Z8470 Z80® DART Dual Asynchronous Receiver/Transmitter

ZiLOG

Product Specification

FEATURES

- Two independent full-duplex channels with separate modem controls. Modem status can be monitored.
- **In x1 clock mode, data rates are 0 to 800K bits/second with a 4.0 MHz clock, or 0 to 1.2 M bits/second with a 6.0 MHz clock.**
- Receiver data registers are quadruply buffered; the transmitter is doubly buffered.
- Programmable options include 1, 1½, or 2 stop bits; even, odd, or no parity; and x1, x16, x32, and x64 clock modes.
- Break generation and detection as well as parity-, overrun-, and framing-error detection are available.
- Interrupt features include a programmable interrupt vector, a "status affects vector" mode for fast interrupt processing, and the standard Z80 peripheral daisy-chain interrupt structure that provides automatic interrupt vectoring with no external logic.
- On-chip logic for ring indication and carrier-detect status.

2

GENERAL DESCRIPTION

The Z80 DART (Dual-Channel Asynchronous Receiver/Transmitter) is a dual-channel multifunction peripheral component that satisfies a wide variety of asynchronous serial data communications requirements in microcomputer systems. The Z80 DART is used as a serial-to-parallel, parallel-to-serial converter/controller in asynchronous applications. In addition, the device also provides modem controls for both channels. In applications where modem controls are not needed, these lines can be used for general-purpose I/O.

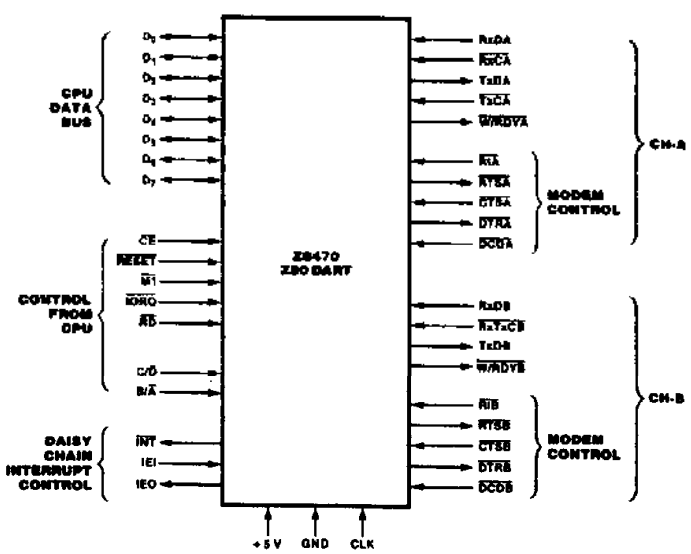


Figure 1. Pin Functions

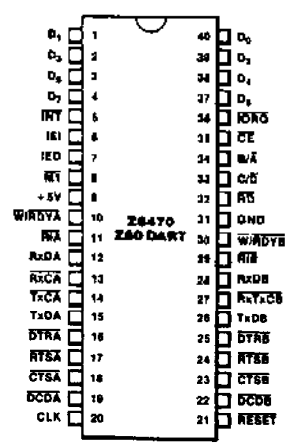


Figure 2. 40-Pin Dual-In-Line Package (DIP), Pin Assignments

Zilog also offers the Z80 SIO, a more versatile device that provides synchronous (Bisync, HDLC, and SDLC) as well as asynchronous operation.

The Z80 DART is fabricated with n-channel silicon-gate depletion-load technology, and is packaged in a 40-pin plastic or ceramic DIP (Figures 1 and 2).

PIN DESCRIPTION

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 Z80 DART.

C/$\overline{D}$. *Control or Data Select* (input, High selects Control). This input specifies the type of information (control or data) transferred on the data bus between the CPU and the Z80 DART.

$\overline{CE}$. *Chip Enable* (input, active Low). A Low at this input enables the Z80 DART 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 Z80 DART uses the standard Z80 single-phase system clock to synchronize internal signals.

$\overline{CTSA}$, $\overline{CTSB}$. *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.

D_0-D_7. *System Data Bus* (bidirectional, 3-state). This bus transfers data and commands between the CPU and the Z80 DART.

$\overline{DCDA}$, $\overline{DCDB}$. *Data Carrier Detect* (inputs, active Low). These pins function as receiver enables if the Z80 DART is programmed for Auto Enables; otherwise they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered.

$\overline{DTRA}$, $\overline{DTRB}$. *Data Terminal Ready* (outputs, active Low). These outputs follow the state programmed into the DTR bit. They can also be programmed as general-purpose outputs.

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 Z80 DART. Thus, this signal blocks lower priority devices from interrupting while a higher priority device is being serviced by its CPU interrupt service routine.

$\overline{INT}$. *Interrupt Request* (output, open drain, active Low). When the Z80 DART is requesting an interrupt, it pulls $\overline{INT}$ Low.

$\overline{M1}$. *Machine Cycle One* (input from Z80 CPU, active Low). When $\overline{M1}$ and $\overline{RD}$ are both active, the Z80 CPU is fetching an instruction from memory; when $\overline{M1}$ is active while $\overline{IORQ}$ is active, the Z80 DART accepts $\overline{M1}$ and $\overline{IORQ}$ as an interrupt acknowledge if the Z80 DART is the highest priority device that has interrupted the Z80 CPU.

$\overline{IORQ}$. *Input/Output Request* (input from CPU, active Low). $\overline{IORQ}$ is used in conjunction with B/$\overline{A}$, C/$\overline{D}$, $\overline{CE}$, and $\overline{RD}$ to transfer commands and data between the CPU and the Z80 DART. When $\overline{CE}$, $\overline{RD}$, and $\overline{IORQ}$ are all active, the channel selected by B/$\overline{A}$ transfers data to the CPU (a read operation). When $\overline{CE}$ and $\overline{IORQ}$ are active, but $\overline{RD}$ is inactive, the channel selected by B/$\overline{A}$ is written to by the CPU with either data or control information as specified by C/$\overline{D}$.

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.

$\overline{RD}$. *Read Cycle Status* (input from CPU, active Low). If $\overline{RD}$ is active, a memory or I/O read operation is in progress.

RxDA, RxDB. *Receive Data* (inputs, active High).

$\overline{RESET}$. *Reset* (input, active Low). Disables both receivers and transmitters, forces TxDA and TxDB marking, forces the modem controls High, and disables all interrupts.

$\overline{RIA}$, $\overline{RIB}$. *Ring Indicator* (inputs, active Low). These inputs are similar to CTS and DCD. The Z80 DART detects both logic level transitions and interrupts the CPU. When not used in switched-line applications, these inputs can be used as general-purpose inputs.

$\overline{RTSA}$, $\overline{RTSB}$. *Request to Send* (outputs, active Low). When the RTS bit is set, the $\overline{RTS}$ output goes Low. When the RTS bit is reset, the output goes High after the transmitter empties.

$\overline{TxCA}$, $\overline{TxCB}$. *Transmitter Clocks* (inputs). TxD changes on the falling edge of $\overline{TxC}$. The Transmitter Clocks may be 1, 16, 32, or 64 times the data rate; however, the clock multiplier for the transmitter and the receiver must be the same. The Transmit Clock inputs are Schmitt-trigger buffered. Both the Receiver and Transmitter Clocks may be driven by the Z80 CTC Counter Time Circuit for programmable baud rate generation.

TxDA, TxDB. *Transmit Data* (outputs, active High).

$\overline{W/RDYA}$, $\overline{W/RDYB}$. *Wait/Ready* (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 Z80 DART data rate. The reset state is open drain.

FUNCTIONAL DESCRIPTION

The functional capabilities of the Z80 DART can be described from two different points of view: as a data communications device, it transmits and receives serial data, and meets the requirements of asynchronous data communications protocols; as a Z80 family peripheral, it interacts with the Z80 CPU and other Z80 peripheral circuits, and shares the data, address, and control buses, as well as being a part of the Z80 interrupt structure. As a peripheral to other microprocessors, the Z80 DART offers valuable features such as nonvectored interrupts, polling, and simple handshake capability.

The first part of the following functional description introduces Z80 DART data communications capabilities; the second part describes the interaction between the CPU and the Z80 DART.

The Z80 DART offers RS-232 serial communications support by providing device signals for external modem control. In addition to dual-channel Request To Send, Clear To Send, and Data Carrier Detect ports, the Z80 DART also features a dual channel Ring Indicator (RIA, RIB) input to facilitate local/remote or station-to-station communication capability. Figure 3 is a block diagram.

Communications Capabilities. The Z80 DART provides two independent full-duplex channels for use as an asynchronous receiver/transmitter. The following is a short description of receiver/transmitter capabilities. For more details, refer to the Asynchronous Mode section of the *Z80 SIO Technical Manual* (03-3033-01).

The Z80 DART offers transmission and reception of five to eight bits per character, plus optional even or odd parity. The transmitter can supply one, one and a half, or two stop bits per character and can provide a break output at any time. The receiver break detection logic interrupts the CPU both at the start and 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 is not started.

Framing errors and overrun errors are detected and buffered together with the character on which they occurred. Vectored interrupts allow fast servicing of interrupting 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 Z80 DART does not require symmetric Transmit and Receive Clock signals, a feature that allows it to be used with a Z80 CTC or any other clock source. 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. When using Channel B, the bit rates for transmit and receive operations must be the same because $\overline{RxC}$ and $\overline{TxC}$ are bonded together ($\overline{RxTxCB}$).

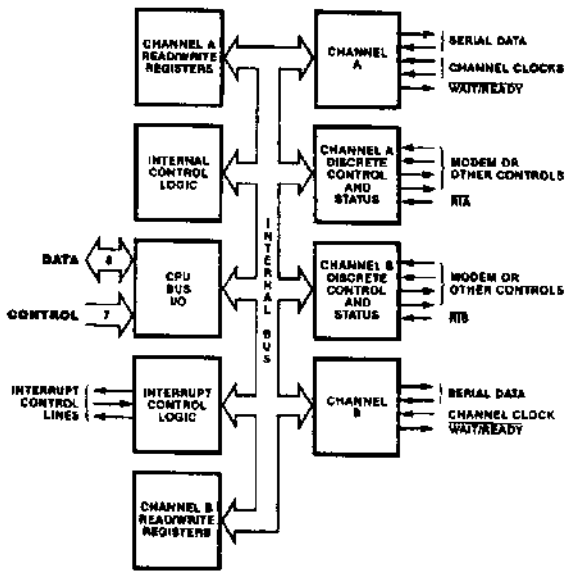


Figure 3. Block Diagram

I/O Interface Capabilities. The Z80 DART offers the choice of Polling, Interrupt (vectored or non-vectored) 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. There are no interrupts in the Polled mode. Status registers RR0 and RR1 are updated at appropriate times for each function being performed. All the interrupt modes of the Z80 DART must be disabled to operate the device in a Polled environment.

While in its Polling sequence, the CPU examines the status contained in RR0 for each channel; the RR0 status bits serve as an acknowledge to the Poll inquiry. The two RR0 status bits D_0 and D_2 indicate that a data transfer is needed. The status also indicates Error or other special status conditions. The Z80 DART Programming section contains more information. The Special Receive Condition status contained in RR1 does not have to be read in a Polling sequence because the status bits in RR1 are accompanied by a Receive Character Available status in RR0.

Interrupts. The Z80 DART offers an elaborate interrupt scheme that provides fast interrupt response in real-time applications. As a member of the Z80 family, the Z80 DART can be daisy-chained along with other Z80 peripherals for peripheral interrupt-priority resolution. In addition, the internal interrupts of the Z80 DART are nested to prioritize the various interrupts generated by Channels A and B. Channel B registers WR2 and RR2 contain the interrupt vector that points to an interrupt service routine in the memory. To eliminate the necessity of writing a status analysis routine, the Z80 DART can modify the interrupt vector in RR2 so it points directly to one of eight interrupt service routines. This is done under program control by setting a program bit (WR1, D_2) in Channel B called "Status

Affects Vector." When this bit is set, the interrupt vector in RR2 is modified according to the assigned priority of the various interrupting conditions.

Transmit 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 Receiver, 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.) When enabled, the receiver can interrupt the CPU in one of three ways:

- Interrupt on the first received character
- Interrupt on all received characters
- Interrupt on a Special receive condition

Interrupt On First Character is typically used with the Block Transfer mode. Interrupt On All Received Characters can optionally modify the interrupt vector in the event of a parity error. The Special Receive Condition interrupt can occur on a character basis. The Special Receive Condition can cause an interrupt can occur on a character basis. 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 Receive 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 $\overline{CTS}$, $\overline{DCD}$, and $\overline{RI}$ pins; however, an External/Status interrupt is also caused by the detection of a Break sequence in the data stream. The interrupt caused by the Break sequence has a special feature that allows the Z80 DART to interrupt when the Break 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 Break condition.

CPU/DMA Block Transfer. The Z80 DART provides a Block Transfer mode to accommodate CPU block transfer functions and DMA block transfers (Z80 DMA or other designs). The Block Transfer mode uses the $\overline{W}/\overline{RDY}$ output in conjunction with the Wait/Ready bits of Write Register 1. The $\overline{W}/\overline{RDY}$ output can be defined under software control 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 Z80 DART Ready output indicates that the Z80 DART is ready to transfer data to or from memory. To the CPU, the Wait output indicates that the Z80 DART is not ready to transfer data, thereby requesting the CPU to extend the I/O cycle.

INTERNAL ARCHITECTURE

The device internal structure includes a Z80 CPU interface, internal control and interrupt logic, and two full-duplex channels. Each channel contains read and write registers, and discrete control and status logic that provides the interface to modems or other external devices.

The read and write register group includes five 8-bit control 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 Read Register 2 in Channel B. The registers for both channels are designated as follows:

WR0-WR5	Write Registers 0 through 5
RR0-RR2	Read Registers 0 through 2

The bit assignment and functional grouping of each register is configured to simplify and organize the programming process.

The logic for both channels provides formats, bit synchronization, and validation for data transferred to and from the channel interface. The modem control inputs Clear to Send ($\overline{CTS}$), Data Carrier Detect ($\overline{DCD}$), and Ring Indicator ($\overline{RI}$) are monitored by the control logic under program control. All the modem control signals are general purpose in nature and can be used for functions other than modem control.

For automatic interrupt vectoring, the interrupt control logic determines which channel and which device within the channel has the highest priority. Priority is fixed with Channel A assigned a higher priority than Channel B; Receive, Transmit, and External/Status interrupts are prioritized in that order within each channel.

Data Path. The transmit and receive data path illustrated for Channel A in Figure 4 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 a Receive Character Available interrupt in a high-speed data transfer.

The transmitter has an 8-bit transmit data register that is loaded from the internal data bus, and a 9-bit transmit shift register that is loaded from the transmit data register.

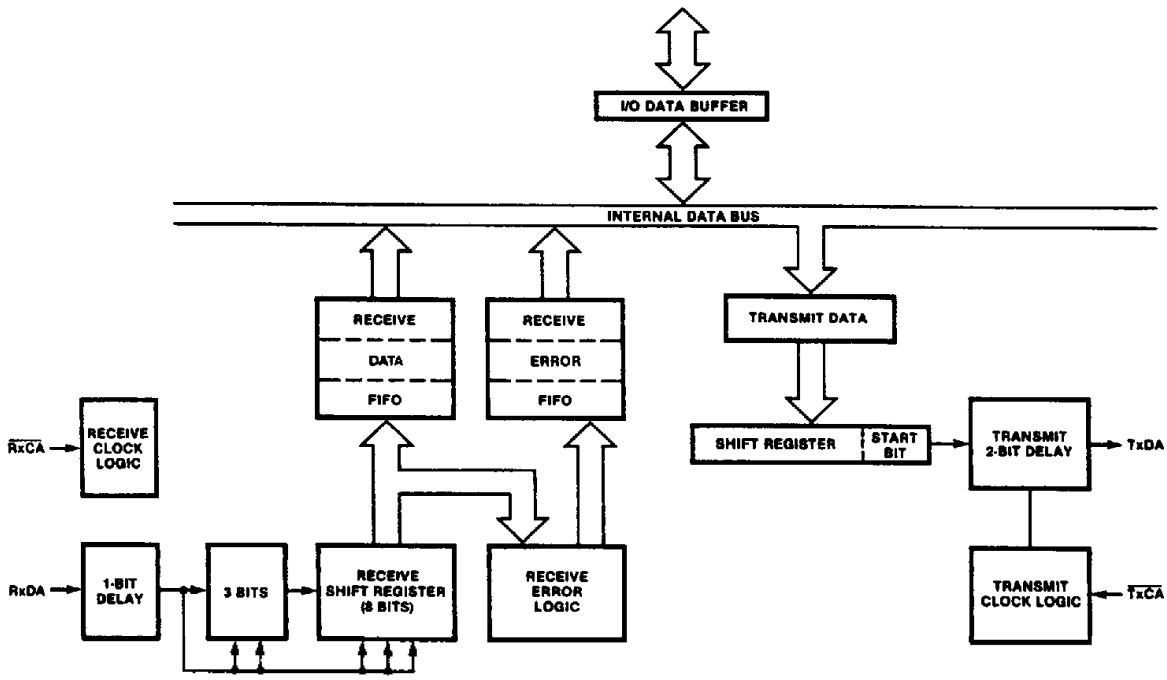


Figure 4. Data Path

READ, WRITE AND INTERRUPT TIMING

Read Cycle. The timing signals generated by a Z80 CPU input instruction to read a Data or Status byte from the Z80 DART are illustrated in Figure 5.

Write Cycle. Figure 6 illustrates the timing and data signals generated by a Z80 CPU output instruction to write a Data or Control byte into the Z80 DART.

Interrupt Acknowledge Cycle. (Figure 7) After receiving an Interrupt Request signal ($\overline{INT}$ pulled Low), the Z80 CPU sends an Interrupt Acknowledge signal ($\overline{M1}$ and $\overline{IORQ}$ both Low). The daisy-chained interrupt circuits determine the highest priority interrupt requestor. The IEI of the highest priority peripheral is terminated High. For any peripheral that has no interrupt pending or under service, IEO = IEI. Any peripheral that does have an interrupt pending or under service forces its IEO Low.

To insure stable conditions in the daisy chain, all interrupt status signals are prevented from changing while $\overline{M1}$ is Low. When $\overline{IORQ}$ 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.

Refer to the *Technical Manual* (03-3033-01) for additional details on the interrupt daisy chain and interrupt nesting.

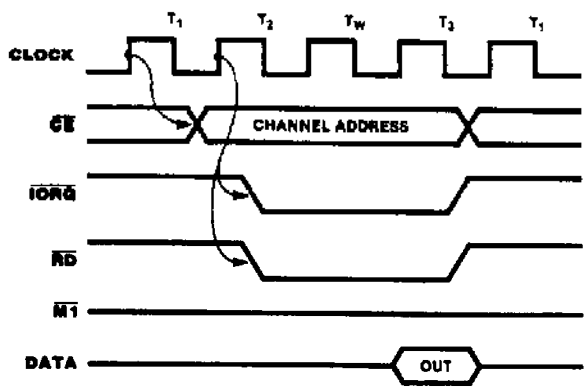


Figure 5. Read Cycle

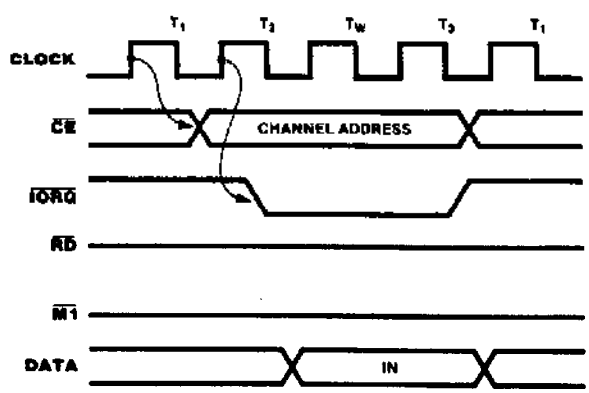


Figure 6. Write Cycle

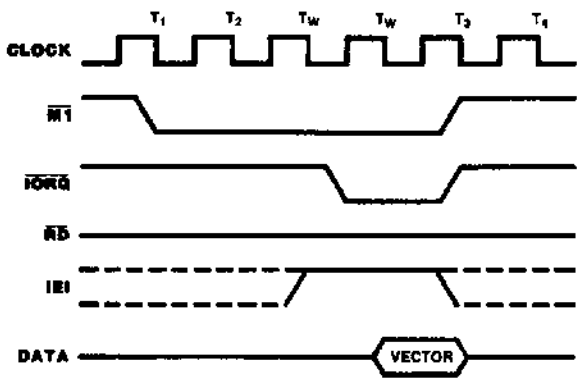


Figure 7. Interrupt Acknowledge Cycle

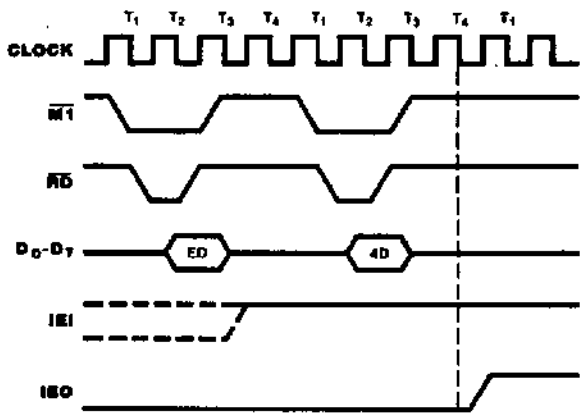


Figure 8. Return from Interrupt Cycle

Return From Interrupt Cycle. (Figure 8) Normally, the Z80 CPU issues an RETI (Return From Interrupt) instruction at the end of an interrupt service routine. RETI is a 2-byte opcode (ED-4D) that resets the interrupt-under-service latch to terminate the interrupt that has just been processed.

When used with other CPUs, the Z80 DART allows the user to return from the interrupt cycle with a special command called "Return From Interrupt" in Write Register 0 of Channel A. This command is interpreted by the Z80 DART in exactly the same way it would interpret an RETI command on the data bus.

Z80 DART PROGRAMMING

To program the Z80 DART, the system program first issues a series of commands that initialize the basic mode and then other commands that qualify conditions within the selected mode. For example, the character length, clock rate, number of stop bits, even or odd parity are first set, then the interrupt mode and, finally, receiver or transmitter enable.

Both channels contain command registers that must be programmed via the system program prior to operation. The Channel Select input (B/$\bar{A}$) and the Control/Data input (C/$\bar{D}$) are the command structure addressing controls, and are normally controlled by the CPU address bus.

Write Registers. The Z80 DART contains six registers (WR0-WR5) in each channel that are programmed separately by the system program to configure the functional personality of the channels (Figure 4). With the exception of WR0, programming the write registers requires two bytes. The first byte 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 Z80 DART.

WR0 is a special case in that all the basic commands (CMD_0-CMD_2) can be accessed with a single byte. Reset (internal or external) initializes the pointer bits D_0-D_2 to point to WR0. This means that a register cannot be pointed to in the same operation as a channel reset.

Write Register Functions	
WR0	Register pointers, initialization commands for the various modes
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

Read Registers. The Z80 DART contains three registers (RR0-RR2) that can be read to obtain the status information for each channel (except for RR2, which applies to Channel B only). The status information includes error conditions, interrupt vector, and standard communications-interface signals.

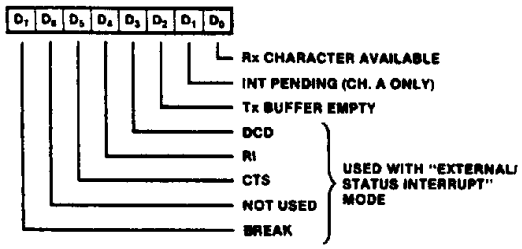
To read the contents of a selected read register other than RR0, the system program must first write the pointer byte to WR0 in exactly the same way as a write register operation. Then, by executing an input instruction, the contents of the addressed read register can be read by the CPU.

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 from a single register (RR1).

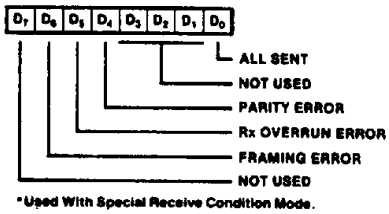
Read Register Functions	
RR0	Transmit/Receive buffer status, interrupt status and external status
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only)

Z80 DART READ AND WRITE REGISTERS

READ REGISTER 0

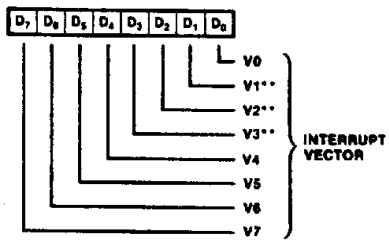


READ REGISTER 1*



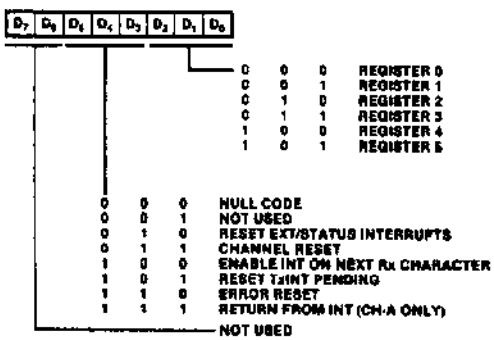
*Used With Special Receive Condition Mode.

READ REGISTER 2

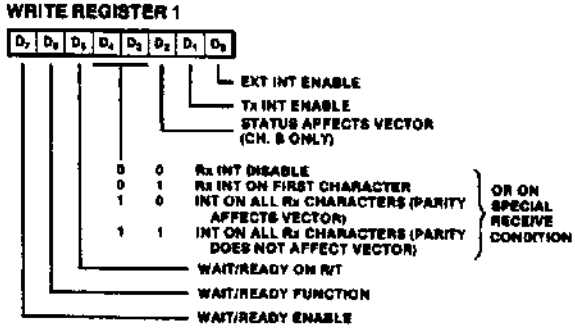


**Variable if "Status Affects Vector" is Programmed.

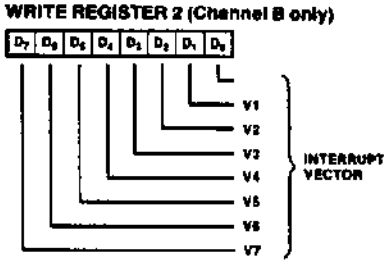
WRITE REGISTER 0



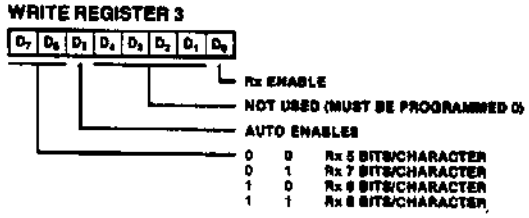
WRITE REGISTER 1



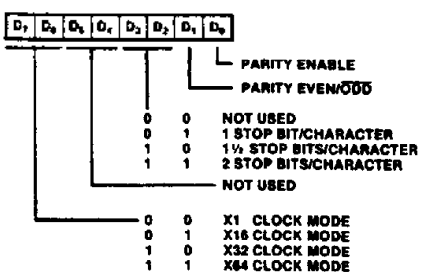
WRITE REGISTER 2 (Channel B only)



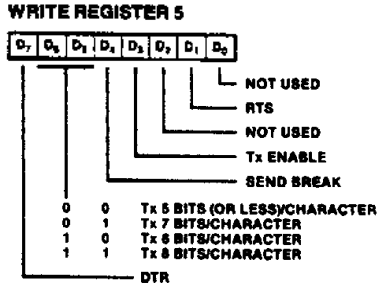
WRITE REGISTER 3



WRITE REGISTER 4



WRITE REGISTER 5



2

ABSOLUTE MAXIMUM RATINGS

Voltages on all pins with respect to GND . . . . . −0.3V to +7V
Operating Ambient Temperature . . . . . See Ordering Information
Storage Temperature . . . . . −65°C to +150°C

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

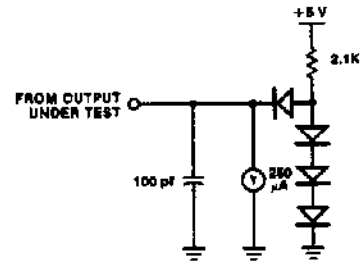
STANDARD TEST CONDITIONS

The DC characteristics and capacitance sections listed below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND (0V). Positive current flows into the referenced pin.

Available operating temperature ranges are:

■ S = 0°C to +70°C, +4.75V ≤ V_{CC} ≤ +5.25V

The Ordering Information section lists package temperature ranges and product numbers. Package drawings are in the Package Information section. Refer to the Literature List for additional documentation.



DC CHARACTERISTICS

Symbol	Parameter	Min	Max	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	−0.3	+0.45	V	
V_{IHC}	Clock Input High Voltage	V_{CC} −0.6		V	V_{cc} + 0.3V
V_{IL}	Input Low Voltage	−0.3	+0.8	V	
V_{IH}	Input High Voltage	+2.0	+5.5	V	
V_{OL}	Output Low Voltage		+0.4	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	+2.4		V	I_{OH} = −250 µA
I_L	Input/3-State Output Leakage Current	−10	+10	µA	0.4 < V_{IN} < 2.4V
$I_{L(RI)}$	RI Pin Leakage Current	−40	+10	µA	0.4 < V_{IN} < 2.4V
I_{CC}	Power Supply Current		100	mA	

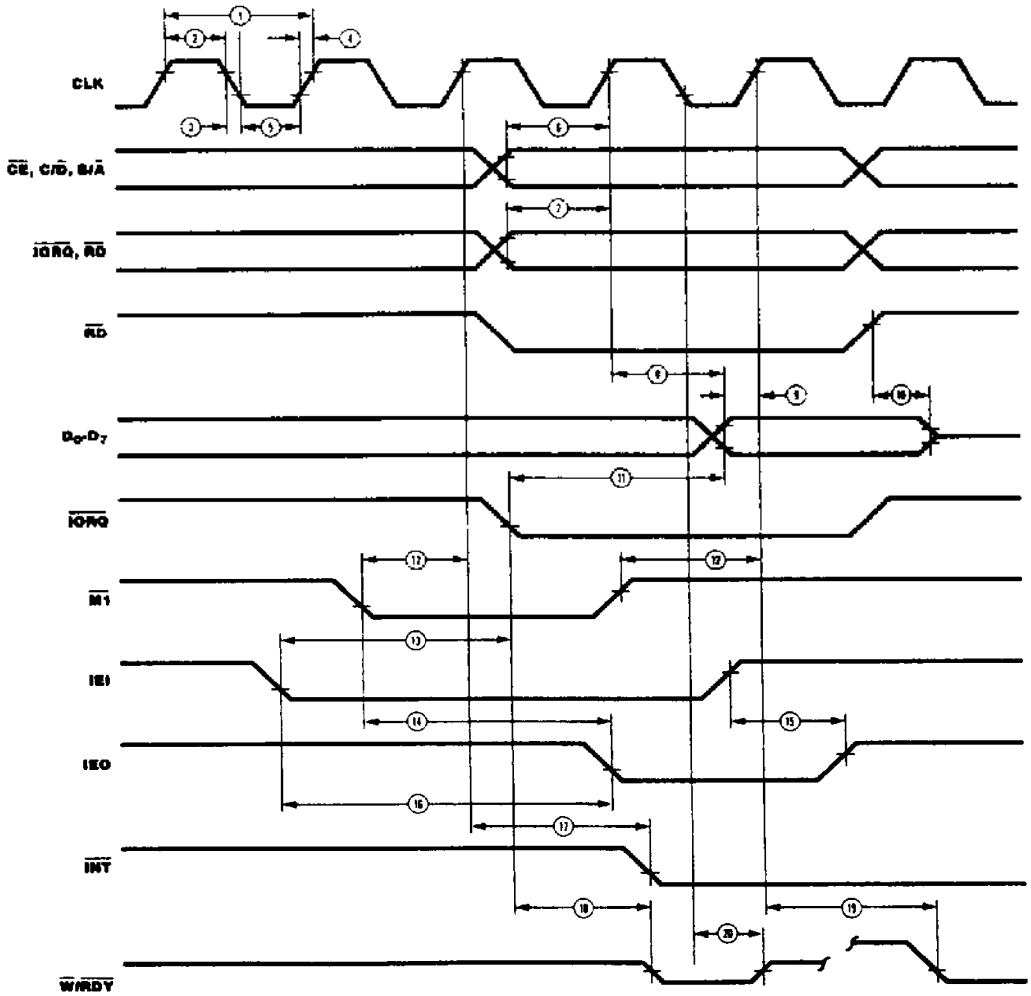
T_A = 0°C to 70°C, V_{CC} = +5V, ±5%.

CAPACITANCE

Symbol	Parameter	Min	Max	Unit
C	Clock Capacitance		40	pf
C_{IN}	Input Capacitance		5	pf
C_{OUT}	Output Capacitance		15	pf

Over specified temperature range; f = 1 MHz.
Unmeasured pins returned to ground.

AC CHARACTERISTICS TIMING



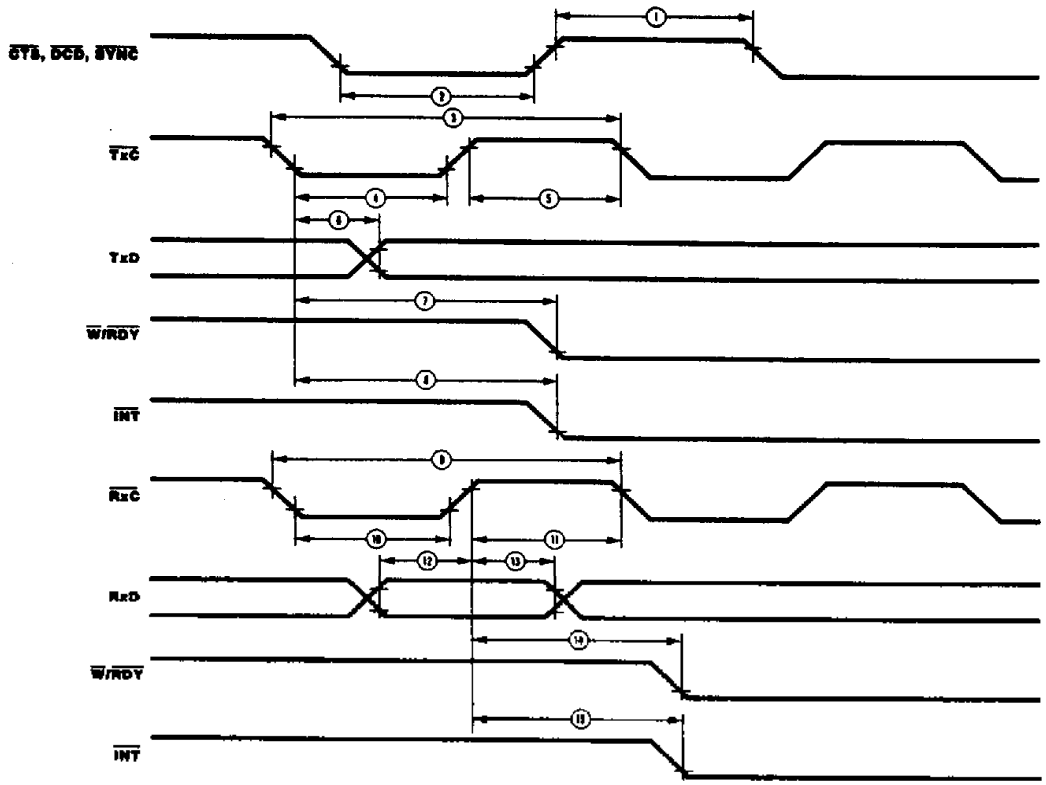
2

AC CHARACTERISTICS

Number	Symbol	Parameter	Z0847004 Min	Z0847004 Max	Z0847006 Min	Z0847006 Max
1	TcC	Clock Cycle Time	250	4000	165	4000
2	TwCh	Clock Width (High)	105	2000	70	2000
3	TfC	Clock Fall Time		30		15
4	TrC	Clock Rise Time		30		15
5	TwCl	Clock Width (Low)	105	2000	70	2000
6	TsAD(C)	$\overline{CE}$, C/$\overline{D}$, B/$\overline{A}$ to Clock ↑ Setup Time	145		60	
7	TsCS(C)	$\overline{IORQ}$, $\overline{RD}$ to Clock ↑ Setup Time	115		60	
8	TdC(DO)	Clock ↑ to Data Out Delay		220		150
9	TsDI(C)	Data In to Clock ↑ Setup (Write or M1 Cycle)	50		30	
10	TdRD(DOz)	$\overline{RD}$ ↑ to Data Out Float Delay		110		90
11	TdIO(DOI)	$\overline{IORQ}$ ↓ to Data Out Delay (INTACK Cycle)		160		100
12	TsM1(C)	$\overline{M1}$ to Clock ↑ Setup Time	90		75	
13	TsIEI(IO)	IEI to $\overline{IORQ}$ ↓ Setup Time (INTACK Cycle)	140		120	
14	TdM1(IEO)	$\overline{M1}$ ↓ to IEO ↓ Delay (interrupt before M1)		190		160
15	TdIEI(IEOr)	IEI ↑ to IEO ↑ Delay (after ED decode)		100		70
16	TdIEI(IEOf)	IEI ↓ to IEO ↓ Delay		100		70
17	TdC(INT)	Clock ↑ to INT ↓ Delay		200		150
18	TdIO(W/RWf)	$\overline{IORQ}$ ↓ or $\overline{CE}$ ↓ to $\overline{W/RDY}$ ↓Delay (Wait Mode)		210		175
19	TdC(W/RR)	Clock ↑ to $\overline{W/RDY}$ ↓ Delay (Ready Mode)		120		100
20	TdC(W/RWz)	Clock ↓ to $\overline{W/RDY}$ Float Delay (Wait Mode)		130		110

*Units in nanoseconds (ns).

AC CHARACTERISTICS (Continued)



Number	Symbol	Parameter	Z0847004 Min	Z0847004 Max	Z0847006 Min	Z0847006 Max	Notes*
1	TwPh	Pulse Width (High)	200		200		2
2	TwPl	Pulse Width (Low)	200		200		2
3	TcTxC	$\overline{TxC}$ Cycle Time	400	∞	330	∞	2
4	TwTxCl	$\overline{TxC}$ Width (Low)	180	∞	100	∞	2
5	TwTxCh	$\overline{TxC}$ Width (High)	180	∞	100	∞	2
6	TdTxC(TxD)	$\overline{TxC}$ ↓ to TxD Delay		300		220	2
7	TdTxC(W/RRf)	$\overline{TxC}$ ↓ to $\overline{W/RDY}$ ↓ Delay (Ready Mode)	5	9	5	9	1
8	TdTxC(INT)	$\overline{TxC}$ ↓ to $\overline{INT}$ ↓ Delay	5	9	5	9	1
9	TcRxC	$\overline{RxC}$ Cycle Time	400	∞	330	∞	2
10	TwRxCl	$\overline{RxC}$ Width (Low)	180	∞	100	∞	2
11	TwRxCh	$\overline{RxC}$ Width (High)	180	∞	100	∞	2
12	TsRxD(RxC)	RxD to $\overline{RxC}$ ↑ Setup Time (x1 Mode)	0		0		2
13	ThRxD(RxC)	RxD Hold Time (x1 Mode)	140		100		2
14	TdRxC(W/RRf)	$\overline{RxC}$ ↑ to $\overline{W/RDY}$ ↓ Delay (Ready Mode)	10	13	10	13	1
15	TdRxC(INT)	$\overline{RxC}$ ↑ to $\overline{INT}$ ↓ Delay	10	13	10	13	1

* In all modes, the System Clock rate must be at least five times the maximum data rate. $\overline{RESET}$ must be active a minimum of one complete clock cycle.
1. Units equal to System Clock Periods.
2. Units in nanoseconds (ns).