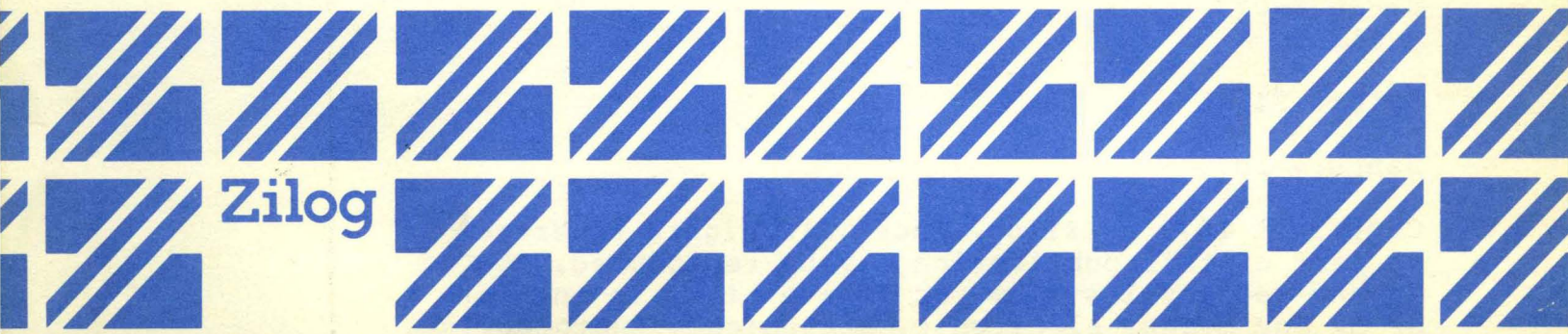
Zilog

ZDS/ASPIO

Hardware Reference Manual



03-3028-02
Revision A

Price: $4.50

January 1979

ZDS/ASPIO

HARDWARE REFERENCE MANUAL

TABLE OF CONTENTS

SECTION 1

INTRODUCTION

1.1 General Information

The Z-80 Zilog Development System Serial/Parallel I/O (ZDS/ASPIO) Board provides the Development System with one additional programmable serial communications interface and one additional programmable parallel interface (see Figure 1-1, Block Diagram). The Serial Interface is implemented by means of an 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART) with EIA level translators to provide a RS-232 compatible output. The Parallel Interface consists of the Z-80 PIO, which has been configured as a Centronics Line Printer Interface.

1.2 Installation

The ZDS/ASPIO card is inserted into slot J3 on the ZDS-1/40 and slot J9 on the ZDS-1/25.

Two flat ribbon cables are provided for internal connection between the card base plate and the connector area on the rear panel of the system chassis.

1.3 The USART

The USART is programmed by means of a control byte loaded by the Z-80 CPU. Controllable parameters include bits per character (5, 6, 7 or 8), number of stop bits for asynchronous operation (1.0, 1.5, 2.0), parity insertion/ checking (odd, even or none) and clocking of transmitted or received data at frequencies of 1, 16 or 64 times the data rates. Each USART accepts data characters from the Z-80 in a parallel format and converts them into a continous serial data stream for transmission. The USART can simultaneously receive serial data streams and convert them into parallel data characters for the Z-80.

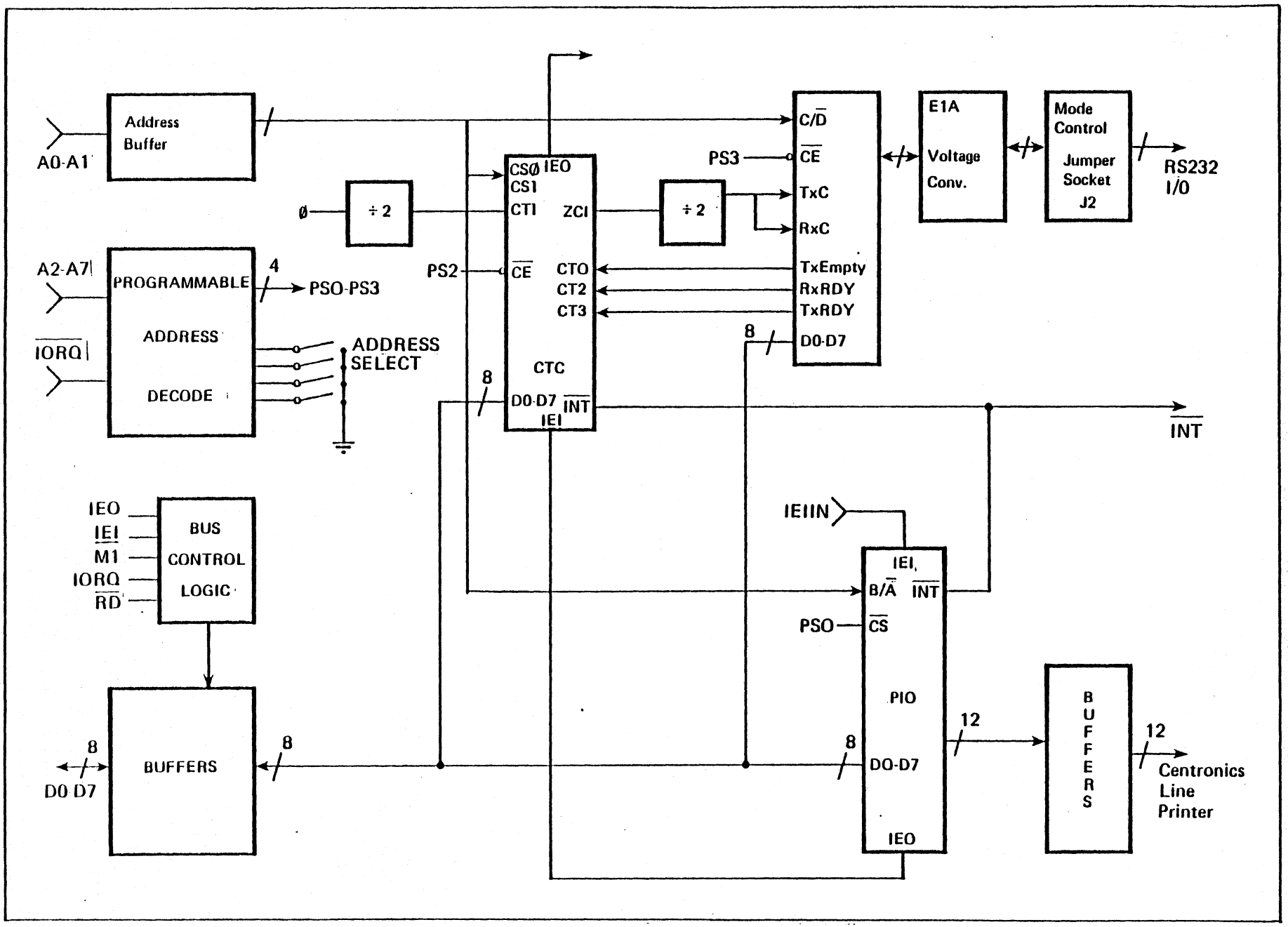


Figure 1-1. ZDS/ASPIO Block Diagram

1.4 The Z-80 PIO

The Z-80 PIO provides a software programmable, two-port parallel I/O device for standard hardware interface between peripheral devices and the Z-80 CPU. The PIO contains two independent 8-bit ports with full handshake control that can be configured by the CPU to operate in any of four modes:

Output Mode	(Mode 0)
Input Mode	(Mode 1)
Bidirectional Mode	(Mode 2)
Control Mode	(Mode 3)

In Output Mode (Mode 0), data is written to the ports from the Z-80 and onto the port data bus. In Input Mode (Mode 1), the peripheral device supplies data to the port. The Bidirectional Mode (Mode 2) allows one port (Port A) to be directional, using the handshake signals from the other port. The Control Mode (Mode 3) allows for direct bit set and reset capability. This mode also allows any bit in either ort to be individually programmed--either input bit or an output bit. Vectored interrupt-communication with the CPU is included to facilitate data transfer. A unique feature of the PIO is that it can be programmed to interrupt the Z-80 CPU on the occurrence of specified status conditions in the peripheral device. One port has the ability to source a minimum of 1.5ma of current at 1.5 volts, allowing Darlington transistors to be driven directly (for printer and alphanumeric displays, for example).

1.5 The CTC

The CTC provides a data rate clock for the USART with all the common communication frequencies between 75Hz and 38.4KHz. Unused area on the card is filled with plated-through holes on .10 inch centers for insertion of wire-wrap sockets.

SECTION 2

SERIAL INTERFACE

2.1 Overview

The Serial Interface can support synchronous or asynchronous full-duplex communications. The interface occupies two port addresses: the Control/Status Address for writing control words or reading device status, and the Data Address for writing output data or reading input. Typically, the interface will be programmed immediately after system RESET by writing an appropriate control-word pattern. Subsequently, the status word may be checked, or data may be input or output through the interface as necessary. If appropriate, the interface may be reprogrammed during system operation.

2.2 The Mode Instruction

The first word written to the Serial Interface Control Port after system RESET defines whether synchronous or asynchronous operation is required, defines the use of parity, and establishes the number of stop bits for asychronous operation or the synchronization scheme for synchronous operation. This first word is referred to as the "Mode Instruction" and has the format shown below.

7	6	5	4	3	2	1	0
S2	S1	PARITY EVEN/ ODD	PARITY ENABLE/ DISABLE	L2	L1	B2	B1

Where the subfields of the Mode Instruction are defined as:

B2	B1	
0	0	Synchronous operation
0	1	Asynchronous divide by 1
1	0	Asynchronous divide by 16
1	1	Asynchronous divide by 64

L2	L1	Word Length
0	0	5 Bits
0	1	6 Bits
1	0	7 Bits
1	1	8 Bits

Parity Enable/Disable: If set, a parity bit is inserted after each word transmitted and checked after each word received.

Parity Even/Off: If set, parity will be even.

S2	S1	Asynchronous Mode
0	0	Invalid
0	1	1 stop bit generated in transmission
1	0	1 1/2 stop bits generated in transmission
1	1	2 stop bits generated in transmission

S1 (Synchronous Mode): If set, external hardware synchronization is to be used. If clear, internal character synchronization will be established.

S2 (Synchronous Mode, Internal Synchronization): If set, a single synchronization character will be used. If clear, two synchronization characters will be used.

If the Mode Instruction specifies synchronous operation, then it must be followed by the one or two synchronization characters as indicated by S2. If two synchronization characters are required, they should be supplied in the same order as they would be expected to appear in a data stream. If asynchronous operation is specified, no synchronization characters need be supplied.

2.3 Command Instruction

After the Mode Instruction and any required synchronization characters have been written, command instructions may be written. Unless the system is RESET, or an Internal Reset command is issued, all subsequent writes to the Control/Status Port Address will be interpreted as Command Instructions. The format of a Command Instruction is shown below:

7	6	5	4	3	2	1	0
ENTER HUNT MODE	INTERNAL RESET	REQ TO SEND	ERROR RESET	SEND BREAK	RCV ENABLE DISABLE	DATA TERM RDY	XMIT ENABLE/ DISABLE

WHERE:

XMIT & RCV ENABLE/DISABLE

Enable with 1's and disable with 0's.

DATA TERM RDY & REQ TO SEND

Force the corresponding hardware signals low when set to 1's and high when set to 0's.

SEND BREAK

Forces transmit data low when set. Long, low states are traditionally interpreted in asynchronous systems as "Breaks".

ERROR RESET

Clears all internal errors when set.

INTERNAL RESET

Forces the USART back to the state where a MODE INSTRUCTION is expected, which permits changing mode "on the fly".

ENTER HUNT MODE (SYNCHRONOUS ONLY)

Terminates assembly of receive data in synchronous mode and returns the USART to the state where it searches for synchronization characters.

2.4 Control/Status Port Address

During the operation of the serial interface, it is possible to read its current status from the control or status port address. The format of the status word is as follows:

7	6	5	4	3	2	1	0
DATA SET READY	SYNCH DETECT	FRAMING ERROR	OVERRUN ERROR	PARITY ERROR	XMITTER EMPTY	RCVR READY	XMITTER READY

WHERE:

DATA SET READY

Corresponds to the control line state.

SYNCH DETECT

Indicates internal or external synch detection.

FRAMMING ERROR, OVERRUN ERROR & PARITY ERROR

Will be set when the corresponding error occurs. Framing errors only occur if less than one stop bit is received.

XMITTER EMPTY

Indicates that both the transmit buffer and shift registers are empty.

RECEIVER READY

Indicates the receive buffer is full and ready to be input by the CPU.

XMITTER READY

Indicates the transmit buffer is empty and ready for output from the CPU.

Data input and output proceeds by reading and writing the data port address.

SECTION 3

PARALLEL INTERFACE

3.1 Overview

The ZDS/ASPIO is configured to allow direct interface to a Standard Centronics interface. Included with the card is a utility diskette that contains the driver software to be used with the Zilog Development System (ZDS-1/25 or 1/40). Zilog supplies printer driver software for use with either version of the development system. This disk-based software driver and user manual is part of the RIO package that comes with each system.

3.2 System Interconnect

Sheet 2 of Appendix B (Logic Diagrams) details the PIO and the Centronics Interface signals. Appendix A identifies the appropriate pin numbers and the required cables for connecting the parallel interface into the system.

SECTION 4

ZDS/ASPIO INTERRUPT STRUCTURE

Interrupt requests may orginate from the Z-80 PIO or the Z-80 CTC on the ZDS/ASPIO. The order of priority is the PIO first then the CTC. This priority is established with the IEO line of the PIO tied to the IEI line of the CTC.

When both the PIO and CTC will be generating interrupts a jumper between points C and D (see Schematics, Sheet 2) will allow data bus control during interrupt acknowledge transfer cycles (see schematics Sheet 1). If only the PIO is to generate interrupts, then a jumper between points A and D will perform the same bus control. The factory-configured board will permit interrupts to be generated by both the CTC and PIO.

When interrupts occur from the CTC the order of priority is channel 0 first (USART transmitter empty), Channel 2 second (USART Receiver ready) and Channel 3 last (USART Transmitter ready). When interrupts occur from the PIO, the order of priority is Channel A first and Channel B second.

SECTION 5

ZDS/ASPIO ADDRESS DECODING

The PIO, CTC and USART can be placed anywhere in the I/O address space by setting the 4-bit switch on the board. Position Number One on the switch corresponds to address line A7, while position Number Four corresponds to A4 (see schematics, Sheet 1). A 4-bit comparator (A14 - 7485 on Sheet 1) compares data from the address bus during I/O operations to the switch setting. When a match is found the comparator produces a true output on Pin 6, which fixes the address range from OX-FX.

Further decoding is performed with Address Lines A2 and A3 to select four contiguous address (PS0 - PS4) within the selected range. These signals are assigned to peripherals according to the information shown below:

PS0 - PIO Select
PS1 - Unused
PS2 - CTC Select
PS3 - USART Select

SECTION 6

BAUD RATE GENERATION

6.1 General Information

The USART can operate in the 1X, 16X or 64X mode and receives its baud rate clock from the Z-80 CTC, which can operate either in a timer or counter mode. Figures 6-1 and 6-2 symbolically represent the function performed by the CTC for each mode. The flip-flop between the CTC and USART is used to convert the pulse signal from the CTC into a 50% duty cycle signal required by the USART. Table 6-1 identifies the time constant and operating mode of CTC used to generate the indicated baud rate. In general, it is best to choose the mode of operation that has the largest time constant in order to most accurately approximate the true frequency.

TABLE 6-1. Time Constants For Baud Rate

BAUD RATE	COUNTER MODE x16	COUNTER MODE x64	TIMER MODE (PRESCALE = 16) x16	TIMER MODE (PRESCALE = 16) x64
50	---	192	96	24
75	---	128	64	16
110	---	87	44	11
134.5	---	71	36	9
150	256	64	32	8
200	192	48	24	6
300	128	32	16	4
600	64	16	8	2
1200	32	8	4	1
2400	16	4	2	---
4800	8	2	1	---
9600	4	1	--	---
19200	2	---	--	---
38400	1	---	--	---

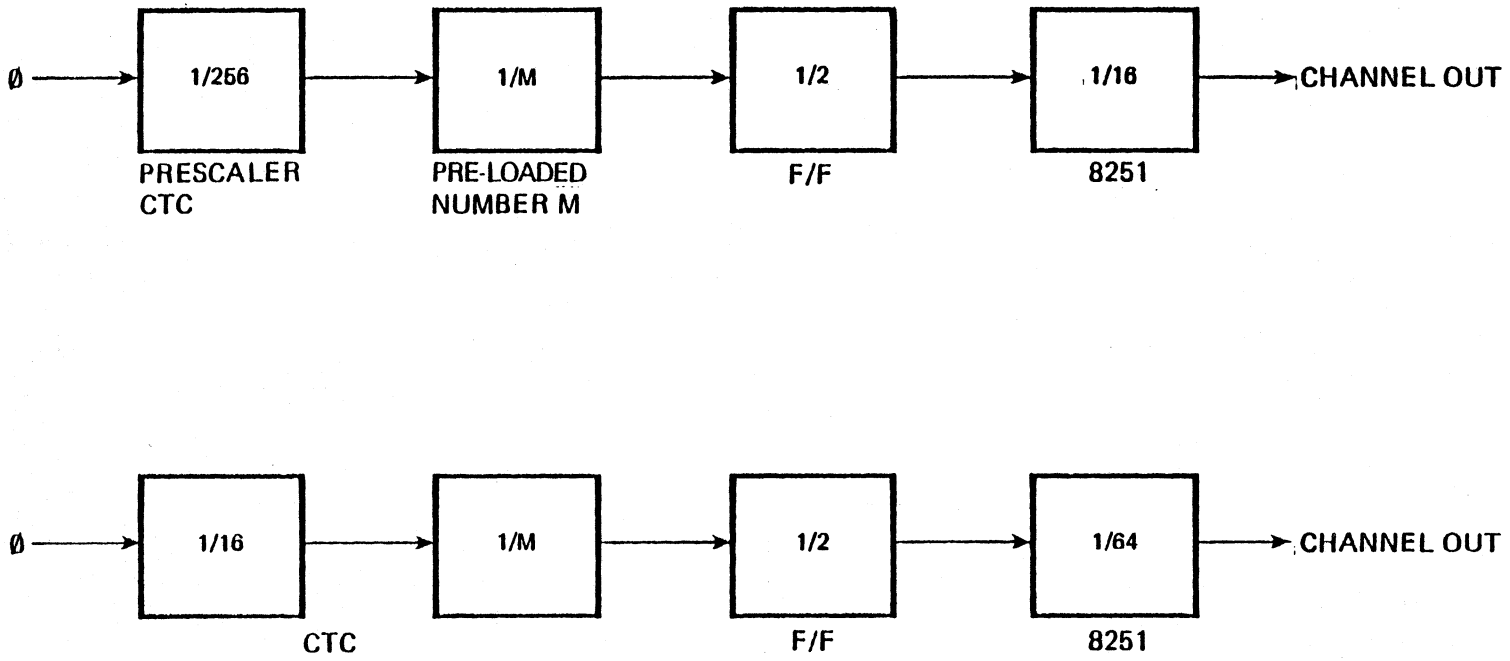


Figure 6-1. Sample Configurations with CTC in Timer Mode

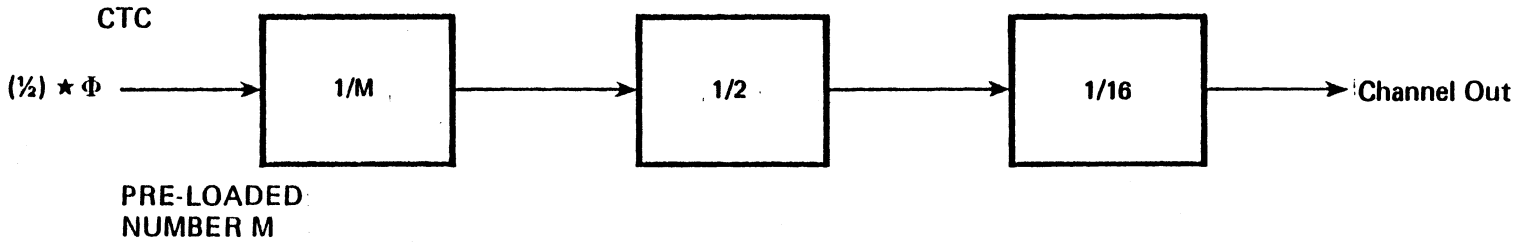


Figure 6-2. Sample Configuration with CTC in Counter Mode

6.2 Timer Mode Sample Calculation

The information found in the Timer Mode Column of Table 6-1 corresponds to configurations as indicated in Figure 6-1. In the Timer Mode, a prescaler is loaded into the CTC, in addition to the Time Constant. This prescaler divides the System clock by 16 or 256 before decrementing the down counter (see CTC Product Specification).

For example, if the first configuration is required, the Time Constant is calculated to obtain a baud rate of 300 as follows.

1) Determine the 16X baud rate frequency to the USART:

$$300\ (16) = 4.8\ \text{kHz}$$

2) Divide the system clock (2.4576 mHz) by the above frequency to determine the ratio:

$$\frac{2.4576\text{MHZ}}{4.8\ \text{kHz}} = 512$$

3) Divide 512 by the CTC prescaler of 16 and half again for the squaring Flip Flop.

$$\frac{512}{32} = 16 = 10\ \text{Hex}$$

In summary, for the Divide-by-16 Mode, the Time Constant in decimal can be found from:

$$M = \frac{76800}{PS\ BR}\ \text{(Decimal)}$$

Where PS is the prescaler and BR is the Baud Rate; likewise, for the Divide-by-64 Mode, M can be found from:

$$M = \frac{19{,}200}{PS\ BR}\ \text{(Decimal)}$$

6.3 Counter Mode Sample Calculation

In the counter mode an external clock is used to decrement the CTC down counter. This clock is obtained by dividing the 0 clock in half and applying it to the CT1 clock input of the CTC (See schematics, Sheet 3).

The information found in the two Counter Mode columns of Table 6-1 represent sample numbers required for each division mode and baud rate. For example, if the USART is in the Divide-by-64 Mode, the pre-load number for a baud rate of 9600 is calculated from:

$$M = \frac{19.6608}{16} \cdot \frac{1}{128} \cdot \frac{1}{9600} = 1$$

$$= \frac{9600}{BR}$$

In general, the relation between baud rate and the pre-load number is:

$$M = \frac{9600}{BR} \quad \text{(x64 mode)}$$

A relation can also be found for the divide by 16 mode:

$$M = \frac{36400}{BR} \quad \text{(x16 mode)}$$

SECTION 7

ZDS/ASPIO SERIAL I/O INTERFACE

The ZDS/ASPIO may be customized to meet specified periphera requirements. The USART output is provided with a jumper area (J2) for interchanging, I/O line drivers and receivers for interface to either a terminal or a modem. Table 7-1 below identifies the jumper points on J2 needed when the ZDS/ASPIO is interfaced to a terminal.

Table 7-1. Modem Mode (ZDS/ASPIO talks to a terminal)

FROM SIGNAL	J2 PIN NO.	TO SIGNAL	J2 PIN NO.
TXDB	1	TRXD	15
RXDB	2	TTXD	16
RTSB	5	TCTS	11
CTSB	6	TRTS	12
DTRB	8	TDSR	9
DSRB	7	TDTR	10

Table 7-2 below identifies the jumper points on J2 needed when the ZDS/ASPIO is interfaced to a modem.

Table 7-2. Terminal Mode (ZDS/ASPIO talks to a modem)

FROM SIGNAL	JX PIN NO.	TO SIGNAL	JX PIN NO.
TXDB	1	TTXD	16
RXDB	2	TRXD	15
RTSB	5	TRTS	12
CTSB	6	TCTS	11
DSRB	7	TDSR	9
DTRB	8	TDTR	10

SECTION 8

PROGRAMMING THE SERIAL INTERFACE

8.1 General Information

This section describes programming techniques for the ZDS/ASPIO serial interface. Before any data can be transmitted or received, the USART must be programmed. The ZDS/ASPIO may be interrupt driven. The utility ZDS/ASPIO.TEST is an example of such a driver.

8.2 Disabling Interrupts

It is important to start any program by disabling the CTC interrupt generator. This may not need to be done if the previously run program terminated normally and disabled the CTC and if the USART is enabled and cleared before interrupts are reenabled. Programming the USART with interrupts enabled can often cause spurious interrupts during the enabling process.

8.3 Enabling the USART

Enabling the USART requires careful attention. It should be enabled before the baud-rate generator is turned on, because its initial state is indeterminate and it may try to transmit or cause interrupts. The process involves five steps, which should be performed in the following order:

1) Reset the USART;

2) Send a mode byte (see 8251 Product Spec.) to the control port;

3) Send a command byte to the control port;

4) Read one character from the data port (the data port address is the control port address minus on by convention).

A USART may be reset by sending three zero bytes followed by a Reset command (40H). The zero bytes are recommended because the USART would treat the first two characters as synce characters under some circumstances. Some USART application notes suggest that sending three reset commands will cause a reset and enable the USART to receive a mode

byte. Because the Development System reset button is attached to the USART's external reset pin, pushing the system Reset button followed by three internal Reset commands will not cause the USART to accept a mode byte, but rather causes the USART to treat the intended mode byte as a command byte.

The next step is to send a mode byte to the USART. The value 8EH sets the mode to 1.5 stop bits, no parity, eight bit characters, and a 16x baud rate factor. The mode byte must be followed by a command byte. The value 37H enables the USART for both transmitting and receiving. Both REQUEST TO SEND (RTS) (bit 5) and DATA TERMINAL READY (DTR) (bit 1) should be turned on because USARTS and most modems only work if CLEAR TO SEND (CTS) is low and one of these pins will be connected to CTS. One character must be read from the USART data port to clear it. Omitting this step causes the first character receiver to be the one previously in the USART.

The routine should look like:

```
        RESET AND SET UP FOR THE USART

START   LD      C: USART

ULC1    LD      A,0

        OUT     (C), A          ; THREE INTERNAL RESETS = EXT RESET

        OUT     (C), A

        OUT     (C), A

        LD      A, RESCOM

        OUT     (C), A

        LD      A, SIBMOD       ; SET UP TESTING MODE

        OUT     (C), A

        LD      A, SIBCOM       ; AND ENABLE BOTH TR AND RCV

        OUT     (C), A

        DEC     C

        IN

        RET     A, (C)          ; CLEAR OUT GARBAGE CHARACTER

RESCOM  EQU     40H             ; RESET USART TO ACCEPT MODE BYTE
```

```
SIBMOD  EQU     8EH             ; 8 BIT ASYNCHRONOUS

SIBCOM  EQU     27H             ; TRANSMIT AND RECEIVE - NO ERROR RESET
```

8.4 Transmitting and Receiving Characters

Actual character transmission and reception may begin once the USART has been set up. A character is transmitted by writing it to the USART's data port. When the USART is ready to accept another character for transmission, the TxRDY bit in the status byte comes on. This condition may be detected by reading the USART's status register.

Alternatively, transmission may be interrupt-driven. TxRDY is connected to the CTC so that it may generate a proper interrupt and an appropriate vector for the Z-80 CPU.

When the USART receives a character, the RxRDY status bit comes on. This condition may also be detected by reading the USART's status register. The received character may be read, when ready, from the USART's data port. Because RxRDY is also connected to the CTC, the receiver as well as the transmitter may be interrupt-driven.

Note that the USART is buffered internally; therefore, when it interrupts, it may in fact be ready for two data transfers rather than just one. In order to avoid possible loss of data, interrupt handling routines should check for the condition.

APPENDIX A

ZDS/ASPIO PINOUT CHART

PINOUT FOR ZDS/ASPIO

J09

PIN #	SIGNAL NAME (DESCRIPTION)
001	(+05V.PRINTED.DISTRIBUTION)
002	(+05V.PRINTED.DISTRIBUTION)
003	(+05V.PRINTED.DISTRIBUTION)
004	TTXD.2
005	TRXD.2
006	TRTS.2
007	TCTS.2
008	TDSR.2
009	A06
010	A07
011	.
012	.
013	.
014	.
015	.
016	A00
017	A01
018	A02
019	A03
020	A04
021	A05
022	.
023	.
024	PHI.(SYSTEM.CLOCK)
025	TERM.BUSY.2
026	POWER.ON.CLR-
027	.
028	M1-
029	.
030	IORQ-
031	.
032	.
033	.
034	.
035	.
036	.
037	.
038	.
039	.
040	CARRIER.DETECT.2
041	.
042	.
043	.

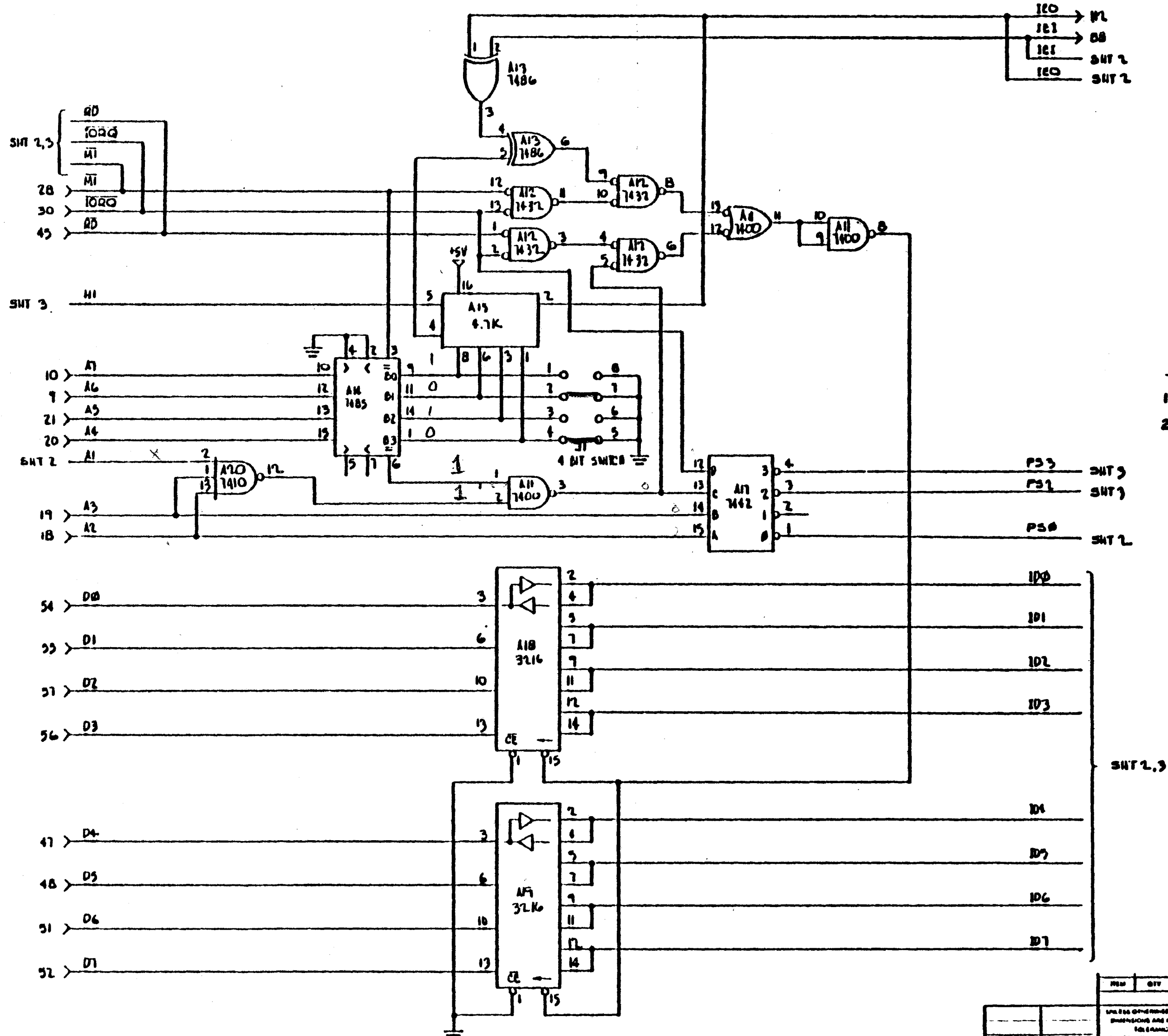
Appendix A, continued

```
044   WR-
045   RD-
046   .
047   D4
048   D5
049   SYNC.2
050   TDTR.2
051   D6
052   D7
053   .
054   D0
055   D1
056   D3
057   D2
058   .
059   (+05V.PRINTED.DISTRIBUTION)
060   (+05V.PRINTED.DISTRIBUTION)
061   (+05V.4.PRINTED.DISTRIBUTION)
062   (GND.9.062)
063   (GND.9.063)
064   (GND.9.064)
065   AB0.D1.LP.RX  (LP.DATA.1  RXP.DATA.1)
066   AB1.D2.LP.RX
067   AB2.D3.LP.RX
068   AB3.D4.LP.RX
069   AB4.D5.LP.RX
070   AB5.D6.LP.RX
071   AB6.D7.LP.RX
072   AB7.D8.LP.RX
073   AA0.LP.RX (LP.DATA.STROBE- RXP.PUNCH.COMMAND)
074   .
075   AA5.LP.RX (LP.SELECT RXP.PUNCH.SYSTEM.READY)
076   AA4.LP.RX (LP.BUSY RXP.PUNCH.READY)
077   AA3.LP.RX (LP.PAPER.EMPTY RXP.TAPE.LOW)
078   AA2.LP.RX (LP.FAULT- RXP.ERROR)
079   .
080   .
081   AA1.LP.RX (LP.INPUT.PRIME- RXP.DIRECTION)
082   ABS.LP (LP ACKNOWLEDGE-)
083   .
084   .
085   .
086   .
087   .
088   IEO.2 (BOARD IEI)
089   RING.IND.2
090   +12V.3
091   ORIGINATE.MODE.2
092   LOCAL.MODE.2
093   .
```

```
094   .
095   .
096   .
097   .
098   .
099   .
100   .
101   .
102   .
103   .
104   .
105   .
106   .
107   -5V.12
108   .
109   INT-
110   .
111   .
112   IEO.3 (BOARD IEO)
113   .
114   .
115   .
116   .
117   .
118   .
119   .
120   (GND.9.120)
121   (GND.9.121)
122   (GND.9.122)
```

APPENDIX B

LOGIC DIAGRAMS



REVISIONS			
LTR	DESCRIPTION	DATE	APPROVED
1	EAR 040	9.2.78	RF
2	EAR 161, 190	12 6 78	RF
3	EAR 361	12.15.78	RRD
4	EAR 413, 414	1.16.79	RRD
5	EAR 510	-7.79	RRD
5.1	EAR 584		

NOTES: UNLESS OTHERWISE SPECIFIED

1. RESISTORS ARE IN OHMS, 1/4W, 5%.
2. CAPACITORS ARE IN MICROFARADS.

LAST REF. DES. USED					
A20	J2	R2	C23	CR1	
REF. DES. NOT USED					
		R1			

INTEGRATED CIRCUIT'S		
TYPE		SPARE
270-330 RES NET	A10	9
7486	A13	2
4.7K RES NET	A15	3
7407	A16	2
7410	A20	2

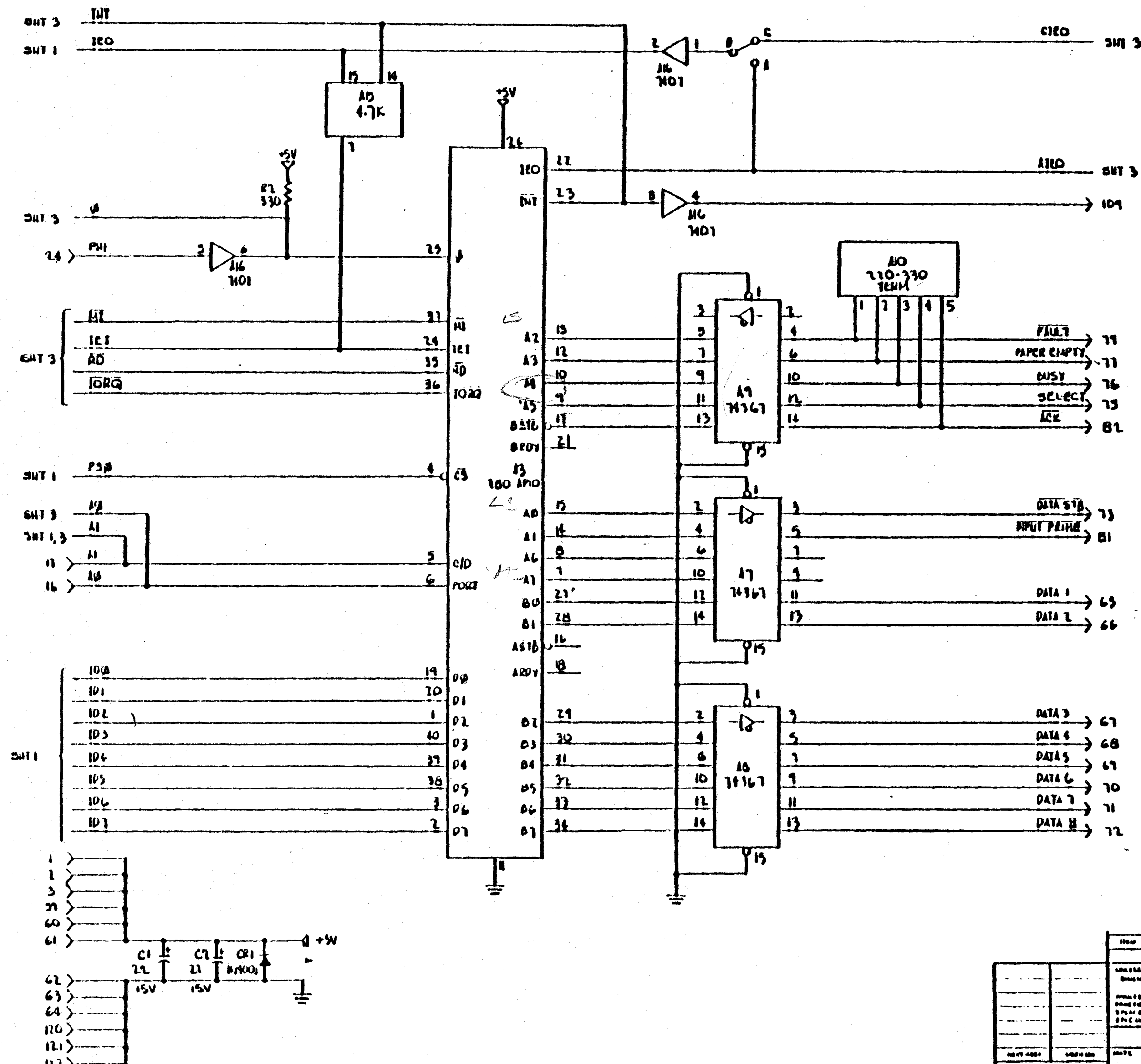
ZILOG INC.

LOGIC DIAGRAM, ZDS/ASPIO

SIZE D | DWG NO. DL-0108-02 | REV 5 | SHEET 1 OF 3

REVISIONS

SEE SHEET 1

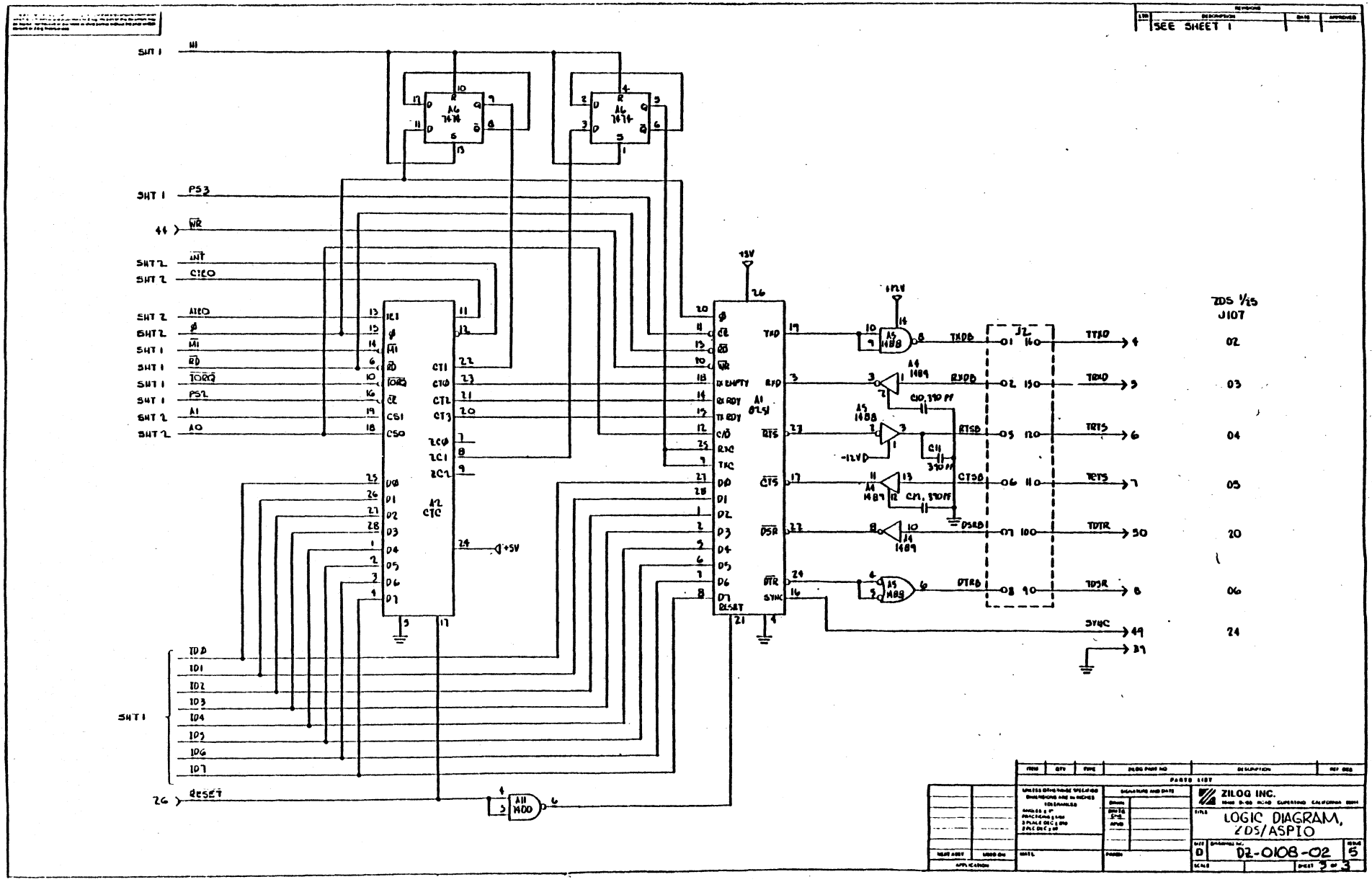


ZILOG INC.

LOGIC DIAGRAM,
ZDS/ASPIO

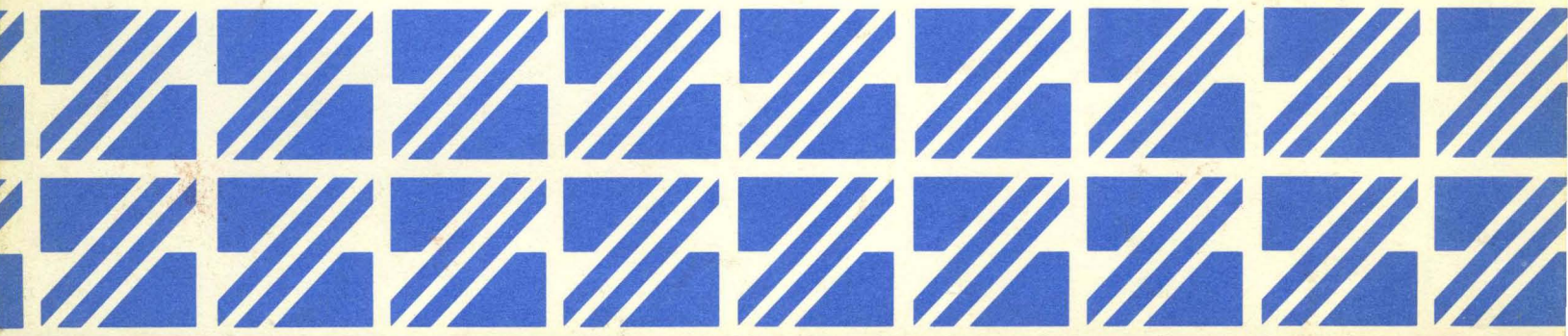
D DZ-0108-02 5

SHEET 2 OF 3


SEE SHEET 1
ZDS 1/15
J107
TTXD
TRXD
TRTS
TCTS
TDTR
TDSR
SYNC
RESET
+5V
+12V
-12V
ZILOG INC.
LOGIC DIAGRAM,
ZDS/ASPIO
02-0108-02

Zilog

10340 Bubb Road
Cupertino, California 95014
Telephone (408) 446-4666
TWX 910-338-7621



Printed in USA