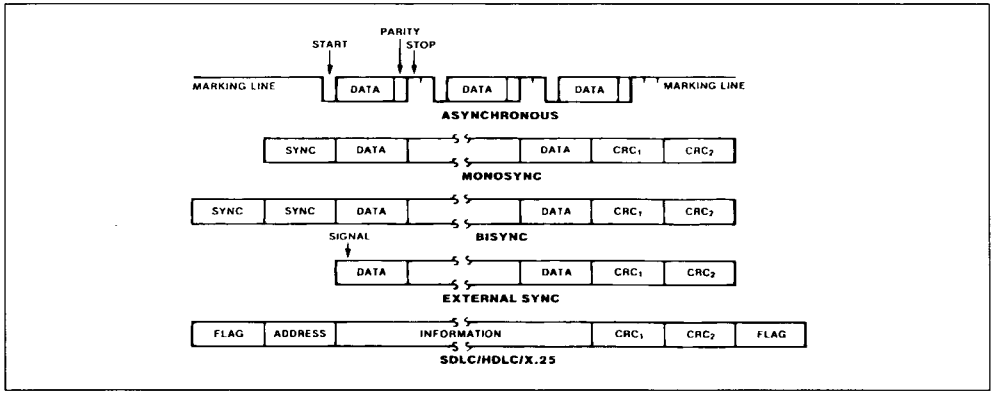
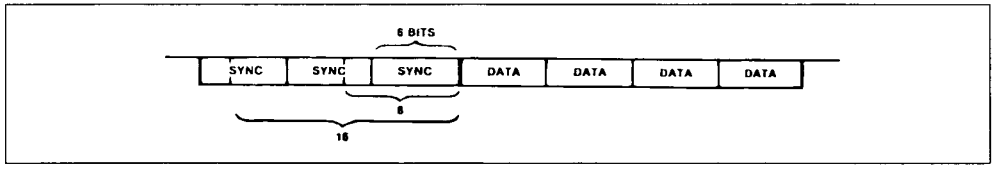


Figure 11 : Six Bit Sync Character Recognition.



STATUS FLOW-CHART

Figure 12a : Status Flowchart.

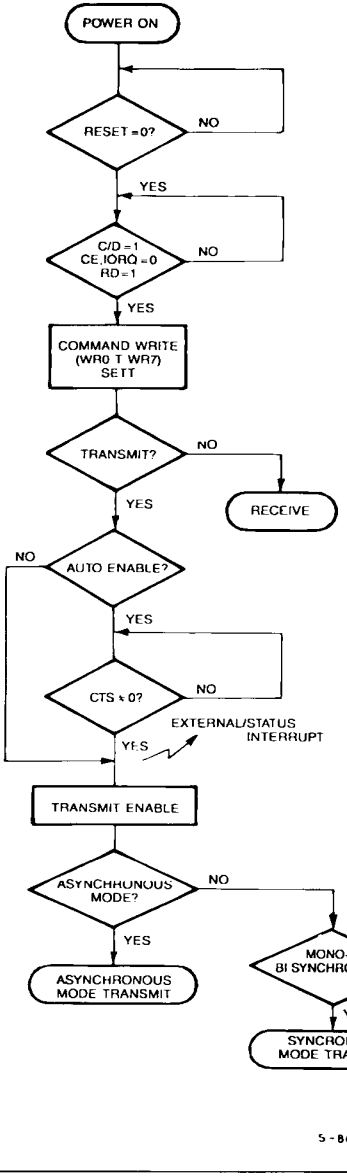


Figure 12b : Status Flowchart.

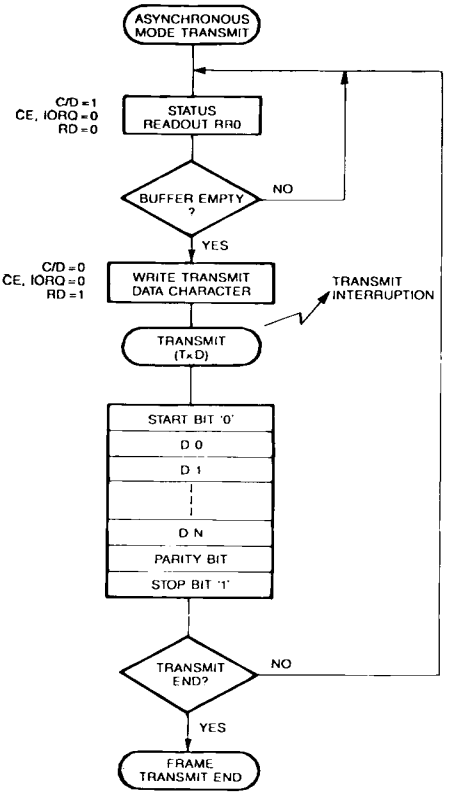


Figure 12c : Status Change Flowchart.

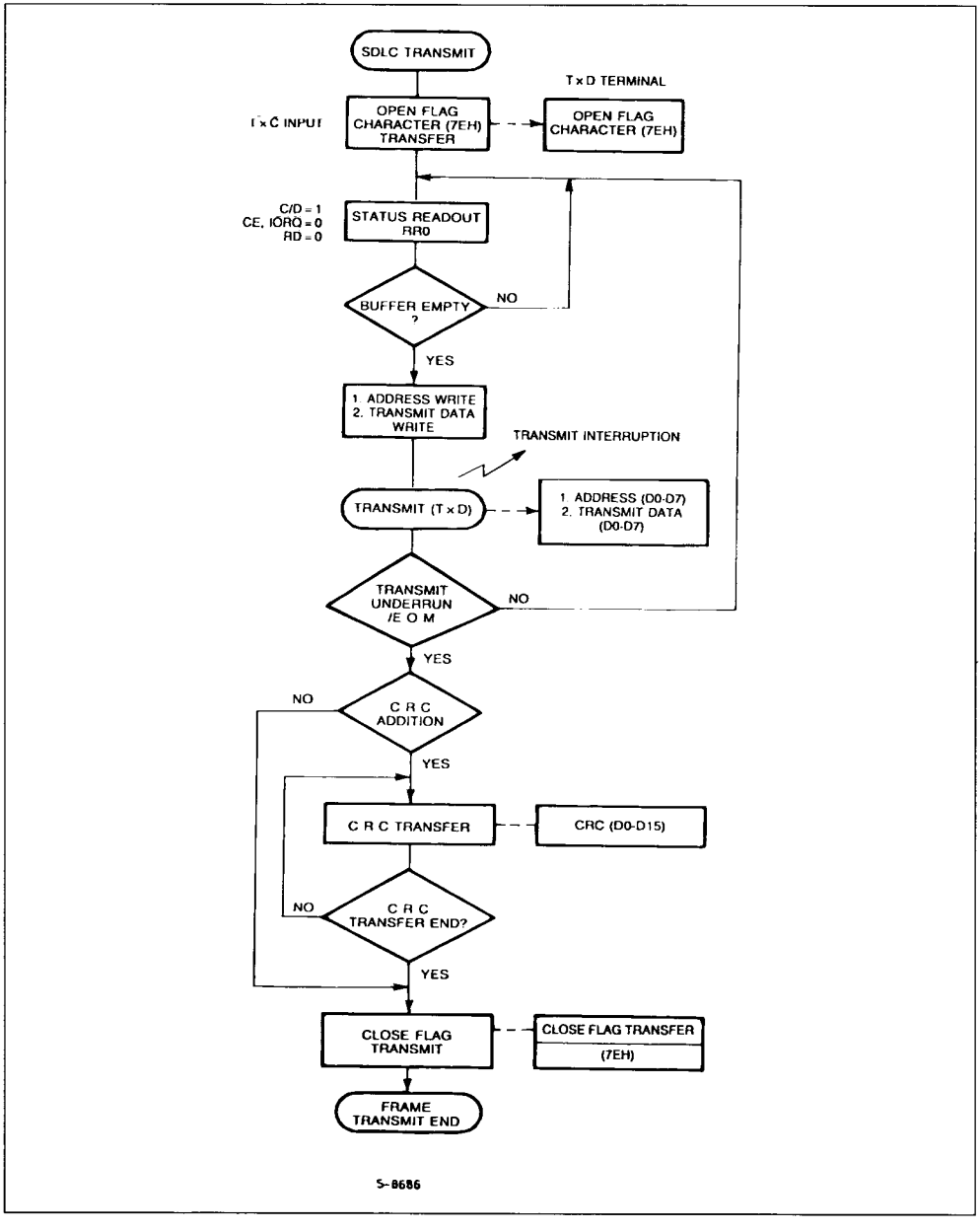


Figure 12d : Status Change Flowchart.

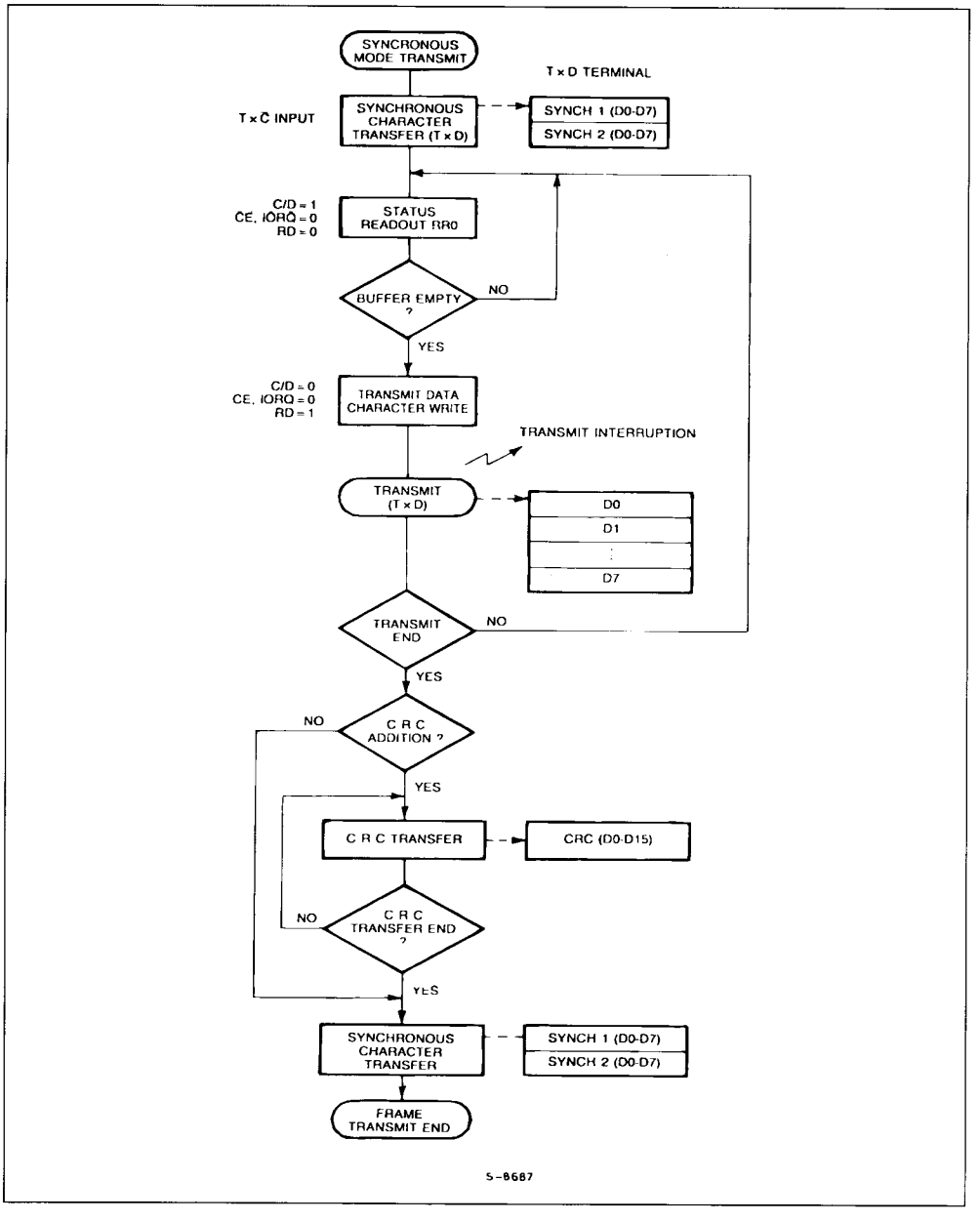
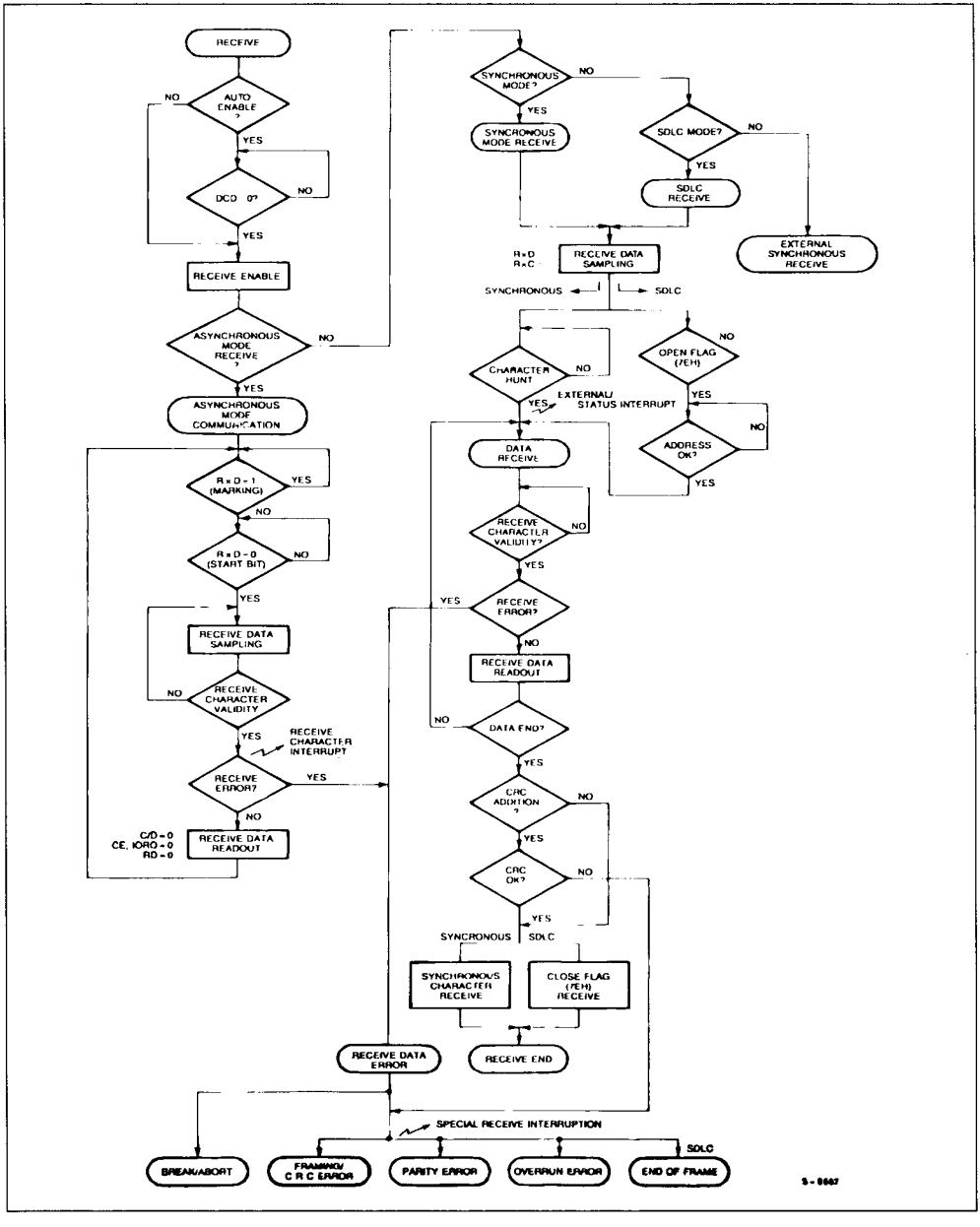


Figure 12e : Status Flowchart.



I/O INTERFACE CAPABILITIES

The SIO offers the choice of polling, interrupt, (vectored or non-vectored) and block-transfers modes to transfer data, status and control information to and from the CPU. The block-transfer mode can also be implemented under DMA control.

POLLING

Two status registers are updated at appropriate times for each function being performed (for example, CRC error-status valid at the end of a message). When the CPU is operated in a polling fashion, one of the SIO's two status registers is used to indicate whether the SIO has some data or needs some data.

Depending on the contents of this register, the CPU will either write data, read data, or just go on. Two bits in the register indicate that a data transfer is needed. In addition, error and other conditions are indicate. The second status register (special receive conditions) does not have to be read in a polling sequence, until a character has been received. All interrupt modes are disabled when operating the device in a polled environment.

INTERRUPTS

The SIO has an elaborate interrupt scheme to provide fast interrupt service in real-time applications. A control register and a status register in Channel B contain the interrupt vector. When programmed to do so, the SIO can modify three bits of the interrupt vector in the status register so that it points directly to one of eight interrupt service routines in memory, thereby servicing conditions in both channels and eliminating most of the needs for a status-analysis routine.

Transmitt interrupts, receive interrupts and external/status interrupts are the main sources of interrupts. Each interrupt source is enabled under program control, with Channel A having a 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 by the transmit buffer becoming empty. (This implies that the transmitter must have had a data character written into it so it can become empty). The receiver can interrupt the CPU in one or two ways :

- Interrupt on first received character
- Interrupt on all received characters

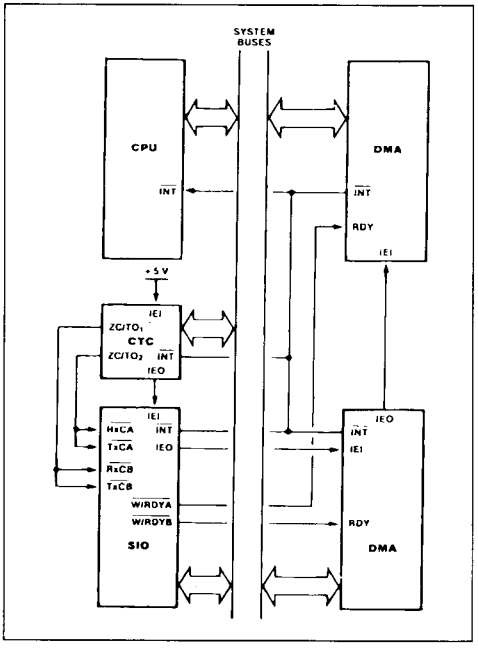
Interrupt-on-first-received-character is typically used with the block-transfer mode. Interrupt-on-all-received-characters has the option of modifying the interrupt vector in the event of a parity error. Both of these interrupt modes will also interrupt under special receive conditions on a character or message basis (end-of-frame interrupt in SDLC, for example).

This means that the special-receive condition can cause an interrupt only if the interrupt-on-first-received-character or interrupt-on-all-received-characters mode is selected. In interrupt-on-first-received-character, an interrupt can occur from special-receive conditions (except parity error) after the first-received-character interrupt (example : receive-overrun interrupt).

The main function of the external/status interrupt is to monitor the signal transitions of the Clear To Send (CTS), Data Carrier Detect (DCD) and Synchronization (SYNC) pins (figures 1 through 6). In addition, an external/status interrupt is also caused by a CRC-sending condition or by the detection of a break sequence (asynchronous mode) or abort sequence (SDLC mode) in the data stream.

The interrupt caused by the break/abort sequence allows the SIO to interrupt when the break/abort sequence is detected or terminated. This feature facilitates the proper temination of the current message,

Figure 13 : Typical Z80C Environment.



correct initialization of the next message, and the accurate timing of the break/abort condition in external logic.

In a Z80C CPU environment (figure 13), SIO interrupt vectoring is "automatic" : the SIO passes its internally-modificable 8-bit interrupt vector to the CPU, which adds an additional 8 bits from its interrupt-vector (I) register to from the memory address of the interrupt-routine table. This table contains the address of the beginning of the interrupt routine itself. The process entails an indirect transfer or CPU control to the interrupt routine, so that the next instruction executed after an interrupt acknowledge by the CPU is the first instruction of the interrupt routine itself.

CPU/DMA BLOCK TRANSFER

The SIO's block-transfer mode accommodates both CPU block transfers and DMA controllers (Z80C DMA or other designs). The block-transfer mode uses the Wait/Ready output signal, which is selected with three bits in an internal control register. The Wait/Ready output signal can be programmed as a WAIT line in the CPU block-transfer mode or as a READY line in the DMA block-transfer mode.

To a DMA controller, the SIO READY output indicates that the SIO is ready to transfer data to or from memory. To the CPU, the WAIT output indicates the SIO is not ready to transfer data, thereby requesting the CPU to extend the I/O cycle.

INTERNAL STRUCTURE

The internal structure of the device includes a Z80C CPU interface, internal control and interrupt logic, and two full-duplex channels.

Each channel contains its own set of control and status (write and read) registers, and control and status logic that provides the interface to modems or other external devices.

The registers for each channel are designated as follows :

WR0-WR7 - Write Registers 0 through 7
RR0-RR2 - Read Register 0 through 2

The register group includes five 8-bit control registers, two sync-character registers and two status registers. The interrupt vector is written into an additional 8-bit register (Write Register 2) in Channel B that may be read through another 8-bit register (Read Register 2) in Channel B. The bit assignment and functional grouping of each register is configured to simplify and organize the programming process. Table 1 list the functions assigned to each read or write register.

The logic for both channels provides formats, synchronization and validation for data transferred to and from the channel interface. The modem control inputs, Clear To Send (CTS) and Data Carrier Detect (DCD), are monitored by the external control and status logic under program control. All external control-and-status-logic signals are general-purpose in nature and can be used for functions other than modem control.

DATA PATH

The transmit and receive data path illustrated for Channel A in figure 13 is identical for both channels. The receiver has three 8-bit buffer registers in a FIFO arrangement, in addition to the 8-bit receive shift register.

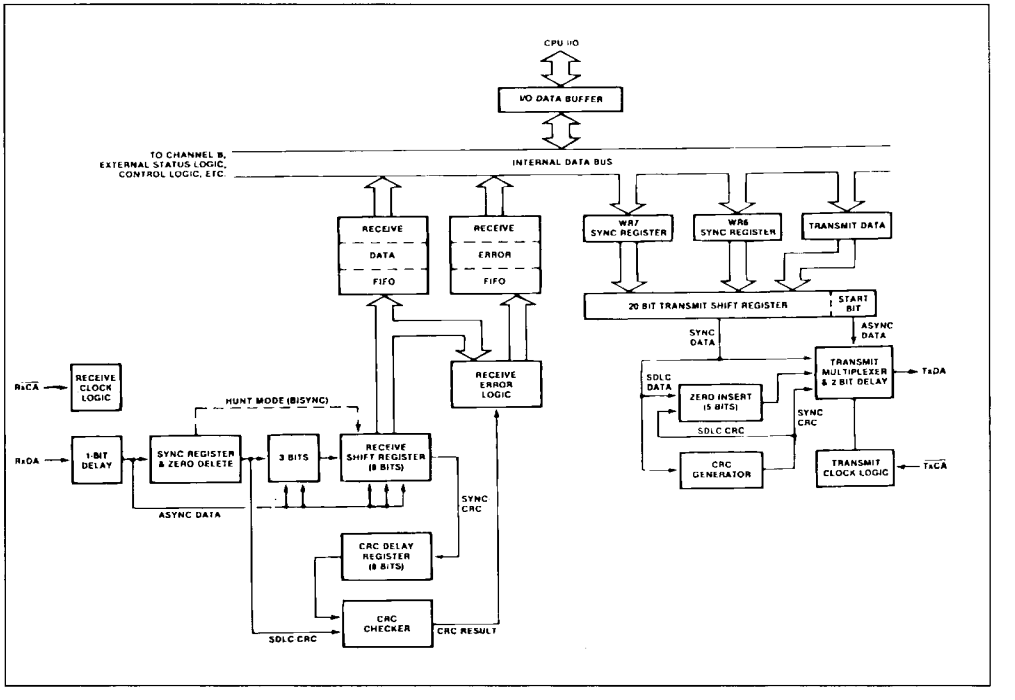
This scheme creates additional time for the CPU to service an interrupt at the beginning of a block of high-speed data. Incoming data is routed through one of several paths (data or CRC) depending on the selected mode andin asynchronous modesthe character length.

Read Register Functions	
RR0	Transmit/Receive Buffer Status, Interrupt Status and External Status
RR1	Special Receive Condition Status
RR2	Modified Interrupt Vector (channel B only)
Write Register Functions	
WR0	Register pointers, CRC initialize, initialization commands for the various modes, etc.
WR1	Transmit/Receive Interrupt and Data Transfer Mode Definition
WR2	Interrupt Vector (channel B only)
WR3	Receive Parameters and Control
WR4	Transmit/Receive Miscellaneous Parameters and Modes
WR5	Transmit Parameters and Controls
WR6	Sync Character or SDLC Address Field
WR7	Sync Character or SDLC Flag

The transmitter has an 8-bit transmit data buffer register that is loaded from the internal data bus, and a 20-bit transmit shift register that can be loaded from the sync-character buffers or from the transmit data register.

Depending on the operational mode, outgoing data is routed throught one of four main paths before it is transmitted from the Transmit Data output TxD).

Figure 14 : Transmit and Receive Data Path (channel A).



PROGRAMMING

The system program first issues a series of commands that initialize the basic mode of operation and then other commands that 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 ; and finally, receiver or transmitter enable.

Both channels contain registers that must be programmed via the system program prior operation. The channel-select input (B/$\overline{A}$) and the control/data input (C/$\overline{D}$) are the command-structure addressing controls, and are normally controlled by the CPU address bus. Figures 17 and 18 illustrate the timing relationships for programming the write registers and transfering data and status.

READ REGISTER

The SIO contains three read registers for Channel B and two read registers for Channel A (RR0-RR2 in figure 14) that can be to obtain the status information ; RR2 contains the internally-modifiable interrupt vector and is only in the Channel B register set. The status information includes error conditions, interrupt vector and standard communications-interface signals.

To read the contents of a selected read register other than RR0, the system program must first write the pointer byte to WR0 in exactly the same way as a write register operation. Then, by executing a read instruction, the contents of the addressed read register can be read by the CPU.

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

WRITE REGISTERS

The SIO contains eight write registers for Channel B and seven write registers for Channel A (WR0-WR7 in figure 15) that are programmed separately to configure the functional personality of the channels ; WR2 contains the interrupt vector for both channels and is only in the Channel B register set. With the exception of WR0, programming the write registers requires two bytes. The first byte is to WR0 and contains three bits (D_0-D_2) that point to the selected register ; the second byte is the actual control word that is written into the register to configure the SIO.

WR0 is a special case in that all of the basic commands can be written to it with a single byte. Reset (internal or external) initializes the pointer bits D_0-D_2 to point to WR0. This implies that a channel reset must not be combined with the pointing to any register.

Figure 15 : Read Register Bit Functions.

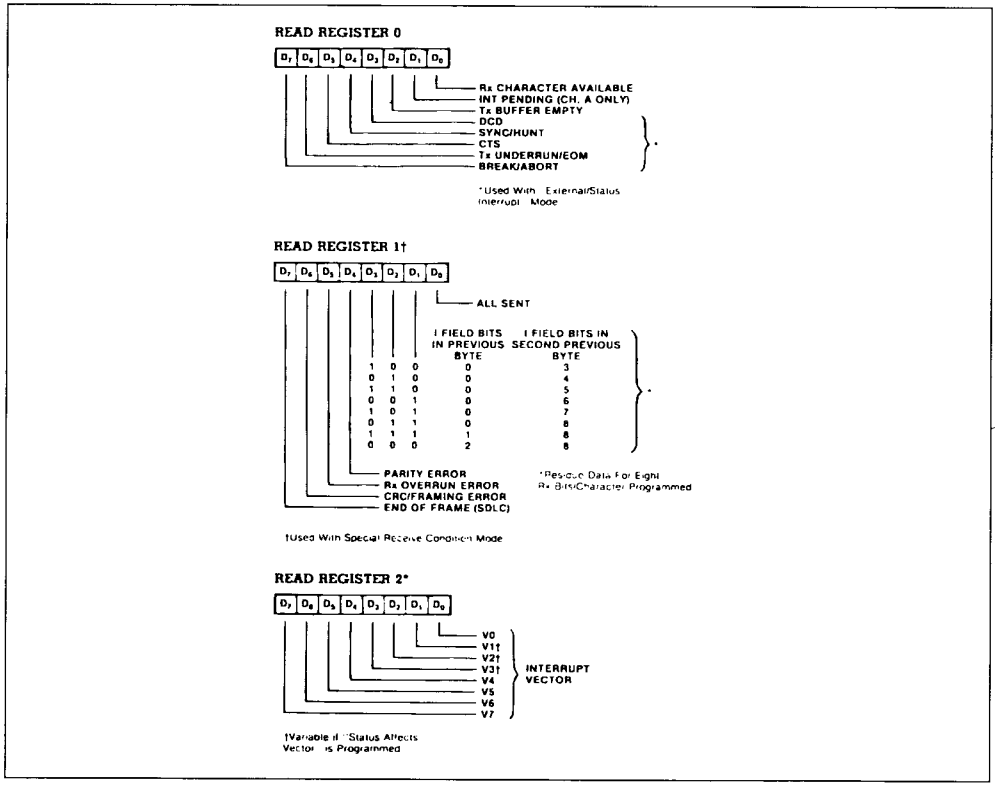
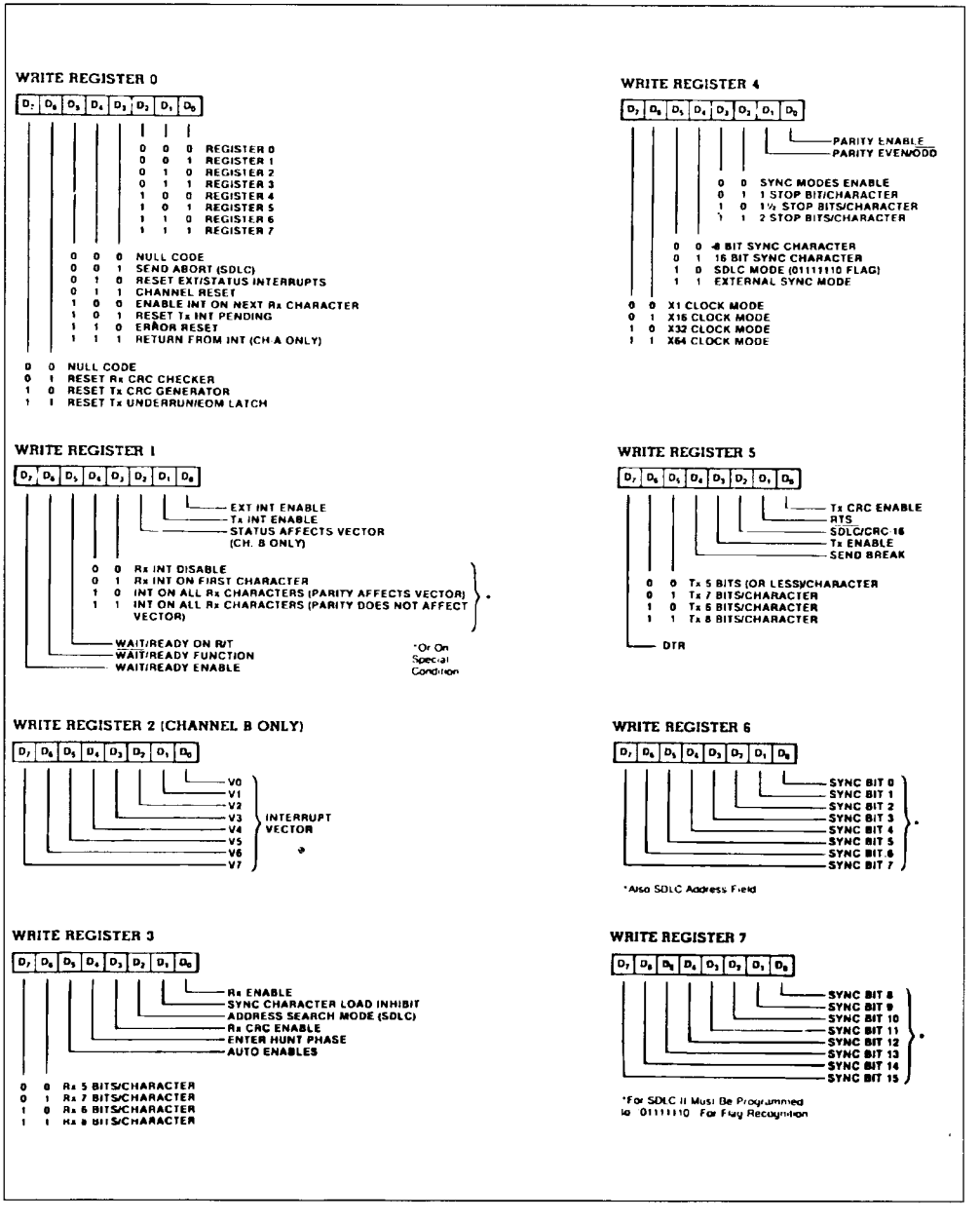


Figure 16 : Write Register Bit Functions.



TIMING

The SIO must have the same clock as the CPU (same phase and frequency relationship, not necessarily the same driver).

READ CYCLE

The timing signals generated by a Z80C CPU input instruction to read a data or status byte from the SIO are illustrated in figure 16.

WRITE CYCLE

Figure 16 illustrates the timing and data signals generated by a Z80C CPU output instruction to write a data or control byte into the SIO.

INTERRUPT-ACKNOWLEDGE CYCLE

After receiving an interrupt-request signal from an SIO (INT pulled Low), the Z80C CPU sends an interrupt-acknowledge sequence (M1 Low, and IORQ Low a few cycles later) as in figure 18.

The SIO contains an internal daisy-chained interrupt structure for prioritizing nested interrupts for the various functions of its two channels, and this structure can be used within an external user-defined daisy chain that prioritizes several peripheral circuits.

The IEI of the highest-priority device is terminated High. A device that has an interrupt pending or under service forces its IEO Low. For devices with no interrupt pending or under service, IEO = IEI.

To insure stable conditions in the daisy chain, all interrupt status signals are prevented from changing while M1 is Low. When IORQ is Low, the highest priority interrupt requestor (the one with IEI High) places its interrupt vector on the data bus and sets its internal interrupt-under-service latch.

RETURN FROM INTERRUPT CYCLE

Figure 19 illustrates the return form interrupt cycle. Normally, the Z80C CPU issues a RETI (Return From Interrupt) instruction at the end of an interrupt sevice routine. RETI is a 2-byte opcode (ED-4D) that resets the interrupt-under-service latch in the SIO to terminate the interrupt that has just been processed. This is accomplished by manipulating the daisy chain in the following way.

The normal daisy-chain operation can be used to detect a pending interrupt ; however, it cannot distinguish between an interrupt under service and a pending unacknowledged interrupt of a higher priority. Whenever "ED" is decoded, the daisy chain is modified by forcing High the IEO of any interrupt that has not yet been acknowledged. Thus the daisy chain identifies the device presently under service as the only one with an IEI High and an IEO Low. If the next opcode byte is "4D", the interrupt-under-service latch is reset.

The ripple time of the interrupt daisy chain (both the High-to-Low and the Low-to-High transitions) limits the number of devices that can be placed in the daisy chain. Ripple time can be improved with carry-look-ahead, or by extending the interrupt-acknowledge cycle.

For further information about techniques for increasing the number of daisy-chained devices, refer to the Z80C CPU Data Sheet.

Figure 17 : Read Cycle.

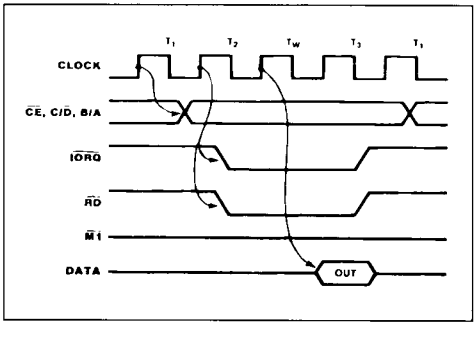


Figure 18 : Write Cycle.

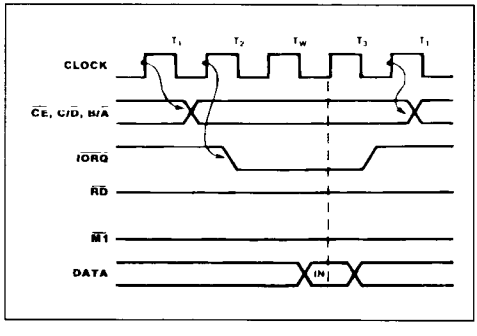


Figure 19 : Interrupt Acknowledge Cycle.

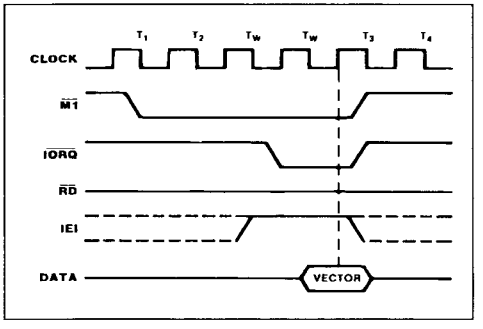


Figure 20 : Return from Interrupt Cycle.

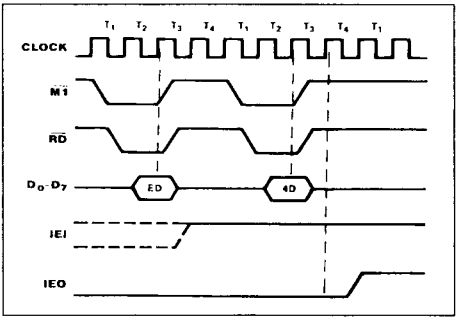
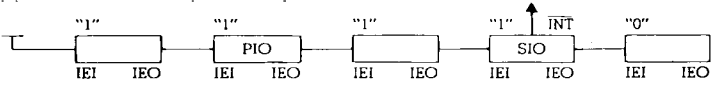
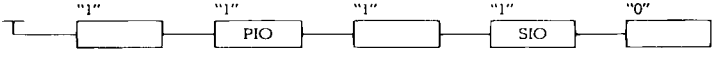


Figure 21 : Daisy chain at RETI Instruction.

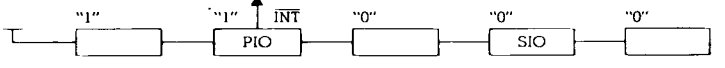
[1] Status where SIOs Request Interruption.



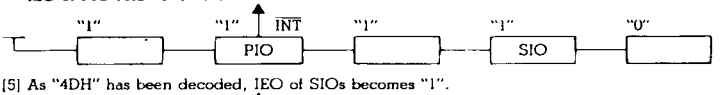
[2] SIOs are under Interruption Service



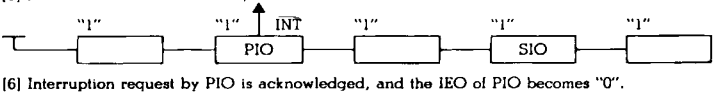
[3] Status where PIO has requested interruption immediately before SIOs decode "ED". By the request of PIO for interruption IEO of PIO becomes "0".



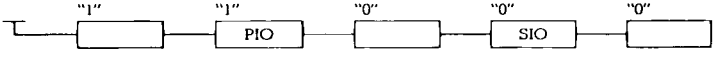
[4] As "EDH" has been decoded, the interruption request of PIO is not acknowledged. Therefore, IEO of PIO returns to "1".



[5] As "4DH" has been decoded, IEO of SIOs becomes "1".



[6] Interruption request by PIO is acknowledged, and the IEO of PIO becomes "0".



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	V_{CC} Supply Voltage with Respect to V_{SS}	– 0.5 to 7	V
V_{IN}	Input Voltage	– 0.5 to V_{CC} + 0.5	V
P_D	Power Dissipation (T_A = 85 °C)	250	mW
T_{SOLDER}	Soldering Temperature (soldering time 10 sec)	260	°C
T_{stg}	Storage Temperature	– 65 to 150	°C
T_{op}	Operating Temperature	– 40 to 85	°C

CAPACITANCE (T_A = 25°C)

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Unit
C_{CLOCK}	Clock Capacitance	f = 1MHz	–	–	7	pF
C_{IN}	Input Capacitance		–	–	5	pF
C_{OUT}	Output Capacitance		–	–	10	pF

DC CHARACTERISTICS (T_A = – 40 °C to 85 °C, V_{CC} = 5 V ± 10 %, V_{SS} = 0 V)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_{ILC}	Clock Input Low Voltage		– 0.3	–	0.6	V
V_{IHC}	Clock Input High Voltage		V_{CC} – 0.6	–	V_{CC} + 0.3	V
V_{IL}	Input Low Voltage (except CLK)		– 0.5	–	0.8	V
V_{IH}	Input High Voltage (except CLK)		2.2	–	V_{CC}	V
V_{OL}	Output Low Voltage	I_{OL} = 2.0 mA	–	–	0.4	V
V_{OH1}	Output High Voltage (1)	I_{OH} = – 1.6 mA	2.4	–	–	V
V_{OH2}	Output High Voltage (2)	I_{OH} = – 250 µA	V_{CC} – 0.8	–	–	V
I_{LI}	Input Leakage Current	$V_{SS} \le V_{IN} \le V_{CC}$	–	–	± 10	µA
I_{LO}	3-State Output Leakage Current in Float	V_{SS} + 0.4 ≤ $V_{OUT} \le V_{CC}$	–	–	± 10	µA
$I_{L(SY)}$	$\overline{\text{SYNC}}$ Pin Leakage Current	V_{SS} + 0.4 ≤ $V_{OUT} \le V_{CC}$ V_{CC} = 5 V, CLK = 4 MHz	– 40	–	10	µA
I_{CC1}	Operating Supply Current : 4 MHz 6 MHz	$V_{IH} = V_{IHC} = V_{CC}$ – 0.2 V $V_{IL} = V_{ILC}$ = 0.2 V V_{CC} = 5 V, fCLK = $1/T_CC$(min)	 – –	 2.5 4	 6 10	 mA mA
I_{CC2}	Stand-by Supply Current (except SYNC pin)	$V_{IH} = V_{CC}$ – 0.2 V CLK = V_{IL} = 0.2 V, V_{CC} = 5 V	–	–	10	µA

TEST CONDITIONS

T_A = – 40 °C to + 85 °C

V_{CC} = 5 V ± 10 %

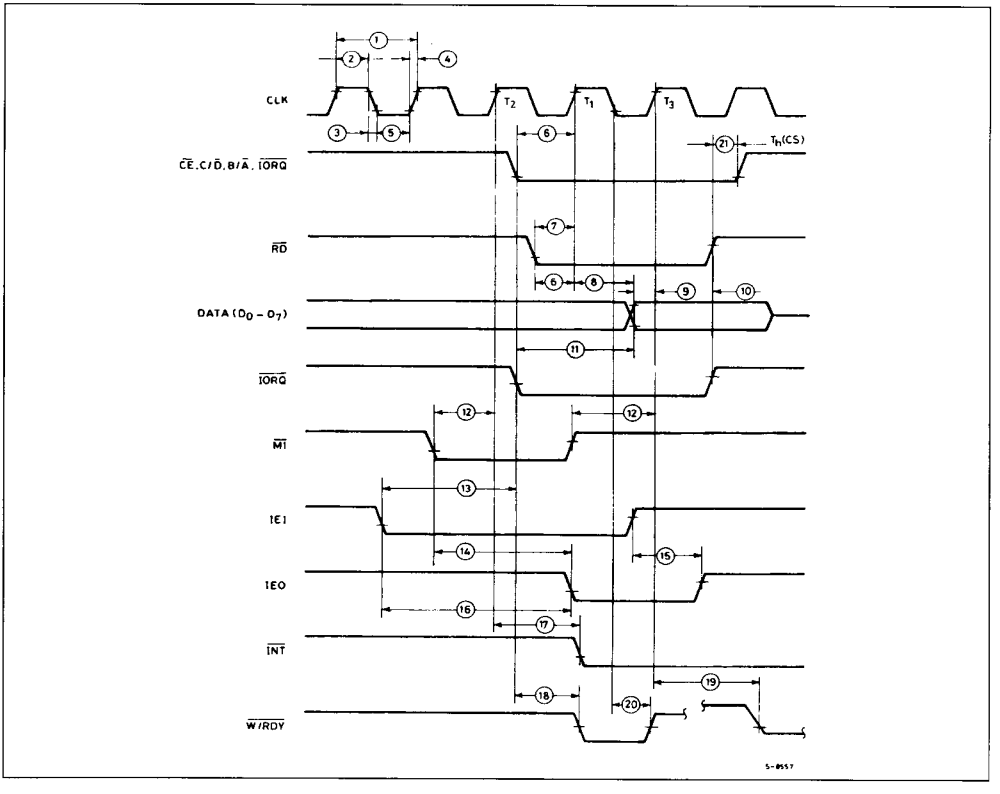
V_{SS} = 0 V

AC TEST CONDITIONS

- Inputs except CLK (clock) are driven at 2.4 V for a logic "1" and 0.4 V for a logic "0". Clock input is driven at V_{CC} – 0.6 V for a logic "1" and 0.6 V for a logic "0".
- Timing measurements are made at 2.2 V for a logic "1" and 0.8 V for a logic "0".

All AC parameters assume a load capacitance of 100 pF

AC CHARACTERISTICS

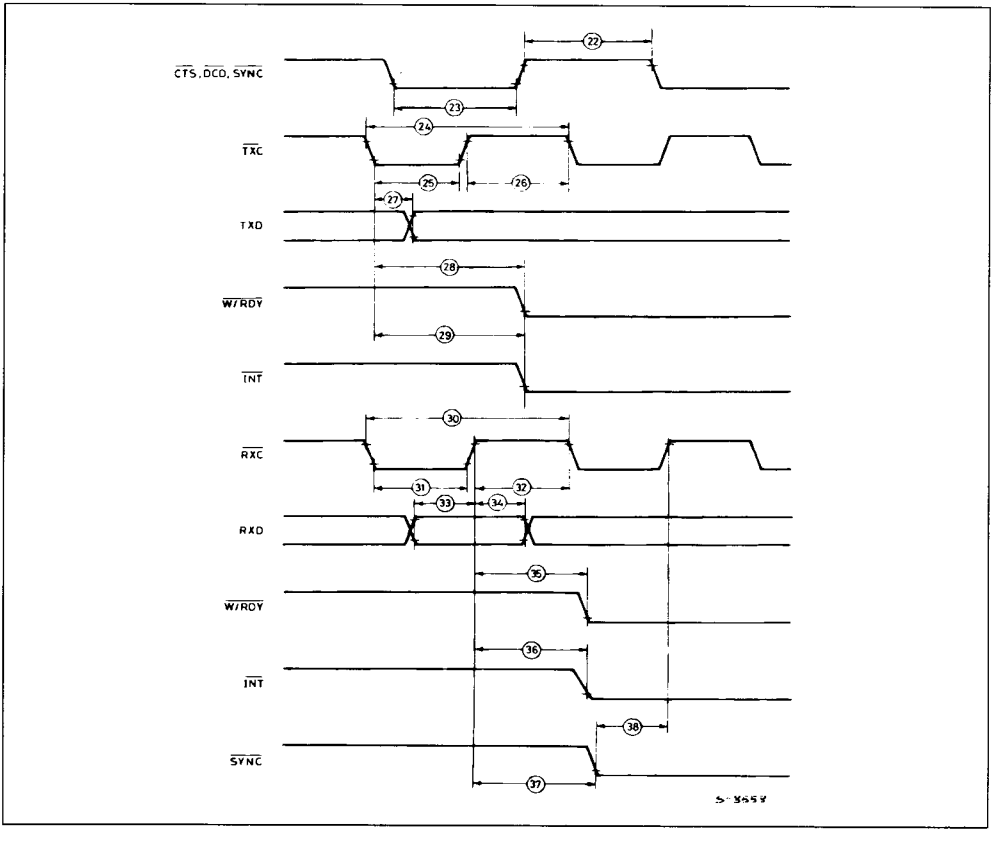


AC CHARACTERISTICS (continued)

N°	Symbol	Parameter	Z84C40 /1/2A		Z84C40 /1/2B		Unit
			Min.	Max.	Min.	Max.	
1	TcC	Clock Cycle Time	250	DC	165	DC	ns
2	TwCh	Clock Width (high)	105	DC	70	DC	ns
3	TfC	Clock Fall Time		30		15	ns
4	TrC	Clock Rise Time		30		15	ns
5	TwCl	Clock Width (low)	105	DC	70	DC	ns
6	TsCS(C)	$\overline{CE}$, C/$\overline{D}$, B/$\overline{A}$, $\overline{IORQ}$ to Clock ↑ Setup Time	145		60		ns
7	TsRD(C)	$\overline{RD}$ to Clock ↑ Setup Time	115		60		ns
8	TdC(D0)	Clock ↑ to Data Out Delay		220		150	ns
9	TsDI(C)	Data In to Clock ↑ Setup (write or MI Cycle)	50		30		ns
10	TdRD(D02)	RD ↑ to Data Out Float Delay		110		90	ns
11	TdI0(D01)	$\overline{IORQ}$ ↓ to Data Out Delay (INTACK cycle)		160		120	ns
12	TsMI(C)	$\overline{M1}$ to Clock ↑ Setup Time	90		75		ns
13	TsIEI(IO)	IEI to $\overline{IORQ}$ ↓ Setup Time (INTACK cycle)	140		120		ns
14	TdMI(IEO)	$\overline{M1}$ ↓ to IEO ↓ Delay (interrupt before M1)		190		160	ns
15*	TdIEI(IEOr)	IEI ↑ to IEO ↑ Delay (after ED decode)		160		110	ns
16	TdIEI(IEOf)	IEI ↓ to IEO ↓ Delay		100		70	ns
17	TdC(INT)	Clock ↑ to $\overline{INT}$ ↓ Delay		200		150	ns
18	TdILO(W/RWf)	$\overline{IORQ}$ ↓ or CE ↓ to $\overline{W}$/$\overline{RDY}$ ↓ Delay (wait mode)		210		175	ns
19	TdC(W/RR)	Clock ↑ to $\overline{W}$/$\overline{RDY}$ ↓ Delay (ready mode)		120		100	ns
20	TdC(W/RWZ)	Clock ↓ to $\overline{W}$/$\overline{RDY}$ Float Delay (wait mode)		130		110	ns
21	Th, Th(CS)	Any unspecified hold when setup is specified.	0		0		ns

Note : * Not compatible with NMOS Specifications.

AC CHARACTERISTICS (continued)



AC CHARACTERISTICS (continued)

N°	Symbol	Parameter	Z84C40 /1/2A		Z84C40 /1/2B		Unit
			Min.	Max.	Min.	Max.	
22	TwPH	Pulse Width (high)	200		200		ns
23	TwPl	Pulse Width (low)	200		200		ns
24	TcTxc	$\overline{Txc}$ Cycle Time	400		330		ns
25	TwTxcl	$\overline{Txc}$ Width (low)	180	∞	100	∞	ns
26	TwTxch	$\overline{Txc}$ Width (high)	180	∞	100	∞	ns
27	TdTxC(TxD)	$\overline{TXC}$ ↓ to TxD Delay (X1 mode)		300		220	ns
28	TdTxC(W/RRf)	$\overline{TXC}$ ↓ to $\overline{W/RDY}$ ↓ Delay (ready mode)	5	9	5	9	CLK Periods
29	TdTxC(INT)	$\overline{TXC}$ ↓ to $\overline{INT}$ ↓ Delay	5	9	5	9	CLK Periods
30	TcRxC	$\overline{Rxc}$ Cycle Time	400	∞	330	∞	ns
31	TwRxCl	$\overline{Rxc}$ Width (low)	180	∞	100	∞	ns
32	TwRxCh	$\overline{Rxc}$ Width (high)	180	∞	100	∞	ns
33	TsRxD(RxC)	RxD to RxC ↑ Setup Time (xl mode)	0		0		ns
34	ThRxD(RxC)	$\overline{RxC}$ ↑ to RxD Hold Time (xl mode)	140		100		ns
35	TdRxC(W/RRf)	$\overline{RxC}$ ↑ to $\overline{W/RDY}$ ↓ Delay (ready mode)	10	13	10	13	CLK Periods
36	TdRxC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ ↓ Delay	10	13	10	13	CLK Periods
37	TdRxC(SYNC)	$\overline{RxC}$ ↑ to $\overline{SYNC}$ ↓ Delay (output modes)	4	7	4	7	CLK Periods
38	TsSYNC(RxC)	$\underline{SYNC}$ ↓ to $\underline{RxC}$ ↑ Setup (external sync modes)	−100		100		ns

ORDERING INFORMATION

Type	Package	Temp.	Clock	Description
Z84C40/1/2AB6	DIP-40 (plastic)	−40/+ 85°C	4 MHz	Z80C Serial I/O Controller
Z84C40/1/2AD6	DIP-40 (ceramic)	−40/+ 85°C		
Z84C40/1/2AD2	DIP-40 (ceramic)	−55/+125°C		
Z84C44AC6	PLCC44 (plastic chip-carrier)	−40/+ 85°C		
Z84C40/1/2BB6	DIP-40 (plastic)	−40/+ 85°C	6 MHz	
Z84C40/1/2BD6	DIP-40 (ceramic)	−40/+ 85°C		
Z84C40/1/2BD2	DIP-40 (ceramic)	−55/+125°C		
Z84C44BC6	PLCC44 (plastic chip-carrier)	−40/+ 85°C		