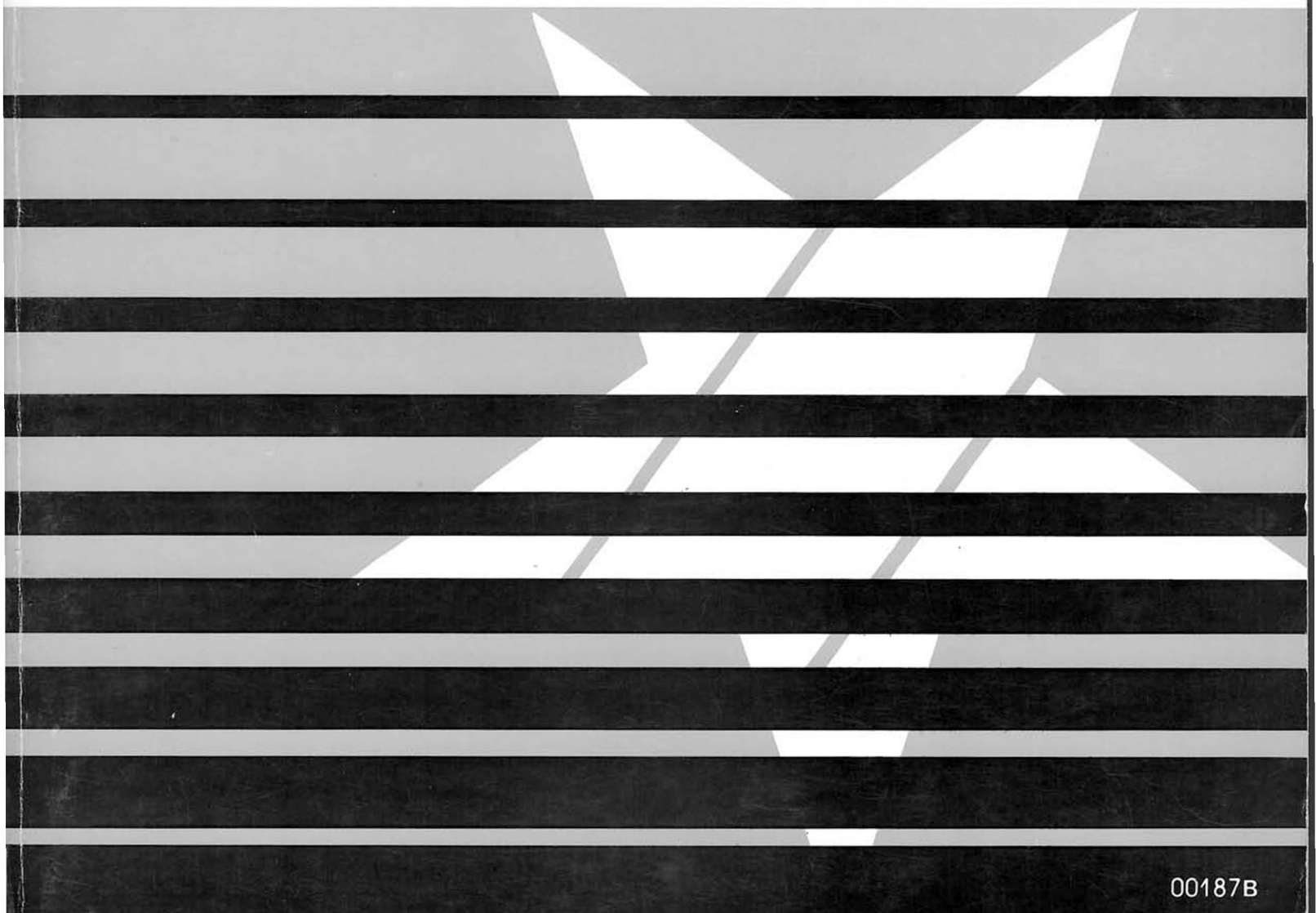
NorthStar™

HSIO-4

HORIZON
Serial Input/Output - Four Port
USER/TECHNICAL MANUAL



00187B

NorthStar™
North Star Computers, Inc.
14440 Catalina St., San Leandro, CA 94577 USA
(415) 357-8500 TWX/Telex (910) 366-7001

HSIO-4
HORIZON Serial Input/Output — Four Port
USER/TECHNICAL MANUAL

00261B

00187B

TABLE OF CONTENTS

SECTION 1 GENERAL INFORMATION

1.1 INTRODUCTION

The HSIO-4 is a 4-port serial input/output board designed for use with the HORIZON. It can also be used with some other S-100 bus systems.

This manual provides basic instructions on how to install the HSIO-4 board in your HORIZON; it also provides technical information and detailed instructions for implementing a range of HSIO-4 features and operating characteristics.

SECTION 1 describes HSIO-4 features and capabilities and contains warranty information.

SECTION 2 provides basic instructions on how to unpack and install the HSIO-4 board in a HORIZON.

SECTION 3 describes interface connections and requirements; it also provides detailed instructions on how to use the HSIO-4 to establish communication links between the HORIZON and a variety of peripheral devices.

SECTION 4 provides a functional block diagram of the HSIO-4 and discusses the theory of operation.

SECTION 5 provides troubleshooting procedures to aid in isolating problems.

The APPENDICES furnish detailed technical information, including schematic drawings, RS-232 pin assignments, and a manufacturer data sheet for the USART.

Every effort has been made to ensure the accuracy of this manual. If you find errors or omissions in this material or if you have suggestions on how it can be improved, please write:

North Star Computers, Inc.
North Star Technical Services Group
14440 Catalina Street
San Leandro, CA 94577

1.2 WARRANTY

North Star Computers, Inc., warrants the electrical and mechanical parts and workmanship of this product to be free of defects for a period of 90 days from date of purchase. If such defects occur, North Star Computers, Inc., will repair the defect at no cost to the purchaser. This warranty does not extend to defects resulting from improper use or assembly by purchaser, nor does it cover transportation to the factory. Also, the warranty is invalid if all instructions included in the accompanying documentation are not carefully followed.

Should a unit returned for warranty repair be deemed by North Star Computers, Inc. to be defective due to purchaser's action, then a repair charge (not to exceed $50 without purchaser's consent) will be assessed. ANY UNIT(S) OR PART(S) RETURNED FOR WARRANTY REPAIR MUST BE ACCOMPANIED BY A DATED COPY OF THE ORIGINAL SALES RECEIPT. The item should be returned to the dealer from whom the product was purchased, for implementation of the warranty. When sending the item to the factory for repair, the dealer must call the North Star Technical Hotline to receive a Return Material Authorization (RMA) number to accompany the item to the factory. Terminals and printers are covered under separate warranties.

The following warranty limitation applies to units located outside the United States of America: All costs and arrangements for transportation of the product to and from the factory are borne entirely by the customer.

No warranty, expressed or implied, is extended concerning completeness, correctness, or suitability of the North Star equipment for any particular application. There are no warranties which extend beyond those expressly stated herein. This limited warranty is made in lieu of all other warranties, expressed or implied, and is limited to repair or replacement of the product.

1.3
FEATURES OF THE HSIO-4

The HSIO-4 is a 4-port serial input/output board designed by North Star Computers, Inc., for use with the HORIZON and North Star multi-user systems. As supplied, the HSIO-4 is configured for use with the North Star TSS/A Multi-User System for Application Software. Additional capabilities provide flexibility and permit operation in a variety of hardware and software environments.

Synchronous/Asynchronous

The HSIO-4 provides four serial input/output ports which may be configured for synchronous or asynchronous communications.

Baud Rate Selection

Baud rates are programmable and may be selected from among ten available asynchronous baud rates and five available synchronous baud rates. Simple wiring changes provide additional baud rates.

Asynchronous Range: 75 baud to 19.2K baud
Synchronous Range: 1.76K baud to 55.8K baud

Mode Configuration

All four ports use RS-232 signal levels. A hardware change permits 20mA current loop operation. As supplied, each port is wired for connection to a standard terminal or printer. Each port has a configuration header which may be changed to permit connection to a modem or another computer.

Buffer Full Signals

The HSIO-4 accomodates printers and other peripheral devices using 'buffer full' signals.

Interrupts

Each port provides four sources of interrupt; three are maskable, the fourth is armed by a jumper. The resultant interrupt from each port may be connected to any of the S-100 bus interrupt lines.

SECTION 2 INSTALLING THE HSIO-4

2.1 UNPACKING THE HSIO-4

The HSIO-4 board is shipped in a carton containing:

1. North Star HSIO-4 board
2. Four ribbon cables
3. Plastic envelope containing eight jackscrews, lockwashers, and nuts.
4. HSIO-4 User/Technical Manual

Examine the contents of the carton to make sure they match this list. If anything appears to be damaged due to shipping, please notify your dealer immediately.

CAUTION

Before handling the board, you should ground yourself to discharge whatever static electricity may have built up on your body. The electronic components on the board are sensitive to static electricity. So, just before you handle the board, touch a grounded metal object, such as the chassis of a HORIZON that is plugged into a wall outlet. This will prevent a surge of static electricity from damaging the components.

When handling the board, touch it only by the edges to avoid bending pins on the components. If you lay the board down, put it on a flat surface with the components facing up.

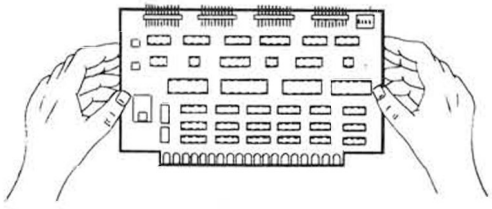
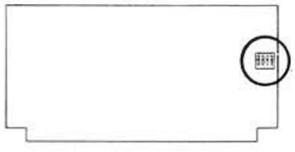


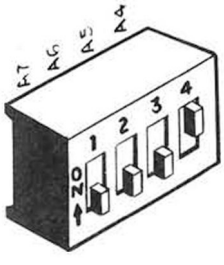
Figure 2-1

2.2
SETTING I/O ADDRESS SWITCHES FOR THE NORTH STAR TSS/A



As supplied, the I/O address switches on the HSIO-4 board are set for use with the North Star TSS/A (Multi-User System for Application Software). The I/O addresses for each port are set by the four switches located in the upper right corner of the board.

Figure 2-2



To make sure the switches have not been inadvertently changed, check them against the setting shown in Figure 2-3.

If you are using different software, refer to Section 3 for instructions on how to set the I/O address switches.

Figure 2-3

2.3
CONFIGURATION CHANGES

As supplied, the HSIO-4 is ready for immediate connection to four asynchronous RS-232 terminals or printers.

If you are using the North Star TSS/A Multi-User System for Application Software with asynchronous RS-232 terminals and printers, the HSIO-4 can be installed directly in the HORIZON without modification.

If you are using different software or using hardware with different operating characteristics, you may need to implement configuration changes and/or programming routines and should refer to Section 3 (IMPLEMENTING HSIO-4 FEATURES) for general guidelines and specific instructions before continuing this installation procedure.

IMPORTANT NOTE

Any necessary configuration changes must be completed BEFORE installing the HSIO-4 board in the HORIZON.

2.4
COMPLETING INSTALLATION

> **WARNING**
>
> Turn off the power switch on the back of the HORIZON, and unplug the power cord from the wall outlet before you remove the HORIZON cover. The high voltages used in the computer can be dangerous unless you observe this precaution.
>
> Do NOT remove the cover from the HORIZON until the power is off, the fan has stopped, and the red indicator light on the front panel has fully dimmed.

Remove the four screws that hold the HORIZON cover in place. Lift off the cover.

Hold the HSIO-4 board in one hand and touch the metal chassis of the HORIZON with the other hand. This will eliminate any difference in static potential between the HSIO-4 board and the computer.

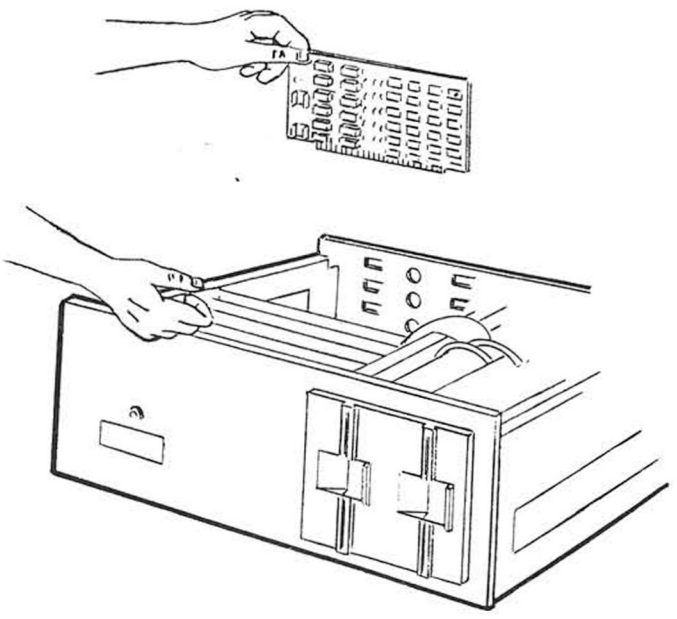


Figure 2-4

Hold the board as shown in figure 2-5, with the component side of the board toward the front of the HORIZON. Slide the board into the rear-most slot in the card cage. If that slot is already occupied by another board, remove that board and place it in an unused slot.

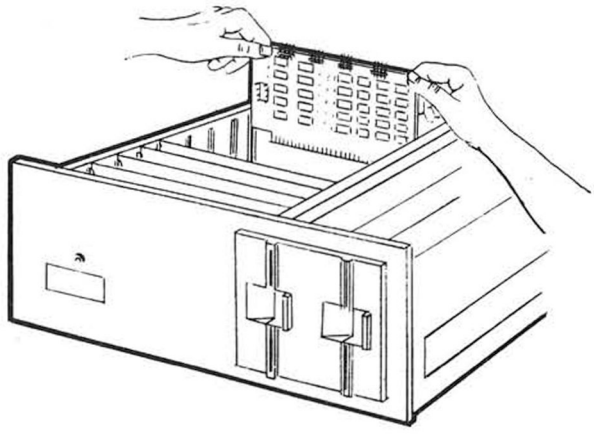


Figure 2-5

The row of metallic strips or "fingers" on the bottom of the board should fit into the connector at the base of the slot. Press firmly on the top of the board to make sure it is fully seated in the connector. The top of the HSIO-4 board should be even with the other installed North Star boards.

If a second HSIO-4 board is utilized, it should be installed in the slot immediately in front of the first HSIO-4 board.

Four flat ribbon cables are supplied with the HSIO-4 board. You will notice that each cable has a connector at either end. The Terminal/ Modem Mating Connector is shaped like an elongated "D" and has curved corners. The Board Mating Connector is rectangular.

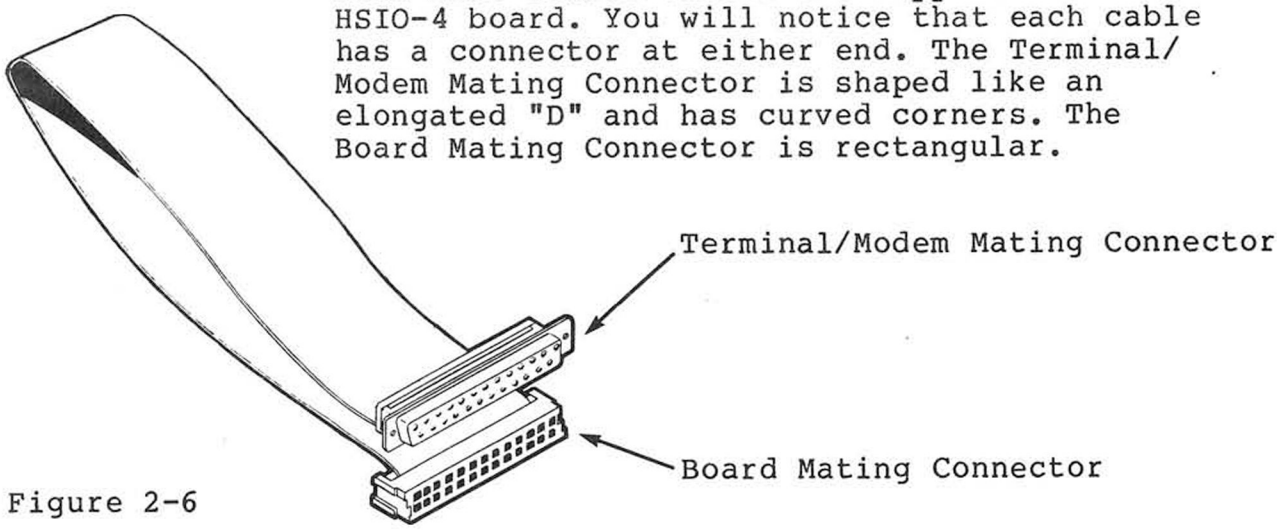


Figure 2-6

Each Board Mating Connector must be plugged into one of the four port connectors on the HSIO-4; each Terminal/Modem Mating Connector must be seated in one of the connector holes in the back panel of the HORIZON.

It is recommended that you utilize the outer-most column of connector holes, as shown in Figure 2-7. The other column of connector holes should be utilized only if a second HSIO-4 board is installed.

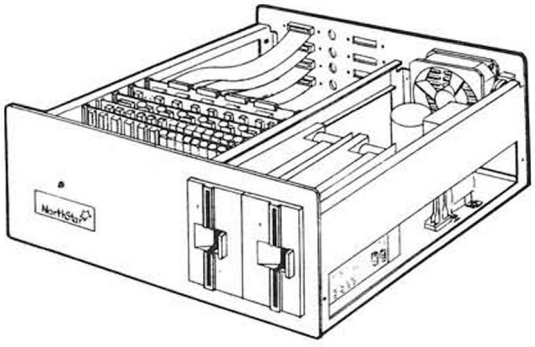


Figure 2-7

Before connecting the cables, you should number each connector hole on the HORIZON back panel. For use with the TSS/A, the connector holes should be numbered as shown in Figure 2-8.

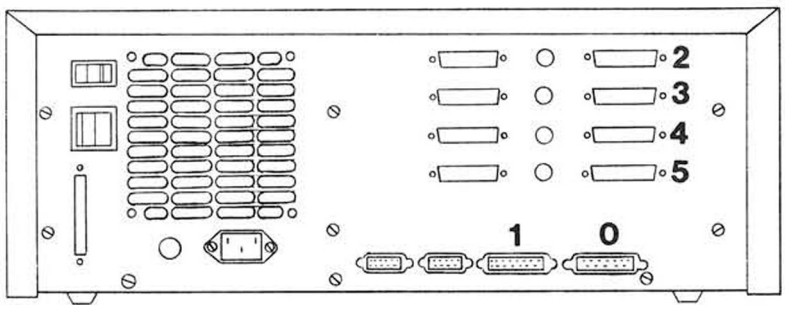


Figure 2-8

From outside the back panel of the HORIZON, thread each ribbon cable -- rectangular Board Mating Connector first -- through a connector hole. Gently pull the cable through until the Terminal/Modem Mating Connector touches the sides of the hole. From outside the back panel, push firmly to seat the connector in the hole.

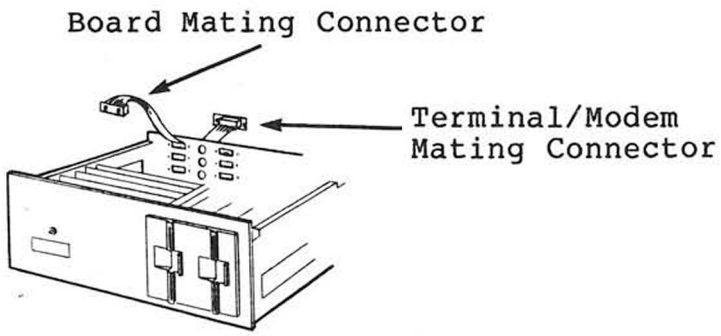


Figure 2-9

Using the jackscrews provided, bolt the D-connector in place, as shown in Figure 2-10. From outside the back panel, insert each screw through the connector mounting. Slip a lockwasher over the screw, and secure it with a nut.

Repeat these installation procedures for all four ribbon cables.

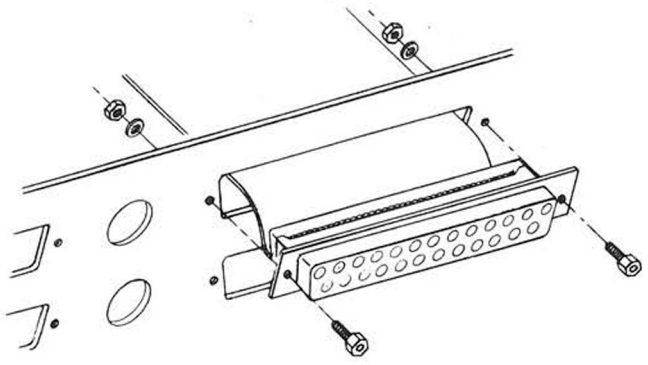


Figure 2-10

Taking care not to twist the cables, plug each Board Mating Connector into the appropriate 26-pin header at the top of the HSIO-4 board. See Figure 2-11. For use with the TSS/A, the cables should be connected as follows:

J1 to Port 2
J2 to Port 3
J3 to Port 4
J4 to Port 5

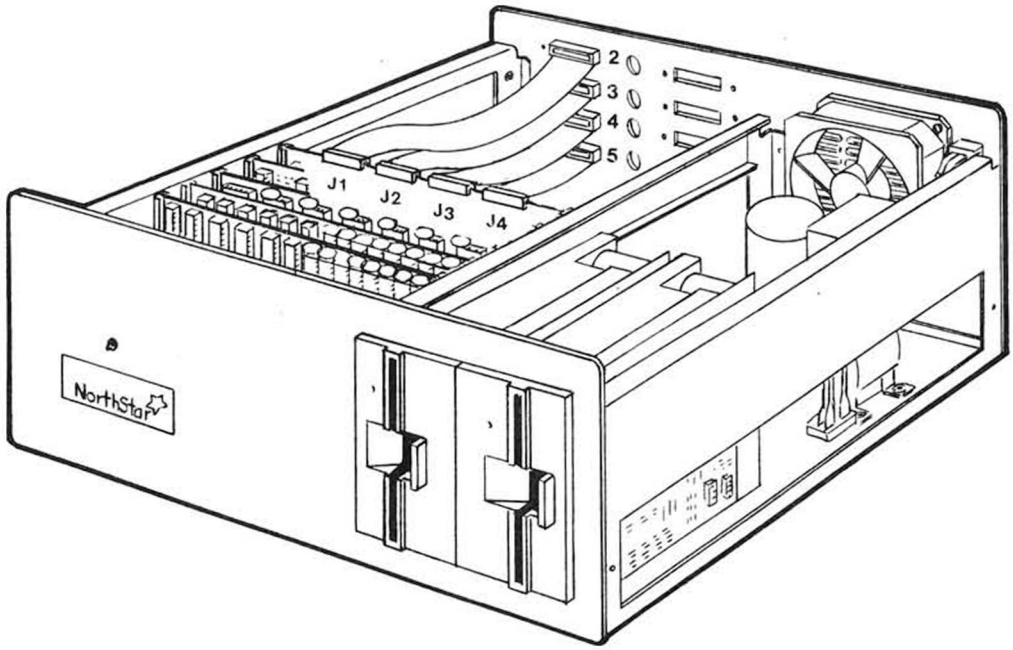


Figure 2-11

Gently flex each cable to fit it into the space between the card cage and the back panel, taking care to position the cables so they will not be crushed when you replace the lid.

Replace the cover on the HORIZON, and replace the screws.

Connect the terminals and/or printers to the appropriate I/O port sockets on the HORIZON back panel.

Plug the HORIZON line cord into the wall outlet. Plug the power cords for all connected terminals into the wall outlet. Turn on the power switches for the HORIZON and all terminals.

SECTION 3 IMPLEMENTING HSIO-4 FEATURES

This section provides an overview of serial data communication between computers and other electronic devices; it also provides detailed instructions on how to use the HSIO-4 to establish communication links between the computer and a variety of periperal devices.

As supplied, the HSIO-4 is ready for immediate connection to four asynchronous RS-232 terminals or printers. Connection to most asynchronous terminals and printers usually requires no configuration changes to the HSIO-4. Reading through the discussion that follows can help you to determine what, if any, changes are required by the particular hardware and software you are using. Once you have identified the needed modifications, you should consult the appropriate subsection for specific, detailed instructions.

3.1 GENERAL CONCEPTS

In order to transmit data between a computer and another electronic device, a communication link must be established. Such a communication link is usually comprised of a cable with a connecting plug at each end. One plug is connected to an Input/Output (I/O) port in the computer; the other is connected to an I/O port in the electronic device.

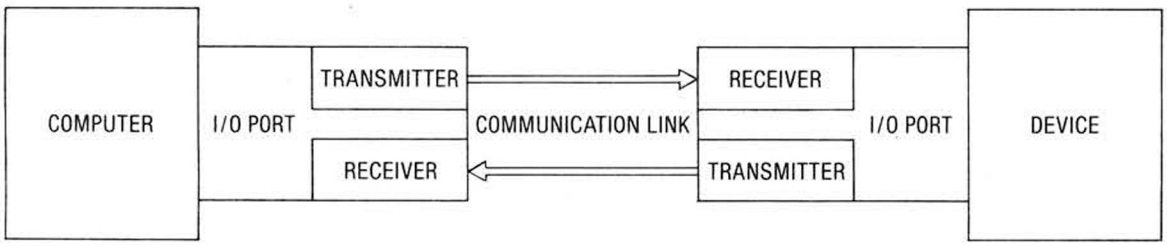


Figure 3-1

3.1.1
ESTABLISHING A COMMUNICATION LINK

To establish a communication link between two electronic devices (such as a computer and a terminal), the operating characteristics of both devices must be compatible.

The HSIO-4 has four serial I/O ports. Each I/O port provides the interface necessary to establish a communication link between a HORIZON computer and another electronic device with a compatible serial I/O port.

RS-232 OR 20mA CURRENT LOOP SIGNALS

Most computers, terminals, and printers use RS-232 signals. A few commercially used terminals, such as teletypes, use 20mA current loop signals.

As supplied, each HSIO-4 port is wired to use RS-232 signals. 20mA current loop signals can be accomodated with a configuration change. Refer to section 3.2.5 for specific instructions.

ASYNCHRONOUS OR SYNCHRONOUS TRANSMISSION

Each port on the HSIO-4 allows two different types of data transmission: asynchronous and synchronous. Typical communication between a computer and a terminal is asynchronous. Synchronous communication is used primarily to facilitate high-speed computer-to-computer data transmission.

ASYNCHRONOUS
See Section 3.2 for specific configuration and programming instructions. As supplied, each port on the HSIO-4 is configured for asynchronous operation.

SYNCHRONOUS
To operate an I/O port in synchronous mode, you must wire the Configuration Header and the Clock Header, and then program the I/O port according to the specific procedures in Section 3.3.

TERMINAL OR MODEM

To establish an RS-232 communication link between two electronic devices, one device must simulate a modem; the other must simulate a terminal. As supplied, each HSIO-4 port simulates a modem.

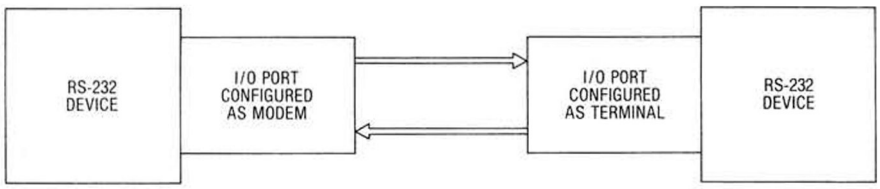


Figure 3-2

COMPUTER TO TERMINAL COMMUNICATION
If the HORIZON is to communicate with a serial terminal (such as a video terminal or a printer), the interfacing port on the HSIO-4 must be configured to simulate a modem.

COMPUTER TO MODEM COMMUNICATION
If the HORIZON is to communicate with a modem or acoustic coupler, the interfacing port on the HSIO-4 must be configured to simulate a terminal. As supplied, each port is configured to simulate a modem. To configure an HSIO-4 port as a terminal, you must wire the Configuration Header for the relevant port according to the specific procedures in Section 3.2.2 for asynchronous operation or Section 3.3 for synchronous operation.

COMPUTER TO COMPUTER COMMUNICATION
The HORIZON can communicate with another computer if one computer simulates a modem while the other simulates a terminal. As supplied, each port on the HSIO-4 board is configured to simulate a modem; it is ready to communicate with a computer that simulates a terminal.

Conversely, to establish a communication link between the HORIZON and another computer that simulates a modem, you must configure an HSIO-4 port to simulate a terminal. To configure an HSIO-4 port as a terminal, you must wire the Configuration Header for the relevant port according to the specific procedures in Section 3.2.2 for asynchronous operation or Section 3.3 for synchronous operation.

BAUD RATE

The data transmission speed (baud rate) of the two communicating devices must be the same. Baud rates for each port can be programmed independently to a range of commonly used baud rates. For a list of available baud rates and specific programming instructions, refer to Section 3.2.3 for asynchronous operation, or Section 3.3.2 for synchronous operation.

FORMAT CONVENTIONS

Both communicating devices must use compatible format conventions. Format conventions include: the number of data bits per character, the type of parity, the baud rate factor (1x, 16x, or 64x), the number of stop bits per character (for asynchronous operation) and the sync character(s) (for synchronous operation).

Format conventions are established by programming the USART, which can accomodate a variety of conventions. Software (such as the North Star TSS/A) that has been written specifically for use with the HSIO-4 contains USART initialization routines, which define and establish the required format conventions. For the convenience of the programmer writing software for the HSIO-4, a

sample USART initialization routine that establishes a commonly used set of format conventions is printed in Section 3.2.4 (asynchronous) and Section 3.3.5 (synchronous). In addition, the complete manufacturer specification for the USART is reproduced in Appendix B, along with notes on the interface to the USART.

3.1.2
HSIO-4 BOARD LAYOUT

Before proceeding to implement any of the features treated in this section, please note:

The four ports on the HSIO-4 board are identically arranged; they are labeled alphabetically from left to right as shown below.

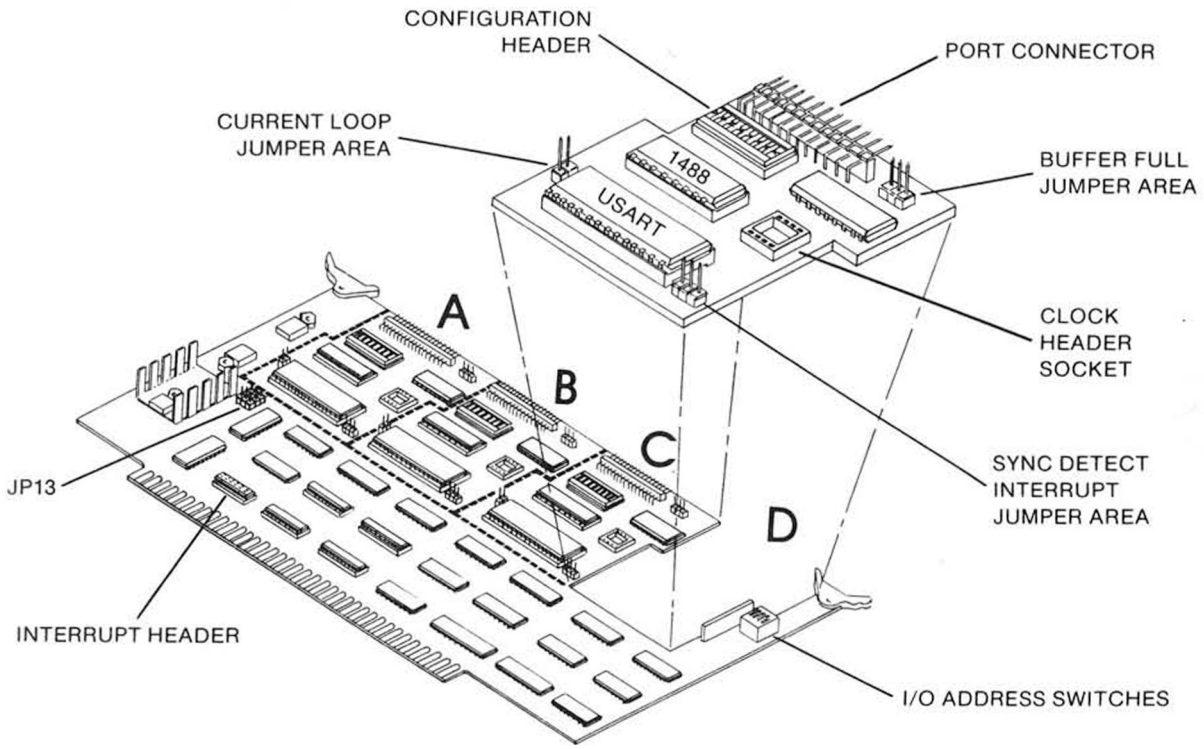


Figure 3-3

3.1.3
I/O ADDRESSES

The HSIO-4 requires 16 consecutive I/O addresses. (All addresses referenced in this section are I/O addresses.) Each of the four ports uses four consecutive addresses. Addresses start at X0H and run to XFH. (X represents a hexadecimal digit which is set by switches on the board.)

As can be seen in Table 3-1, port A starts at X0H; port B starts at X4H; port C starts at X8H; port D starts at XCH. The board address space is used as follows:

Table 3-1

I/O PORT ADDRESS ASSIGNMENTS

Port	Address	Function	Port Access
A	X0H	Baud rate select	write only
A	X1H	Interrupt mask set	write only
A	X2H	USART data	read/write
A	X3H	USART status/control	read/write
B	X4H	Baud rate select	write only
B	X5H	Interrupt mask set	write only
B	X6H	USART data	read/write
B	X7H	USART status/control	read/write
C	X8H	Baud rate select	write only
C	X9H	Interrupt mask set	write only
C	XAH	USART data	read/write
C	XBH	USART status/control	read/write
D	XCH	Baud rate select	write only
D	XDH	Interrupt mask set	write only
D	XEH	USART data	read/write
D	XFH	USART status/control	read/write

Figure 3-4

The hexadecimal digit X used as part of the I/O address is set by the four switches located in the upper right corner of the board (Location 8B).

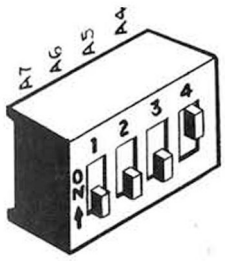


Figure 3-5

The four switches yield a total of 16 possible switch settings; each switch setting selects a starting address for the I/O board. Unless otherwise directed by the software manual you are using, use the switch setting illustrated in Figure 3-5 at left, which sets the starting address at 10H.

If the software you are using requires a different starting address, consult Table 3-2 below for the switch setting that corresponds to the required starting address.

Table 3-2

I/O ADDRESS SWITCH SETTINGS

0 = Off
1 = On

Starting Address	Switch Number				Reserved Address Regions
	A7	A6	A5	A4	
00H	0	0	0	0	Used by the HORIZON motherboard
10H	0	0	0	1	
20H	0	0	1	0	
30H	0	0	1	1	
40H	0	1	0	0	
50H	0	1	0	1	
60H	0	1	1	0	Reserved for North Star use
70H	0	1	1	1	Reserved for North Star use
80H	1	0	0	0	
90H	1	0	0	1	
A0H	1	0	1	0	
B0H	1	0	1	1	
C0H	1	1	0	0	Used by North Star RAM boards
D0H	1	1	0	1	
E0H	1	1	1	0	
F0H	1	1	1	1	

3.1.4 INTERRUPTS

In a multi-user environment, interrupts relieve the processor of the task of polling each peripheral device to see if it is ready to transmit or receive data. Different events generate different interrupt requests. One interrupt request, for example, is generated when the USART has finished sending a byte and is ready to send another; another interrupt request is generated when a byte is ready to be received by the processor.

Each HSIO-4 port provides four interrupt sources. Three may be armed and disarmed under program control; the fourth, Synchronous Detect, is armed by a jumper on the HSIO-4 board. Consult Section 3.4 for instructions on how to program the interrupts and connect interrupt sources to the S-100 bus interrupt lines.

3.2 ASYNCHRONOUS OPERATION

To establish a communication link between two electronic devices, one device must simulate a modem while the other simulates a terminal. If the HORIZON is to communicate with a serial terminal (such as a video terminal, a teletype, or a printer), then the relevant HSIO-4 port must be configured to simulate a modem. Similarly, if the HORIZON is to communicate with a modem, then the relevant HSIO-4 port must simulate a terminal.

3.2.1 CONFIGURING PORT AS MODEM

As supplied, each port on the HSIO-4 is configured as a modem; it is ready for connection to an RS-232 terminal. If a port has ever been reconfigured as a terminal, it can be restored to its original configuration as follows:

1. Remove the Clock Header (if there is one present) from the Clock Header socket.

   Please Note: As supplied, Clock Headers are not installed on the HSIO-4 board. However, if a port has been configured for synchronous operation, then a Clock Header will have been installed and must now be removed.

2. Check the shunt in the Configuration Header socket; it should have the following pins connected. (If the original shunt is no longer available, then make up a header with the same connections.)

pin 1 to pin 16
pin 2 to pin 15
pin 3 to pin 14
pin 4 to pin 13
pin 5 to pin 12
pin 6 to pin 11
pin 7 to pin 10
pin 8 to pin 9

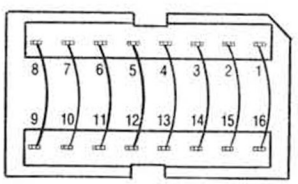


Figure 3-6

3.2.2
CONFIGURING PORT AS TERMINAL

If the HORIZON is to communicate with a modem (or with another computer simulating a modem), the interfacing HSIO-4 port must be configured to simulate a terminal.

Configure an HSIO-4 port as a terminal as follows:

1. Carefully remove the shunt from the Configuration Header socket and replace it with a 16-pin header wired as follows:

pin 1 to pin 2
pin 3 to pin 4
pin 5 to pin 6
pin 7 to pin 10
pin 8 to pin 9
pin 11 to pin 12
pin 13 to pin 14
pin 15 to pin 16

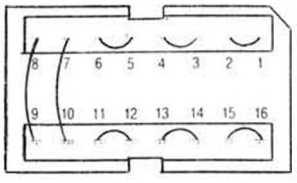


Figure 3-7

2. Remove the Clock Header (if there is one present) from the Clock Header socket.

   Please Note: As supplied, Clock Headers are not installed on the HSIO-4 board. However, if a port has been configured for synchronous operation, then a Clock Header will have been installed and must now be removed.

3.2.3
BAUD RATE SELECTION

The baud rate for each HSIO-4 port is programmable and can be selected from among a total of ten available asynchronous baud rates. See Table 3-3 below.

Table 3-3

ASYNCHRONOUS BAUD RATES

Output Value	Standard	Alternatives (With Jumper Change)
00H	19200	
01H	9600	
02H	4800	
03H	2400	1200 or 150 or 75
04H	1200	2400 or 75
05H	600	
06H	300	
07H	110	

The eight "standard" baud rates in Table 3-3 can be program-selected without jumper changes. Jumper changes can make the "alternative" baud rates available for program selection. (For specific instructions, see Section 3.2.3 B.)

A. PROGRAMMING BAUD RATE SELECTION

The baud rate for each port is programmed independently. To program the desired baud rate for a port, output the Output Value (from Table 3-3) to the Baud Rate Select Address for the relevant port.

To determine the Baud Rate Select Address, consult Tables 3-1 and 3-2 (in Section 3.1.3).

FOR EXAMPLE: Assuming the HSIO-4 board is addressed starting at 10H, you would set the baud rate to 300 on port B by outputting 06H (the Output Value) to I/O address 14H with the following instructions:

```
0000  3E06          MVI   A,6     ; Output Value for 300 Baud Async.
0002  D314          OUT   14H
```

B. ALTERNATIVE BAUD RATES (REQUIRED JUMPER CHANGES)

The baud rates listed in Table 3-3 as requiring a jumper change can be made available as described below. Please Note: Such a jumper change will affect all four ports.

1. Locate JP13, turn the board so that the solder side faces up, and cut one of the traces as directed below.

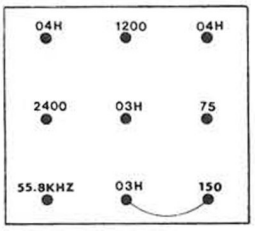


Figure 3-8

A. To implement one of the baud rates for Output Value 03H:

Cut the bottom trace, as shown in Figure 3-9. Then turn the board so that the component side faces up, and install a mini-jump between the pin marked with the speed you wish to specify and the nearest pin marked "03H." Figure 3-8, for example, shows where to install the mini-jump to specify 150 baud.

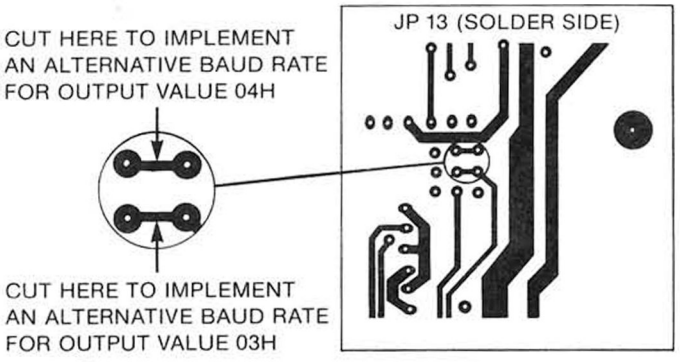


Figure 3-9

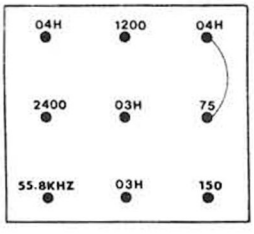


Figure 3-10

B. To implement one of the baud rates for Output 04H:

Cut the top trace, as shown in Figure 3-9. Then, turn the board so that the component side faces up and install a mini-jump between the pin marked with the speed you wish to specify and the nearest pin marked "04H." Figure 3-10, for example, shows where to install the mini-jump to specify 75 baud.

3.2.4
SAMPLE PROGRAM

This section provides an example of how to program the four HSIO-4 ports for use with four asynchronous RS-232 terminals and/or printers. Programming other configurations requires a detailed familiarity with USART operation. For the convenience of the programmer, a complete manufacturer data sheet for the USART is reproduced in Appendix B.

The following sample program:

1. checks input status and inputs a character for each of the four ports;

2. checks output status and outputs a character for each of the four ports;

3. initializes the HSIO-4 board.

Nine routines are given:

```
CINA   inputs a character from port A
CINB   inputs a character from port B
CINC   inputs a character from port C
CIND   inputs a character from port D


COUTA outputs a character to port A
COUTB outputs a character to port B
COUTC outputs a character to port C
COUTD outputs a character to port D


INIT   initializes the HSIO-4 board
```

```
0000                ; Sample asynchronous I/O routines for HSIO-4
0000                ;
0000                ;
0010                BASE    EQU     10H             ;Address of HSIO-4
0010                PORTA   EQU     BASE+00H        ;Address of port A
0014                PORTB   EQU     BASE+04H        ;Address of port B
0018                PORTC   EQU     BASE+08H        ;Address of port C
001C                PORTD   EQU     BASE+0CH        ;Address of port D
0000                ;
0001                BAUDA   EQU     1               ;Baud rate for port A of 9600
0001                BAUDB   EQU     1               ;Baud rate for port B of 9600
0001                BAUDC   EQU     1               ;Baud rate for port C of 9600
0001                BAUDD   EQU     1               ;Baud rate for port D of 9600
0000                ;
0000                BAUD    EQU     0               ;Baud rate set
0001                MADK    EQU     1               ;Interrupt mask set
0002                UDAT    EQU     2               ;USART data
0003                USTA    EQU     3               ;USART status
0000                ;
0000                ;
0000                ; Port A input and output routines
0000                ;
0000  DB13          CINA    IN      PORTA+USTA      ;Check USART status
0002  E602                  ANI     2               ;Get RxReady bit
0004  28FA                  JRZ     CINA            ;Wait till character ready
0006  DB12                  IN      PORTA+UDAT      ;Read character
0008  E67F                  ANI     7FH             ;Mask off top bit
000A  C9                    RET
000B                ;
000B  DB13          COUTA   IN      PORTA+USTA      ;Check USART status
000D  E601                  ANI     1               ;Get TxReady bit
000F  28FA                  JRZ     COUTA           ;Wait till ready
0011  78                    MOV     A,B             ;Output char is in B reg
0012  D313                  OUT     PORTA+USTA      ;Output character
0014  C9                    RET
0015                ;
0015                ;
0015                ; Port B input and output routines
0015                ;
0015  DB17          CINB    IN      PORTB+USTA      ;Check USART status
0017  E602                  ANI     2               ;Get RxReady bit
0019  28FA                  JRZ     CINB            ;Wait till character ready
001B  DB16                  IN      PORTB+UDAT      ;Read character
001D  E67F                  ANI     7FH             ;Mask off top bit
001F  C9                    RET
0020                ;
0020  DB17          COUTB   IN      PORTB+USTA      ;Check USART status
0022  E601                  ANI     1               ;Get TxReady bit
0024  28FA                  JRZ     COUTB           ;Wait till ready
0026  78                    MOV     A,B             ;Output char is in B reg
0027  D317                  OUT     PORTB+USTA      ;Output character
0029  C9                    RET
002A                ;
```

```
002A                    ;
002A                    ; Port C input and output routines
002A                    ;
002A  DB1B      CINC    IN      PORTC+USTA      ;Check USART status
002C  E602              ANI     2               ;Get RxReady bit
002E  28FA              JRZ     CINC            ;Wait till character ready
0030  DB1A              IN      PORTC+UDAT      ;Read character
0032  E67F              ANI     7FH             ;Mask off top bit
0034  C9                RET
0035                    ;
0035  DB1B      COUTC   IN      PORTC+USTA      ;Check USART status
0037  E601              ANI     1               ;Get TxReady bit
0039  28FA              JRZ     COUTC           ;Wait till ready
003B  78                MOV     A,B             ;Output char is in B reg
003C  D31B              OUT     PORTC+USTA      ;Output character
003E  C9                RET
003F                    ;
003F                    ;
003F                    ; Port D input and output routines
003F                    ;
003F  DB1F      CIND    IN      PORTD+USTA      ;Check USART status
0041  E602              ANI     2               ;Get RxReady bit
0043  28FA              JRZ     CIND            ;Wait till character ready
0045  DB1E              IN      PORTD+UDAT      ;Read character
0047  E67F              ANI     7FH             ;Mask off top bit
0049  C9                RET
004A                    ;
004A  DB1F      COUTD   IN      PORTD+USTA      ;Check USART status
004C  E601              ANI     1               ;Get TxReady bit
004E  28FA              JRZ     COUTD           ;Wait till ready
0050  78                MOV     A,B             ;Output char is in B reg
0051  D31F              OUT     PORTD+USTA      ;Output character
0053  C9                RET
```

```
0054                    ;
0054                    ; HSIO-4 initialization routine
0054                    ;
0054  3E01       INIT   MVI    A,BAUDA
0056  D310              OUT    PORTA+BAUD      ;Set baud rates
0058  3E01              MVI    A,BAUDB
005A  D314              OUT    PORTB+BAUD
005C  3E01              MVI    A,BAUDC
005E  D318              OUT    PORTC+BAUD
0060  3E01              MVI    A,BAUDD
0062  D31C              OUT    PORTD+BAUD
0064                    ;
0064                    ; Interrupt masks are cleared at power up
0064                    ;
0064  3E03              MVI    A,3             ;Give USART commands
0066  CD8700            CALL   USTAT           ;to reset.
0069  CD8700            CALL   USTAT
006C  3E40              MVI    A,40H
006E  CD8700            CALL   USTAT
0071  3ECE              MVI    A,0CEH          ;Give mode command
0073  CD8700            CALL   USTAT           ;2 STOP BITS, 16*CLK,
0076  3E27              MVI    A,27H           ;Give command.
0078  CD8700            CALL   USTAT           ;CMD: RTS,ER,RXF,DTR,TXEN
007B  CD7E00            CALL   INJNK           ;Read junk twice
007E  DB12       INJNK  IN     PORTA+UDAT
0080  DB16              IN     PORTB+UDAT
0082  DB1A              IN     PORTC+UDAT
0084  DB1E              IN     PORTD+UDAT
0086  C9                RET
0087                    ;
0087  D313       USTAT  OUT    PORTA+USTA      ;Output command to
0089  D317              OUT    PORTB+USTA      ;All HSIO-4 channels
008B  D31B              OUT    PORTC+USTA
008D  D31F              OUT    PORTD+USTA
008F  C9                RET
0090                    ;
0090                    END
```

3.2.5
CURRENT LOOP OPERATION

Most computers, terminals, and printers use RS-232 signal levels. A few commercially used terminals, such as teletypes, use 20 mA current loop signals.

A teletype is a passive device; it does not supply current, but relies on current supplied by the HSIO-4. The HSIO-4 is not equipped to communicate with active current loop devices such as computers that produce current loop signals.

As supplied, each HSIO-4 port is configured to use RS-232 signals. To configure an HSIO-4 port for current loop operation:

1. Install a mini-jump at the current loop jumper area for the relevant port. See Table 3-4 below.

Table 3-4

CURRENT LOOP JUMPER AREAS

Port	Jumper Area
A	JP5
B	JP6
C	JP7
D	JP8

2. Ensure that the original Configuration Header is in place. The pins should be connected as follows:

pin 1 to pin 16
pin 2 to pin 15
pin 3 to pin 14
pin 4 to pin 13
pin 5 to pin 12
pin 6 to pin 11
pin 7 to pin 10
pin 8 to pin 9

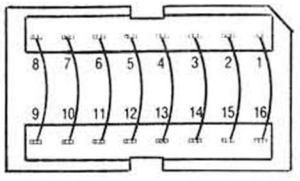


Figure 3-11

3. Remove the 1488 for the port and replace it with a header constructed as follows:

a. Connect a 2N3904 transistor to the 14-pin header with the E lead connected to pin 7, the B lead connected to pin 5, and the C lead connected to pin 6.

b. Solder a 5.6K ohm 1/4 Watt resistor (green-blue-red) between pin 4 and pin 12 on the header.

c. Solder a 1K ohm 1/4 Watt resistor (brown-black-red) between pin 8 and pin 14 on the header.

4. Connect a 25-pin D-type connector to the terminal cable as follows:

pin 9 to the printer "+" lead
pin 3 to the printer "-" lead
pin 2 to the keyboard "+" lead
pin 10 to the keyboard "-" lead

3.2.6
BUFFER FULL SIGNALS

The HSIO-4 supports printers that indicate buffer full status on pin 20 (DTR) or on pin 19 (SCA). Consult the manual for your printer to determine which pin it uses to indicate buffer full status. Depending upon the maufacturer, this signal may be identified as "Printer Ready" or "Buffer Full." The HSIO-4 expects this signal to be high when it is required to send characters and low when it is required to wait.

As supplied, the HSIO-4 expects the buffer full signal on pin 20. Some printers, however, use pin 19. Printers using pin 19 require the following modification to the HSIO-4 board:

1. Locate the 3-pin jumper area to the right of the connector for the port to be modified. (See Table 3-5 below).

Table 3-5

BUFFER FULL JUMPER AREAS

Port	Jumper Area
A	JP1
B	JP2
C	JP3
D	JP4

2. Turn the board over so the solder side faces up, and cut the trace that connects the center pin and pin #2 (See Figure 3-12).

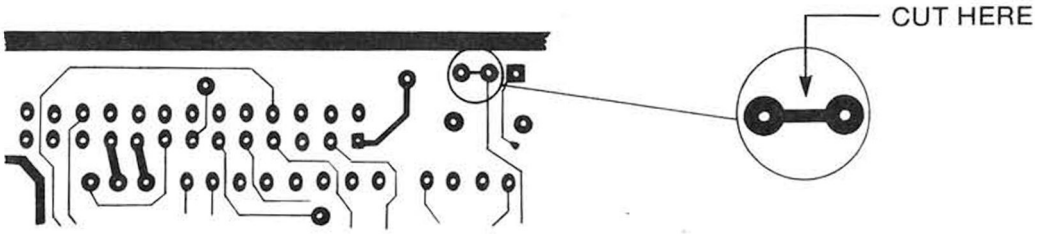


Figure 3-12

3. On the component side of the board, install a mini-jump between the center pin and pin #1. (The pin numbers are printed on the component side of the board.)

3.3
SYNCHRONOUS OPERATION

Used primarily to facilitate rapid data communication between two computers, synchronous transmission is also used to communicate with devices that are capable only of synchronous operation, such as high-speed modems.

During synchronous operation, data transmission is synchronized and controlled by equally spaced clock signals. The clock signal is generated by the transmitting port and then used by the receiving port. The timing of the clock signal (and speed at which the sending port transmits data) is determined by the baud rate setting for that port. Thus, the baud rate for both the transmitting port and the receiving port is determined by the baud rate selected at the transmitting end.

As supplied, all four HSIO-4 ports are configured for asynchronous operation. Each port can be reconfigured for synchronous operation (see Sections 3.3.1. and 3.3.2). Sections 3.3.3 and 3.3.5 provide examples of how to program synchronous operation.

Each HSIO-4 port may be configured for synchronous operation in one of two possible ways:

FIXED BAUD RATE (55.8KHz)
The first method allows a synchronous port to transmit at a single fixed baud rate of 55.8KHz, which is the maximum possible synchronous speed provided on the HSIO-4. This method requires two configuration changes, which effect only the modified port.

PROGRAMMABLE BAUD RATES
The second method permits a synchronous port to use any one of five programmable baud rates ranging from 1.76KHz to 38.4KHz. A programmable 55.8KHz baud rate is also possible, but requires an additional jumper change, which effects all four ports.

3.3.1
FIXED BAUD RATE OF 55.8KHz

To configure an individual port for synchronous operation at a single fixed baud rate of 55.8KHz, make the following changes:

Please Note: At the same time that a port is configured for synchronous operation, it must also be configured to simulate either a modem or a terminal. To determine whether your particular application requires the MODEM or TERMINAL configuration, consult Section 3.1.1.

3.3.1 A
CONFIGURING PORT AS MODEM (FIXED BAUD RATE)

1. Disconnect the links between pins 7 and 10 and between pin 8 and 9 on the Configuration Header.

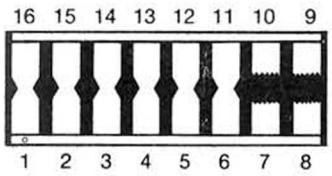


Figure 3-13

2. Take an 8-pin shunt and install it, with all four links connected (as shown in Figure 3-14), in the Clock Header socket.

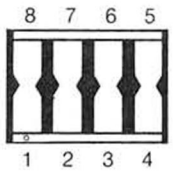


Figure 3-14

3.3.1 B
CONFIGURING PORT AS TERMINAL (FIXED BAUD RATE)

1. Wire a 16-pin header as follows, and install it in the Configuration Header socket.

pin 1 to pin 2
pin 3 to pin 4
pin 5 to pin 6
pin 11 to pin 12
pin 13 to pin 14
pin 15 to pin 16

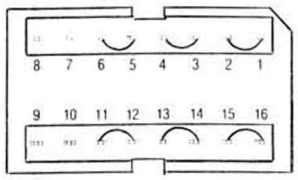


Figure 3-15

2. Wire an 8-pin header as follows, and install it in the Clock Header socket.

pin 1 to pin 2
pin 3 to pin 6
pin 4 to pin 5
pin 7 to pin 8

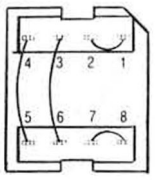


Figure 3-16

3.3.2
PROGRAMMABLE BAUD RATES

To configure an individual port for synchronous operation with programmable baud rates, make the following changes:

Please Note: When a port is configured for synchronous operation, it must also be configured to simulate either a MODEM or a TERMINAL. (To determine whether your particular application requires the MODEM or TERMINAL configuration, consult Section 3.1.1.)

3.3.2 A
CONFIGURING PORT AS MODEM (PROGRAMMABLE BAUD RATES)

1. Disconnect the link between pins 8 and 9 on the Configuration Header.

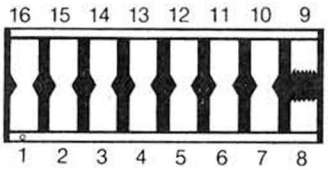


Figure 3-17

2. Take an 8-pin shunt, disconnect the link between pin 3 and pin 6, and install it in the Clock Header socket.

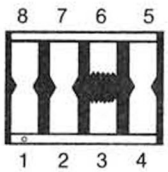


Figure 3-18

3. Refer to Section 3.3.3 for instructions on programming baud rates for synchronous operation.

3.3.2 B
CONFIGURING PORT AS TERMINAL (PROGRAMMABLE BAUD RATES)

1. Wire a 16-pin header as follows, and install it in the Configuration Header socket.

pin 1 to pin 2
pin 3 to pin 4
pin 5 to pin 6
pin 7 to pin 10
pin 11 to pin 12
pin 13 to pin 14
pin 15 to pin 16

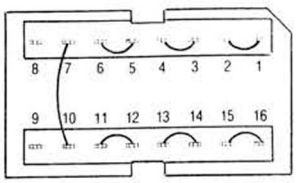


Figure 3-19

2. Wire an 8-pin header as follows, and install it in the Clock Header socket.

pin 1 to pin 2
pin 4 to pin 5
pin 7 to pin 8

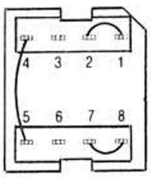


Figure 3-20

2. Refer to Section 3.3.3 for instructions on programming baud rates for synchronous operation.

3.3.3
PROGRAMMING BAUD RATES

When configured for synchronous operation with programmable baud rates, an HSIO-4 port can transmit data at any one of five programmable baud rates. See Table 3-6 below. An additional jumper change enables three additional programmable baud rates. (For specific instructions, see Section 3.3.4.)

During synchronous operation, the baud rate of the receiving port is determined by the clock signal generated by the transmitting port. Thus, each HSIO-4 baud rate selection determines only the transmission speed for a particular port -- not the receiving speed.

Table 3-6

PROGRAMMABLE SYNCHRONOUS BAUD RATES

Output Value	Standard	Alternatives (With Jumper Change)
00H	Not Used	
01H	Not Used	
02H	Not Used	
03H	38400	19200 or 2400 or 1200 or 55.8K
04H	19200	38400 or 1200
05H	9600	
06H	4800	
07H	1760	

Please Note: Output values 00H, 01H and 02H cannot be used in synchronous mode as they exceed USART speed capabilities.

The baud rate for each port is programmed independently. To program the desired baud rate for a port, output the Output Value (from Table 3-6) to the Baud Rate Select Address for the relevant port.

To determine the Baud Rate Select Address, consult Table 3-1 (in Section 3.1.3).

For example, assuming the HSIO-4 board is addressed starting at 10H, you would set the baud rate to 4800 on port B by outputting 06H (the Output Value) to I/O address 14H with the following instructions:

```
0000 3E06          MVI   A,6    ; Output Value for 4800 Baud Sync.
0002 D314          OUT   14H
```

The baud rate must be selected BEFORE executing software instructions to set up the USART for synchronous transmission.

3.3.4
ALTERNATIVE BAUD RATES (REQUIRED JUMPER CHANGES)

The baud rates listed in Table 3-6 as "Alternatives (With Jumper Change)" can be made available as described below. Please Note: Such a jumper change effects all four HSIO-4 ports.

1. Locate JP13, turn the board so that the solder side faces up, and cut one of the traces as directed below.

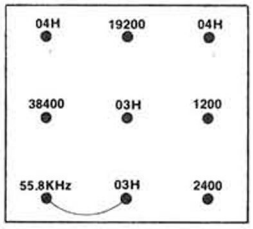


Figure 3-21

A. To implement one of the baud rates for Output Value 03H:

Cut the bottom trace, as shown in Figure 3-22. Then turn the board so that the component side faces up, and install a mini-jump between the pin marked with the speed you wish to specify and the nearest pin marked "03H." Figure 3-21, for example, shows where to install the mini-jump to specify 55.8KHz.

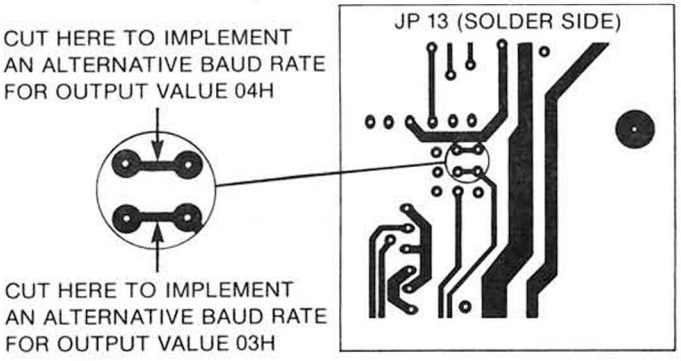


Figure 3-22

B. To implement one of the baud rates for Output Value 04H:

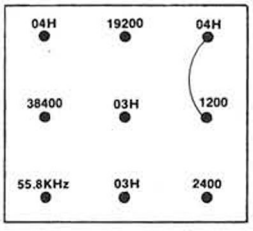


Figure 3-23

Cut the top trace, as shown in Figure 3-22. Then turn the board so that the component side faces up and install a mini-jump between the pin marked with the speed you wish to specify and the nearest pin marked "04H." Figure 3-23, for example, shows where to install the mini-jump to specify 1200 baud.

3.3.5
SAMPLE PROGRAM

This section provides a very basic example of how to program a USART in the HSIO-4 for synchronous operation. Programming for specific applications requires a detailed familiarity with USART operation. It is beyond the scope of this manual to provide specific instructions on programing for such applications. However, for the convenience of the programmer, the manufacturer data sheet for the USART is reproduced in Appendix B.

Three routines are given:

INIT sets up the USART in synchronous mode, waiting to receive sync characters.

SYNI loads a received message into RAM starting at the address given in HL.

SYNO transmits a message from RAM starting at the address given in HL. The message length (number of bytes) is given in BC. The message is preceded by 255 DLE-SYN sync character pairs and a DLE-STX; this indicates the start of the message. The message is terminated by a DLE-ETX; this indicates the end of the message.

As the data transferred is binary and may contain any character, an escape character must be used to indicate the presence of control characters such as End-of-text, Start-of-text, and Sync. The escape character used here is DLE (10H). If a DLE character occurs in the data, this is replaced by two DLEs in sequence.

```
0000                ; Sample routines to use HSIO-4 for synchronous operation
0000                ;
0000                ;
0002                STX     EQU     2       ; Start of text character
0003                ETX     EQU     3       ; End of text character
0010                DLE     EQU     10H     ; Data Link Escape character
0016                SYN     EQU     16H     ; Sync character
0000                ;
0001                TXRDY   EQU     1       ; USART status bits
0002                RXRDY   EQU     2
0000                ;
0010                PORTA   EQU     10H     ; Set for HSIO-4 port A
0010                BAUD    EQU     PORTA   ; Baud rate for port A
0012                DATA    EQU     PORTA+2 ; USART data address
0013                CTRL    EQU     PORTA+3 ; USART control/status
0000                ;
0000                ;
0000  3E03          INIT    MVI     A,3     ; Set data rate to 2400*16
0002  D310                  OUT     BAUD    ; for HSIO-4
0004  3E80                  MVI     A,80H   ; Ensure USART is cleared
0006  D313                  OUT     CTRL    ; as specified by manufacturers
0008  D313                  OUT     CTRL
000A  3E40                  MVI     A,40H   ; do reset
000C  D313                  OUT     CTRL
000E                ;
000E  3E0C                  MVI     A,0CH   ; Double sync, no parity
0010  D313                  OUT     CTRL
0012  3E10                  MVI     A,DLE   ; Sync character #1
0014  D313                  OUT     CTRL
0016  3E16                  MVI     A,SYN   ; Sync character #2
0018  D313                  OUT     CTRL
001A  3EB7                  MVI     A,0B7H  ; Hunt,RTS,Error reset,RxE,DTR,TxE
001C  D313                  OUT     CTRL
001E  DB12                  IN      DATA    ; Read junk
0020  C9                    RET
```

```
0021                    ;
0021                    ; Synchronous input routine (RAM address in HL)
0021                    ;
0021  CD0000    SYNI    CALL    INIT    ; Set USART into hunt mode and
0024  CD5100            CALL    GETCH   ; reset errors
0027  FE10              CPI     DLE
0029  20F6              JRNZ    SYNI    ; Wait for DLE to appear
002B  CD5100            CALL    GETCH
002E  FE16              CPI     SYN     ; If SYNC, try again
0030  28EF              JRZ     SYNI
0032  FE02              CPI     STX     ; Check for start of text,
0034  20EB              JRNZ    SYNI    ; if bad, try again
0036                    ;
0036                    ; Transfer message into RAM
0036                    ;
0036  CD5100    SDATA   CALL    GETCH
0039  FE10              CPI     DLE
003B  2010              JRNZ    RAMLD   ; If not DLE then data
003D  CD5100            CALL    GETCH   ; Get second char of DLE seq
0040  FE10              CPI     DLE     ; If DLE-DLE then use one
0042  2809              JRZ     RAMLD   ; of them as data
0044  FE16              CPI     SYN     ; Check for padding (SYNC chars)
0046  28EE              JRZ     SDATA   ; ignore if it is
0048  FE03              CPI     ETX     ; End yet ?
004A  C8                RZ      ;       ; If not done, then bad DLE
004B  18E9              JR      SDATA   ; sequence found, ignore it
004D                    ;
004D  77        RAMLD   MOV     M,A     ; Insert byte into RAM at (HL)
004E  23                INX     H
004F  18E5              JR      SDATA   ; Get next byte
0051                    ;
0051  DB13      GETCH   IN      CTRL    ; Get char from serial port
0053  E602              ANI     RXRDY
0055  28FA              JRZ     GETCH   ; Wait till done
0057  DB12              IN      DATA
0059  C9                RET
```

```
005A                    ;
005A                    ; Synchronous output routine
005A                    ; Outputs BC characters starting at address in HL
005A                    ;
005A  CD0000      SYNO  CALL  INIT    ; Reset USART
005D  C5                PUSH  B       ; Save byte count
005E  0600              MVI   B,0     ; Send 255 DLE-SYNCs
0060  3E10        HEADR MVI   A,DLE   ; before message
0062  CD9100            CALL  OPCH
0065  3E16              MVI   A,SYN
0067  CD9100            CALL  OPCH
006A  10F4              DJNZ  HEADR
006C  C1                POP   B       ; Restore byte count
006D                    ;
006D  3E10              MVI   A,DLE   ; Send message header of
006F  CD9100            CALL  OPCH    ; DLE STX
0072  3E02              MVI   A,STX
0074  CD9100            CALL  OPCH
0077                    ;
0077                    ; Transfer message contents
0077                    ;
0077  7E          NCHO  MOV   A,M
0078  CD9100            CALL  OPCH    ; Output byte of data
007B  3E10              MVI   A,DLE   ; DLE for comparison
007D  EDA1              CPII  ;       ; Check if char was DLE and count
007F  CC9100            CZ    OPCH    ; Output second DLE if it was
0082  EA7700            JPE   NCHO    ; Loop till done
0085  CD9100            CALL  OPCH    ; Output DLE from A
0088  3E03              MVI   A,ETX   ; Send End of text
008A  CD9100            CALL  OPCH    ;
008D  CD0000            CALL  INIT    ; Stop SYNC characters
0090  C9                RET   ;       ; Return to calling program
0091                    ;
0091  F5          OPCH  PUSH  PSW     ; Output Character
0092  DB13        WTX   IN    CTRL    ; Get USART status
0094  E601              ANI   TXRDY   ; Check if ready for character
0096  28FA              JRZ   WTX     ; Wait till it is
0098  F1                POP   PSW     ; Get character back and
0099  D312              OUT   DATA    ; output
009B  C9                RET
009C                    ;
009C                    END
```

3.4
INTERRUPTS

Each HSIO-4 port provides four sources of interrupt from the following signals:

Table 3-7

INTERRUPT SIGNALS

Signal	Function
Tx Empty	Transmitter has finished sending characters
Tx Ready	Transmitter is ready for next character
Rx Ready	Receiver has a character ready for computer
Sync Detect	Synchronization has been achieved (in synchronous mode only)

The first three interrupt sources are armed and disarmed under program control; the fourth interrupt source, Synchronous Detect, is armed by a mini-jump on the HSIO-4 board. The resultant interrupt from each port can be connected to any of the S-100 interrupt lines. As supplied, the interrupts for all HSIO-4 ports are connected to S-100 bus interrupt line VI2.

3.4.1
ARMING AND DISARMING MASKABLE INTERRUPTS

To arm maskable interrupts for a port, set the particular bit(s) in the mask that corresponds to the interrupt -- or combination of interrupts -- you wish to use. This is done by outputting the value for the required bit pattern to the Interrupt Mask Set I/O address for the relevant port.

As shown in Table 3-9, there are a total of eight possible bit patterns. When output to the Interrupt Mask Set address for a particular port, the bit pattern establishes the status (arm/disarm) of all three maskable interrupts for that port. Table 3-9 shows the values that correspond to each of the eight possible bit patterns. Table 3-1 shows the Interrupt Mask Set addresses for each port.

For example: Assuming that the HSIO-4 board is addressed starting at 10H, to arm the Tx Ready interrupt (and disarm the other two interrupts) for Port C, you would output 02H to I/O address 19H (the Interrupt Mask Set I/O Address for Port C).

Similarly, to disarm all three maskable interrupts for port C, you would output 00H to I/O address 19H.

Table 3-8

INTERRUPT MASK BIT ALLOCATION

Value	Bit	Signal	Function
1	0	Tx Empty	Transmitter has finished sending characters
2	1	Tx Ready	Buffer is ready to receive the next character
4	2	Rx Ready	Receiver has a character to be read by the computer

Bit 0 is the least significant bit.

All interrupt mask bits are disarmed when the computer is powered up or reset.

Table 3-9

1 = Arm
0 = Disarm

EIGHT POSSIBLE BIT PATTERNS

Value	Rx Ready	Tx Ready	Tx Empty
00H	0	0	0
01H	0	0	1
02H	0	1	0
03H	0	1	1
04H	1	0	0
05H	1	0	1
06H	1	1	0
07H	1	1	1

3.4.2
ARMING THE SYNCHRONOUS DETECT INTERRUPT

As supplied, the Synchronous Detect interrupt for each HSIO-4 port is disarmed. The Synchronous Detect interrupt can be armed by a mini-jump.

To arm a Synchronous Detect interrupt:

1. Locate the 3-pin jumper area to the right of the USART for the relevant port, and install a mini-jump between the center pin and pin 2.

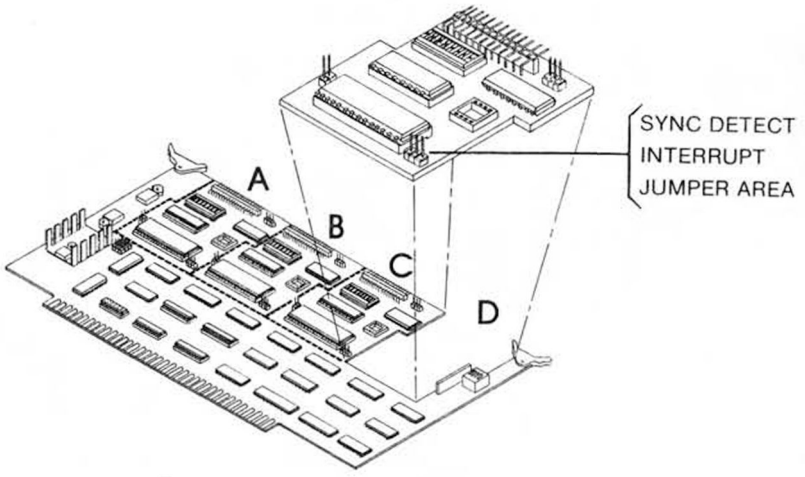


Figure 3-24

3.4.3
CONNECTING THE INTERRUPT LINES

The HSIO-4 board provides access to ten S-100 bus interrupt lines. These can cause eight vectored interrupts (VI0 - VI7), a non-vectored interrupt (PINT), and a non-maskable interrupt (NMI).

Each port provides four sources of interrupt. The four interrupt sources from each port are combined into a single interrupt source for each port. These combined interrupt sources -- one for each port -- are brought to the Interrupt Header. As shown in Figure 3-25, the interrupt sources for ports A, B, C, and D are made available on pins 1, 2, 3, and 4, respectively.

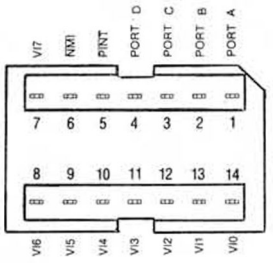


Figure 3-25

Any one or more of these combined interrupt sources can be connected to an S-100 interrupt line at the Interrupt Header. The specific connections to be made are determined by the particular software in use; consult the manual for the software you intend to use. If you are writing your own software, it is recommended that you utilize the standard Interrupt Header wiring with which the HSIO-4 is supplied. This wiring is illustrated in Figure 3-27.

To connect the interrupt sources to the S-100 interrupt lines:

1. Take a 14-pin header and wire the interrupt source for the desired port(s) to the pin carrying the required interrupt line.

2. Install this header in the Interrupt Header Socket (Location 2F).

For example, to cause all sources of interrupt from Port A to generate Vectored Interrupt 2 (VI2), you would wire pin 1 to pin 12 on a 14-pin header, as shown below:

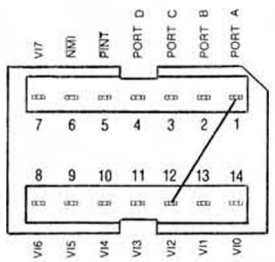


Figure 3-26

As supplied, the interrupts for each HSIO-4 port are all wired to S-100 bus interrupt line VI2, as shown in Figure 3-27.

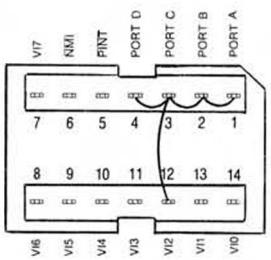
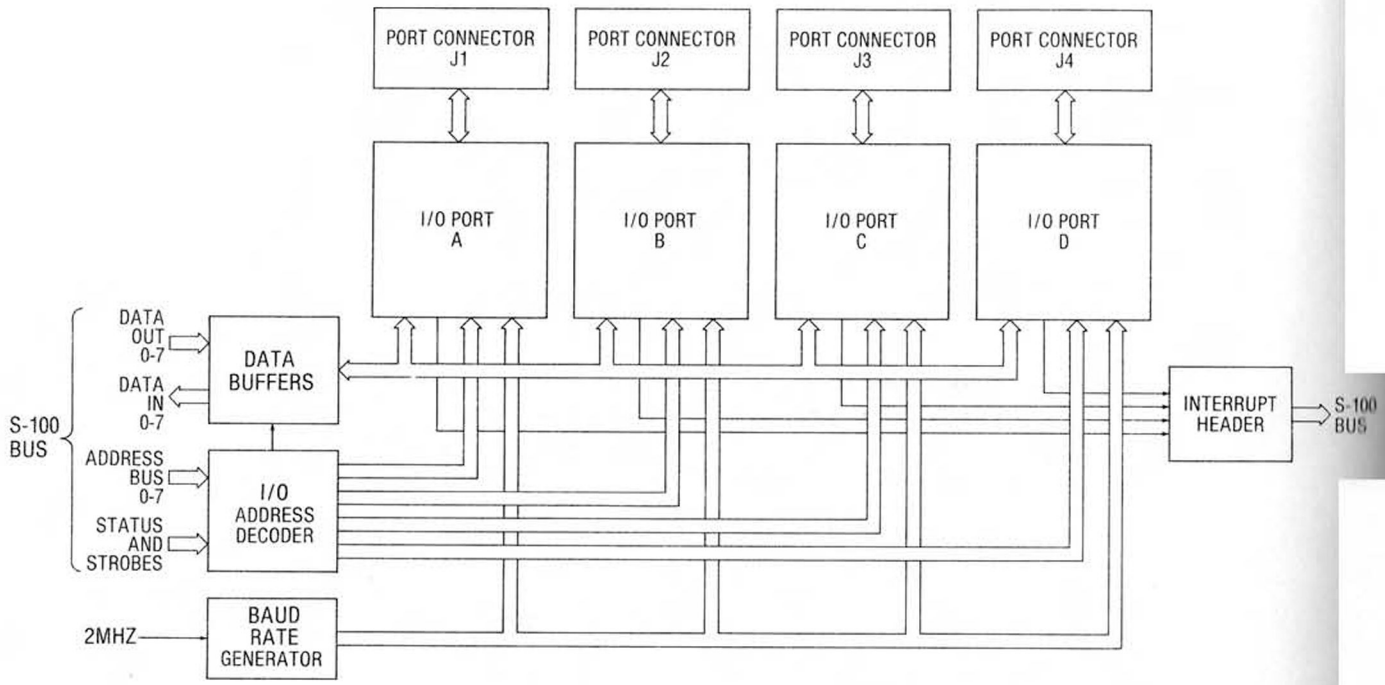


Figure 3-27

The HSIO-4 board consists of four independent serial I/O ports, and some common control logic. The major common functional blocks are the I/O Address Decoder, the Data Buffers, the Baud Rate Generator, and the Interrupt Header. (See Figure 4-1.)



FUNCTIONAL BLOCK DIAGRAM OF THE HSIO-4

Figure 4-1

4.1
ADDRESS DECODER

The Address Decoder converts I/O address and timing signals from the processor into a set of signals for each of the four I/O ports.

The HSIO-4 board is selected by address bits A4 - A7. These address bits are compared with the address switches. If a match occurs, the board is selected. Address bits A0 - A3 select functions within the board. Address bits A2 and A3 select one of the four I/O ports as follows:

A3	A2	
0	0	Selects Port A
0	1	Selects Port B
1	0	Selects Port C
1	1	Selects Port D

Address bits A0 and A1 select the operation to be performed by the selected port as follows:

A0	A1	
0	0	Write to Baud Rate Select Register
0	1	Write to Interrupt Mask Register
1	0	Write or read USART data
1	1	Write or read USART control or status

4.2
DATA BUFFERS

The Data Buffers interface the S-100 bus Data Out (DO) and Data In (DI) lines to a single bidirectional bus on the HSIO-4. During S-100 bus writes, the Data Buffers gate data from the S-100 DO bus onto the HSIO-4 internal bus. During USART reads, the buffers gate the read data onto the S-100 DI bus.

4.3
BAUD RATE GENERATOR

Each of the four ports can be programmed to select one of eight baud rates. These eight baud rates are derived from eleven frequencies produced by the Baud Rate Generator as shown in Table 4-1:

Table 4-1

1760	Hz or	(110 x 16)
4800	Hz or	(300 x 16)
9600	Hz or	(600 x 16)
76800	Hz or	(4800 x 16)
153600	Hz or	(9600 x 16)
307200	Hz or	(19200 x 16)
610	Hz or	(75 x 16)
1220	Hz or	(150 x 16)
19200	Hz or	(1200 x 16)
38400	Hz or	(2400 x 16)
55800	Hz	

The first six frequencies go directly to the I/O ports. Of the next four frequencies, only two go to the I/O ports. The two frequencies that are allowed to go to the I/O ports for program selection are selected by mini-jumps at JP13. The eleventh frequency (55800 Hz) can be selected by a mini-jump at JP13. When selected, this frequency goes directly to all four Clock Headers.

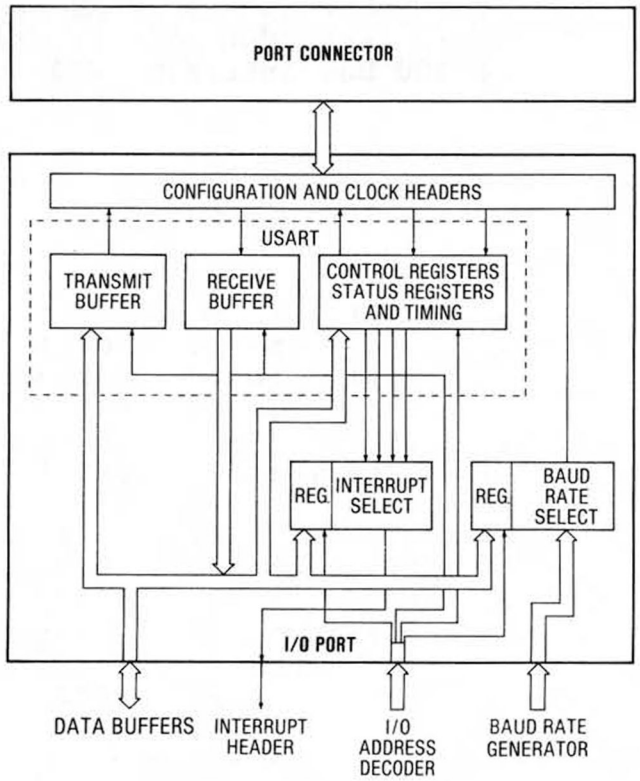
4.4
INTERRUPT HEADER

The USART in each I/O port generates four flags which, under software and hardware control, can generate an interrupt signal. The selected flags are ORed together within each I/O port. These four signals, one from each I/O port, are brought to the Interrupt Header.

The Interrupt Header can be wired to connect the interrupt signals to any of the S-100 bus interrupt lines. More than one I/O port interrupt signal may be connected to one S-100 bus interrupt line. As supplied, all four port interrupt signals are connected to the S-100 bus interrupt VI2.

4.5
I/O PORT

Each I/O port contains a USART, Configuration and Clock Headers, a Baud Rate Select Register, and an Interrupt Select Register. (See Figure 4-2.)



FUNCTIONAL BLOCK DIAGRAM OF AN INDIVIDUAL I/O PORT

Figure 4-2

4.5.1
USART

USART is an acronym for Universal Synchronous/Asynchronous Receiver/Transmitter. The USART can be programmed from the processor to communicate in most commonly used types of serial data transmission. The USART receives serial data streams and converts them into parallel data bytes for input to the processor. While receiving serial data, the USART will also accept data bytes from the processor in parallel format, convert them to serial format and transmit them. By performing conversion and formatting services automatically, the USART appears to the processor as a simple device for input or output of byte-oriented parallel data. A control register within the USART determines parameters. A complete USART status byte, including data format, error and status signals such as Transmitter Empty, Transmitter Ready, and Synchronous Detect, is available to the processor at any time. For the convenience of users requiring more detailed information, a manufacturer data sheet for the USART is reproducd in Appendix B, along with notes on the Interface to the USART.

4.5.2
CONFIGURATION AND CLOCK HEADERS

Various connections on the Configuration Header and the Clock Header for a port permit several types of serial transmission. The header configurations determine how the USART connects to the port connector and the baud rate signals.

4.5.3
BAUD RATE SELECT REGISTER

Each I/O port has a three-bit Baud Rate Select Register, which selects the baud rate. Physically, each bit is in a different IC (Locations 5F, 5E, and 5D). Logically, the Baud Rate Select Register is a single register loaded from the three least significant data bits of an output instruction from the processor. This register selects one of the eight frequencies to be sent to the Configuration Header.

To use the 55800 Hz signal, the Configuraton and Clock Headers can be wired to use the 55800 Hz signal ONLY and not the selectable rates.

4.5.4
INTERRUPT SELECT REGISTER

Each port has a three-bit Interrupt Select Register. Physically, each bit is in a different IC (Locations 5F, 5E, 5D). Logically, the Interrupt Select Register is a single register loaded from the three least significant data bits of an output instruction from the processor. Each bit in the register controls the arming/disarming of one flag from the USART. Only if a flag is armed is it ORed into the interrupt signal for the I/O port.

Bit 0 arms the TRANSMIT EMPTY SIGNAL
Bit 1 arms the TRANSMIT READY SIGNAL
Bit 2 arms the RECEIVE READY SIGNAL

The Synchronous Detect Flag, which is armed by a mini-jump, is only used for synchronous transmission.

If you lose communication with any peripheral device connected to the HSIO-4 board, you must first determine whether the cause of the failure is in a specific peripheral device and its cabling to the computer or in the HSIO-4 board itself.

Use the following procedure to trace the cause of a communications failure:

1. If peripheral devices are connected to more than one I/O port and communication to all the connected peripherals has failed, this indicates that some part of the HSIO-4 not specific to a particular I/O port may be at fault. In this case, follow the procedure outlined in Section 5.3.

2. If you lose communication with only one peripheral, you should check that the configuration of the peripheral and the cable connection to the computer have not been altered. Make sure the connectors at both ends of the cable are firmly seated. Retry establishing communication with the peripheral.

3. If communication with the peripheral is still lost, disconnect the cable from the malfunctioning port, re-connect it to one of the other I/O ports that is functioning properly, and retry establishing communication through the new I/O port.

4. If communication is still lost, this indicates that the problem is with the peripheral device itself or the connecting cable.

5. If communication is successfully established, this indicates that some part of the HSIO-4 board specific to the I/O port first tested is at fault. Follow the procedures in Sections 5.1 and 5.2.

5.1
CREATING A TEST DISKETTE

Before testing an individual I/O port, you must create a test diskette. The procedure for creating such a test diskette is described in Sections 5.1.1 - 5.1.3 below. This procedure utilizes a feature, supported by DOS and BASIC, known as "auto-start," which automatically loads BASIC and runs the HSIO-4 test. Once a test diskette has been set up, restarting the computer with the reset switch prompts DOS to load BASIC, which, in turn, starts the HSIO-4 test.

5.1.1
CREATING AN AUTO-START DOS DISKETTE

To create an auto-start DOS diskette:

1. Make a copy of the North Star DOS and BASIC diskette (version 5.2 and up).

2. Put the copy of the DOS and BASIC diskette in drive one and restart the computer with the reset switch.

3. Type: GO BASIC

4. Type the following commands:

```
10  OPEN #1 %0,"DOS"
20  WRITE #1 %48,&0,NOENDMARK
30  CLOSE #1

RUN
```

5. Restart the computer with the reset switch. The word READY should appear on the screen, indicating that auto-start of BASIC has been successfully established.

5.1.2
CREATING THE TEST PROGRAM

1. Type in the BASIC program that is listed on the next page. In order to simplify the program entry, delete the REMark statements (which have been added only to explain the test process).

2. List the program to verify that it has been correctly entered into BASIC.

3. Save this test program on the diskette by typing: NSAVE HSIOTEST

```
11 REM PROGRAM NAME=   HSIOTEST     8-April-81
12 !TAB(10),"North Star HSIO-4 Test" \!
13                 REM Set up port addresses for first HSIO-4 port--
14                                  REM Edit B for non-standard address!
15 B=16                            \REM Base address of HSIO-4 =10 hex.
16 D=B+2                           \REM USART data port address.
17 C=D+1                           \REM USART command port address.
18                 REM set HSIO-4 baud rates--
19 FOR P=B TO B+12 STEP 4
20   OUT P,1                       \REM 1=9600 baud, 4=1200 baud.
21 NEXT P                          \REM P= baud rate select address.
22                 REM initialize the USARTs--
23 FOR A=1 TO 5                    \REM Send 5 bytes of...
24   READ N                        \REM initialization code to...
25   FOR P=C TO C+12 STEP 4        \REM each command port.
26     OUT P,N
27   NEXT P
28 NEXT A
29                 REM Clear each HSIO-4 port--
30 FOR P=D TO D+12 STEP 4
31   X=INP(P)                      \REM clear by reading each data port.
32 NEXT P
33 !\!"(note: connect pin 2 to pin 3 on the port to be tested!)"\!\!
34 ! CHR$(13),"Test port  [A,B,C,D]: ", \T$=INCHAR$(0) \!\!
35 IF T$>="A" THEN IF T$<="D" THEN 36 \GOTO 34
36 !"Press <ESC>  to abort current test..."  \!\! \!"Testing port: ",T$
37 P=4*(ASC(T$)-64)-2+B            \REM Convert ASCII 'A','B','C','D'...
38                                  REM into actual data port address...
39                                  REM  (B+2, B+6, B+10, B+14).
40   Z=0                           \REM Initialize error counter.
41   FOR T=0 TO 255                \REM Test all possible bytes.
42     PRINT ".",                  \REM Show some life!
43     X=INP(P+1) \IF INT(X*.5)*2 = X THEN 43 \REM Wait until port ready.
44     OUT P,T                     \REM Send it a character.
45     FOR I=1 TO 50 \X=INT( INP(P+1) *.5)
46       IF INT(X*.5)*2<>X THEN EXIT 50  \REM Exit if character received.
47     NEXT
48       !CHR$(7),"Input status not ready.  Not receiving data!"
49       GOTO 53
50     R=INP(P)                    \REM Get transmitted character.
51     IF T=R THEN 54              \REM If transmitted= received, then AOK!
52       !\! CHR$(7),"Port ",T$,": transmitted",T," received as",R,
53       Z=Z+1
54     I=INP(2) \I=I-128*(I>127) \IF I=27 THEN EXIT 56 \REM Abort test?
55   NEXT T
56 !\! \!"There were",Z," errors encountered." \GOTO 33
57 DATA 3,3                        \REM Ensure acceptance of...
58 DATA 64                         \REM Internal Reset.
59 DATA 206                        \REM Mode- 2 stop, no parity,
60                                 \REM       8 data, 16*clock.
61 DATA 39                         \REM Cmd- RTS and DTR low,
62                                  REM enable Receive and Transmit.
63 END
```

5.1.3
CONFIGURING BASIC FOR AUTO-START OPERATION

1. With the diskette still in drive one, restart the computer with the reset switch. The word READY should appear on the screen.

2. Type the following commands:

```
10  CHAIN "HSIOTEST"

BYE

+SF BASIC E00 (Standard Origin of 5.2 DOS and BASIC

+JP E00

10 OPEN #1 %1, "BASIC"
20 WRITE #1 %15,&0,NOENDMARK
30 CLOSE #1

RUN
```

3. Restart the computer with the reset switch. The HSIO-4 test sign-on prompt ("North Star HSIO-4 Test") should appear on the screen. If this sign-on prompt does not display, check to make sure that the procedures in Section 5.1. have been properly executed.

5.2
SOFTWARE TESTING

To test an individual I/O port, use the test diskette created in Section 5.1 as follows:

1. Insert the test diskette in drive one and restart the computer with the reset switch. The HSIO-4 sign-on message ("North Star HSIO-4 Test") should appear on the screen, followed by this prompt:

Test port [A, B, C, D]:

2. Before selecting the port to be tested, you must connect pin 2 and pin 3 at the connector for the port to be tested. A test fixture to do this may be constructed by using a DP25-P connector with pin 2 and pin 3 soldered together. (See Figure 5-1.)

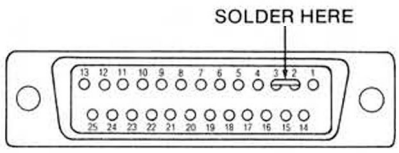


Figure 5-1

3. As prompted by the screen display, select the port to be tested (A, B, C, or D).

4. The test program will respond with:

Press <ESC> to abort current test ...

Testing port: [A, B, C, D]

4. If the port is functioning properly, a dot displays on the screen for each of the 256 test characters sent to the port. At successful conclusion, the following displays:

There were 0 errors encountered

5. If there is a malfunction, one of two error messages will display:

A.
Input status not ready. Not receiving data!

Using an oscilloscope and referring to the schematic, this test may be repeated to trace the signal path from pin 19 (TXD) of the USART (8251)--to the RS-232 driver (1448)--to the test connector and back to the receiver (1489)--to pin 3 (RXD) of the USART. Check to make sure that the test fixture is installed properly.

B.
Port [A,B,C,D] transmitted [value] received as [value]

If the value received differs from the value transmitted, the suspected components are 6F, 7F or the USART.

5.3
OSCILLOSCOPE TESTING

If all four I/O ports fail to communicate with peripheral devices or fail to pass the HSIO-4 test, this indicates that the problem is probably in the power circuitry or the Baud Rate Generator circuitry. The following procedure for checking this circuitry requires the use of an oscilloscope with a bandwidth of at least 10 Megahertz.

> **WARNING**
>
> The screw and nut on the top regulator in the upper left corner (on the component side of the board) are NOT at ground potential and should not be used for oscilloscope or meter grounds, as this may damage the HSIO-4 board or the HORIZON motherboard.

1. Turn off the computer.

2. Place the HSIO-4 board on an extender board, and turn on the computer. Check for the following DC voltages:

   +5 volts at pin 14 of the 1489 at 8A

   +12 volts at pin 14 of the 1488 at 7B

   -12 volts at pin 1 of the 1488 at 7B

3. Trace the source of the clock pulses from pin 49 of the motherboard to the various signals used to provide baud rates. If the signals are not as listed in Table 5-1, refer to the schematic drawings and trace backward to locate the problem.

All wave forms described in Table 5-1 repeat every period, T3, (see Figure 5-2) -- except the wave form for 19200 baud, which repeats every two periods (see Figure 5-3).

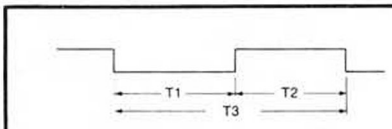


Figure 5-2

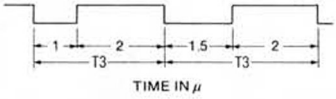


Figure 5-3

Table 5-1

WAVE FORMS

Signal	Location	T1	T2	T3
2 Mhz	8F pin 2	250ns	250ns	500ns
19200 baud	3E pin 4	See Figure 5-3 (above)		
9600 baud	3E pin 3	3.3us	3.2us	6.5us
4800 baud	3E pin 2	6.5us	6.5us	13us
2400 baud	3E pin 1	13us	13us	26us
1200 baud	3E pin 15	27us	26us	53us
600 baud	3E pin 14	53us	52us	105us
300 baud	3E pin 13	105us	105us	210us
110 baud	3E pin 12	210us	365us	575us

5.4
REPAIR PROCEDURES

If you identify malfunctions or defective components that you are unable to rectify or replace yourself, you can return a defective HSIO-4 board to any North Star Authorized Service Center for repair.

APPENDIX A

RS-232 CONNECTOR PIN ASSIGNMENTS

PIN #	SIGNAL NAME	DIRECTION T = TERMINAL M = MODEM	ABBREVIATION
1	Chassis Ground	Not Connected	AA
2	Transmit Data	T to M	BA
3	Receive Data	M to T	BB
4	Request to Send	T to M	CA
5	Clear to Send	M to T	CB
6	Data Set Ready	M to T	CC
7	Signal Ground		AB
8	Carrier Detect	M to T	CF
9	+12 Volts	Used in Current Loop	
10	-12 Volts	Used in Current Loop	
11	No Connection		
12	No Connection		
13	No Connection		
14	No Connection		
15	No Connection		
16	No Connection		
17	Receive Clock	M to T	DD
18	No Connection		
19	Secondary RTS	Optional Use	SCA
20	Data Terminal Ready	T to M	CD
21	No Connection		
22	No Connection		
23	No Connection		
24	Transmit Clock	T to M	DA
25	No Connection		

APPENDIX B

NOTES ON THE INTERFACE TO THE USART

CONNECTIONS TO THE USART

The USART data sheet describes all possible USART connections. Not all of these connections, however, are supported by the HSIO-4. Specifically, the SYNDET pin can be used only as an output with the HSIO-4. It cannot be used as an input to initiate HUNT mode. When HUNT mode is required, it must be set by the appropriate command instruction.

The RESET signal for the USART is derived from the S-100 POC/ line, and is asserted when the computer is powered up or when the computer is restarted with the reset switch. If a RESET is required at any other time, the USART can be internally reset with the appropriate software instructions. (Refer to the USART data sheet for specific command instructions.)

SETTING UP THE USART

It is necessary to pause between resetting a USART and issuing the set up commands. This is is a problem only if ports are reset and set up one at a time. This problem can be avoided, and the necessary pause provided, by resetting the USART for ports 1, 2, 3, and 4 in sequence and then setting up each USART in similar sequence. To achieve the required delay when setting up only one port, execute two XTHL instructions after issuing the RESET command.

PROGRAMMING THE USART

The following sequence of command words will accomplish a soft reset and ready an individual USART for a mode instruction.

```
Output  3 to USART Status Control address
Output  3 to USART Status Control address
Output 64 to USART Status Control address
```

After programming the USART with mode and command words, the USART data should be read to remove any extraneous character.

PROGRAMMABLE COMMUNICATION INTERFACE

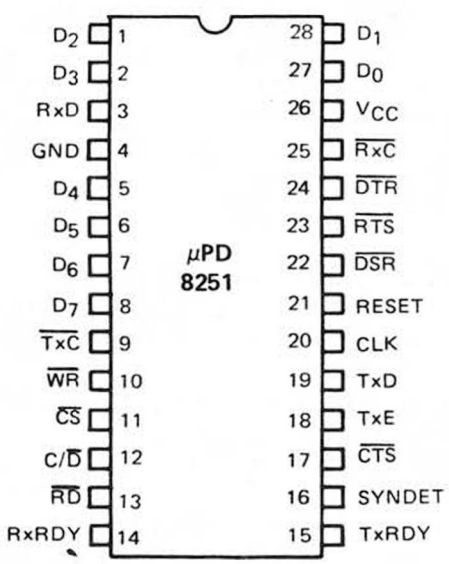
DESCRIPTION

The μPD8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is designed for microcomputer systems data communications. The USART is used as a peripheral and is programmed by the μPD8080 or other processor to communicate in commonly used serial data transmission techniques including IBM Bi-Sync. The USART receives serial data streams and converts them into parallel data characters for the processor. While receiving serial data, the USART will also accept data characters from the processor in parallel format, convert them to serial format and transmit. The USART will signal the processor when it has completely received or transmitted a character and requires service. Complete USART STATUS including data format errors and control signals such as TxE and SYNDET is available to the processor at any time.

FEATURES

- Asynchronous or Synchronous Operation
  - Asynchronous:
    - 5-8 Bit Characters
    - Clock Rate – 1, 16 or 64 x Baud Rate
    - Break Character Generation
    - Select 1, 1-1/2, or 2 Stop Bits
    - False Start Bit Detector
  - Synchronous:
    - 5-8 Bit Characters
    - Internal or External Character Synchronization
    - Automatic Sync Insertion
    - Single or Double Sync Characters
- Baud Rate – Synchronous – DC to 56K Baud
  – Asynchronous – DC to 9.6K Baud
- Full Duplex, Double Buffered Transmitter and Receiver
- Parity, Overrun and Framing Flags
- Fully Compatible with 8080
- All Inputs and Outputs are TTL Compatible
- Single +5 Volt Supply
- Separate Device, Receive and Transmit TTL Clocks
- 28 Pin Plastic DIP Package
- N-Channel MOS Technology

PIN CONFIGURATION



PIN NAMES

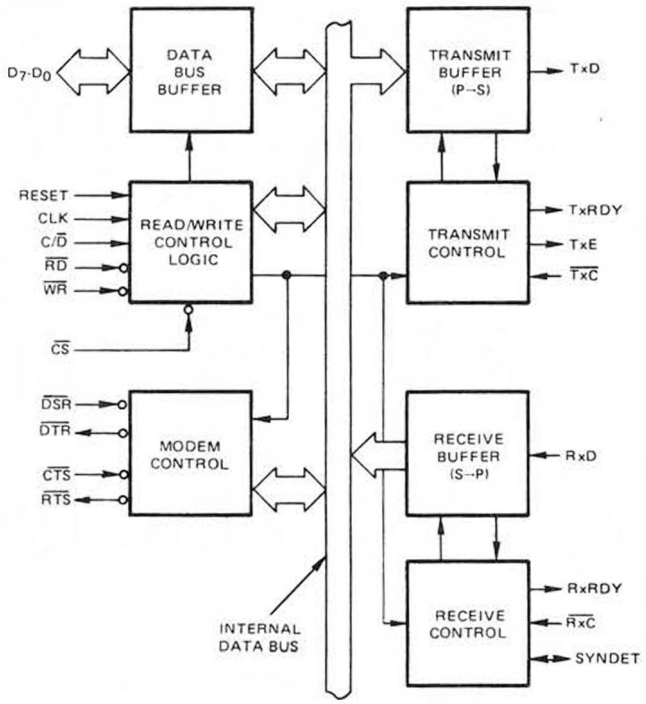
D_7-D_0	Data Bus (8 bits)
C/D	Control or Data is to be Written or Read
RD	Read Data Command
WR	Write Data or Control Command
CS	Chip Enable
CLK	Clock Pulse (TTL)
RESET	Reset
TxC	Transmitter Clock (TTL)
TxD	Transmitter Data
RxC	Receiver Clock (TTL)
RxD	Receiver Data
RxRDY	Receiver Ready (has character for 8080)
TxRDY	Transmitter Ready (ready for char. from 8080)
DSR	Data Set Ready
DTR	Data Terminal Ready
SYNDET	Sync Detect
RTS	Request to Send Data
CTS	Clear to Send Data
TxE	Transmitter Empty
V_{CC}	+5 Volt Supply
GND	Ground

FUNCTIONAL DESCRIPTION

The μPD8251 Universal Synchronous/Asynchronous Receiver/Transmitter is designed specifically for 8080 microcomputer systems but works with most 8-bit processors. Operation of the 8251, like other I/O devices in the 8080 family, is programmed by system software for maximum flexibility.

In the receive mode, a communication interface device must convert incoming serial format data into parallel data and make certain format checks on the data. And in the transmit mode, the device must format data into serial data. The device must also supply or remove characters or bits that are unique to the communication format in use. By performing conversion and formatting services automatically, the USART appears to the processor as a simple or "transparent" input or output of byte-oriented parallel data.

BLOCK DIAGRAM



BASIC OPERATION

C/D	RD	WR	CS	
0	0	1	0	8251 → Data Bus
0	1	0	0	Data Bus → 8251
1	0	1	0	Status → Data Bus
1	1	0	0	Data Bus → Control
X	X	X	1	Data Bus → 3-State
X	1	1	0	

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature	-0°C to +70°C
Storage Temperature	-65°C to +125°C
All Output Voltages	-0.5 to +7 Volts
All Input Voltages	-0.5 to +7 Volts
Supply Voltages	-0.5 to +7 Volts

COMMENT: Stress above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

*$T_a = 25°C$

DC CHARACTERISTICS

T_a = 0°C to 70°C; V_{CC} = 5.0V ± 5%; GND = 0V

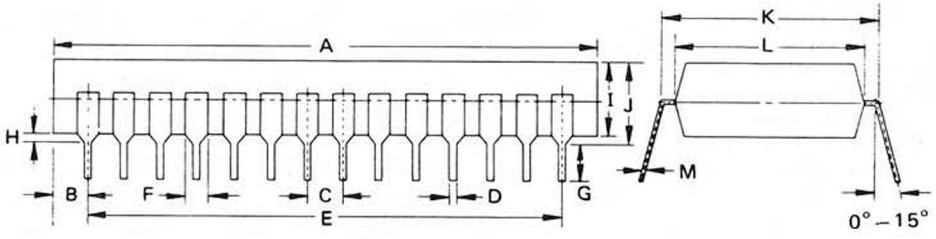
PARAMETER	SYMBOL	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT	TEST CONDITIONS
Input Low Voltage	V_{IL}	GND −.5		0.8	V	
Input High Voltage	V_{IH}	2.0		V_{CC}	V	
Output Low Voltage	V_{OL}			0.45	V	I_{OL} = 1.7 mA
Output High Voltage	V_{OH}	2.4			V	I_{OH} = −100 μA
Data Bus Leakage	I_{DL}			−50	μA	V_{OUT} = 0.45V
				10		V_{OUT} = V_{CC}
Input Load Current	I_{IL}			10	μA	@5.5V
Power Supply Current	I_{CC}		45	80	mA	

CAPACITANCE

T_a = 25°C; V_{CC} = GND = 0V

PARAMETER	SYMBOL	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT	TEST CONDITIONS
Input Capacitance	C_{IN}			10	pF	fc = 1 MHz
I/O Capacitance	$C_{I/O}$			20	pF	Unmeasured pins returned to GND

PACKAGE OUTLINE μPD8251C



ITEM	MILLIMETERS	INCHES
A	38.0 MAX.	1.496 MAX.
B	2.49	0.098
C	2.54	0.10
D	0.5 ± 0.1	0.02 ± 0.004
E	33.02	1.3
F	1.5	0.059
G	2.54 MIN.	0.10 MIN.
H	0.5 MIN.	0.02 MIN.
I	5.22 MAX.	0.205 MAX.
J	5.72 MAX.	0.225 MAX.
K	15.24	0.6
L	13.2	0.52
M	$0.25^{+0.10}_{0.05}$	$0.01^{+0.004}_{0.002}$

BUS PARAMETERS: ①

T_a = 0°C to 70°C; V_{CC} = 5.0V ± 5%; GND = 0V

PARAMETER	SYMBOL	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT	TEST CONDITIONS
READ						
Address Stable before $\overline{READ}$, ($\overline{CS}$, C $\overline{D}$)	t_{AR}	50			ns	
Address Hold Time for $\overline{READ}$, ($\overline{CS}$, $\overline{CD}$)	t_{RA}	5			ns	
$\overline{READ}$ Pulse Width	t_{RR}	430			ns	
Data Delay from $\overline{READ}$	t_{RD}			350	ns	C_L = 100 pF
$\overline{READ}$ to Data Floating	t_{DF}	25		200	ns	C_L = 100 pF C_L = 15 pF
Recovery Time Between WRITES ②	t_{RV}	6			t_{CY}	
WRITE						
Address Stable before $\overline{WRITE}$	t_{AW}	20			ns	
Address Hold Time for $\overline{WRITE}$	t_{WA}	20			ns	
$\overline{WRITE}$ Pulse Width	t_{WW}	400			ns	
Data Set-Up Time for $\overline{WRITE}$	t_{DW}	200			ns	
Data Hold Time for $\overline{WRITE}$	t_{WD}	40			ns	
OTHER TIMING						
Clock Period ③	t_{CY}	.420		1.35	µs	
Clock Pulse Width	t_W	220		0.7t_{CY}	ns	
Clock Rise and Fall Time	$t_{R,tF}$	0		50	ns	
TxD Delay from Falling Edge of $\overline{TxC}$	t_{DTx}			1	µs	C_L = 100 pF
Rx Data Set-Up Time to Sampling Pulse	t_{SRx}	2			µs	C_L = 100 pF
Rx Data Hold Time to Sampling Pulse	t_{HRx}	2			µs	C_L = 100 pF
Transmitter Input Clock Frequency 1X Baud Rate 16X and 64X Baud Rate	f_{Tx}	 DC DC		 56 520	 KHz KHz	
Transmitter Input Clock Pulse Width 1X Baud Rate 16X and 64X Baud Rate	t_{TPW}	 12 1			 t_{CY} t_{CY}	
Transmitter Input Clock Pulse Delay 1X Baud Rate 16X and 64X Baud Rate	t_{TPD}	 15 3			 t_{CY} t_{CY}	
Receiver Input Clock Frequency 1X Baud Rate 16X and 64X Baud Rate	f_{Rx}	 DC DC		 56 520	 KHz KHz	
Receiver Input Clock Pulse Width 1X Baud Rate 16X and 64X Baud Rate	t_{RPW}	 12 1			 t_{CY} t_{CY}	
Receiver Input Clock Pulse Delay 1X Baud Rate 16X and 64X Baud Rate	t_{RPD}	 15 3			 t_{CY} t_{CY}	
TxRDY Delay from Center of Data Bit	t_{Tx}			16	t_{CY}	C_L = 50 pF
RxRDY Delay from Center of Data Bit	t_{Rx}			20	t_{CY}	
Internal Syndet Delay from Center of Data Bit	t_{IS}			25	t_{CY}	
External Syndet Set-Up Time before Falling Edge of $\overline{RxC}$	t_{ES}	16		16	t_{CY}	
TxEMPTY Delay from Center of Data Bit	t_{TxE}			16	t_{CY}	C_L = 50 pF
Control Delay from Rising Edge of WRITE (TxE, $\overline{DTR}$, $\overline{RTS}$)	t_{WC}				t_{CY}	
Control to READ Set-Up Time ($\overline{DSR}$, $\overline{CTS}$)	t_{CR}	16			t_{CY}	

Notes: ① AC timings measured at V_{OH} = 2.0, V_{OL} = 0.8, and with load circuit of Figure 1.

② This recovery time is for initialization only, when MODE, SYNC1, SYNC2, COMMAND and first DATA BYTES are written into the USART. Subsequent writing of both COMMAND and DATA are only allowed when TxRDY = 1.

③ The TxC and RxC frequencies have the following limitations with respect to CLK.
For 1X Baud Rate, f_{Tx} or $f_{Rx} \leq 1/(30\ t_{CY})$
For 16X and 64X Baud Rate, f_{Tx} or $f_{Rx} \leq 1/(4.5\ t_{CY})$

④ Reset Pulse Width = 6 t_{CY} minimum.

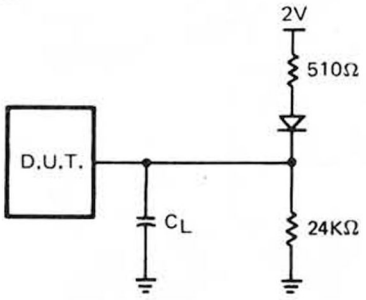
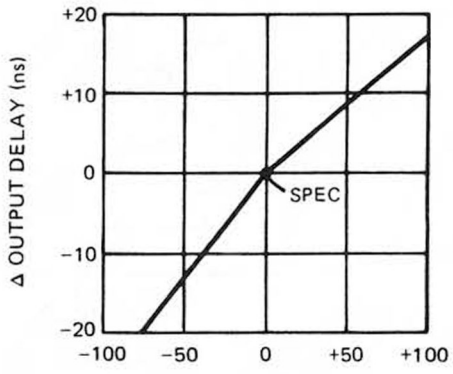


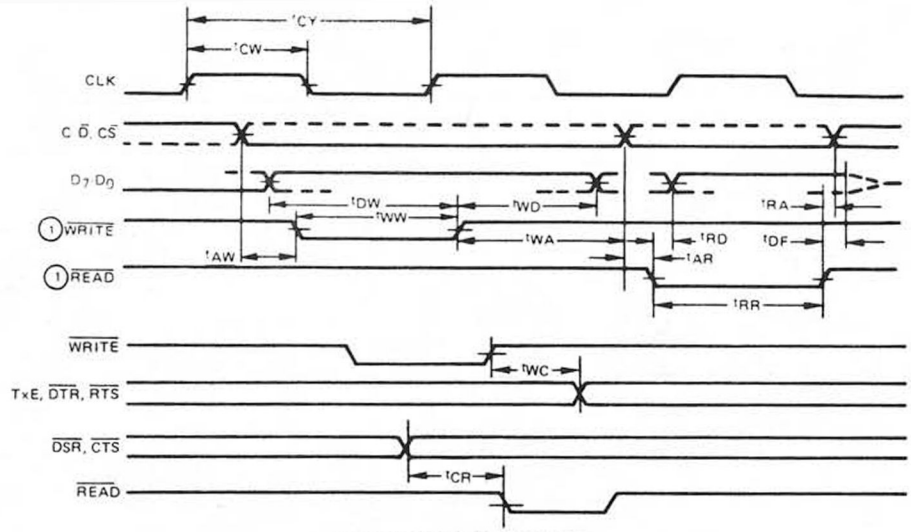
Figure 1.

TEST LOAD CIRCUIT

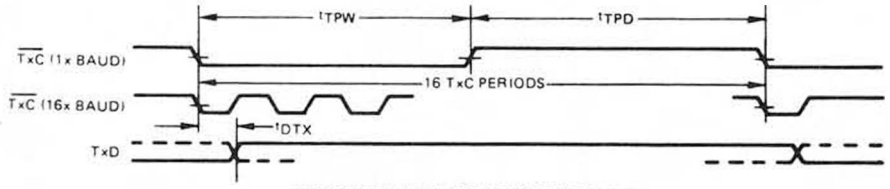


Typical Δ Output
Delay Versus Δ Capacitance (pF)

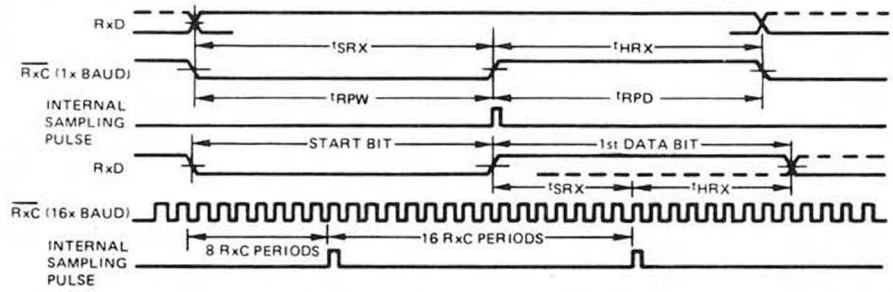
TIMING WAVEFORMS



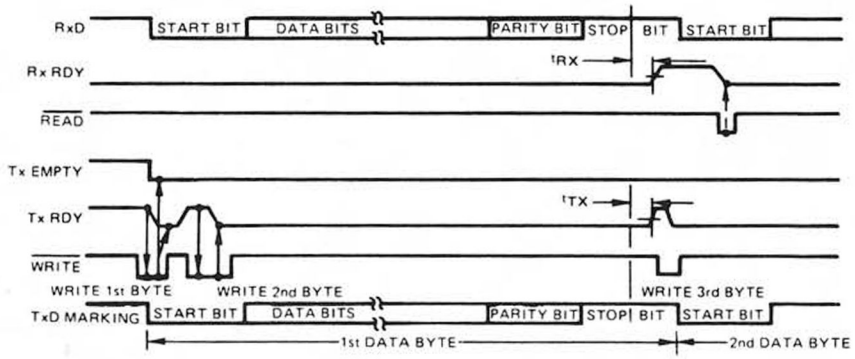
READ AND WRITE TIMING



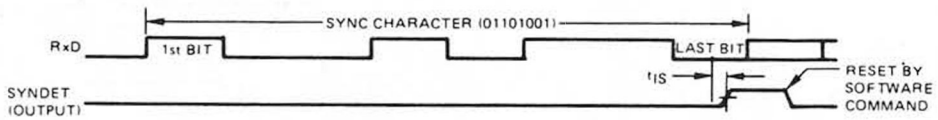
TRASNMITTER CLOCK AND DATA



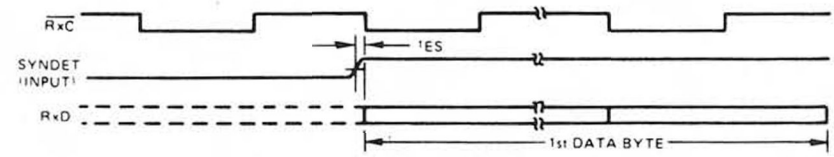
RECEIVER CLOCK AND DATA



TxRDY AND RxRDY TIMING (ASYNC MODE)



INTERNAL SYNC DETECT



EXTERNAL SYNC DETECT

Note: (1) Write and Read pulses have no timing limitation with respect to CLK.

PIN			FUNCTION
NO.	SYMBOL	NAME	
1, 2, 27, 28 5 – 8	$D_7 - D_0$	Data Bus Buffer	An 8-bit, 3-state bi-directional buffer used to interface the 8251 to the processor data bus. Data is transmitted or received by the buffer in response to input/output or Read/Write instructions from the processor. The Data Bus Buffer also transfers Control words, Command words, and Status.
26	V_{CC}	V_{CC} Supply Voltage	+5 volt supply
4	GND	Ground	Ground
Read/Write Control Logic			This logic block accepts inputs from the processor Control Bus and generates control signals for overall USART operation. The Mode Instruction and Command Instruction registers that store the control formats for device functional definition are located in the Read/ Write Control Logic.
21	RESET	Reset	A "one" on this input forces the USART into the "Idle" mode where it will remain until reinitialized with a new set of control words. Minimum RESET pulse width is t_{CY}.
20	CLK	Clock Pulse	The CLK input provides for internal device timing and is usually connected to the Phase 2 (TTL) output of the μPB8224 Clock Generator. External inputs and outputs are not referenced to CLK, but the CLK frequency must be 30 times the Receiver or Transmitter clocks in the synchronous mode and 4.5 times for the asynchronous mode.
10	$\overline{WR}$	Write Data	A "zero" on this input instructs the μPD8251 to accept the data or control word which the processor is writing out to the USART via the data bus.
13	$\overline{RD}$	Read Data	A "zero" on this input instructs the μPD8251 to place the data or status information onto the Data Bus for the processor to read.
12	C/$\overline{D}$	Control/Data	The Control/Data input, in conjunction with the $\overline{WR}$ and $\overline{RD}$ inputs, informs USART to accept or provide either a data character, control word or status information via the Data Bus. 0 = Data; 1 = Control.
11	$\overline{CS}$	Chip Select	A "zero" on this input enables the USART for reading and writing to the processor.
Modem Control			The μPD8251 has a set of control inputs and outputs which may be used to simplify the interface to a Modem.
22	$\overline{DSR}$	Data Set Ready	The Data Set Ready input can be tested by the processor via Status information. The $\overline{DSR}$ input is normally used to test Modem Data Set Ready condition.
24	$\overline{DTR}$	Data Terminal Ready	The Data Terminal Ready output can be controlled via the Command word. The $\overline{DTR}$ output is normally used to drive Modem Data Terminal Ready or Rate Select lines.
23	$\overline{RTS}$	Request to Send	The Request to Send output can be controlled via the Command word. The $\overline{RTS}$ output is normally used to drive the Modem Request to Send line.
17	$\overline{CTS}$	Clear to Send	A "zero" on the Clear to Send input enables the USART to transmit serial data if the TxEN bit in the Command Instruction register is enabled (one).

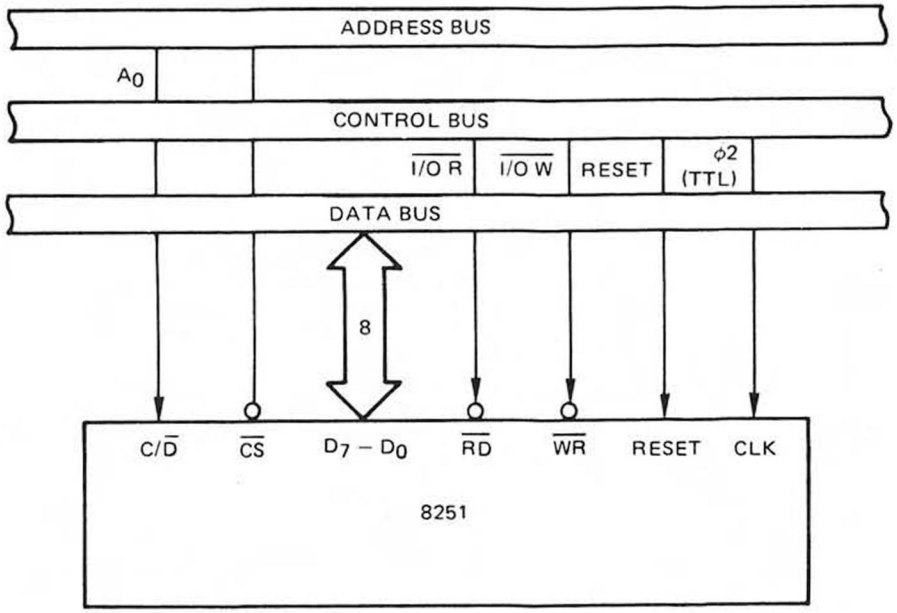
TRANSMIT BUFFER/ CONVERTER

The Transmit Buffer/Converter receives parallel data from the Data Bus Buffer via the internal data bus, converts parallel to serial data, inserts the necessary characters or bits needed for the programmed communication format and outputs composite serial data on the TxD output.

PIN IDENTIFICATION (CONT.)

PIN			FUNCTION
NO.	SYMBOL	NAME	
Transmit Control Logic			The Transmit Control Logic accepts and outputs all external and internal signals necessary for serial data transmission.
15	TxRDY	Transmitter Ready	Transmitter Ready signals the processor that the transmitter is ready to accept a data character. TxRDY can be used as an interrupt or may be tested through the Status information for Polled operation. Loading a character from the processor automatically resets TxRDY.
18	TxE	Transmitter Empty	The Transmitter Empty output signals the processor that the USART has no further characters to transmit. TxE is automatically reset upon receiving a data character from the processor. In half-duplex, TxE can be used to signal end of a transmission and request the processor to "turn the line around." The TxEn bit in the command instruction does not effect TxE. In the Synchronous mode, a "one" on this output indicates that a Sync character or characters are about to be automatically transmitted as "fillers" because the next data character has not been loaded.
9	$\overline{TxC}$	Transmitter Clock	The Transmitter Clock controls the serial character transmission rate. In the Asynchronous mode, the $\overline{TxC}$ frequency is a multiple of the actual Baud Rate. Two bits of the Mode Instruction select the multiple to be 1x, 16x, or 64x the Baud Rate. In the Synchronous mode, the $\overline{TxC}$ frequency is automatically selected to equal the actual Baud Rate. Note that for both Synchronous and Asynchronous modes, serial data is shifted out of the USART by the falling edge of $\overline{TxC}$.
19	TxD	Transmitter Data	The Transmit Control Logic outputs the composite serial data stream on this pin.

8251 INTERFACE TO 8080 STANDARD SYSTEM BUS



RECEIVER BUFFER

The Receiver Buffer accepts serial data input at the $\overline{RxD}$ pin and converts the data from serial to parallel format. Bits or characters required for the specific communication technique in use are checked and then an eight-bit "assembled" character is readied for the processor. For communication techniques which require less than eight bits, the μPD8251 sets the extra bits to "zero."

PIN IDENTIFICATION (CONT.)

PIN NO.	PIN SYMBOL	PIN NAME	FUNCTION
Receiver Control Logic			This block manages all activities related to incoming data.
14	RxRDY	Receiver Ready	The Receiver Ready output indicates that the Receiver Buffer is ready with an "assembled" character for input to the processor. For Polled operation, the processor can check RxRDY using a Status Read or RxRDY can be connected to the processor interrupt structure. Note that reading the character to the processor automatically resets RxRDY.
25	$\overline{RxC}$	Receiver Clock	The Receiver Clock is the rate at which the incoming character is received. In the Asynchronous mode, the $\overline{RxC}$ frequency may be 1,16 or 64 times the actual Baud Rate but in the Synchronous mode the $\overline{RxC}$ frequency must equal the Baud Rate. Two bits in the mode instruction select Asynchronous at 1x, 16x or 64x or Synchronous operation at 1x the Baud Rate. Unlike $\overline{TxC}$, data is sampled by the μPD8251 on the rising edge of $\overline{RxC}$. (1)
3	RxD	Receiver Data	A composite serial data stream is received by the Receiver Control Logic on this pin.
16	SYNDET	Sync Detect	The SYNC Detect pin is only used in the Synchronous mode. The μPD8251 may be programmed through the Mode Instruction to operate in either the internal or external Sync mode and SYNDET then functions as an output or input respectively. In the internal Sync mode, the SYNDET output will go to a "one" when the μPD8251 has located the SYNC character in the Receive mode. If double SYNC character (bi-sync) operation has been programmed, SYNDET will go to "one" in the middle of the last bit of the second SYNC character. SYNDET is automatically reset to "zero" upon a Status Read or RESET. In the external SYNC mode, a "zero" to "one" transition on the SYNDET input will cause the μPD8251 to start assembling data character on the next falling edge of $\overline{RxC}$. The length of the SYNDET input should be at least one $\overline{RxC}$ period, but may be removed once the μPD8251 is in SYNC.

Note: (1) Since the μPD8251 will frequently be handling both the reception and transmission for a given link, the Receive and Transmit Baud Rates will be same. $\overline{RxC}$ and $\overline{TxC}$ then require the same frequency and may be tied together and connected to a single clock source or Baud Rate Generator.

Examples:

If the Baud Rate equals 110 (Async):
- $\overline{RxC}$ or $\overline{TxC}$ equals 110 Hz (1x)
- $\overline{RxC}$ or $\overline{TxC}$ equals 1.76 KHz (16x)
- $\overline{RxC}$ or $\overline{TxC}$ equals 7.04 KHz (64x)

If the Baud Rate equals 300:
- $\overline{RxC}$ or $\overline{TxC}$ equals 300 Hz (1x) A or S
- $\overline{RxC}$ or $\overline{TxC}$ equals 4800 Hz (16x) A only
- $\overline{RxC}$ or $\overline{TxC}$ equals 19.2 KHz (64x) A only

OPERATIONAL DESCRIPTION

A set of control words must be sent to the μPD8251 to define the desired mode and communications format. The control words will specify the BAUD RATE FACTOR (1x, 16x, 64x), CHARACTER LENGTH (5 to 8), NUMBER OF STOP BITS (1, 1-1/2, 2), ASYNCHRONOUS or SYNCHRONOUS MODE, SYNDET (IN or OUT), PARITY, etc.

After receiving the control words, the μPD8251 is ready to communicate. TxRDY is raised to signal the processor that the USART is ready to receive a character for transmission. When the processor writes a character to the USART, TxRDY is automatically reset.

Concurrently, the μPD8251 may receive serial data; and after receiving an entire character, the RxRDY output is raised to indicate a completed character is ready for the processor. The processor fetch will automatically reset RxRDY.

Note: The μPD8251 may provide faulty RxRDY for the first read after power-on or for the first read after receive is re-enabled by a command instruction (RxE). A dummy read is recommended to clear faulty RxRDY. But this is not the case for the first read after hardware or software reset after the device operation has once been established.

The μPD8251 cannot transmit until the TxEN (Transmitter Enable) bit has been set by a Command Instruction and until the $\overline{CTS}$ (Clear to Send) input is a "zero". TxD is held in the "marking" state after Reset awaiting new Command Words.

μPD8251 PROGRAMMING

The USART must be loaded with a group of two to four control words provided by the processor before data reception and transmission can begin. A Reset (internal or external) must immediately proceed the control words which are used to program the complete operational description of the communications interface. If an external RESET is not available, three successive 00 Hex or two successive 80 Hex command instructions ($C/\overline{D}$ = 1) followed by a software reset command instruction (40 Hex) can be used to initialize the 8251.

There are two control word formats:

1. Mode Instruction
2. Command Instruction

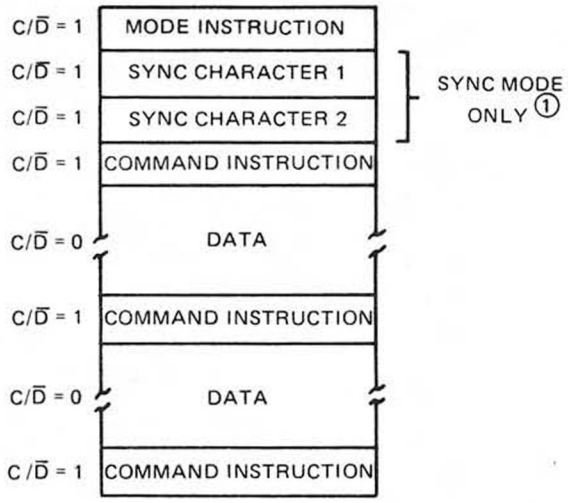
MODE INSTRUCTION

This control word specifies the general characteristics of the interface regarding the SYNCHRONOUS or ASYNCHRONOUS MODE, BAUD RATE FACTOR, CHARACTER LENGTH, PARITY, and NUMBER OF STOP BITS. Once the Mode Instruction has been received, SYNC characters or Command Instructions may be inserted depending on the Mode Instruction content.

COMMAND INSTRUCTION

This control word will be interpreted as a SYNC character definition if immediately preceded by a Mode Instruction which specified a Synchronous format. After the SYNC character(s) are specified or after an Asynchronous Mode Instruction, all subsequent control words will be interpreted as an update to the Command Instruction. Command Instruction updates may occur at any time during the data block. To modify the Mode Instruction, a bit may be set in the Command Instruction which causes an internal Reset which allows a new Mode Instruction to be accepted.

TYPICAL DATA BLOCK



NOTE ① The second SYNC character is skipped if MODE instruction has programmed the 8251 to single character Internal SYNC Mode. Both SYNC characters are skipped if MODE instruction has programmed the 8251 to ASYNC mode.

MODE INSTRUCTION DEFINITION

The μPD8251 can operate in either Asynchronous or Synchronous communication modes. Understanding how the Mode Instruction controls the functional operation of the USART is easiest when the device is considered to be two separate components, one asynchronous and the other synchronous, which share the same support circuits and package. Although the format definition can be changed at will or "on the fly", the two modes will be explained separately for clarity.

ASYNCHRONOUS TRANSMISSION

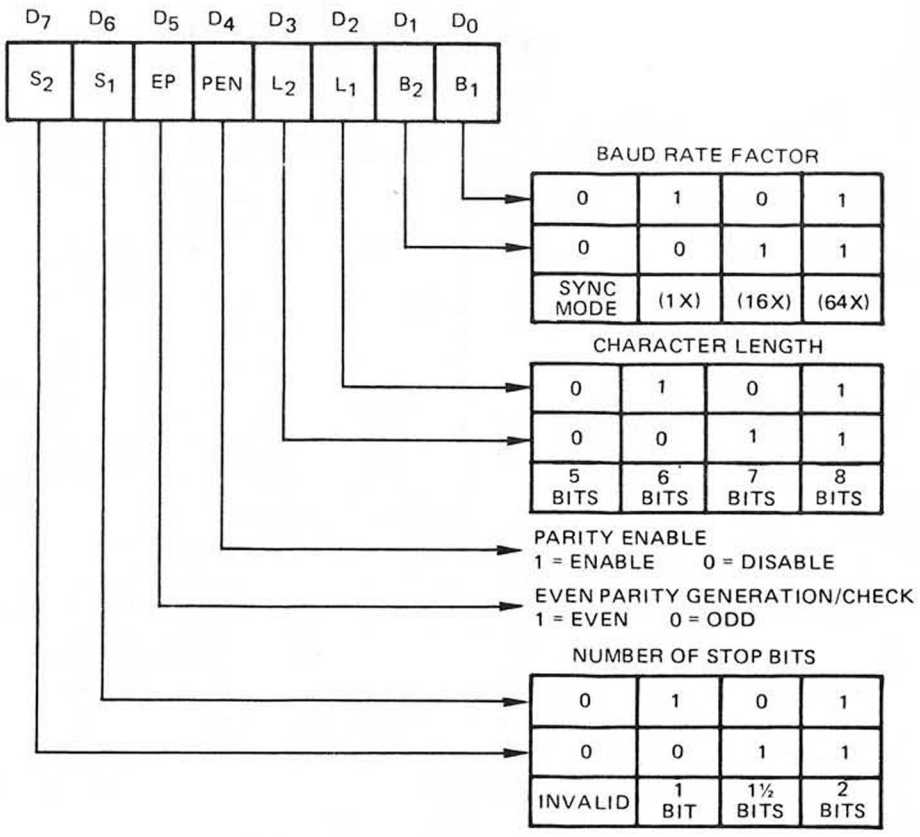
When a data character is written into the μPD8251, the USART automatically adds a START bit (low level or "space") and the number of STOP bits (high level or "mark") specified by the Mode Instruction. If Parity has been enabled, an odd or even Parity bit is inserted just before the STOP bits(s), as specified by the Mode Instruction. Then, depending on $\overline{CTS}$ and TxEN, the character may be transmitted as a serial data stream at the TxD output. Data is shifted out by the falling edge of $\overline{TxC}$ at $\overline{TxC}$, $\overline{TxC}$/16 or $\overline{TxC}$/64, as defined by the Mode Instruction.

If no data characters have been loaded into the μPD8251, or if all available characters have been transmitted, the TxD output remains "high" (marking) in preparation for sending the START bit of the next character provided by the processor. TxD may be forced to send a BREAK (continuously low) by setting the correct bit in the Command Instruction.

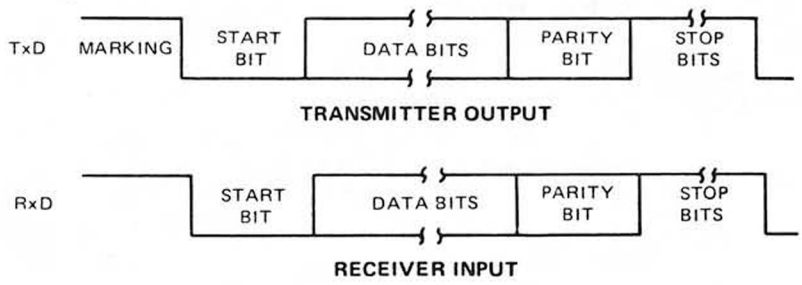
ASYNCHRONOUS RECEIVE

The RxD input line is normally held "high" (marking) by the transmitting device. A falling edge at RxD signals the possible beginning of a START bit and a new character. The START bit is checked by testing for a "low" at its nominal center as specified by the BAUD RATE. If a "low" is detected again, it is considered valid, and the bit assembling counter starts counting. The bit counter locates the approximate center of the data, parity (if specified), and STOP bits. The parity error flag (PE) is set, if a parity error occurs. Input bits are sampled at the RxD pin with the rising edge of $\overline{RxC}$. If a high is not detected for the STOP bit, which normally signals the end of an input character, a framing error (FE) will be set. After a valid STOP bit, the input character is loaded into the parallel Data Bus Buffer of the μPD8251 and the RxRDY signal is raised to indicate to the processor that a character is ready to be fetched. If the processor has failed to fetch the previous character, the new character replaces the old and the overrun flag (OE) is set. All the error flags can be reset by setting a bit in the Command Instruction. Error flag conditions will not stop subsequent USART operation.

MODE
INSTRUCTION FORMAT
ASYNCHRONOUS MODE

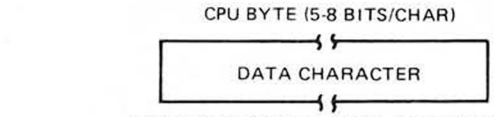


TRANSMIT/RECEIVE
FORMAT
ASYNCHRONOUS MODE



TRANSMITTER OUTPUT

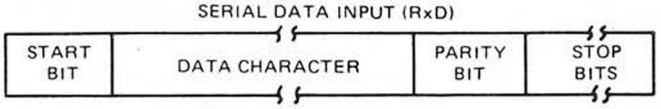
RECEIVER INPUT



ASSEMBLED SERIAL DATA OUTPUT (TxD)

TRANSMISSION FORMAT

SERIAL DATA INPUT (RxD)



NOTE ①: IF CHARACTER LENGTH IS DEFINED AS 5, 6, OR 7 BITS; THE UNUSED BITS ARE SET TO "ZERO."

RECEIVE FORMAT

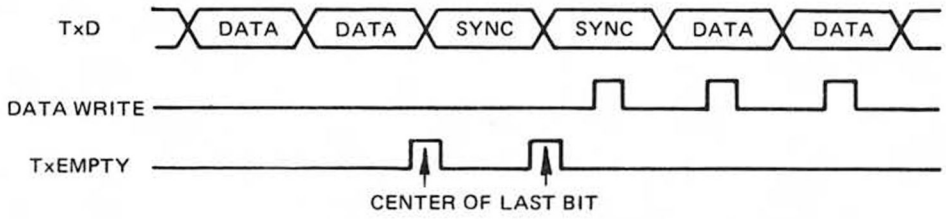
SYNCHRONOUS TRANSMISSION

As in Asynchronous transmission, the TxD output remains "high" (marking) until the μPD8251 receives the first character from the processor which is usually a SYNC character. After a Command Instruction has set TxEN and after Clear to Send ($\overline{CTS}$) goes low, the first character is serially transmitted. Data is shifted out on the falling edge of $\overline{TxC}$ and the same rate as $\overline{TxC}$.

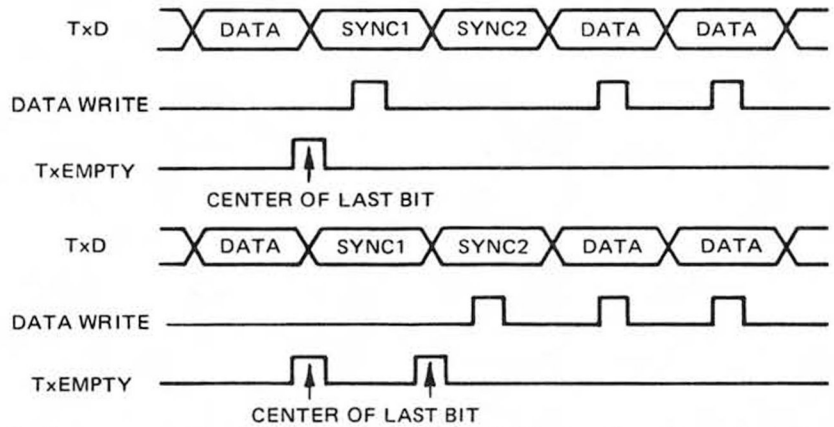
Once transmission has started, Synchronous Mode format requires that the serial data stream at TxD continue at the $\overline{TxC}$ rate or SYNC will be lost. If a data character is not provided by the processor before the μPD8251 Transmitter Buffer becomes empty, the SYNC character(s) loaded directly following the Mode Instruction will be automatically inserted in the TxD data stream. The SYNC character(s) are inserted to fill the line and maintain synchronization until new data characters are available for transmission. If the μPD8251 becomes empty, and must send the SYNC character(s), the TxEMPTY output is raised to signal the processor that the Transmitter Buffer is empty and SYNC characters are being transmitted. TxEMPTY is automatically reset by the next character from the processor.

TxEMPTY go high at the middle of the last data bit when the Transmit Register is EMPTY. TxEMPTY goes low again as sync characters are transmitted. See figure below.

FOR SINGLE SYNC CHARACTER OPERATION



FOR DOUBLE SYNC CHARACTER OPERATION (BISYNC)



SYNCHRONOUS RECEIVE

In Synchronous Receive, character synchronization can be either external or internal. If the internal SYNC mode has been selected, and the Enter HUNT (EH) bit has been set by a Command Instruction, the receiver goes into the HUNT mode.

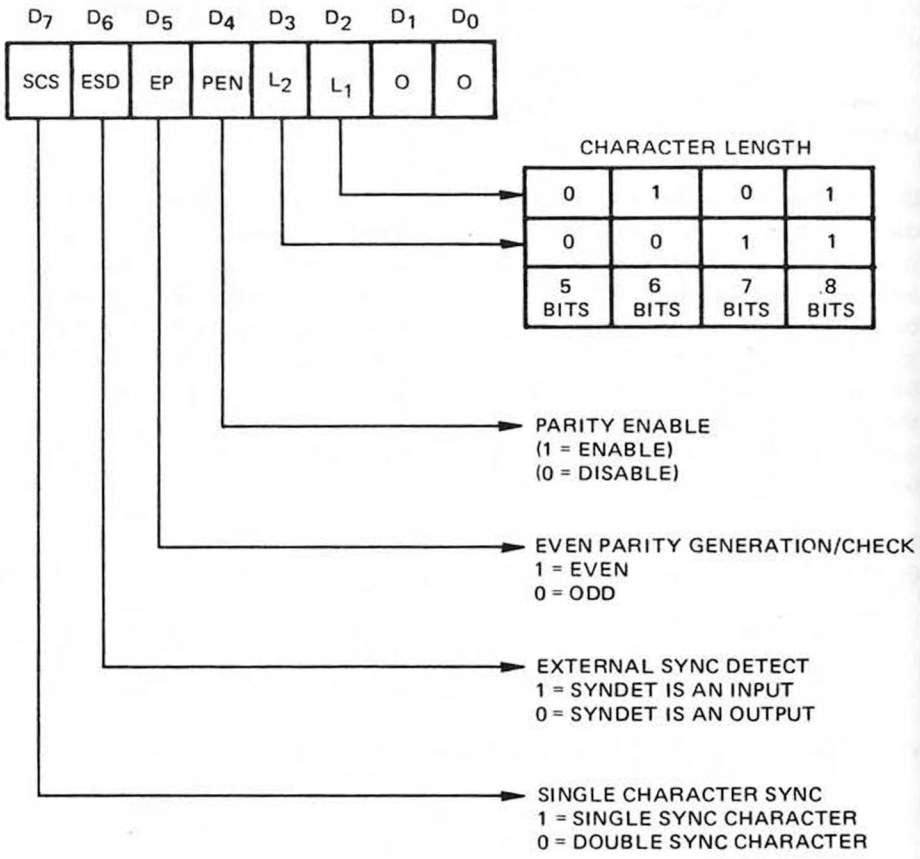
Incoming data on the RxD input is sampled on the rising edge of $\overline{RxC}$, and the Receiver Buffer is compared with the first SYNC character after each bit has been loaded until a match is found. If two SYNC characters have been programmed, the next received character is also compared. When the SYNC character(s) programmed have been detected, the μPD8251 leaves the HUNT mode and is in character synchronization. At this time, the SYNDET (output) is set high. SYNDET is automatically reset by a STATUS READ.

If external SYNC has been specified in the Mode Instruction, a "one" applied to the SYNDET (input) for at least one $\overline{RxC}$ cycle will synchronize the USART.

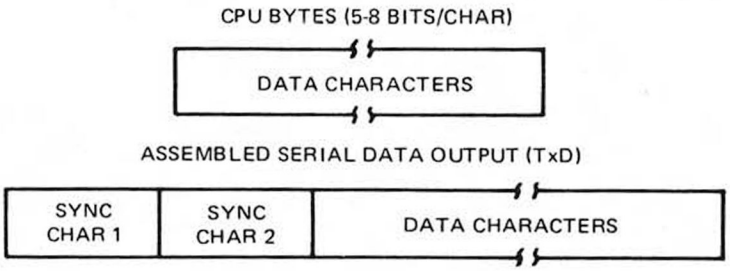
Parity and Overrun Errors are treated the same in the Synchronous as in the Asynchronous Mode. Framing errors do not apply in the Synchronous format.

The processor may command the receiver to enter the HUNT mode with a Command Instruction which sets Enter HUNT (EH) if synchronization is lost.

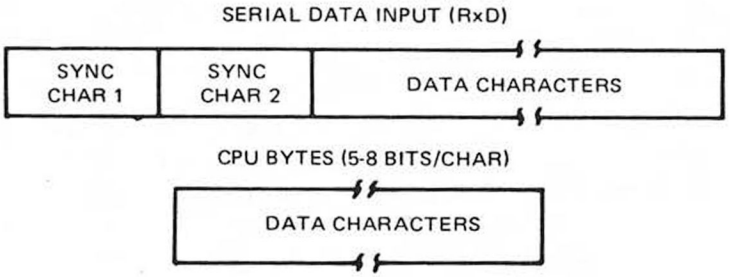
MODE INSTRUCTION FORMAT SYNCHRONOUS MODE



TRANSMIT/RECEIVE FORMAT SYNCHRONOUS MODE



TRANSMIT FORMAT



RECEIVE FORMAT

COMMAND INSTRUCTION FORMAT

After the functional definition of the μPD8251 has been specified by the Mode Instruction and the SYNC character(s) have been entered, if in SYNC mode, the USART is ready to receive Command Instructions and begin communication. A Command Instruction is used to control the specific operation of the format selected by the Mode Instruction. Enable Transmit, Enable Receive, Error Reset and Modem Controls are controlled by the Command Instruction.

After the Mode Instruction and the SYNC character(s), as needed, are loaded, all subsequent "control writes" ($C/\overline{D} = 1$) will load or overwrite the Command Instruction register. A Reset operation (internal via CMD IR or external via the RESET input) will cause the μPD8251 to interpret the next "control write", which must immediately follow the reset, as a Mode Instruction.

STATUS READ FORMAT

It is frequently necessary for the processor to examine the "status" of an active interface device to determine if errors have occurred or to notice other conditions which require a response from the processor. The μPD8251 has features which allow the processor to "read" the device status at any time. A data fetch is issued by the processor while holding the $C/\overline{D}$ input "high" to obtain device Status Information. Many of the bits in the status register are copies of external pins. This dual status arrangement allows the μPD8251 to be used in both Polled and interrupt driven environments. Status update can have a maximum delay of a 16 clock period.

PARITY ERROR

When a parity error is detected, the PE flag is set, and is cleared by setting the ER bit in a subsequent Command Instruction. PE being set does not inhibit USART operation.

OVERRUN ERROR

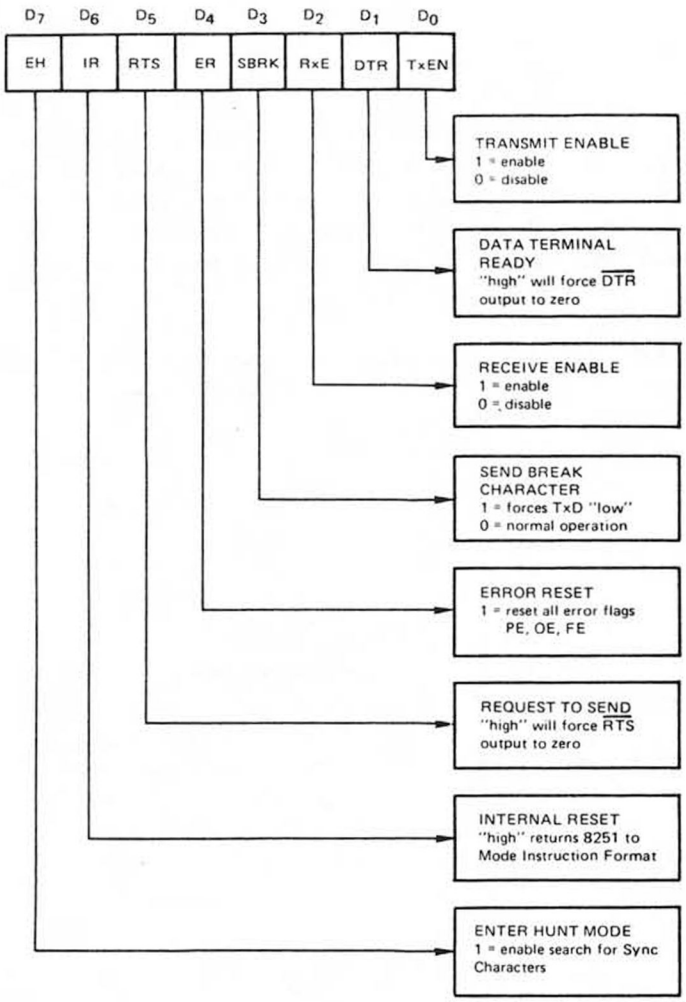
If the processor fails to read a data character before the one following is available, the OE flag is set, and is cleared by setting the ER bit in a subsequent Command Instruction. Although OE being set does not inhibit USART operation, the previously received character is overwritten and lost.

FRAMING ERROR ①

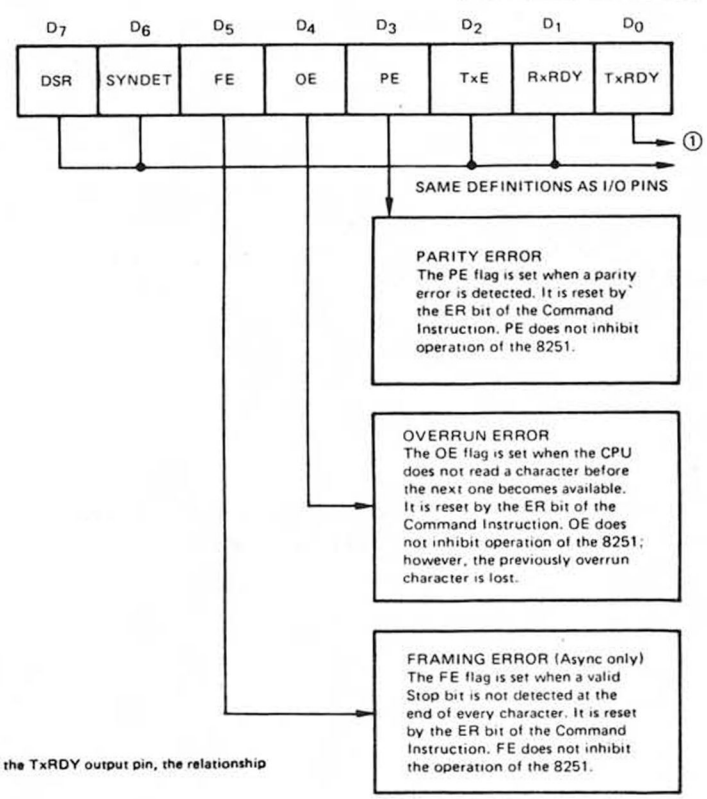
If a valid STOP bit is not detected at the end of a character, the FE flag is set, and is cleared by setting the ER bit in a subsequent Command Instruction. FE being set does not inhibit USART operation.

Note: ① ASYNC mode only.

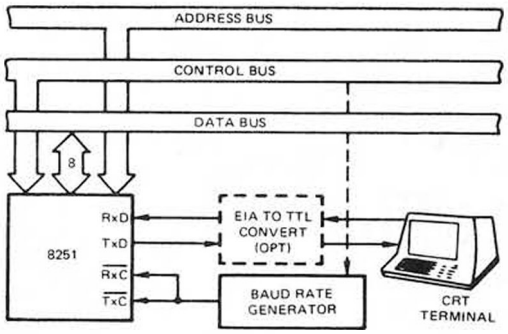
COMMAND INSTRUCTION FORMAT



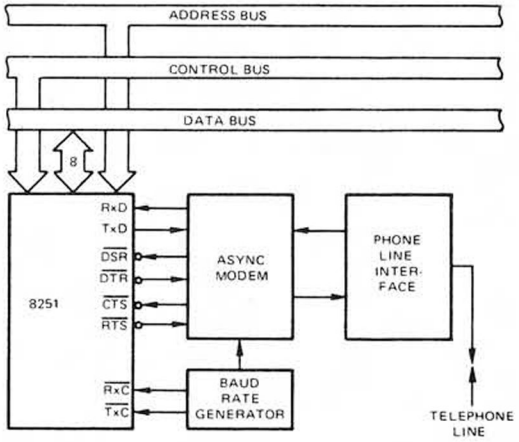
STATUS READ FORMAT



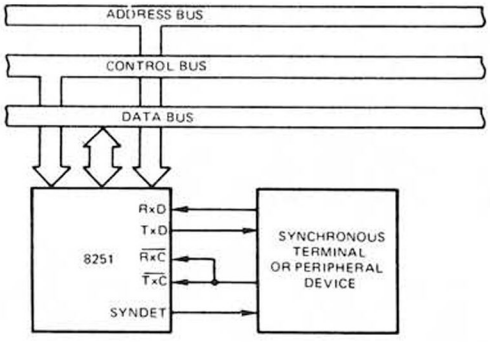
Note: ① TxRDY status bit is not totally equivalent to the TxRDY output pin, the relationship is as follows:
TxRDY status bit = DB Buffer Empty
TxRDY (pin 15) = DB Buffer Empty • CTS • TxEn



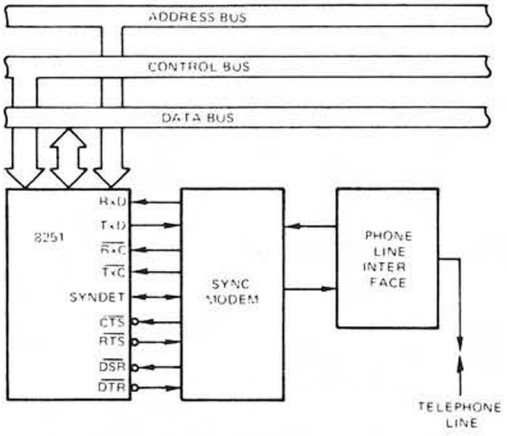
ASYNCHRONOUS SERIAL INTERFACE TO CRT TERMINAL, DC to 9600 BAUD



ASYNCHRONOUS INTERFACE TO TELEPHONE LINES



SYNCHRONOUS INTERFACE TO TERMINAL OR PERIPHERAL DEVICE



SYNCHRONOUS INTERFACE TO TELEPHONE LINES

APPENDIX C

HSIO-4 PARTS LIST
PL 00135A

ITEM	P/N	QTY	DESCRIPTION	REF
1	00133	1	HSIO-4 PC BOARD	
2	43003	4	IC 74LS03	3D,4D,6D,8D
3	43008	1	IC 74LS13	8E
4	43009	1	IC 74LS14	8F
5	43012	1	IC 74LS32	6E
6	43020	1	IC 74LS136	7E
7	43022	1	IC 74LS139	7D
8	43024	4	IC 74LS151	3E,3F,4E,4F
9	43027	2	IC 74LS161	2D, 1F
10	43029	1	IC 741S169	1D
11	43033	2	IC 74LS241	6F,7F
12	43038	3	IC 74LS259	5D,5E,5F
13	43044	1	IC 74LS393	2E
14	43070	4	IC MC1488	1B,3B,5B,7B
15	43071	4	IC MC1489	2A,4A,6A,8A
16	43148	4	IC 8251 (FOR 19.2K BAUD)	1C,3C,5C,7C
17	01001	25	CAPACITOR, .047 uF CER	BY-PASS
18	01005	16	CAPACITOR, 470 pF CER	C7-C22
19	01022	1	CAPACITOR, 6.8 uF 35V TANT	C5
20	01043	5	CAPACITOR, 2.2 uF 35V TANT	C1-C4,C6
21	13025	4	SOCKET, IC 8-PIN	
22	13026	5	SOCKET, IC 14-PIN	
23	13028	8	SOCKET, IC 16-PIN	
24	13034	4	SOCKET, IC-28 PIN	
25	13065	4	CONNECTOR, 26-PIN RIGHT ANGLE	J1,J2,J3,J4
26	13091	4	HEADER, SINGLE ROW, 2-PIN	JP5,6,7,8
27	13092-03	11	HEADER, SINGLE ROW, 3-PIN	JP1,2,3,4, JP9,10,11, JP12,13(3)
28	13064	4	SHUNT 16-PIN	1A,3A,5A,7A
29	43149	1	HEADER, HSIO-4 INTERRUPT(14-PIN)	2F
30	65002	1	VOLTAGE REGULATOR, 7805	
31	65003	1	VOLTAGE REGULATOR, 7812	
32	65018	1	VOLTAGE REGULATOR, 7912	
33	61013	4	RESISTOR, 220 OHM, 1/4 W, 5%	R5,R10,R15,R20
34	61016	4	RESISTOR, 560 OHM	R4,R9,R14,R19
35	61022	4	RESISTOR, 3.6K OHM	R3,R8,R13,R16
36	61025	4	RESISTOR, 5.6K OHM	R2,R7,R12,R17
37	61035	4	RESISTOR, 1K, 1/2 W	R1,R6,R11,R16
38	61024	4	RESISTOR, 4.7K	RN2,RN3,RN4,RN5
39	61004	1	RESISTOR NETWORK 2.2K	RN1

ITEM	P/N	QTY	DESCRIPTION	REF
40	68011	1	SWITCH, 4-POLE DIP	S1
41	38002	3	LOCKWASHER, #6	
42	38010	3	HEX NUT, 6-32	
43	38017	3	SCREW, 6-32 x 5/16	
44	38042	1	HEATSINK, 6106B-14	
45	38080	2	EJECTOR, CIRCUIT CARD	
46	38064	4	JACKSCREW KITS, D-CONNECTOR (Each kit contains: 2 jackscrews, 2 nuts, and 2 bolts)	
47	00144	4	RIBBON CABLE, HSIO-4	
48	00187B	1	HSIO-4 USER/TECHNICAL MANUAL	

North Star Computers, Inc.
1444O Catalina St., San Leandro, CA 94577 USA
(415) 357-85OO TWX/Telex (91O) 366-7OO1

00262B

North Star Computers, Inc.
1444O Catalina St., San Leandro, CA 94577 USA
(415) 357-85OO TWX/Telex (91O) 366-7OO1

00262B