IO8

8 Port Serial I/O Interface
S-100 Bus

USER'S MANUAL

AMTECH

IO8

8 Port Serial I/O Interface
S-100 Bus

User's Manual

SSM MICROCOMPUTER PRODUCTS, INC.
2190 Paragon Drive
San Jose, California 95131

(408) 946-7400 • Telex: 171171
TWX: 910-338-2077

Written by Malcolm T. Wright

Part No. MN0034
February 1982

TABLE OF CONTENTS

APPENDIX:

1.0 INTRODUCTION

The IO8 interface board offers maximum flexibility for your serial communication needs by providing eight RS-232 asynchronous interfaces for an S-100 or IEEE 696 system. Each serial interface is individually accessible from the bus as I/O in a Status/Data port pair. Baud rates are individually selectable for 110, 134, 300, 1200, 2400, 4800, 9600 and 19200 baud.

The IO8 makes it possible to support a wide variety of multi-user applications on a single board, including the following:

- Special purpose business applications (word processing, hotel management, banking terminals, training systems).
- Real-time control systems (engraving, machine tools, robotics, energy management).
- Data collection (quality control, laboratory data acquisition, communications message switching).
- Add-in/upgrading of current S-100 systems.

Each serial interface can be set through software for:

- 5 to 8 data bits
- 1 to 2 stop bits
- parity enable
- parity even or odd
- sending a break character
- control of the RTS and DTR lines

The RS-232 connector for each serial interface has been provided with 7 standard signals:

TD	(Transmit Data)
RD	(Receive Data)
RTS	(Request To Send)
CTS	(Clear To Send)
DTR	(Data Terminal Ready)
DSR	(Data Set Ready)
SG	(Signal Ground)

The eighth serial interface has two additional lines (TC and RC) for synchronous operation with a modem.

For real-time applications, the IO8 provides a 50/60 Hz interrupt clock on-board. To support I/O and timer interrupts, the IO8 has a very flexible priority interrupt circuit on-board to support 8080 RST instructions, Z80 Mode 2, 8086 byte identifier, Z8000 jump vector, or 68000 table vector applications. The IO8 is designed to synchronize with several high speed processors.

Optional cable assemblies are available for terminal, printer or modem interfacing to the IO8.

2.0 SETTING UP YOUR IO8

2.1 ADDRESSING THE BOARD

2.1.1 I/O Boundary

The IO8 board can be set up as 16 or 18 I/O ports depending on whether the interrupt timer is used. Each serial interface (8251A) uses two ports for bus communication, and the timer occupies two ports. To accommodate the 18-port setting, the IO8 is addressable to 32-port boundaries but only occupies 16 to 18 ports in that range. The starting address is selected by jumpers E8 thru E13.

Mini-jumper installed = 1
Mini-jumper removed = 0

I/O STARTING ADDRESS (Hex)	JUMPERS E8 to E9	 E10 to E11	 E12 to E13
00	0	0	0
*20	0	0	1
40	0	1	0
60	0	1	1
80	1	0	0
A0	1	0	1
C0	1	1	0
E0	1	1	1

*Used by SSM for the IO8.

The timer function, which adds two ports to the 16 ports used by the serial interfaces, can be enabled by jumpering pins E3 and E4 together. If pins E3 and E4 are not jumpered, the IO8 occupies only 16 ports, not 18, and there is no access to the timer. (Even if the timer is not accessible, it can still interrupt the CPU. Refer to Section 2.3 on how to disable the timer if necessary.)

The I/O ports can be further broken down into addresses by function in the following table:

ADDRESS (Hex)	FUNCTION	MODE	CONNECTOR
0,1	Serial-A (8251A)	Async	J0
2,3	Serial-B	"	J1
4,5	Serial-C	"	J2
6,7	Serial-D	"	J3
8,9	Serial-E	"	J4
A,B	Serial-F	"	J5
C,D	Serial-G	"	J6
E,F	Serial-H	Async/Sync	J7
10,11	Timer*	-	None

Async = Asynchronous serial interface
Sync = Synchronous serial interface for MODEMs only.
* = Only addressable if jumper selected

2.1.2 Status/Data Order

The serial and parallel interfaces on the IO8 have status and data port pairs which can be reversed by jumpers E5 thru E7. Some system standards require a status port first (even address) with the data port following (odd address). All status and data port addresses, except the timer function, are affected by this jumper.

Status port (even address), Data port (odd address)
Jumper E5 to E6

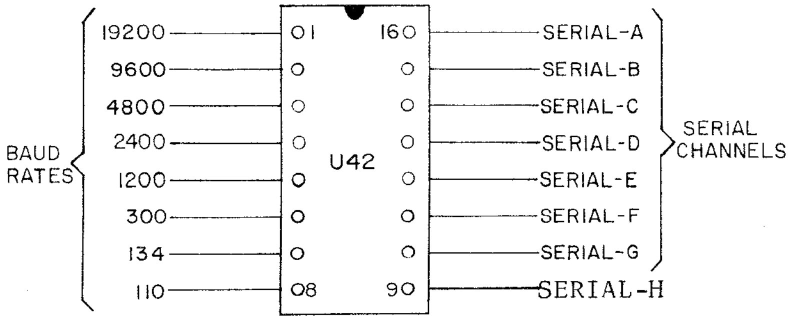
Data port (even address), Status port (odd address)
Jumper E6 to E7

This jumper selection **MUST BE MADE** for the board to work properly.

2.2 SERIAL INTERFACES

2.2.1 Baud Rate Selection

The serial channels can be strapped for any one of the 8 baud rates from 110 to 19200 baud. The IO8 has a 16-pin socket for using an IC header to make the baud rate selection, which requires only that a wire on the header be run from the desired baud rate to the serial channel pin.



Only one baud rate can be selected for each serial channel, but all channels can have the same baud rate.

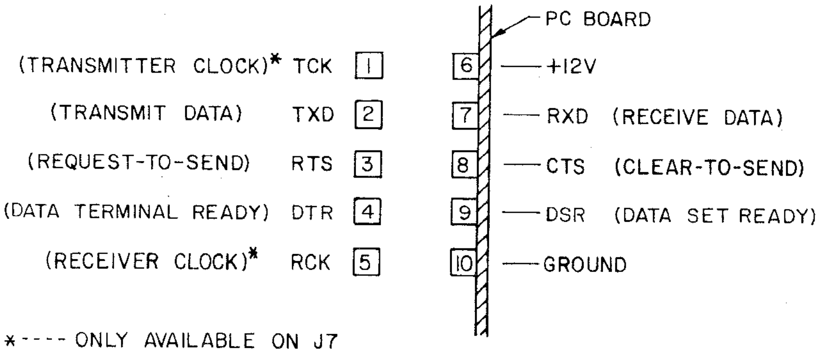
Example Setups:

a. To set all 8 serial interfaces to 9600 Baud, connect U42, pin 2 to U42 pins 9 thru 16.

b. To set serial "B" to 1200 Baud, and all other serial interfaces to 9600, connect U42, pin 5 to U42, pin 15; then connect U42, pin 2 to U42, pins 9 thru 14 and 16.

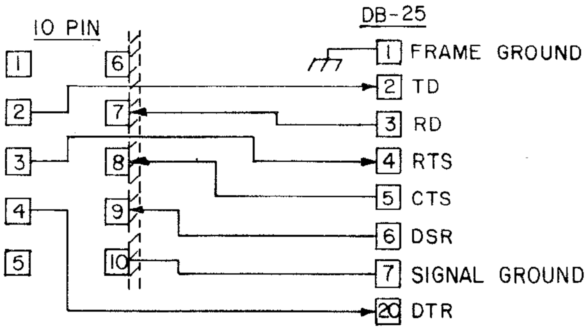
2.2.2 Interfacing a Modem

The 8 serial interfaces (A thru H) on the IO8 come off to a 10-pin male ribbon cable connector. Each connector provides six RS-232 signals and signal ground for terminal or modem applications.

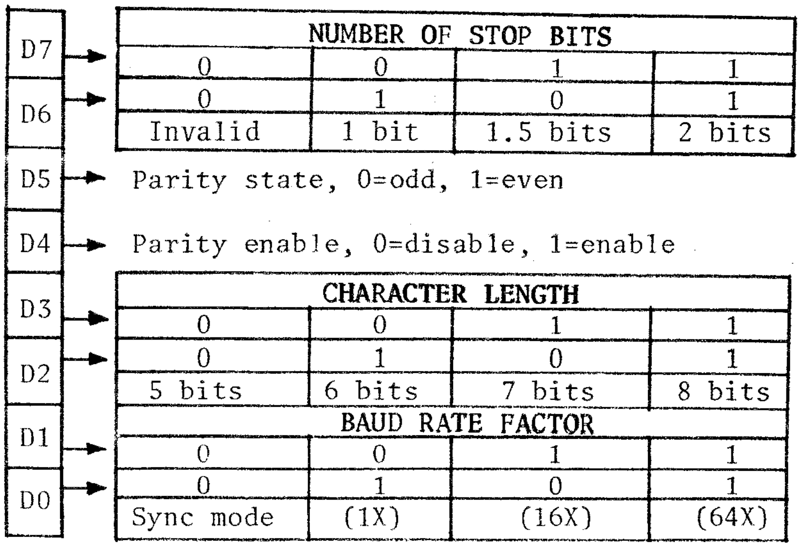


SERIAL CONNECTOR (JØ THRU J7)

The interconnection from this 10-pin connector to a DB-25 connector for a modem would be as follows:



Frame ground normally connects to the chassis ground of the computer mainframe, while signal ground for the DB-25 connector comes from the 10-pin serial connector.



D7, D6 →

NUMBER OF STOP BITS			
0	0	1	1
0	1	0	1
Invalid	1 bit	1.5 bits	2 bits

D5 → Parity state, 0=odd, 1=even

D4 → Parity enable, 0=disable, 1=enable

D3, D2 →

CHARACTER LENGTH			
0	0	1	1
0	1	0	1
5 bits	6 bits	7 bits	8 bits

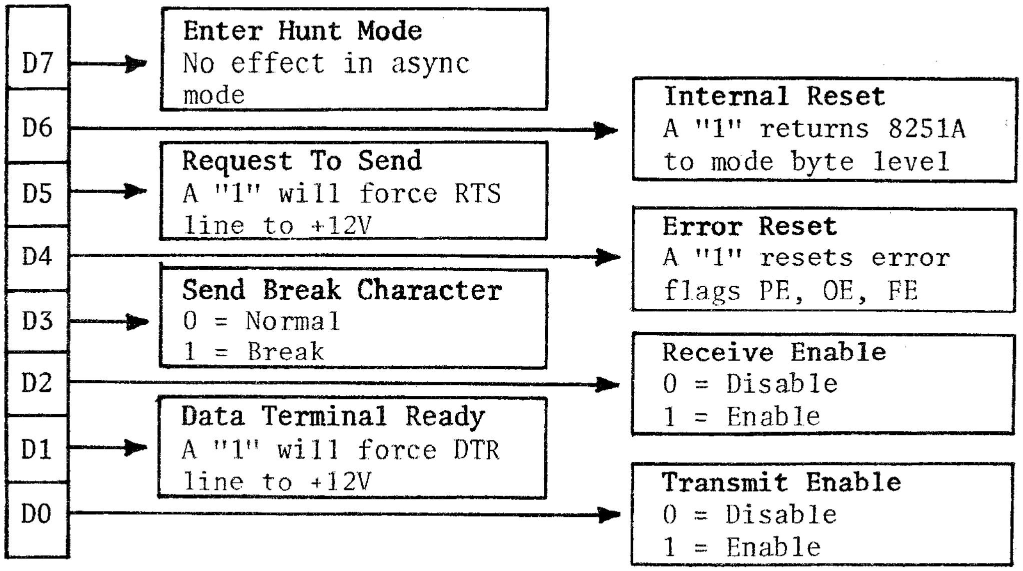
D1, D0 →

BAUD RATE FACTOR			
0	0	1	1
0	1	0	1
Sync mode	(1X)	(16X)	(64X)

MODE BYTE FORMAT

The IO8's baud rate circuit generates 16 times the data bit clock rate, so that D=0 and D1=1 in the mode byte. Typical character length is 8 bits; thus, D2=1 and D3=1. Parity selection will vary with the interfacing equipment and hardware application. If baud rates above 300 are selected, then 1 stop bit is typical, and D6=1 and D7=0. Therefore, a standard mode byte could be 4E Hex in most applications.

Once the mode byte is entered, a command byte should be sent. This byte will control the RS-232 handshaking lines DTR and RTS, can enable transmit and receive modes, clear error flags, send a "break" character, or reset the 8251A back to the mode byte level.



D7 → **Enter Hunt Mode**
No effect in async mode

D6 → **Internal Reset**
A "1" returns 8251A to mode byte level

D5 → **Request To Send**
A "1" will force RTS line to +12V

D4 → **Error Reset**
A "1" resets error flags PE, OE, FE

D3 → **Send Break Character**
0 = Normal
1 = Break

D2 → **Receive Enable**
0 = Disable
1 = Enable

D1 → **Data Terminal Ready**
A "1" will force DTR line to +12V

D0 → **Transmit Enable**
0 = Disable
1 = Enable

COMMAND BYTE FORMAT

A typical command byte will enable receive and transmit functions, force RTS and DTR high, not send a break character to start, and leave the error flags enabled for any possible communication problems. The command byte will be 27 Hex in most cases.

The same port address that a mode or command byte is written to is also the status port for normal communication. This port will indicate the arrival of data and the completion of transmisson of data.

STATUS BYTE FORMAT

D7	→ 1 = DSR line is high
D6	→ Don't care
D5	→ 1 = Framing error detected (no stop bit)
D4	→ 1 = Overrun error detected (buffer overflow)
D3	→ 1 = Parity error detected
D2	→ 1 = Transmit buffer empty (acknowledges sending)
D1	→ 1 = Data received (data available)
D0	→ 1 = Transmit buffer ready

If interrupts are active on the serial interfaces by strapping the INTERRUPT PRIORITY header, then masking these interrupts will become important. Bits 0 and 2 in the control byte are the interrupt mask bits. The 8251A has two output pins (RXRDY & TXRDY) which are "ORed" into one common interrupt line to the INTERRUPT PRIORITY header.

Serial Interrupts

- RXRDY line is true:
  - . Receive enable bit must be true (enable).
  - . Receive buffer has a character.

  NOTE: Receive enable bit must be true to sense the start bit on an incoming character.

- TXRDY line is true:
  - . Transmit enable must be true (enable).
  - . CTS input line must be greater than +3 volts.
  - . Transmit buffer is empty.

If either output line (RXRDY or TXRDY) is true, an interrupt can be generated. Software must check the status byte to determine if the interrupt is for receiving or transmitting.

EXAMPLE SOFTWARE

Serial-A for simple I/O with no interrupts. Jumper E5 to E6 installed and board address set to 20 Hex.

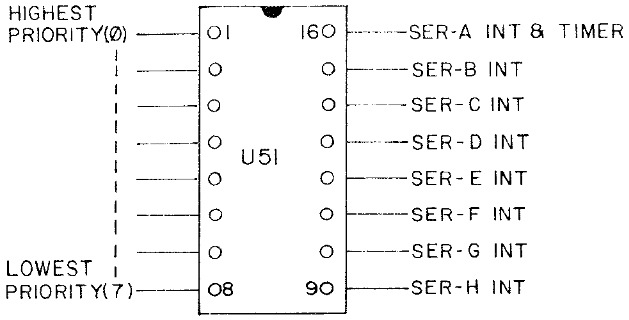
```
; PORT ADDRESSES
  KSTAT            EQU 20H; STATUS/COMMAND PORT
  KDATA            EQU 21H; DATA PORT

; LOCATION OF EXAMPLE DRIVER
  LOC              EQU 0E000H    ; USER DEFINED ADDRESS

                   ORG LOC
                   JMP INTZ      ; INITIALIZE 8251
                   JMP GET       ; GET DATA
                   JMP PUT       ; SEND DATA

; INITIALIZATION ROUTINE

; HELP GUARANTEE THE STATE OF THE 8251 PART
  INTZ:            MVI B,3
                   XRA A
  INTZ1:           OUT KSTAT
                   DCR B
                   JNZ INTZ1

; AT THIS POINT THE 8251A IS DEFINITELY WAITING FOR A COMMAND BYTE
                   MVI  A,40H    ; RESET TO MODE LEVEL
                   OUT  KSTAT
                   MVI  A,4EH    ; MODE BYTE
                   OUT  KSTAT
                   MVI  A,37H    ; COMMAND + ERROR RESET
                   OUT  KSTAT
                   MVI  A,27H    ; COMMAND
                   OUT  KSTAT
                   RET

; INPUT ROUTINE, REG.-A=DATA

; EXITS WITH:  A=DATA RECEIVED
 GET:              IN KSTAT      ; GET STATUS
                   ANI  2        ; TEST DAV
                   JZ GET
                   IN KDATA      ; GET DATA
                   RET

; OUTPUT ROUTINE, REG.-C=DATA

; ENTER WITH:  C=DATA TO SEND
 PUT:              IN KSTAT      ; GET STATUS
                   ANI 4         ; TEST FOR READY
                   JZ PUT
                   MOV A,C
                   OUT KDATA     ; SEND DATA
                   RET
```

In writing other software for the 8251A, you must have greater than 3.26 microseconds between writes during initialization; 4.34 microseconds between writes once in asynchronous mode; and 8.68 microseconds between writes if the part is running synchronous.

2.2.5 Serial Interrupts

2.2.5.1 General

The interrupt signals for DAV (Data Available) and DAK (Data Acknowledge) for each serial interface (A thru H) are sent to an INTERRUPT PRIORITY header, U51. Each serial interrupt can be jumpered to the priority desired (0 thru 7) or not connected if no interrupts are desired.



Only one serial interrupt signal can be tied to a particular priority level on U51 (i.e., you cannot connect two or more serial interrupts to one priority level input).

2.2.5.2 Maskable Output Interrupts

It may be desirable to have interrupts on serial inputs, but none on the DAK signal for serial outputs. The ability to send serial data but not generate an interrupt is possible with the 8251A under software. The TX-ENABLE bit in the command byte to the 8251A must be zero until a byte to be transmitted is loaded into the 8251A. If the TX-ENABLE bit is low, no DAK interrupt will be sent. With the output buffer full in the 8251A, no DAK interrupt will be sent. Next, with the buffer full, set TX-ENABLE bit equal to a one in the command byte; then immediately set TX-ENABLE bit low again. This change of state of the TX-ENABLE bit will cause transmission of the serial data without allowing the DAK interrupt.

An example routine for initializing the 8251A, checking DAK status, outputting serial data, enabling interrupts on DAK, and disabling interrupts on DAK in 8080 code follows.

```
;ROUTINE TO ENABLE OR DISABLE TRANSMITTER INTERRUPTS
;WRITTEN BY MALCOLM T. WRIGHT,11-24-81


;Board address.
IO8     EQU     20H     ;START ADDRESS

;Serial port addresses(Serial-A).
MODE0   EQU     IO8
CMD0    EQU     IO8
STS0    EQU     IO8     ;STATUS PORT
DATO    EQU     STS0+1  ;DATA PORT

;8251 bits
DAV     EQU     2
DAK     EQU     4       ;MUST BE TX-EMPTY BIT

;Flag bits.
DEV0    EQU     1       ;BIT MASK,SERIAL-A
DEV1    EQU     2       ;BIT MASK,SERIAL-B
DEV2    EQU     4       ;BIT MASK,SERIAL-C
DEV3    EQU     8       ;BIT MASK,SERIAL-D
DEV4    EQU     10H     ;BIT MASK,SERIAL-E
DEV5    EQU     20H     ;BIT MASK,SERIAL-F
DEV6    EQU     40H     ;BIT MASK,SERIAL-G
DEV7    EQU     80H     ;BIT MASK,SERIAL-H

;Mode bytes for 8251.
MBYT0   EQU     4EH     ;BAUD=16X,BITS=8,STOP=1

;Command bytes for 8251.
CBYT0   EQU     22H     ;RX & TX DISABLED




;MAIN ENTRY POINTS.
        JMP     INTZ    ;INITIALIZE 8251
        JMP     STOUT0  ;OUTPUT STATUS
        JMP     SOUT0   ;SEND DATA
        JMP     EINT0   ;ENABLE INTERRUPTS
        JMP     DINT0   ;DISABLE INT.(MASK)

;INITIALIZE THE IO8 (SERIAL A ONLY)
;Following sequence is unique to the 8251.
;       Force the 8251 into Command level.
INTZ:   MVI     B,3
        XRA     A
INTZ1:  OUT     CMD0
        DCR     B
        JNZ     INTZ1
;       Now force the 8251 to Mode level.
        MVI     A,40H   ;RESET TO MODE LEVEL
        OUT     CMD0
;       Set-up number of bits and baud.
        MVI     A,MBYT0 ;BAUD=16X,BITS=8,STOP=1
        OUT     MODE0
        MVI     A,CBYT0+10H;RESET ERROR,DISABLE RX&TX
        OUT     CMD0
        MVI     A,CBYT0
        OUT     CMD0      ;REMOVE RESET
        STA     WORD0     ;SAVE CMD BYTE
        RET

;Serial status check for DAK.
STOUT0: IN      STS0      ;GET STATUS
        ANI     DAK
        RZ                ;RET. IF NOT READY
        MVI     A,0FFH
        RET               ;RET. IF READY

;Serial output routine. (SERIAL-A ONLY)
SOUT0:  PUSH    H
SO1:    CALL    STOUT0    ;CHECK STATUS
        JZ      SO1
        MOV     A,C
        OUT     DAT0      ;OUTPUT DATA
        LXI     H,WORD0   ;PT. TO CMD WORD
        CALL    INTE      ;DAK ENABLE
        OUT     CMD0      ;NO INT. AS YET
                          ;SINCE BUFFER IS
                          ;FULL.
        LDA     INTF      ;INT. WANTED ON DAK?
        ANI     DEV0      ;TEST SERIAL-A BIT
        JNZ     SO2       ;JUMP IF YES.
        CALL    INTD      ;INT. DISABLE
        OUT     CMD0
SO2:    POP     H
        RET

;Command byte for Serial-A.
WORD0:  DB      CBYT0

;Output INT. DAK flag for 8 serial ports.
INTF:   DB      0         ;ALL DISABLED.

;Enable DAK interrupts.
EINT0:  LDA     INTF      ;GET INT FLAG
        ORI     DEV0      ;SET INT FLAG
        STA     INTF      ;SAVE FLAG
        LXI     H,WORD0
        CALL    INTE      ;CHANGE SER.-A CMD WORD
        OUT     CMD0      ;SEND COMMAND
        RET

;Disable DAK interrupts.
DINT0:  LDA     INTF      ;GET INT FLAG
        ANI     NOT DEV0 AND 0FFH;CLEAR INT FLAG
        STA     INTF      ;SAVE FLAG
        LXI     H,WORD0
        CALL    INTD      ;CHANGE SER.-A CMD WORD
        OUT     CMD0      ;SEND COMMAND
        RET

;TRANSMIT ENABLE.
;IF USING INTERRUPTS THIS WILL ALSO ENABLE
;DAK INTERRUPTS.
INTE:   MOV     A,M       ;GET CMD. WORD
```

```
        ORI     1       ;SET TX-READY
        MOV     M,A     ;SAVE IT.
        RET

;TRANSMIT DISABLE.
;IF USING INTERRUPTS THIS WILL ALSO DISABLE
;DAK INTERRUPTS.
INTD:   MOV     A,M     ;GET CMD. WORD
        ANI     0FEH    ;CLEAR TX-READY
        MOV     M,A     ;SAVE IT.
        RET
```

2.3 TIMER

The I08 has a simple interrupt timer on-board for multi-tasking or real-time system applications requiring a 50 Hz or 60 Hz clock. The timer is read or cleared by accessing the 17th or 18th port of the I08. The interrupt request from the timer is combined with the interrupt request from Serial-A (U6) because only 8 interrupt levels can be handled by the 74LS148 (U49). On a timer interrupt, you must test Serial-A status for the Data Available bit because an incoming character could look like a time interrupt.

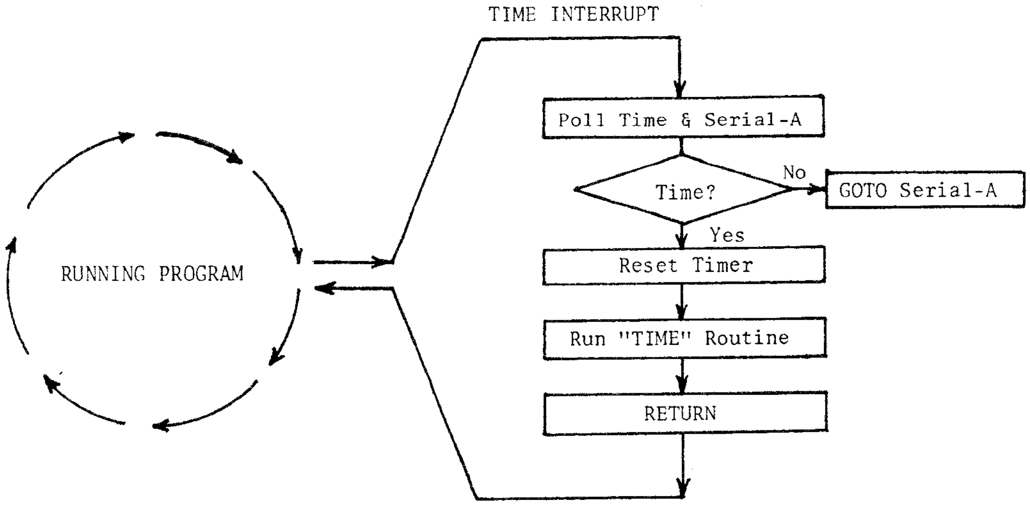
Near U12 is a jumper marked E1 & E2. This jumper selects 50 Hz (20 msec) or 60 Hz (16.66 msec) clock rates for the interrupt timer.

50 Hz = E1 connected to E2
60 Hz = E1 to E2 open

Once the interrupt timer indicates that a time interval has passed, the interrupt signal is maintained until the timer is reset. Outputting **anything** (00 to FF) to the 17th port of the I08 will reset the timer and ready it for the next time interval. The 50 Hz or 60 Hz clock is always running and the interrupt timer is tripped off of each complete cycle of the clock.

When the timer is tripped, the D0 bit for the 17th port is set to a zero. While timing or if the timer is reset, bit D0 will be a one when read by the CPU.

To disable the timer circuit if your applications do not require it, remove IC U13 (74LS107) from the I08 board. If you are not using interrupts off of the timer or Serial-A, you may ignore the timer function and leave IC U13 on the board.



TIMER PROCESS

2.4 INTERRUPT CIRCUIT

2.4.1 Introduction

If your applications are for a real-time multi-user or multi-tasking system, the flexibility of the I08 interrupt circuitry should meet a wide variety of 8 and 16 bit CPU requirements for the IEEE 696 bus. The interrupt circuitry is set up to address three main areas of system integration:

A. **Data is passed to the bus during an interrupt.**
The I08 can pass a one byte instruction, one byte code or a one byte address during an interrupt acknowledge.

B. **One of nine interrupt lines used.**
The I08 can drive the main interrupt line or one of the 8 vector interrupt lines on the IEEE 696 bus.

C. **Board arbitration for multiple interrupt cards.**
The I08 provides the ability to arbitrate by daisy chain, vector priority or address code during an interrupt acknowledge.

For **Item A**, on the passing of data, it is important to know what type of microprocessor will be used in the system. Some processors will accept a vector address byte, while others will have to poll (scan) the boards to find out who interrupted and then call a routine for service. The passing of one byte during an interrupt acknowledge for different processors can be listed as follows:

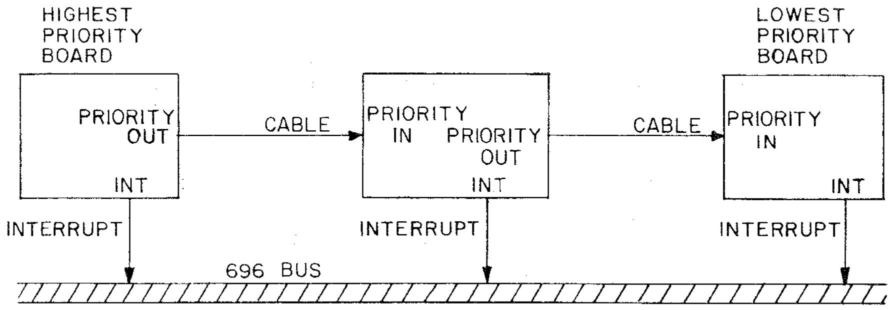
<table>
<tr><th>PROCESSOR</th><th>INTERRUPT MODE</th></tr>
<tr><td><b>8080</b></td><td>Responds to a one byte instruction during an interrupt. The instruction is called a "restart instruction" and is a one byte call. The restart code vectors the 8080 to one of 8 address locations per the code.
<table>
<tr><th>Mnemonic</th><th>Hex</th><th>Vectored to Address</th></tr>
<tr><td>RST0</td><td>C7</td><td>00 Hex</td></tr>
<tr><td>RST1</td><td>CF</td><td>08 Hex</td></tr>
<tr><td>RST2</td><td>D7</td><td>10 Hex</td></tr>
<tr><td>RST3</td><td>DF</td><td>18 Hex</td></tr>
<tr><td>RST4</td><td>E7</td><td>20 Hex</td></tr>
<tr><td>RST5</td><td>EF</td><td>28 Hex</td></tr>
<tr><td>RST6</td><td>F7</td><td>30 Hex</td></tr>
<tr><td>RST7</td><td>FF</td><td>38 Hex</td></tr>
</table>
</td></tr>
<tr><td><b>Z80</b></td><td>Responds in one of three ways:
<b>Mode 0:</b> The same as 8080.
<b>Mode 1:</b> No code passed, the Z-80 vectors to address 0038 Hex.</td></tr>
</table>

PROCESSOR	INTERRUPT MODE
Z80 (con't)	**Mode 2:** One byte is passed to form the least significant part of a 16-bit vector address. The most significant part of the address is stored in the I-Register. This 16-bit address points into a table where the address of the service routine must be stored. All three modes of the Z80 are software programmable and are set to MODE 0 during a reset. The most powerful mode is mode 2, and it is supported by the IO8.
8085	Responds to a one byte instruction during an interrupt like the 8080. See 8080 processor interrupt mode.
8088/8086	Responds to a one byte identifier during an interrupt acknowledge. This one byte identifier is multiplied by 4 to give a table address at which the vector address to the service routine can be found. Identifiers of 20 Hex to FF Hex are allowed for external interrupts. This mode is easily supported by the IO8.
68000	Responds to a one byte table address during an interrupt acknowledge. The one byte table address is multiplied by 4 and expanded with additional zeros to give a 24-bit table address at which the address of the service routine can be found. The 68000 also puts out a priority code on address lines A1, A2 & A3 to tell which peripheral it will service during an interrupt acknowledge. The table byte can be 40 Hex to FF Hex for about 192 vector addresses. The IO8 supports A1 thru A3 board arbitration as well as the one byte table address.
Z8000	Responds to a one byte jump vector during an interrupt acknowledge. The one byte jump vector is doubled, 30 is added to it, and then the program status area pointer is added to it. The new computed vector points to a table which has the address of the service routine. The one byte jump vector can be 00 Hex to FF Hex. The IO8 can pass one byte jump vectors during an interrupt acknowledge.

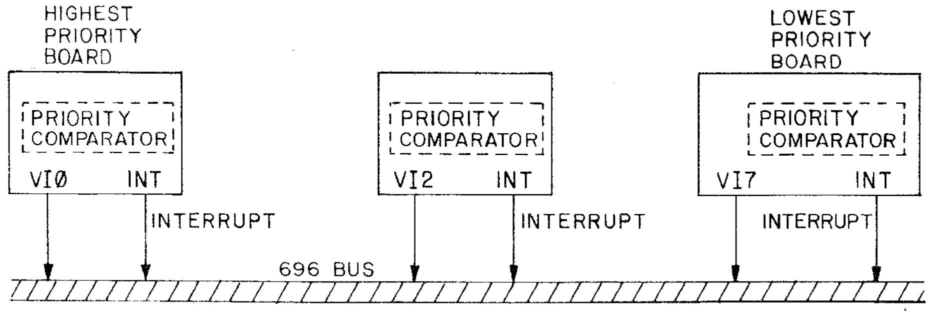
The setup in **Item B**, on the selection of interrupt lines, will vary from system to system depending on the interrupt circuitry available on the CPU and other I/O boards. The most commonly supported interrupt line is INT (pin 73) on the IEEE 696 bus. The line NMI (non-maskable interrupt) is reserved for major errors within the system and power failure and thus was not considered as a jumper option on the IO8. The vector interrupt lines (VI0 thru VI7) are mainly used with an interrupt controller circuit present on the CPU or external board. The IO8 can use the vector interrupt lines to arbitrate between multiple IO8's or IO5's.

Item C, on arbitration, will become a major issue when multiple interrupting boards are present on the bus. If no interrupt controller is present within the computer, the boards themselves must arbitrate. Two methods are available on the IO8:

o "Daisy chain" cable connection between boards.
o Arbitrate on the VI lines by priority.



DAISY CHAIN SYSTEM



VI ARBITRATION SYSTEM

The IO8's interrupt circuit is divided into three sections: a priority encoder, an arbitration circuit, and a one byte code buffer. Refer to the Interrupt Circuit Block Diagram.

PRIORITY ENCODER

To control the order of transfer of the 8 on-board interrupting devices (serial and timer), a priority encoder circuit is utilized. The priority encoder receives the device interrupts through the INTERRUPT PRIORITY HEADER so that the highest to lowest priority is user-selectable. The output interrupt from the priority encoder is user-strappable to the MAIN

INTERRUPT or the VECTOR INTERRUPT lines of the IEEE 696 bus. The priority encoder has an INTERRUPT ENABLE jumper which must be strapped for interrupts to be passed to the CPU.

Two daisy chain signal pins have been provided to the priority encoder called PRIORITY IN and PRIORITY OUT, if daisy chaining is desired between boards.

ARBITRATION CIRCUIT

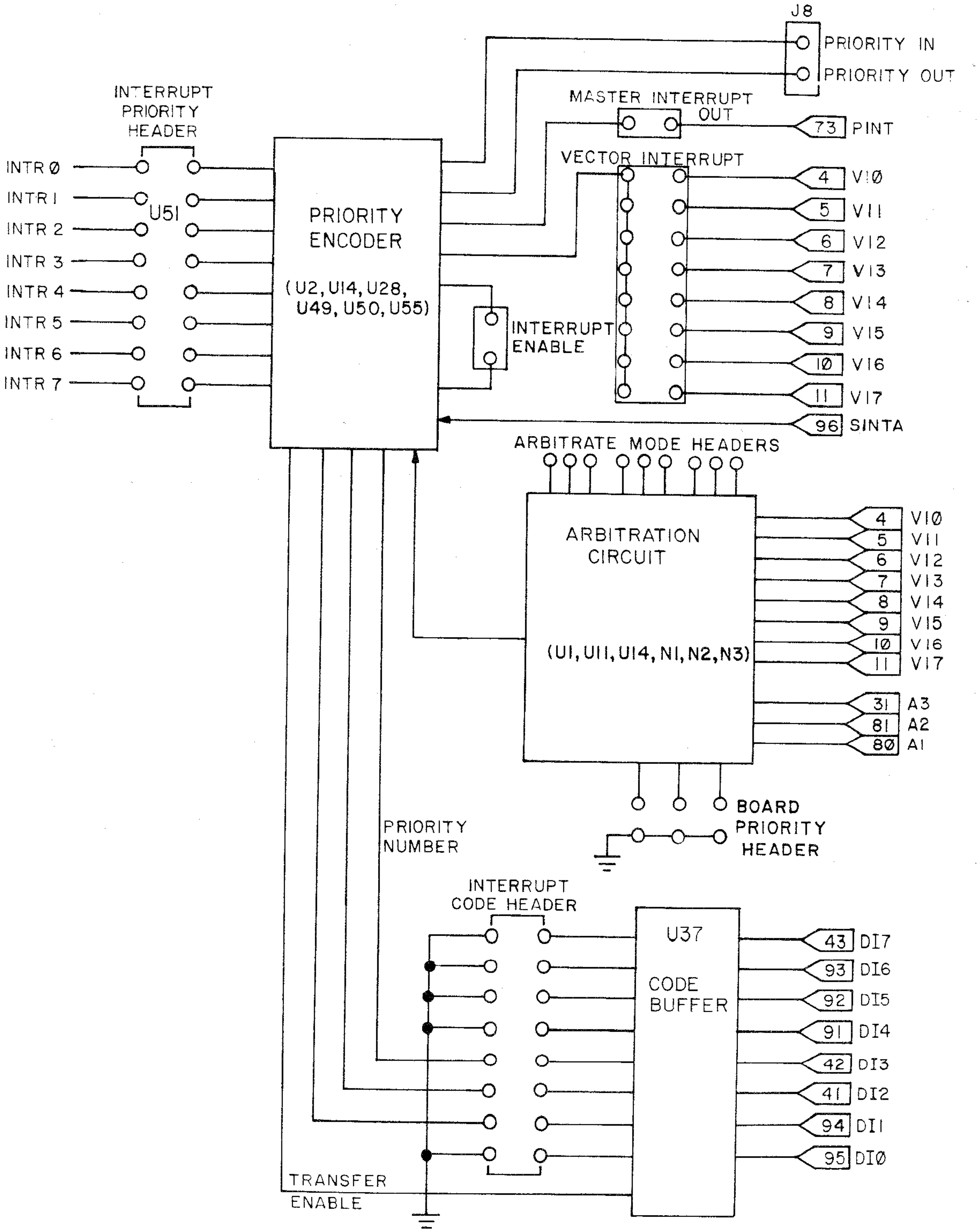
The arbitration circuit is used to prevent a bus conflict between multiple interrupting IEEE 696 boards. The arbitration circuit can be set up for one of three modes of operation:

a. Arbitrator disabled for daisy chain only operation.
b. Arbitrate by priority of the vector interrupt lines.
c. Arbitrate by priority number sent back on the address lines.

The arbitration circuit basically prevents the board from sending back an INTERRUPT CODE during an interrupt acknowledge from the CPU unless one of the three modes are met. The ARBITRATE MODE header and BOARD PRIORITY header sets the main arbitration mode.

CODE BUFFER

The one byte code buffer is used to pass an instruction or vector address byte to the CPU during an interrupt acknowledge. The code is user-selectable since it will depend on the capability of the microprocessor chip used. The INTERRUPT CODE header receives three lines from the priority encoder to pass the number of the on-board interrupting device wanting service. Only if the arbitrator's conditons are met will the code buffer be enabled to transfer data to the main CPU during an interrupt acknowledge cycle.



INTERRUPT CIRCUIT BLOCK DIAGRAM

2.4.3 CPU Interrupt Line

The IEEE 696 bus has 10 interrupt input lines; the IO8 can drive 9 of these interrupt lines. The line NMI (non-maskable interrupt) is reserved for major errors within the system and power failure and thus was not considered as a jumper option on the IO8.

Interrupt Lines

INT (pin 73)
VI0 (pin 4)
VI1 (pin 5)
VI2 (pin 6)
VI3 (pin 7)
VI4 (pin 8)
VI5 (pin 9)
VI6 (pin 10)
VI7 (pin 11)

The interrupt lines used by the IO8 can be strapped several ways; the following are some examples:

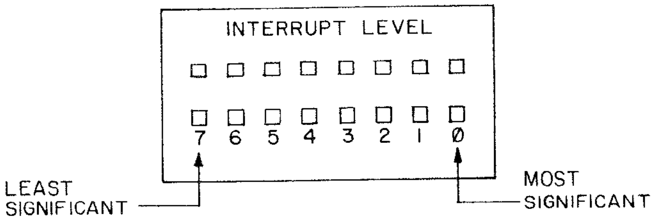
- **One interrupt board only, no vector interrupt controller.**
  Only INT needs to be driven, so jumper MSTR INT OUT.

- **Multiple interrupt boards, no vector interrupt controller, daisy chain protocol.**
  Only INT needs to be driven, so jumper MSTR INT OUT. Connect up the daisy chain cable between boards.

- **Multiple interrupt boards, no vector interrupt controller, VI arbitration.**
  Jumper MSTR INT OUT for INT. A different "VI" line is selected for each board by jumpering the VECTOR INTERRUPT header.

- **Multiple interrupt boards, vector interrupt controller, VI arbitration.**
  Leave MSTR INT OUT open. A different "VI" line is selected for each board by jumpering the VECTOR INTERRUPT header.

The described interconnections are only examples, and may vary depending on system applications.

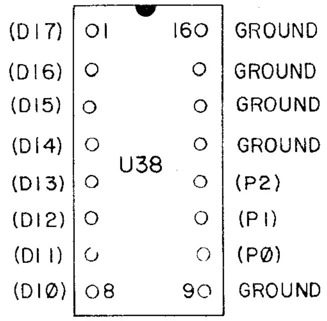


VECTOR INTERRUPT (VI Lines)

Only **ONE** vector interrupt level should be selected per board.

2.4.2 Pass Data Codes

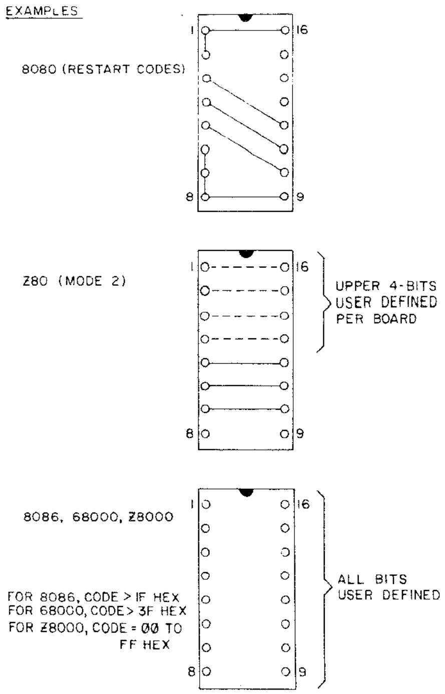
As was indicated in Section 2.4.1 in Item A, a wide variety of microprocessors will accept a one byte code during an interrupt acknowledge cycle. The IO8 has an INTERRUPT CODE header which can provide a fixed code or a variable code, depending on its strapping.



INTERRUPT CODE HEADER

If pins 1 thru 8 are left unconnected, the data code will be 00 Hex during an interrupt acknowledge. If pins 1 thru 8 are connected to 16 (ground), the data code becomes an FF Hex.

The lines P0 thru P2 on the interrupt code header give a priority number for the on-board device requesting service on the IO8. If the P0 thru P2 lines are mixed with ground connections to the data (DI) side of the interrupt code header, one byte instructions or vector addresses can be formed.



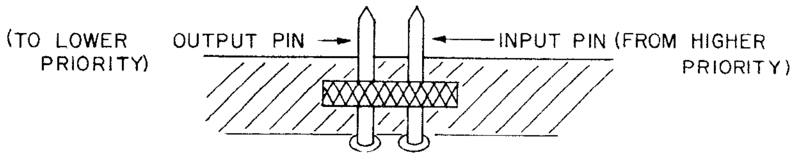
2.4.4 Board Arbitration

2.4.4.1 Daisy Chain

To support an old scheme used by several S-100 manufacturers, a two-pin daisy chain connector has been provided on the IO8. Daisy chaining is a method (external from the IEEE 696 bus) of arbitrating between multiple boards equipped with interrupts by an inter-board cable. The highest priority board can send a signal to the next lowest priority board to disable its interrupt until the highest priority is serviced by the CPU. Boards are literally strung together by a cable from the highest priority down to the lowest.

One major weakness of daisy chaining is the time necessary to propagate a disable signal down through all the boards on the interconnecting cable. During interrupt acknowledge time interval, part of the time is used for daisy chain propagation and part of the time for passing the interrupt data code to the CPU. If too many boards are daisy chained, the CPU may not see the data code. The number of boards that can be daisy chained is governed by the CPU's speed and number of cycles allowed for an interrupt acknowledge.

The daisy chain connector has two pins; one is for the interrupt disable signal from a higher priority board, and the other to go to the next lowest priority board.



DAISY CHAIN CONNECTOR

If only daisy chaining is used for interrupt arbitration, the "VI" arbitration scheme must be defeated on the IO8 board in the following way:

Setup for Daisy Chaining

- No jumpers should be present on N1 thru N3 of the ARBITRATE MODE headers.
- Jumper 1 of the Priority Number.
- Jumper 2 " " "
- Jumper 3 " " "

2.4.4.2 VI Lines

The VI (Vector Interrupt) lines can be used by multiple IO8 or IO5 interface boards for interrupt priority arbitration. This scheme is faster than the daisy chain concept, so the number of interrupting boards does not affect performance of the system. In the VI arbitration scheme, each of the I/O boards watches all 8 VI lines and only transfers interrupt data to the bus when that board's priority number is generated from an on-board priority encoder.

Each I/O board should be set up to drive **only one** VI line on the bus. The BOARD PRIORITY header should be set to match the number of that VI line. Up to eight I/O boards can use this arbitration scheme, and if each board is like the IO8, up to 64 serial interrupts can be prioritized and independently address vectored.

a. To enable this mode, first set the ARBITRATE MODE header as follows:

   Jumper A to C on N1
   Jumper A to C on N2
   Jumper A to C on N3

b. Next, set the priority of the board on the VECTOR INTERRUPT header by referring to the vector interrupt figure in Section 2.4.3.

c. Set the BOARD PRIORITY header to match the vector interrupt line selected in Step 2.

VI SELECTED	**CONNECT PRIORITY NUMBER**			**PRIORITY NUMBER**	**PRIORITY LEVEL**
	3	**2**	**1**		
0	no	no	no	7	Highest
1	no	no	yes	6	.
2	no	yes	yes	5	.
3	no	yes	yes	4	.
4	yes	no	no	3	.
5	yes	no	yes	2	.
6	yes	yes	no	1	.
7	yes	yes	yes	0	Lowest

2.4.4.3 Address Line

Presently, only the 68000 microprocessor is using this mode. The 68000 will send out on three of its address lines (A1,A2,A3) the priority level it will service during an interrupt acknowledge. Only the I/O board that matches this priority level can pass back a vector address byte.

a. To enable this mode, first set the ARBITRATE MODE header as follows:

   Jumper B to D on N1.
   Jumper B to D on N2.
   Jumper B to D on N3.

b. Next, set the priority of the board on the VECTOR INTERRUPT header by referring to the vector interrupt figure in Section 2.4.3. Be careful with your vector interrupt selection because restrictions may exist depending on the 68000 CPU board used.

VI SELECTED	CONNECT PRIORITY NUMBER			PRIORITY NUMBER	PRIORITY LEVEL
	1	2	3		
0	no	no	no	7	Highest
1	no	no	yes	6	.
2	no	yes	no	5	.
3	no	yes	yes	4	.
4	yes	no	no	3	.
5	yes	no	yes	2	.
6	yes	yess	no	1	.
7	yes	yes	no	0	Lowest

2.5 STANDARD SETUP

To simplify the use of the IO8 board, a few standard setup examples are listed. First, the jumpers and headers on the board relate to the following functions:

General

E8 thru E13 . . . . . . Set the board's I/O address.
E5 thru E7 . . . . . . Select Status/Data order for serial.
E3 to E4 . . . . Enable timer access.
E1 to E2 . . . . Set timer's rate to 50 or 60 Hz.
E14 thru E21 . . . . . For serial-H only, synchronous modem.
U42 . . . . Baud rate selection.

Interrupts

Interrupt Enable . . . . . Must be jumpered if interrupts desired.
MSTR INT OUT . . . . Drives PINT bus line.
Interrupt Level . . . . . . Drives VI0 thru VI7 bus lines.
Interrupt Number . . . . . Select Arbitrate mode.
Priority Number . . . . . . Board's priority number.
U38 - Interrupt Code . . . Byte to be passed to the CPU.
U51 - Interrupt Priority . Select which serial interface can interrupt.

2.5.1 No Interrupts, No Timer

CONNECT	COMMENT
E8 thru E13	Set board address per Section 2.1.1
E5 to E6	Status/Data Order selected
Jumper U42	Select baud rate per Section 2.2.1

2.5.2 Console Interrupt Only (8080 Mode)

This setup connects only one interrupt line for the console (serial-A) on the INTERRUPT PRIORITY header (U51). We will assume, as a simple example, that a 8080 or 8085 CPU will be used. The INTERRUPT CODE to be passed will be a RST 1 instruction (see Section 2.4.1). Only the PINT line (bus pin 73) will be driven, so daisy chaining must be used if more than one interrupting board is used in the computer.

CONNECT	COMMENT
E8 thru E13	Set board address per Section 2.1.1.
E5 to E6	Status/Data Order selected.
Jumper U42	Select baud rate per Section 2.2.1.
Remove U13	Disable timer function.
E16 to E17	Run Serial-H asynchronous.
E18 to E17	" " "
Interrupt Enable	Enable interrupt circuitry.
MSTR INT OUT	IO8 will drive PINT line.
Priority Number 1	Setup for daisy chain only.
Priority Number 2	" " " " "
Priority Number 3	" " " " "
U51 pin 16 to pin 1	Interrupt from Serial-A (console) per Section 2.2.5.
Set U38	Set up a RST-1 instruction (CF Hex).
Pin 1 to 16	DI7=1
Pin 1 to 2	DI6=1
Pins 3, 4 open	DI4 & DI5=0
Pin 5 to 6	DI3=1
Pin 6 to 7	DI2=1
Pin 7 to 8	DI1=1
Pin 8 to 9	DI0=1

2.5.3 Serial and Timer Interrupts (8080 Mode)

This setup connects all 8 interrupt lines on the INTERRUPT PRIORITY header (U51). We will assume, as a simple example, that a 8080 or 8085 CPU is used. The INTERRUPT CODE passed to the CPU will be a RST-2 instruction (see Section 2.4.1) for Serial-E thru H, and a RST-3 instruction for the Timer and Serial-A thru D. Only the PINT line (bus pin 73) will be driven, so daisy chaining must be used if more than one interrupting board is used in the computer.

CONNECT	COMMENT
E8 thru E13	Set board address per Section 2.1.1.
E5 to E6	Status/Data Order selected.
Jumper U42	Select baud rate per Section 2.2.1.
E3 to E4	Enable Timer port access.
E1 open	Set 60 Hz timer interrupts.
E16 to E17	Run Serial-H asynchronous.
E18 to E19	" " "
Interrupt Enable	Enable interrupt circuitry.
MSTR INT OUT	IO8 will drive PINT line.
Priority Number 1	Setup for daisy chain only.
Priority Number 2	" " " " "
Priority Number 3	" " " " "
Set U51	Set up INTERRUPT PRIORITY.
Pin 1 to 1	Timer, Serial-A
Pin 2 to 15	Serial-B
Pin 3 to 14	Serial-C
Pin 4 to 13	Serial-D
Pin 5 to 12	Serial-E
Pin 6 to 11	Serial-F
Pin 7 to 10	Serial-G
Pin 8 to 9	Serial-H
Set U38	Set up RST-2 & RST-3
Pin 1 to 16	DI7=1
Pin 1 to 2	DI6=1
Pin 3 open	DI5=0
Pin 4 to 13	DI4=1
Pin 5 to 12	DI3=0 or 1 by priority
Pin 6 to 7	DI2=1
Pin 7 to 8	DI1=1
Pin 8 to 9	DI0=1

2.5.4 Serial and Timer Interrupts (Z80 Mode 2)

This setup connects all 8 interrupt lines on the INTERRUPT PRIORITY header (U51). The INTERRUPT CODE passed to the Z80 in Mode 2 will be addresses 00 thru 0F Hex to be appended to the value in the I-Register. The Z80 vector table will look like this:

XX = I-Register's value

TABLE ADDRESS	INTERRUPT FROM
XX00	Serial-H
XX02	Serial-G
XX04	Serial-F
XX06	Serial-E
XX08	Serial-D
XX0A	Serial-C
XX0C	Serial-B
XX0E	Timer & Serial-A

Each starting table address has two bytes pointing to the address of the interrupt service routine.

VI arbitration will be used for multiple IO8 boards. Assume no interrupt controller is available on the Z80 CPU board or within the system. No daisy chaining cable is needed between boards.

CONNECT	COMMENT
E8 thru E13	Set board address per Section 2.1.1.
E5 to E6	Status/Data Order selected.
Jumper U42	Select baud rate per Section 2.2.1.
E3 to E4	Enable timer port access.
E1 open	Set 60 Hz timer interrupts.
E16 to E17	Run Serial-H asynchronous.
E18 to E19	" " "
Interrupt Enable	Enable interrupt circuitry.
MSTR INT OUT	IO8 will drive PINT line.
Interrupt Level 0	This board highest priority per Section 2.4.4.2.
Arbitrate Mode	Set up VI Arbitration mode.
N1, A to C	
N2, A to C	
N3, A to C	
Priority numbers all open	Set up for highest priority.
Set U51	Set up INTERRUPT PRIORITY
Pin 1 to 16	Timer, Serial-A
Pin 2 to 15	Serial-B
Pin 3 to 14	Serial-C
Pin 4 to 13	Serial-D
Pin 5 to 12	Serial-E
Pin 6 to 11	Serial-F
Pin 7 to 10	Serial-G
Pin 8 to 9	Serial-H
Set U38	Set up Mode 2 table address
Pin 1, 2 open	DI6=DI7=0
Pin 2, 3 open	DI4=DI5=0
Pin 5 to 12	DI3=Address
Pin 6 to 11	DI2=Address
Pin 7 to 10	DI1=Address
Pin 8 open	DI0=0

3.0 THEORY OF OPERATION

3.1 ADDRESS SELECT CIRCUIT (Refer to the ADDRESS SELECT CIRCUITRY diagram)

The address select circuit for the IO8 is formed out of five IC's:

U25 (74LS136) . .	Compares address lines A5 to A7 against jumpers E8 thru E13.
U26 (74LS14) . . .	Inverts U25's output.
U17 (74LS42) . . .	Chip select for each serial interface plus timer.
U15 (74LS32) . . .	OR-gates to enable/disable U17.
U16 (74LS83) . . .	Generates master board enable for 16 or 18 port addresses.

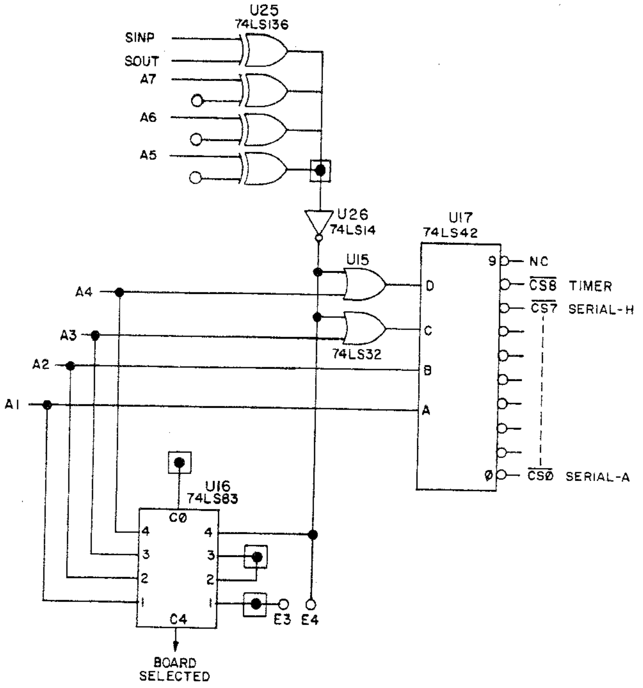
U25 is a quad exclusive-or IC with open collector outputs. The outputs of U25 are tied together to form one main IO8 select line. If the board address (A5 thru A7) matches the address jumpers (E8 thru E13) and it is an I/O operation (SINP or SOUT are high), the outputs of U25 will go high. The outputs of U25 (pins 3,6,8,11) are inverted by U26 to provide a low-going enable to U15 and U16.

U15 is used to block the passage of the A3 and A4 address lines to U17 when the board is not selected. U17 is a 4-to-10 line decoder which drives the select lines for each serial IC (8251A) and the timer (U13). U17 will output a chip select signal when the address lines A1 thru A4 are within the binary number range of 0 thru 9. If U15 pins 2 & 13 receive a high state (board not selected), then U17 receives an offset of a binary number 12 which disables its outputs.

U16 is used to check if the board select line (U26, pin 4) is within a 16 or 18 port range. U16 is a 4-bit adder which adds the number 6 or 7 plus carry to the incoming address A1 thru A4. If there is no final carry (U16, pin 14), the I/O select is within the range of the IO8's ports.

BOARD SELECT	TIMER	ADDRESS LINES				BOARD ENABLE	
(U26, pin 4)	E3-E4	A4	A3	A2	A1	(U16, pin 14)	COMMENT
1	X	X	X	X	X	1	Timer select disabled. Serial selects disabled.
0	Open	0	0	0	0	0	Serial-A
0	"	0	0	0	1	0	Serial-B
0	"	0	0	1	0	0	Serial-C
0	"	0	0	1	1	0	Serial-D
0	"	0	1	0	0	0	Serial-E
0	"	0	1	0	1	0	Serial-F
0	"	0	1	1	0	0	Serial-G
0	"	0	1	1	1	0	Serial-H
0	"	1	0	0	0	1	Overflow. Timer read/write disabled on U4.
0	Closed	1	0	0	0	0	Timer read/write enabled on U4.
0	"	1	X	X	1	1	Overflow. Timer read/write disabled. All serials deselected.

I/O SELECT TABLE



ADDRESS SELECT CIRCUITRY

3.2 **WAIT CIRCUIT** (Refer to the WAIT CIRCUIT Diagram)

To guarantee the operation of the IO8 up to 6.2 MHz (8-bit processors) and up to 12.4 MHz (16-bit processors with two cycle I/O operations), a wait circuit was added. Each I/O read or write access of the IO8 board will generate one cycle of wait back to the main CPU on the PRDY line.

The wait circuit is formed out of four IC's:

U15 (74LS32) . . . Receives PSYNC to start wait.

U26 (74LS14) . . . Drives clock of flip-flop and sets wait state.

U27 (74LS107) . . Flip-flop to clock out one wait cycle.

U3 (75453) . . . Drives PRDY line with wait request.

U15 is used to isolate the PSYNC bus line from an RC filter circuit on-board the IO8. The RC filter removes noise spikes from the PSYNC bus signal to prevent multiple wait cycles from occurring. The filtered PSYNC signal is then inverted by a schmitt trigger gate to drive the reset pin of U27.

U26 is a hex schmitt trigger invert gate. U2 is used to help reject noise spikes on the CPU clock line (pin 24) and the PSYNC line for proper one wait cycle operation of U27.

U27 is the wait cycle flip-flop. At the start of every read/write CPU cycle, PSYNC resets the flip-flop for a wait state level. After PSYNC, one CPU cycle later, the clock input of the flip-flop is strobed to load in a "one" and remove the wait level. The wait state level from the flip-flop goes through one more control gate (U3) before it gets to the bus.

U3 is an open collector OR-gate. Wait cycles from U27 are combined with the board enable signal (U16, pin 14) to provide a PRDY signal to the bus. Both inputs to U3 must be low to make PRDY low for a wait request to the CPU.

3.3 SERIAL CONTROL

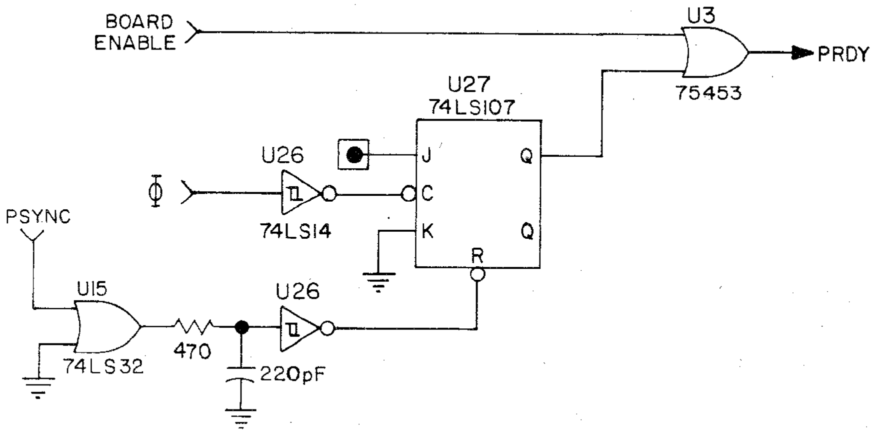
The I08 uses eight 8251A chips for its serial interfaces. Each 8251A must receive:

Pin 20, A clock . . . . 1.8432 MHz used for internal operation.
Pin 11, Chip select . . Enable IC for reading or writing.
Pin 12, C/D line . . . Select command or data registers.
Pin 13, Read . . . . . Read strobe.
Pin 10, Write . . . . . Write strobe.
Pin 21, Reset . . . . . Reset IC to start-up state.
Pins 9 & 25 . . . . . . Baud rate clock for communications.

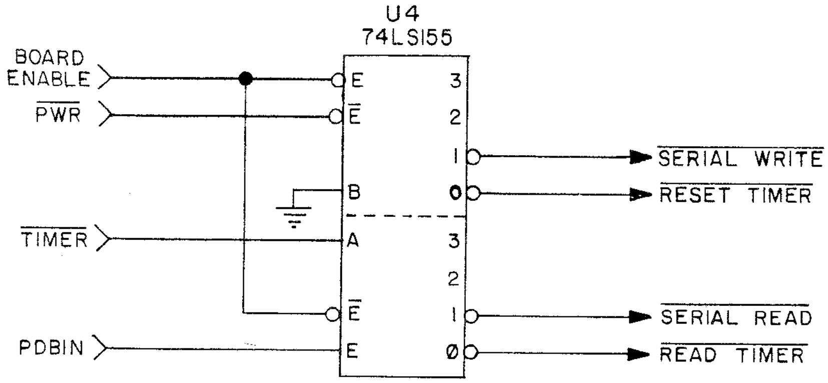
Pin 11 is generated by the Address Select Circuit (see Section 3.1). Pin 12 is related to the A0 address line. The A0 line goes through two inverters (U26) to allow for positive or negative logic control at E5-E7. E5 thru E7 jumpers allow the command port to precede or follow the data port address of the 8251A as mapped to the bus.

Pin signals 10 and 13 are generated from U4 (74LS155). This dual 2-to-4 decoder receives the board enable, PWR, PDBIN, and timer select to create a read and a write strobe signal to the 8251As (see READ/WRITE STROBE Diagram).

The reset signal (pin 21) is formed by a two-input NAND gate (U40) with Power-On Clear (POC) and Processor Reset (PRESET) as its input signals. The output of U40 goes high if either POC or PRESET goes low.



WAIT CIRCUIT



READ/WRITE STROBE

3.4 BAUD RATES

The baud rate circuit for the IO8 is formed out of three IC's:

U39 (MC14411) . . . Master baud rate IC and crystal oscillator.
U41 (74LS163) . . . ÷ 6 counter for 19200 baud.
U40 (74LS00) . . . Load pulse for U41.

U39 is a CMOS baud rate generator IC which will provide rates from 110 to 9600 baud. There is no software control of U39 selection lines RSA and RSB; because **ALL** baud rates including 19200 were desired, a counter (U41) was added. 19200 is not directly available from U39 when 110 baud is also desired. Each baud rate output goes to an option header pin of U42 so that it can be jumpered to any of the 8 serial ICs. U42 has a pull-up resistor on all 8 serial baud rate inputs to reduce the possibility of static electricity damage.

U41 is a synchronous counter used to create 16 x 19200 baud by dividing 1.8432 MHz by 6. (The 8251A must be set by software to divide by 16 all rates for correct data speed.) The counter was set to count from one to 6 for best output waveform symmetry, so U40 detects a binary 6 and loads "one" on the next cycle into U41. The Qc output from the counter is used for 19200 baud. Qc is low during counts 1, 2 and 3, and high during 4, 5 and 6.

3.5 INTERRUPT CIRCUIT (Refer to sheet 2 of the SCHEMATIC)

The interrupt circuit on the IO8 is formed by twelve ICs:

U52 thru U54 . . . . Nor gates to combine send & receive interrupts.
U50 (74LS373) . . . Latch the interrupt state during acknowledge.
U49 (74LS148) . . . Prioritize serial/timer interrupts into numbers.
U55 (74LS04) . . . . Daisy chain input and output control.
U2 (75453) . . . . Drive main & vector interrupt lines.
U14 (74LS04) . . . . Sense board interrupt & 68000 arbitrate control.
U28 (74LS20) . . . . Latch U50 & enable interrupt code on acknowledge.
U37 (74LS240) . . . Buffer to pass interrupt code.

U1 (74LS148) . . . Prioritize the vector interrupt lines.
U11 (74LS136) . . . Compare interrupt number with board priority.

U52 thru U54 combine RX-RDY (pin 14) and TX-RDY (pin 15) from each 8251A IC into a common interrupt signal line. There are 9 device interrupts, but Serial-A and the timer are combined to make only 8 lines to the header socket U51. Each interrupt signal to U51 (pins 9 thru 16) is a low-going signal.

Socket U51 is used for setting the priorities of the 8 interrupting lines to the input latch (U50). U50 is used to hold the interrupt state of each of the 8 lines during an interrupt acknowledge (SINTA). The interrupt state is frozen in order to give a stable condition for arbitration purposes for a multiple interrupting board architecture. Freezing the state also stabilizes the interrupt code to the main CPU.

U49 receives the 8 interrupt lines from the latch and prioritizes it into a 3-bit number 0 thru 7. U49 can be disabled by a higher priority board through the daisy chain connector J8 by driving U55, pin 11 low. U49 passes an interrupt enable to the next board if it has no interrupts through U55, pin 12.

The master interrupt signal for the IO8 comes from U49, pin 14. A jumper called INTERRUPT ENABLE must be installed if interrupts are desired to drive the IC buffer (U2) for signals on PINT or the VI lines. For arbitration puroses, the master interrupt (U49, pin 14) is sensed by U14, pin 13 to determine whether the board should respond to the SINTA signal.

Half of U28 acts as a latch control circuit for U50. If SINTA is low, the latch is open for a change in the interrupt state to pass to U49. If SINTA is high, the latch holds the last state of the signals from U51. The other gate in U28 controls the passing of the interrupt code. If U28, pin 2 is high and U49 is sending an interrupt and the arbitration circuit's (U11) output is high, then U28 pin 6 goes low to turn on the interrupt code tri-state gate U37.

There are three ways to arbitrate between multiple boards on the IO8:

a) Daisy chain cable to connector J8.
b) Priority of the VI lines.
c) A1 thru A3 address match-up (68000 only).

In items "b" and "c", the IO8 has a comparator (U11) to check the board's priority (P1 thru P3) against either the VI lines by U1 or the address lines through U14. If the two inputs to each gate of U11 are in opposite states, then U11 pin 8 will go high allowing the IO8 to respond with an interrupt code during an interrupt acknowledge. Jumpers N1 thru N3 allow the user to select either VI or 68000 mode of arbitration. If N1 thru N3 are left open and the board priority pins (P1 thru P3) are jumpered, only daisy chain arbitration is possible.

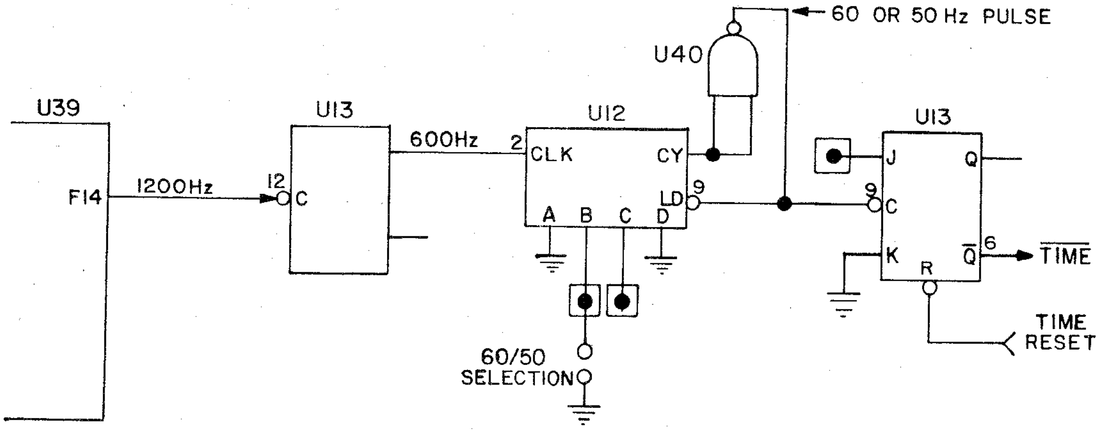
3.6 TIMER (Refer to TIMER CIRCUIT Diagram)

The on-board timer circuit is formed out of four ICs:

U3 (MC14411) . . . Provides a 1200 Hz signal.
U13 (74LS107) . . . Divides by 2 and is the timer flip-flop.
U12 (74LS163) . . . Divides by 10 or 12 the clock.
U40 (74LS00) . . . Load pulse for U12.

U39 provides a crystal-controlled clock of 1200 Hz. This crystal-controlled frequency must be reduced to 50/60 Hz for the timer. First the 1200 Hz is divided by two to 600 Hz by U13. The 600 Hz signal is fed to a 4-bit counter which can be set to divide by 10 or 12. If 60 Hz is selected, the counter is loaded with 6 and reloads after 15. If 50 Hz is selected, the counter is loaded with 4 and reloads after 15.

The 60 or 50 Hz clock is fed to the input of another flip-flop. This flip-flop can be read by the main CPU to check its status on bit D0. If bit D0 is low, one cycle or more of 60/50 Hz has passed. The CPU can reset this flip-flop to time out another 60/50 Hz interval. If the IO8 is set up for interrupts, the timer can provide time-slice interrupts for multi-tasking purposes.



TIMER CIRCUIT

4.0 TROUBLESHOOTING HINTS

To aid in checkout of the IO8, the serial interface connectors have been set up so that mini-jumpers can be used for self-testing the board. J0 thru J7 have input pins across from output pins on the 10-pin 90-degree connector.

Example connector pin-out (see Section 2.2.2):

OUTSIDE PINS	**AGAINST THE BOARD PINS**
1 = (J7 only)	6 = +12 volts
2 = Transmit Data	7 = Receive Data
3 = Request-To-Send	8 = Clear-To-Send
4 = Data Terminal Ready	9 = Data Set Ready
5 = (J7 only)	10 = Ground

By placing a mini-jumper to connect 2 to 7, 3 to 8, and 4 to 9 on J0 or J1 or J2, etc., a self-test program can be entered and run on the system to check that serial interface.

The following listing is a simple check routine to be run under CP/M. Three mini-jumpers should be installed on the serial connector to be tested. The IO8 EQU in the routine for the IO8's address should be changed to match your selected hardware address. The SERIAL EQU should be set for the serial interface you wish to test and should match the connector you have jumpered.

```
                         ;A simple self test of the IO8 Serial interface.
                         ;Written by Malcolm T. Wright,3-10-82
                         ;Technical support from D.B.Maerzke

FFFF =                   TRUE    EQU     0FFFFH  ;LOGIC TRUE
0000 =                   FALSE   EQU     NOT TRUE
8000 =                   TIME    EQU     8000H   ;DELAY TIMER


0020 =                   IO8     EQU     20H     ;STARTING BOARD ADDRESS
0041 =                   SERIAL  EQU     'A'     ;SERIAL PORT TO TEST
FFFF =                   STATUS  EQU     TRUE    ;STATUS PORT FIRST

0100 =                   LOC     EQU     100H    ;START OF PROGRAM
0005 =                   BDOS    EQU     5       ;CP/M COMMAND ENTRY
0000 =                   BOOT    EQU     0       ;WARM CP/M BOOT
                         ;Commands for CP/M.
0001 =                   CONS    EQU     1       ;READ FROM CONSOLE
0009 =                   STNG    EQU     9       ;SEND STRING TO CONSOLE
                         ;Characters.
000D =                   CR      EQU     0DH     ;CARRIAGE-RETURN
000A =                   LF      EQU     0AH     ;LINE-FEED
                         ;Port parameters.
0020 =                   KSTAT   EQU     IO8+(SERIAL-41H)*2+(NOT STATUS AND 01H)
0021 =                   KDATA   EQU     IO8+(SERIAL-41H)*2+(STATUS AND 01H)


0100                             ORG     LOC
                         ;Test DAV and DAK status bits.
0100 CD9101              BEGIN:  CALL    INTZ    ;INITIALIZE SERIAL CHIP
0103 DB21                        IN      KDATA   ;CLEAR AWAY INITIAL JUNK
0105 DB20                        IN      KSTAT   ;GET 8251A STATUS
0107 E601                        ANI     1       ;TEST DAK BIT
0109 11B001                      LXI     D,STR1  ;SET A MESSAGE
010C CA8501                      JZ      ERROR   ;DAK ERROR?
010F DB20                        IN      KSTAT
0111 E602                        ANI     2       ;TEST DAV BIT
0113 11BD01                      LXI     D,STR2  ;
0116 C28501                      JNZ     ERROR   ;DAV ERROR?
                         ;Try to send a character.
0119 3E55                        MVI     A,55H
011B D321                        OUT     KDATA   ;SEND A CHARACTER
011D 210080                      LXI     H,TIME
0120 DB20                TEST1:  IN      KSTAT   ;CHECK STATUS
0122 E602                        ANI     2       ;REC'D YET?
0124 C23301                      JNZ     TEST2   ;DATA REC'D
0127 2B                          DCX     H       ;DECREMENT TIME
0128 7D                          MOV     A,L
0129 B4                          ORA     H
012A C22001                      JNZ     TEST1   ;STILL TIME TO LOOK.
012D 11CA01                      LXI     D,STR3
0130 C38501                      JMP     ERROR   ;REC/SEND DATA ERROR
0133 DB21                TEST2:  IN      KDATA   ;GET DATA
0135 FE55                        CPI     55H     ;IS ALL BITS OK.
0137 11D601                      LXI     D,STR4
```

```
013A C28501             JNZ     ERROR   ;DATA BIT ERROR.
                ;Test the DTR and DSR lines.
013D CD9101             CALL    INTZ    ;INITIALIZE CHIP
0140 DB20               IN      KSTAT
0142 E680               ANI     80H     ;TEST DSR BIT
0144 11E101             LXI     D,STR5
0147 CA8501             JZ      ERROR   ;DSR BIT ERROR
014A 3E25               MVI     A,25H
014C D320               OUT     KSTAT   ;SET DTR LOW
014E E3                 XTHL
014F E3                 XTHL            ;WASTE TIME
0150 DB20               IN      KSTAT   ;GET STATUS
0152 E680               ANI     80H     ;TEST DSR BIT
0154 11F001             LXI     D,STR6
0157 C28501             JNZ     ERROR   ;DSR BIT LOW ERROR
                ;test RTS and CTS lines.
015A 3E27               MVI     A,27H
015C D320               OUT     KSTAT   ;SET DTR HIGH
015E E3                 XTHL
015F E3                 XTHL            ;WASTE TIME
0160 DB20               IN      KSTAT   ;GET STATUS
0162 E601               ANI     1
0164 11FE01             LXI     D,STR7
0167 CA8501             JZ      ERROR   ;CTS BIT ERROR
016A 3E07               MVI     A,7
016C D320               OUT     KSTAT   ;SET RTS LOW
016E D321               OUT     KDATA   ;TEST RTS
0170 E3                 XTHL
0171 E3                 XTHL            ;WASTE TIME
0172 DB20               IN      KSTAT   ;GET STATUS
0174 E601               ANI     1
0176 110D02             LXI     D,STR8
0179 C28501             JNZ     ERROR   ;CTS LOW ERROR
017C 112402             LXI     D,STR10
017F CDAB01             CALL    STRNG   ;IT PASSES!
0182 C30000             JMP     BOOT

                ;Output error message.
0185 CDAB01     ERROR:  CALL    STRNG
0188 111B02             LXI     D,STR9
018B CDAB01             CALL    STRNG
018E C30000             JMP     BOOT    ;EXIT

                ;Initialize 8251A chip.
0191 0603       INTZ:   MVI     B,3     ;SEND THREE
0193 AF                 XRA     A       ;  NULLS.
0194 D320       INTZ1:  OUT     KSTAT   ;SET MODE
0196 05                 DCR     B
0197 C29401             JNZ     INTZ1   ;REPEAT
019A 3E40               MVI     A,40H   ;SET TO MODE LEVEL
019C D320               OUT     KSTAT
019E 3E4E               MVI     A,4EH   ;8BITS/NO PARITY/1 STOP
01A0 D320               OUT     KSTAT
01A2 3E37               MVI     A,37H   ;CLEAR ERROR FLAGS
01A4 D320               OUT     KSTAT
```

```
01A6 3E27               MVI     A,27H   ;REMOVE CLEAR
01A8 D320               OUT     KSTAT
01AA C9                 RET

                ;Output a String to the console.
01AB 0E09       STRNG:  MVI     C,STNG
01AD C30500             JMP     BDOS

                ;Strings.
01B0 0D0A44414BSTR1:    DB      CR,LF,'DAK status$'
01BD 0D0A444156STR2:    DB      CR,LF,'DAV status$'
01CA 0D0A444154STR3:    DB      CR,LF,'DATA send$'
01D6 0D0A444154STR4:    DB      CR,LF,'DATA bit$'
01E1 0D0A445352STR5:    DB      CR,LF,'DSR/DTR high$'
01F0 0D0A445352STR6:    DB      CR,LF,'DSR/DTR low$'
01FE 0D0A525453STR7:    DB      CR,LF,'RTS/CTS high$'
020D 0D0A525453STR8:    DB      CR,LF,'RTS/CTS low$'
021B 206572726FSTR9:    DB      ' error',CR,LF,'$'
0224 0D0A2A2A2ASTR10:   DB      CR,LF,'****** ALL PASSES! ******$'

0240                    END
```

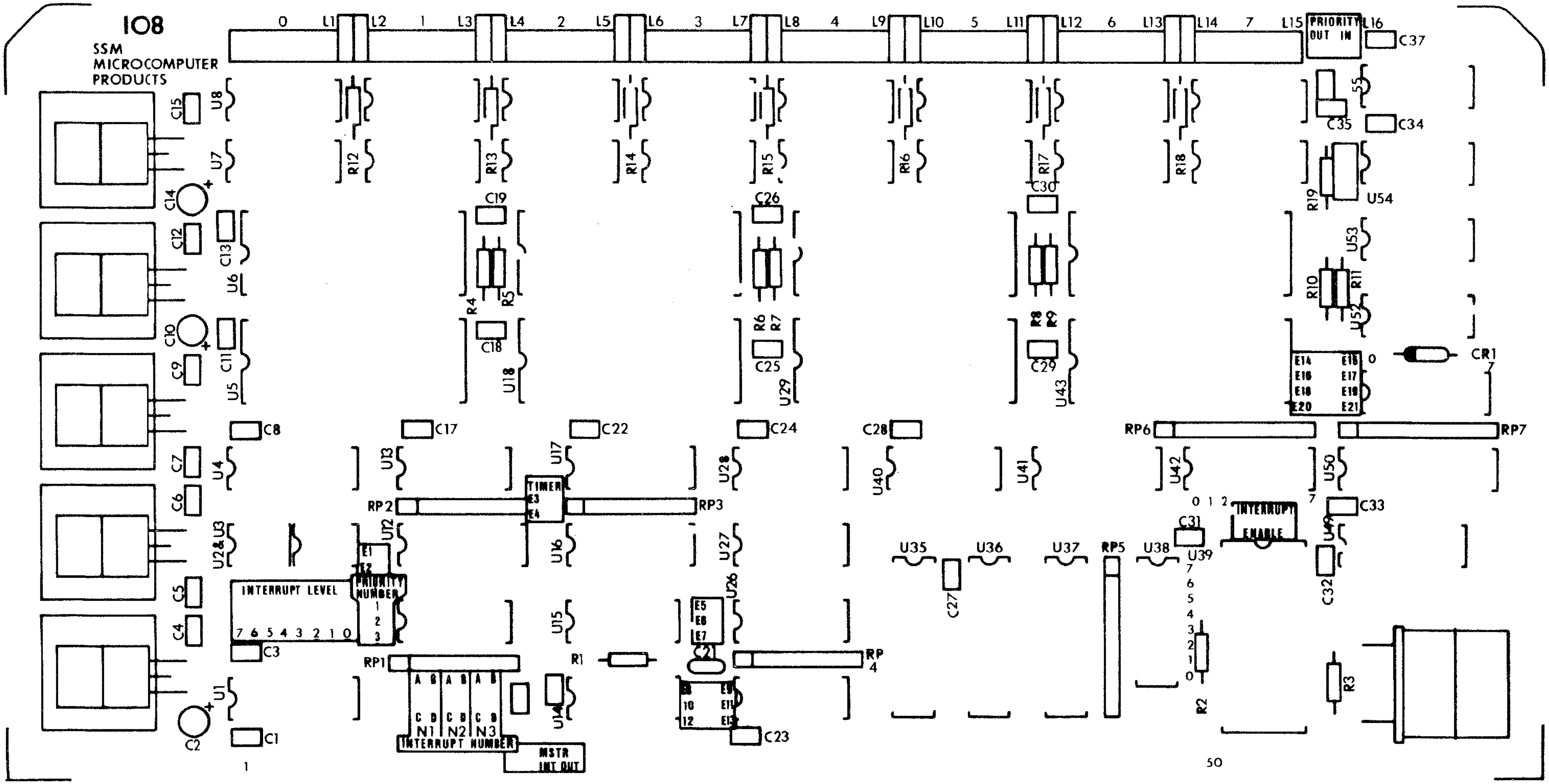
5.0 WARRANTY

SSM Microcomputer Products, Inc., warrants its products to be free from defects in materials and/or workmanship for a period of one (1) year for factory assembled boards. In the event of malfunction or other indication of failure attributable directly to faulty workmanship and/or material, then, upon return of the product (postage paid) to SSM at 2190 Paragon Drive, San Jose, CA 95131, "Attention: Warranty Claims Department", SSM will, at its option, repair or replace the defective part or parts to restore said product to proper operating condition. All such repairs and/or replacements shall be rendered by SSM without charge for parts or labor when the product is returned within the specified period of the date of purchase. This warranty applies only to the original purchaser.

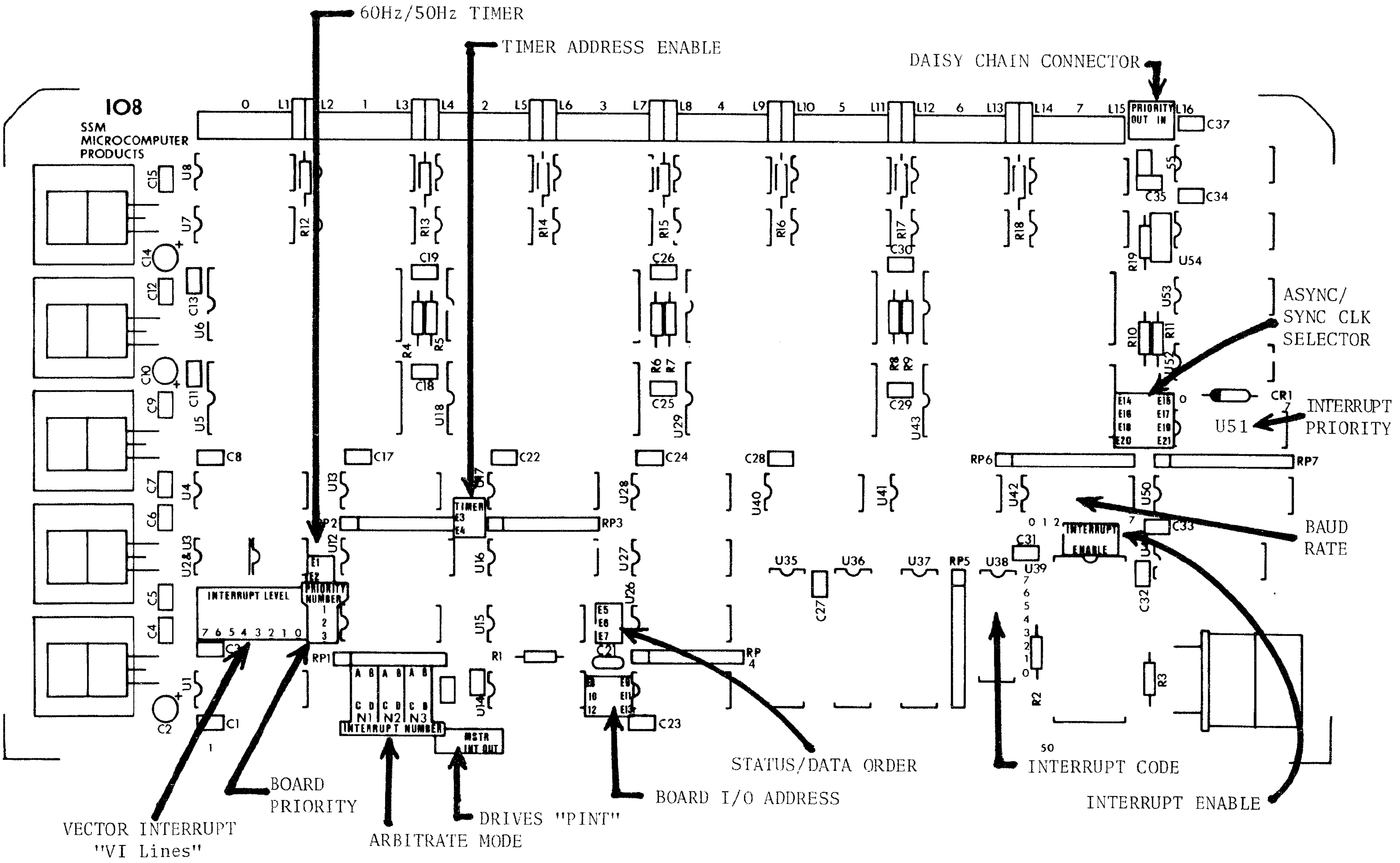
This warranty will not cover the failure of SSM products which at the discretion of SSM shall have resulted from accident, abuse, negligence, alteration, or misapplication of the product. While every effort has been made to provide clear and accurate technical information on the application of SSM products, SSM assumes no liability in any events which may arise from the use of said technical information.

This warranty is in lieu of all other warranties, expressed or implied, including warranties of mercantability and fitness for use. In no event will SSM be liable for incidental and consequential damages arising from or in any way connected with the use of its products. Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

IMPORTANT: Proof of purchase is necessary for products returned for repair under warranty. Before returning any product, please call our Customer Service Department for a return authorization number.



ASSEMBLY DRAWING



JUMPER DRAWING

IO8 PARTS LIST

Chip Pack

1	U40	74L$00
2	U52,53	74L$02
2	U14,55	74L$04
1	U26	74L$14
1	U28	74L$20
1	U54	74L$27
1	U15	74L$32
1	U17	74L$42
1	U16	74L$83A
2	U13,27	74L$107
2	U11,25	74L$136
2	U1,49	74L$148
1	U4	74L$155
2	U12,41	74L$163
1	U37	74L$240
2	U35,36	74L$244
1	U50	74L$373
8	U8,10,22,24,32, 34,46,48	1488
8	U7,9,20,23,31,33, 45,47	1489
1	U39	14411
2	U2,3	75453
8	U5,6,18,19,29,30, 43,44	8251A

Resistor/Diode Pack

1	R1	470 ohm, 1/4W	(yellow, violet, red)
7	R12-18	1.8K ohm, 1/4W	(brown, grey, red)
10	R2,4-10,19,11	2.7K ohm, 1/4W	(red, violet, red)
1	R3	15M ohm, 1/4W	(brown, green, blue)
4	RP1-4	2.7K x 7 SIP	
2	RP5,7	4.7K x 9 SIP	
1	RP6	100K x 9 SIP	
1	CR1	1N749	

Capacitor Pack

1	C21	220 pf disc radial
32	C1,3-9,11,13, 15-20,22-37	.1 uf monolithic .1" spacing radial
4	C2,10,12,14	4.7 uf DIP tantalum radial

Regulator Pack

3	VR1-3	7805
1	VR5	7812
1	VR4	7912
5	VR1-5	Small heatsinks
5	VR1-5	#6 hardware sets (nut, washer, bolt)

Connector Pack

3	U38,42,51	16-pin IC header
4		2 x 1 wire pin headers
1		3 x 1 wire pin header
2		3 x 2 wire pin headers
1		6 x 2 wire pin header
1		8 x 2 wire pin header
8		5 x 2 wire pin headers, 90°
1		2 x 1 wire pin header, 90°
1		4 x 2 wire pin header
17		Mini-jumpers

Miscellaneous Pack

1	Y1	1.8432 MHz xtal
14	L1-14	LED package
1		IO8 PC board
1		User's Manual
1		Warranty card

MOTOROLA Semiconductors
BOX 20912 • PHOENIX, ARIZONA 85036

MC14411

BIT RATE GENERATOR

The MC14411 bit rate generator is constructed with complementary MOS enhancement mode devices. It utilizes a frequency divider network to provide a wide range of output frequencies.

A crystal controlled oscillator is the clock source for the network. A two-bit address is provided to select one of four multiple output clock rates.

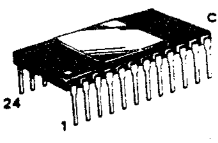
Applications include a selectable frequency source for equipment in the data communications market, such as teleprinters, printers, CRT terminals, and microprocessor systems.

- Single 5.0 Vdc (± 5%) Power Supply
- Internal Oscillator Crystal Controlled for Stability (1.8432 MHz)
- Sixteen Different Output Clock Rates
- 50% Output Duty Cycle
- Programmable Time Bases for One of Four Multiple Output Rates
- Buffered Outputs Compatible with Low Power TTL
- Noise Immunity = 45% of V_{DD} Typical
- Diode Protection on All Inputs
- External Clock May be Applied to Pin 21

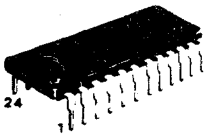
McMOS LSI

(LOW-POWER COMPLEMENTARY MOS)

BIT RATE GENERATOR



L SUFFIX
CERAMIC PACKAGE
CASE 716

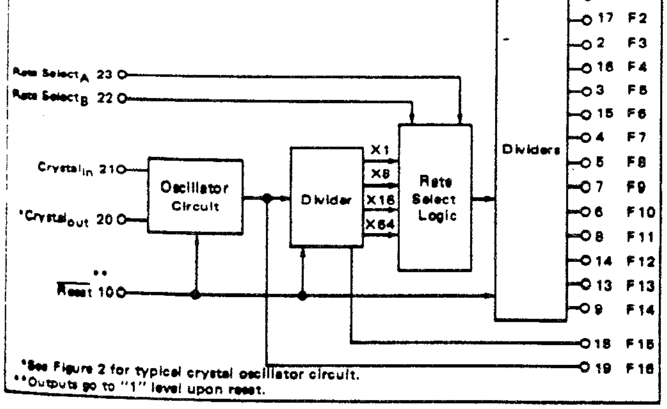


P SUFFIX
PLASTIC PACKAGE
CASE 709

MAXIMUM RATINGS (Voltages referenced to V_{SS}, Pin 12.)

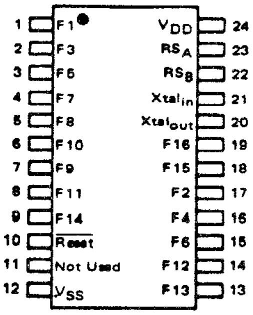
Rating	Symbol	Value	Unit
DC Supply Voltage Range	V_{DD}	5.25 to -0.5	Vdc
Input Voltage, All Inputs	V_{in}	V_{DD} + 0.5 to V_{SS} -0.5	Vdc
DC Current Drain per Pin	I	10	mAdc
Operating Temperature Range	T_A	-40 to +85	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C

BLOCK DIAGRAM



*See Figure 2 for typical crystal oscillator circuit.
**Outputs go to "1" level upon reset.

PIN ASSIGNMENT



V_{DD} = Pin 24
V_{SS} = Pin 12

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedence circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

ELECTRICAL CHARACTERISTICS

Characteristic	Symbol	V_{DD} Vdc	-40°C		25°C			+85°C		Unit
			Min	Max	Min	Typ	Max	Min	Max	
Supply Voltage	V_{DD}	–	4.75	5.25	4.75	5.0	5.25	4.75	5.25	Vdc
Output Voltage "0" Level	V_{out}	5.0	–	0.05	–	0	0.05	–	0.05	Vdc
"1" Level		5.0	4.95	–	4.95	5.0	–	4.95	–	Vdc
Input Voltage (V_O = 4.5 or 0.5 Vdc) "0" Level	V_{IL}	5.0	1.5	–	1.5	2.25	–	1.5	–	Vdc
(V_O = 0.5 or 4.5 Vdc) "1" Level	V_{IH}	5.0	1.5	–	1.5	2.25	–	1.5	–	Vdc
Output Drive Current (V_{OH} = 2.5 Vdc) Source	I_{OH}	5.0	-0.23	–	-0.20	-1.7	–	-0.16	–	mAdc
(V_{OL} = 0.4 Vdc) Sink	I_{OL}	5.0	0.23	–	0.20	0.78	–	0.16	–	mAdc
Input Current	I_{in}	–	–	±0.1	–	±0.00001	±0.1	–	±1.0	µAdc
Input Capacitance (V_{in} = 0)	C_{in}	–	–	–	–	5.0	–	–	–	pF
Quiescent Dissipation	P_Q	5.0	–	2.5	–	0.015	2.5	–	15	mW
Power Dissipation**† (Dynamic plus Quiescent) (C_L = 15 pF)	P_D	5.0	(P_D = (7.5 mW/MHz) f + P_Q							mW
Output Rise Time** t_r = (3.0 ns/pF) C_L + 25 ns	t_r	5.0	–	–	–	70	200	–	–	ns
Output Fall Time** t_f = (1.5 ns/pF) C_L + 47 ns	t_f	5.0	–	–	–	70	200	–	–	ns
Maximum Input Clock Frequency	f_{max}	5.0	–	–	–	1.8432	–	1.85	–	MHz

† For dissipation at different external load capacitance (C_L) refer to corresponding formula:

$$P_T(C_L) = P_D + 2.6 \times 10^{-3} (C_L - 15 \text{ pF}) V_{DD}{}^2 f$$

where: P_T, P_D in mW, C_L in pF, V_{DD} in Vdc, and f in MHz.

**The formula given is for the typical characteristics only.

TABLE 1 – OUTPUT CLOCK RATES

Rate Select		Rate
B	A	
0	0	X1
0	1	X8
1	0	X16
1	1	X64

Output Number	Output Rates (Hz)			
	X64	X16	X8	X1
F1	614.4 k	153.6 k	76.8 k	9600
F2	460.8 k	115.2 k	57.6 k	7200
F3	307.2 k	76.8 k	38.4 k	4800
F4	230.4 k	57.6 k	28.8 k	3600
F5	153.6 k	38.4 k	19.2 k	2400
F6	115.2 k	28.8 k	14.4 k	1800
F7	76.8 k	19.2 k	9600	1200
F8	38.4 k	9600	4800	600
F9	19.2 k	4800	2400	300
F10	12.8 k	3200	1600	200
F11	9600	2400	1200	150
F12	8613.2	2153.3	1076.6	134.5
F13	7035.5	1758.8	879.4	109.9
F14	4800	1200	600	75
F15	921.6 k	921.6 k	921.6 k	921.6 k
F16*	1.843M	1.843M	1.843M	1.843M

*F16 is buffered oscillator output.

FIGURE 1 – DYNAMIC SIGNAL WAVEFORMS

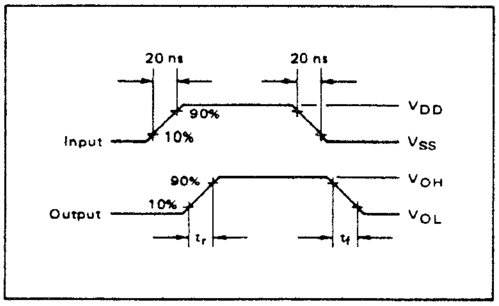
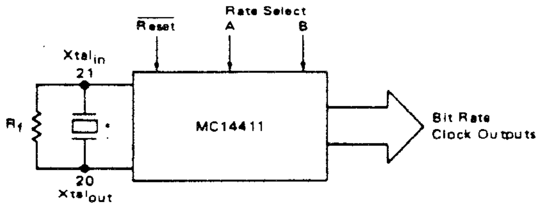


FIGURE 2 – TYPICAL CRYSTAL OSCILLATOR CIRCUIT



R_f = 15 MΩ±10%

CRYSTAL SPECIFICATION	
Crystal Mode	Parallel
Frequency	1.8432 MHz ±0.05% @ 13 pF
R_S	640 Ω max
C_0	7.0 pF max
Temperature Range	0 to 70°C
Test Level	1 mW
Test Set	TS – 330/TSM or Equivalent

*Suggested Crystal Suppliers: Tyco, CTS Knight and Motorola Crystal Products.

Circuit diagrams utilizing Motorola products are included as a means of illustrating typical semiconductor applications; consequently, complete information sufficient for construction purposes is not necessarily given. The information has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, such information does not convey to the purchaser of the semiconductor devices described any license under the patent rights of Motorola Inc. or others.

intel®

8251A/S2657
PROGRAMMABLE COMMUNICATION INTERFACE

- Synchronous and Asynchronous Operation
- Synchronous 5-8 Bit Characters; Internal or External Character Synchronization; Automatic Sync Insertion
- Asynchronous 5-8 Bit Characters; Clock Rate—1, 16 or 64 Times Baud Rate; Break Character Generation; 1, 1½, or 2 Stop Bits; False Start Bit Detection; Automatic Break Detect and Handling
- Synchronous Baud Rate — DC to 64K Baud
- Asynchronous Baud Rate — DC tc 19.2K Baud
- Full Duplex, Double Buffered, Transmitter and Receiver
- Error Detection — Parity, Overrun and Framing
- Fully Compatible with 8080/8085 CPU
- 28-Pin DIP Package
- All Inputs and Outputs are TTL Compatible
- Single +5V Supply
- Single TTL Clock

The Intel® 8251A is the enhanced version of the industry standard, Intel® 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART), designed for data communications with Intel's new high performance family of microprocessors such as the 8085. The 8251A is used as a peripheral device and is programmed by the CPU to operate using virtually any serial data transmission technique presently in use (including IBM "bi-sync"). The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission. Simultaneously, it can receive serial data streams and convert them into parallel data characters for the CPU. The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU. The CPU can read the complete status of the USART at any time. These include data transmission errors and control signals such as SYNDET, TxEMPTY. The chip is constructed using N-channel silicon gate technology.

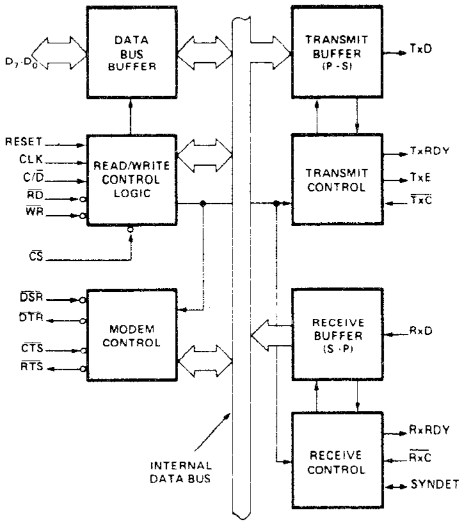


Figure 1. Block Diagram

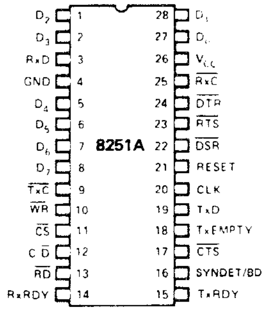


Figure 2. Pin Configuration

FEATURES AND ENHANCEMENTS

8251A is an advanced design of the industry standard USART, the Intel® 8251. The 8251A operates with an extended range of Intel microprocessors that includes the new 8085 CPU and maintains compatibility with the 8251. Familiarization time is minimal because of compatibility and involves only knowing the additional features and enhancements, and reviewing the AC and DC specifications of the 8251A.

The 8251A incorporates all the key features of the 8251 and has the following additional features and enhancements:

- 8251A has double-buffered data paths with separate I/O registers for control, status, Data In, and Data Out, which considerably simplifies control programming and minimizes CPU overhead.
- In asynchronous operations, the Receiver detects and handles "break" automatically, relieving the CPU of this task.
- A refined Rx initialization prevents the Receiver from starting when in "break" state, preventing unwanted interrupts from a disconnected USART.
- At the conclusion of a transmission, TxD line will always return to the marking state unless SBRK is programmed.
- Tx Enable logic enhancement prevents a Tx Disable command from halting transmission until all data previously written has been transmitted. The logic also prevents the transmitter from turning off in the middle of a word.
- When External Sync Detect is programmed, Internal Sync Detect is disabled, and an External Sync Detect status is provided via a flip-flop which clears itself upon a status read.
- Possibility of false sync detect is minimized by ensuring that if double character sync is programmed, the characters be contiguously detected and also by clearing the Rx register to all ones whenever Enter Hunt command is issued in Sync mode.
- As long as the 8251A is not selected, the $\overline{RD}$ and $\overline{WR}$ do not affect the internal operation of the device.
- The 8251A Status can be read at any time but the status update will be inhibited during status read.
- The 8251A is free from extraneous glitches and has enhanced AC and DC characteristics, providing higher speed and better operating margins.
- Synchronous Baud rate from DC to 64K.
- Fully compatible with Intel's new industry standard, the MCS-85.

FUNCTIONAL DESCRIPTION

General

The 8251A is a Universal Synchronous/Asynchronous Receiver/Transmitter designed specifically for the 80/85 Microcomputer Systems. Like other I/O devices in a Microcomputer System, its functional configuration is programmed by the system's software for maximum flexibility. The 8251A can support virtually any serial data technique currently in use (including IBM "bi-sync").

In a communication environment an interface device must convert parallel format system data into serial format for transmission and convert incoming serial format data into parallel system data for reception. The interface device must also delete or insert bits or characters that are functionally unique to the communication technique. In essence, the interface should appear "transparent" to the CPU, a simple input or output of byte-oriented system data.

Data Bus Buffer

This 3-state, bidirectional, 8-bit buffer is used to interface the 8251A to the system Data Bus. Data is transmitted or received by the buffer upon execution of INput or OUTput instructions of the CPU. Control words, Command words and Status information are also transferred through the Data Bus Buffer. The command status and data in, and data out are separate 8-bit registers to provide double buffering.

This functional block accepts inputs from the system Control bus and generates control signals for overall device operation. It contains the Control Word Register and Command Word Register that store the various control formats for the device functional definition.

RESET (Reset)

A "high" on this input forces the 8251A into an "Idle" mode. The device will remain at "Idle" until a new set of control words is written into the 8251A to program its functional definition. Minimum RESET pulse width is 6 t_{CY} (clock must be running).

CLK (Clock)

The CLK input is used to generate internal device timing and is normally connected to the Phase 2 (TTL) output of the 8224 Clock Generator. No external inputs or outputs are referenced to CLK but the frequency of CLK must be greater than 30 times the Receiver or Transmitter data bit rates.

$\overline{WR}$ (Write)

A "low" on this input informs the 8251A that the CPU is writing data or control words to the 8251A.

$\overline{RD}$ (Read)

A "low" on this input informs the 8251A that the CPU is reading data or status information from the 8251A.

$C/\overline{D}$ (Control/Data)

This input, in conjunction with the $\overline{WR}$ and $\overline{RD}$ inputs, informs the 8251A that the word on the Data Bus is either a data character, control word or status information.
1 = CONTROL/STATUS 0 = DATA

$\overline{CS}$ (Chip Select)

A "low" on this input selects the 8251A. No reading or writing will occur unless the device is selected. When $\overline{CS}$ is high, the Data Bus in the float state and $\overline{RD}$ and $\overline{WR}$ will have no effect on the chip.

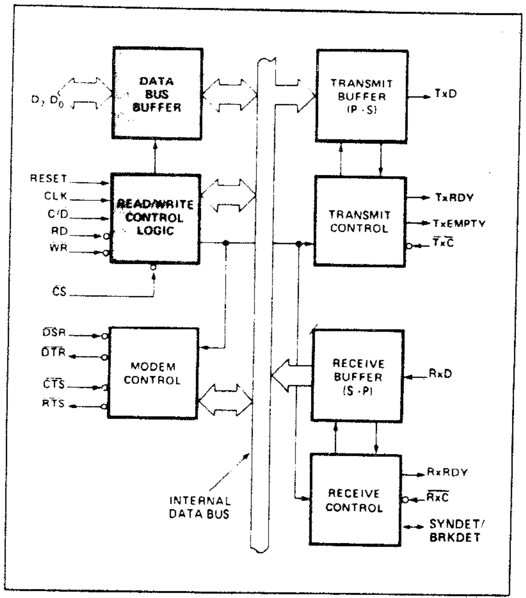


Figure 3. 8251A Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

$C/\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
0	0	1	0	8251A DATA ⇒ DATA BUS
0	1	0	0	DATA BUS ⇒ 8251A DATA
1	0	1	0	STATUS ⇒ DATA BUS
1	1	0	0	DATA BUS ⇒ CONTROL
X	1	1	0	DATA BUS ⇒ 3-STATE
X	X	X	1	DATA BUS ⇒ 3-STATE

Modem Control

The 8251A has a set of control inputs and outputs that can be used to simplify the interface to almost any Modem. The Modem control signals are general purpose in nature and can be used for functions other than Modem control, if necessary.

$\overline{\text{DSR}}$ (Data Set Ready)

The $\overline{\text{DSR}}$ input signal is a general purpose, 1-bit inverting input port. Its condition can be tested by the CPU using a Status Read operation. The $\overline{\text{DSR}}$ input is normally used to test Modem conditions such as Data Set Ready.

$\overline{\text{DTR}}$ (Data Terminal Ready)

The $\overline{\text{DTR}}$ output signal is a general purpose, 1-bit inverting output port. It can be set "low" by programming the appropriate bit in the Command Instruction word. The $\overline{\text{DTR}}$ output signal is normally used for Modem control such as Data Terminal Ready or Rate Select.

$\overline{\text{RTS}}$ (Request to Send)

The $\overline{\text{RTS}}$ output signal is a general purpose, 1-bit inverting output port. It can be set "low" by programming the appropriate bit in the Command Instruction word. The $\overline{\text{RTS}}$ output signal is normally used for Modem control such as Request to Send.

$\overline{\text{CTS}}$ (Clear to Send)

A "low" on this input enables the 8251A to transmit serial data if the Tx Enable bit in the Command byte is set to a "one." If either a Tx Enable off or CTS off condition occurs while the Tx is in operation, the Tx will transmit all the data in the USART, written prior to Tx Disable command before shutting down. On the 8251A/S2657 if CTS off or Tx Enable off condition occurs before the last character written appears in the serial bit stream, that character will be transmitted again upon CTS on or Tx Enable on condition.

Transmitter Buffer

The Transmitter Buffer accepts parallel data from the Data Bus Buffer, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin on the falling edge of $\overline{\text{TxC}}$. The transmitter will begin transmission upon being enabled if $\overline{\text{CTS}}$ = 0. The TxD line will be held in the marking state immediately upon a master Reset or when Tx Enable/$\overline{\text{CTS}}$ off or TxEMPTY.

Transmitter Control

The transmitter Control manages all activities associated with the transmission of serial data. It accepts and issues signals both externally and internally to accomplish this function.

TxRDY (Transmitter Ready)

This output signals the CPU that the transmitter is ready to accept a data character. The TxRDY output pin can be used as an interrupt to the system, since it is masked by Tx Disabled, or, for Polled operation, the CPU can check TxRDY using a Status Read operation. TxRDY is automatically reset by the leading edge of $\overline{\text{WR}}$ when a data character is loaded from the CPU.

Note that when using the Polled operation, the TxRDY status bit is *not* masked by Tx Enabled, but will only indicate the Empty/Full Status of the Tx Data Input Register.

TxE (Transmitter Empty)

When the 8251A has no characters to transmit, the TxEMPTY output will go "high". It resets automatically upon receiving a character from the CPU if the transmitter is enabled. TxEMPTY can be used to indicate the end of a transmission mode, so that the CPU "knows" when to "turn the line around" in the half-duplexed operational mode.

In SYNChronous mode, a "high" on this output indicates that a character has not been loaded and the SYNC character or characters are about to be or are being transmitted automatically as "fillers". TxEMPTY does not go low when the SYNC characters are being shifted out.

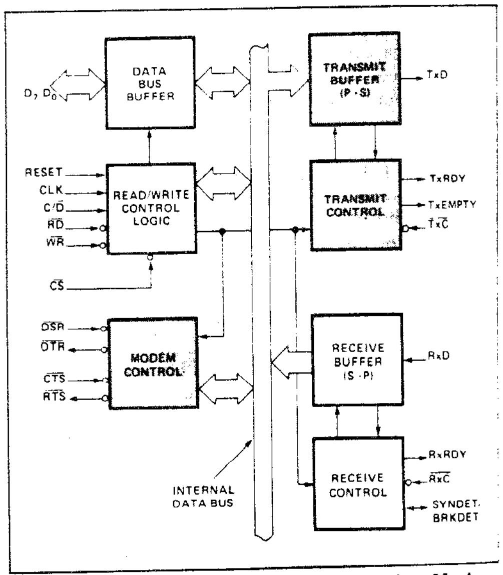


Figure 4. 8251A Block Diagram Showing Modem and Transmitter Buffer and Control Functions

$\overline{\text{TxC}}$ (Transmitter Clock)

The Transmitter Clock controls the rate at which the character is to be transmitted. In the Synchronous transmission mode, the Baud Rate (1x) is equal to the $\overline{\text{TxC}}$ frequency. In Asynchronous transmission mode the baud rate is a fraction of the actual $\overline{\text{TxC}}$ frequency. A portion of the mode instruction selects this factor; it can be 1, 1/16 or 1/64 the $\overline{\text{TxC}}$.

For Example:

If Baud Rate equals 110 Baud,
$\overline{\text{TxC}}$ equals 110 Hz (1x)
$\overline{\text{TxC}}$ equals 1.76 kHz (16x)
$\overline{\text{TxC}}$ equals 7.04 kHz (64x).

The falling edge of $\overline{\text{TxC}}$ shifts the serial data out of the 8251A.

Receiver Buffer

The Receiver accepts serial data, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU. Serial data is input to RxD pin, and is clocked in on the rising edge of $\overline{RxC}$.

Receiver Control

This functional block manages all receiver-related activities which consist of the following features:

The RxD initialization circuit prevents the 8251A from mistaking an unused input line for an active low data line in the "break condition". Before starting to receive serial characters on the RxD line, a valid "1" must first be detected after a chip master Reset. Once this has been determined, a search for a valid low (Start bit) is enabled. This feature is only active in the asynchronous mode, and is only done once for each master Reset.

The False Start bit detection circuit prevents false starts due to a transient noise spike by first detecting the falling edge and then strobing the nominal center of the Start bit (RxD = low).

The Parity Toggle F/F and Parity Error F/F circuits are used for parity error detection and set the corresponding status bit.

The Framing Error Flag F/F is set if the Stop bit is absent at the end of the data byte (asynchronous mode), and also sets the corresponding status bit.

RxRDY (Receiver Ready)

This output indicates that the 8251A contains a character that is ready to be input to the CPU. Rx RDY can be connected to the interrupt structure of the CPU or, for Polled operation, the CPU can check the condition of RxRDY using a Status Read operation.

Rx Enable off both masks and holds RxRDY in the Reset Condition. For Asynchronous mode, to set RxRDY, the Receiver must be Enabled to sense a Start Bit and a complete character must be assembled and transferred to the Data Output Register. For Synchronous mode, to set RxRDY, the Receiver must be enabled and a character must finish assembly and be transferred to the Data Output Register.

Failure to read the received character from the Rx Data Output Register prior to the assembly of the next Rx Data character will set overrun condition error and the previous character will be written over and lost. If the Rx Data is being read by the CPU when the internal transfer is occurring, overrun error will be set and the old character will be lost.

$\overline{RxC}$ (Receiver Clock)

The Receiver Clock controls the rate at which the character is to be received. In Synchronous Mode, the Baud Rate (1x) is equal to the actual frequency of $\overline{RxC}$. In Asynchronous Mode, the Baud Rate is a fraction of the actual $\overline{RxC}$ frequency. A portion of the mode instruction selects this factor; 1, 1/16 or 1/64 the $\overline{RxC}$.

For Example:

Baud Rate equals 300 Baud, if
$\overline{RxC}$ equals 300 Hz (1x)
$\overline{RxC}$ equals 4800 Hz (16x)
$\overline{RxC}$ equals 19.2 kHz (64x).

Baud Rate equals 2400 Baud, if
$\overline{RxC}$ equals 2400 Hz (1x)
$\overline{RxC}$ equals 38.4 kHz (16x)
$\overline{RxC}$ equals 153.6 kHz (64x).

Data is sampled into the 8251A on the rising edge of $\overline{RxC}$.

NOTE: In most communications systems, the 8251A will be handling both the transmission and reception operations of a single link. Consequently, the Receive and Transmit Baud Rates will be the same. Both $\overline{TxC}$ and $\overline{RxC}$ will require identical frequencies for this operation and can be tied together and connected to a single frequency source (Baud Rate Generator) to simplify the interface.

SYNDET (SYNC Detect)/BRKDET (Break Detect)

This pin is used in SYNChronous Mode for SYNDET and may be used as either input or output, programmable through the Control Word. It is reset to output mode low upon RESET. When used as an output (internal Sync mode), the SYNDET pin will go "high" to indicate that the 8251A has located the SYNC character in the Receive mode. If the 8251A is programmed to use double Sync characters (bi-sync), then SYNDET will go "high" in the middle of the last bit of the second Sync character. SYNDET is automatically reset upon a Status Read operation.

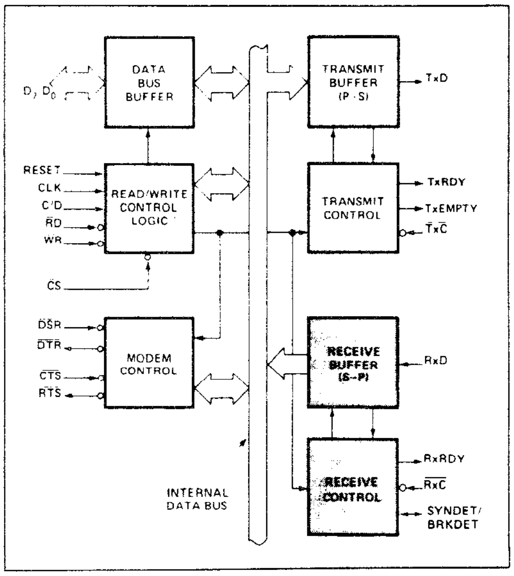


Figure 5. 8251A Block Diagram Showing Receiver Buffer and Control Functions

When used as an input (external SYNC detect mode), a positive going signal will cause the 8251A to start assembling data characters on the rising edge of the next $\overline{RxC}$. Once in SYNC, the "high" input signal can be removed. When External SYNC Detect is programmed, the Internal SYNC Detect is disabled.

BREAK DETECT (Async Mode Only)

This output will go high whenever the receiver remains low through two consecutive stop bit sequences (including the start bits, data bits, and parity bits). Break Detect may also be read as a Status bit. It is reset only upon a master chip Reset or Rx Data returning to a "one" state.

NOTE: On the 8251A/S2657, if the RxData returns to a "one" state during the last bit of the next character after the break, break detect will latch-up, and the device must be cleared by a Chip Reset.

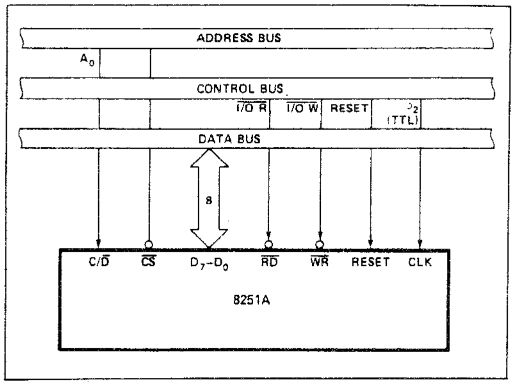


Figure 6. 8251A Interface to 8080 Standard System Bus

DETAILED OPERATION DESCRIPTION

General

The complete functional definition of the 8251A is programmed by the system's software. A set of control words must be sent out by the CPU to initialize the 8251A to support the desired communications format. These control words will program the: BAUD RATE, CHARACTER LENGTH, NUMBER OF STOP BITS, SYNCHRONOUS or ASYNCHRONOUS OPERATION, EVEN/ODD/OFF PARITY, etc. In the Synchronous Mode, options are also provided to select either internal or external character synchronization.

Once programmed, the 8251A is ready to perform its communication functions. The TxRDY output is raised "high" to signal the CPU that the 8251A is ready to receive a data character from the CPU. This output (TxRDY) is reset automatically when the CPU writes a character into the 8251A. On the other hand, the 8251A receives serial data from the MODEM or I/O device. Upon receiving an entire character, the RxRDY output is raised "high" to signal the CPU that the 8251A has a complete character ready for the CPU to fetch. RxRDY is reset automatically upon the CPU data read operation.

The 8251A cannot begin transmission until the Tx Enable (Transmitter Enable) bit is set in the Command Instruction and it has received a Clear To Send ($\overline{CTS}$) input. The TxD output will be held in the marking state upon Reset.

Programming the 8251A

Prior to starting data transmission or reception, the 8251A must be loaded with a set of control words generated by the CPU. These control signals define the complete functional definition of the 8251A and must immediately follow a Reset operation (internal or external).

The control words are split into two formats:

1. Mode Instruction
2. Command Instruction

Mode Instruction

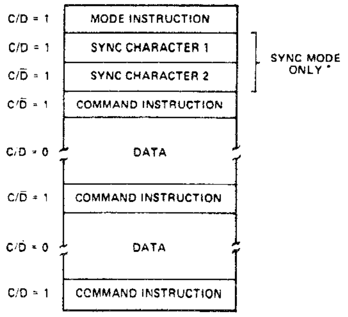
This format defines the general operational characteristics of the 8251A. It must follow a Reset operation (internal or external). Once the Mode Instruction has been written into the 8251A by the CPU, SYNC characters or Command Instructions may be inserted.

Command Instruction

This format defines a status word that is used to control the actual operation of the 8251A.

Both the Mode and Command Instructions must conform to a specified sequence for proper device operation. The Mode Instruction must be inserted immediately following a Reset operation, prior to using the 8251A for data communication.

All control words written into the 8251A after the Mode Instruction will load the Command Instruction. Command Instructions can be written into the 8251A at any time in the data block during the operation of the 8251A. To return to the Mode Instruction format, the master Reset bit in the Command Instruction word can be set to initiate an internal Reset operation which automatically places the 8251A back into the Mode Instruction format. Command Instructions must follow the Mode Instructions or Sync characters.



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

Figure 7. Typical Data Block

Mode Instruction Definition

The 8251A can be used for either Asynchronous or Synchronous data communication. To understand how the Mode Instruction defines the functional operation of the 8251A, the designer can best view the device as two separate components sharing the same package, one Asynchronous the other Synchronous. The format definition can be changed only after a master chip Reset. For explanation purposes the two formats will be isolated.

NOTE: When parity is enabled it is not considered as one of the data bits for the purpose of programming the word length. The actual parity bit received on the Rx Data line cannot be read on the Data Bus. In the case of a programmed character length of less than 8 bits, the least significant Data Bus bits will hold the data; unused bits are "don't care" when writing data to the 8251A, and will be "zeros" when reading the data from the 8251A.

Asynchronous Mode (Transmission)

Whenever a data character is sent by the CPU the 8251A automatically adds a Start bit (low level) followed by the data bits (least significant bit first), and the programmed number of Stop bits to each character. Also, an even or odd Parity bit is inserted prior to the Stop bit(s), as defined by the Mode Instruction. The character is then transmitted as a serial data stream on the TxD output. The serial data is shifted out on the falling edge of $\overline{TxC}$ at a rate equal to 1, 1/16, or 1/64 that of the $\overline{TxC}$, as defined by the Mode Instruction. BREAK characters can be continuously sent to the TxD if commanded to do so.

When no data characters have been loaded into the 8251A the TxD output remains "high" (marking) unless a Break (continuously low) has been programmed.

Asynchronous Mode (Receive)

The RxD line is normally high. A falling edge on this line triggers the beginning of a START bit. The validity of this START bit is checked by again strobing this bit at its nominal center (16X or 64X mode only). If a low is detected again, it is a valid START bit, and the bit counter will start counting. The bit counter thus locates the center of the data bits, the parity bit (if it exists) and the stop bits. If parity error occurs, the parity error flag is set. Data and parity bits are sampled on the RxD pin with the rising edge of $\overline{RxC}$. If a low level is detected as the STOP bit, the Framing Error flag will be set. The STOP bit signals the end of a character. Note that the *receiver* requires only *one* stop bit, regardless of the number of stop bits programmed. This character is then loaded into the parallel I/O buffer of the 8251A. The RxRDY pin is raised to signal the CPU that a character is ready to be fetched. If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer, and the OVERRUN Error flag is raised (thus the previous character is lost). All of the error flags can be reset by an Error Reset Instruction. The occurrence of any of these errors will not affect the operation of the 8251A.

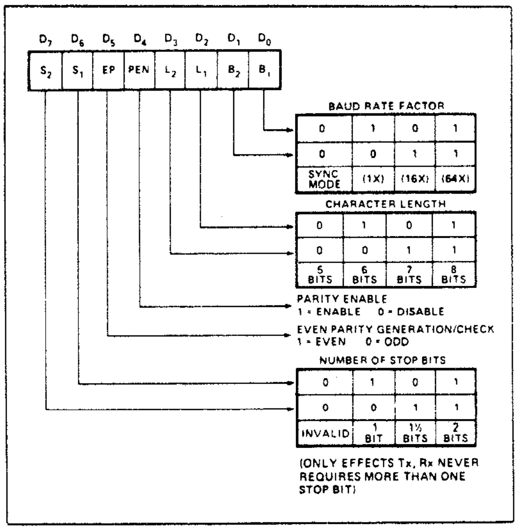

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
S_2 S_1 EP PEN L_2 L_1 B_2 B_1
BAUD RATE FACTOR
0 1 0 1
0 0 1 1
SYNC MODE (1X) (16X) (64X)
CHARACTER LENGTH
0 1 0 1
0 0 1 1
5 BITS 6 BITS 7 BITS 8 BITS
PARITY ENABLE
1 = ENABLE 0 = DISABLE
EVEN PARITY GENERATION/CHECK
1 = EVEN 0 = ODD
NUMBER OF STOP BITS
0 1 0 1
0 0 1 1
INVALID 1 BIT 1½ BITS 2 BITS
(ONLY EFFECTS Tx, Rx NEVER REQUIRES MORE THAN ONE STOP BIT)

Figure 8. Mode Instruction Format, Asynchronous Mode

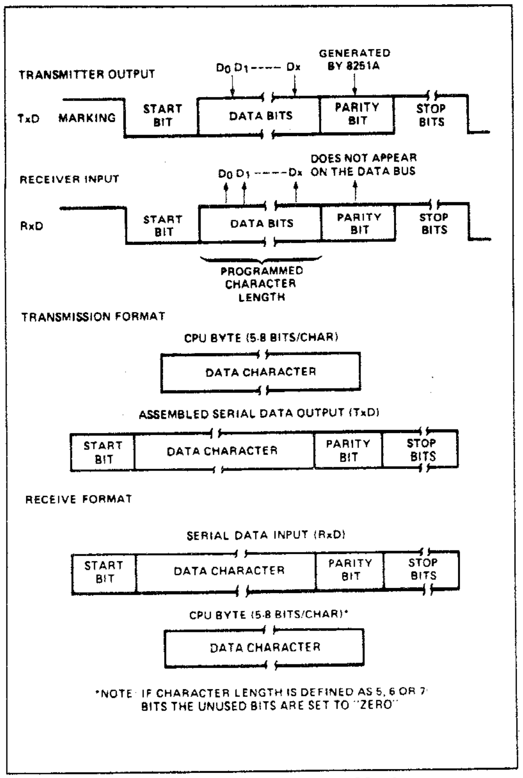
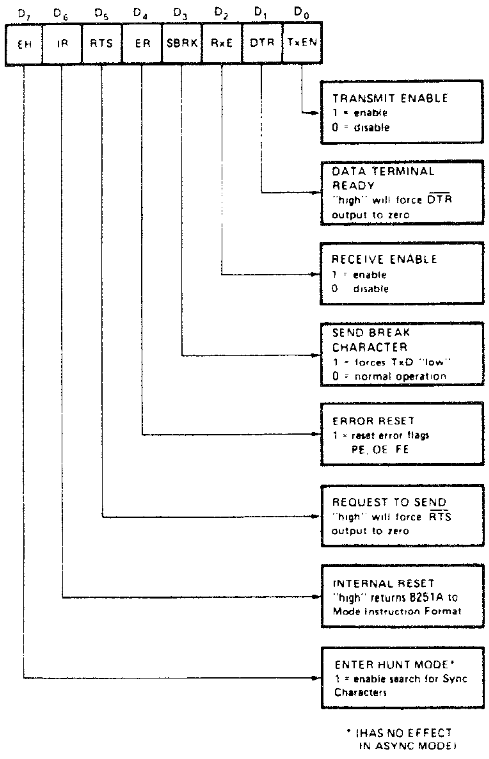


Figure 9. Asynchronous Mode

COMMAND INSTRUCTION DEFINITION

Once the functional definition of the 8251A has been programmed by the Mode Instruction and the Sync Characters are loaded (if in Sync Mode) then the device is ready to be used for data communication. The Command Instruction controls the actual operation of the selected format. Functions such as: Enable Transmit/Receive, Error Reset and Modem Controls are provided by the Command Instruction.

Once the Mode Instruction has been written into the 8251A and Sync characters inserted, if necessary, then all further "control writes" ($C/\overline{D}$ = 1) will load a Command Instruction. A Reset Operation (internal or external) will return the 8251A to the Mode Instruction format.



Note: Error Reset must be performed whenever RxEnable and Enter Hunt are programmed.

Figure 12. Command Instruction Format

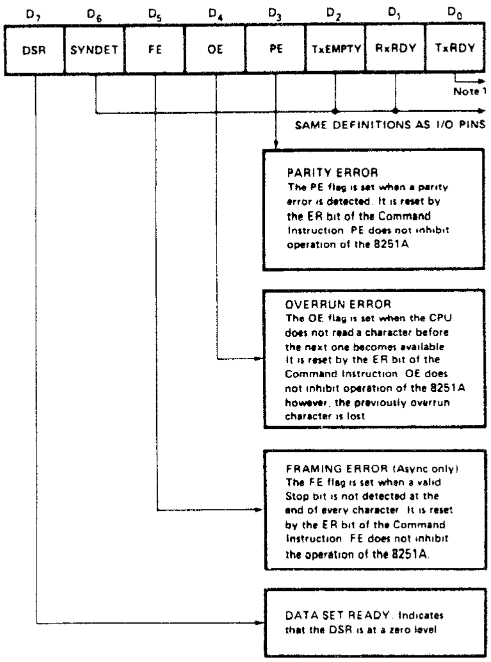
STATUS READ DEFINITION

In data communication systems it is often necessary to examine the "status" of the active device to ascertain if errors have occurred or other conditions that require the processor's attention. The 8251A has facilities that allow the programmer to "read" the status of the device at any time during the functional operation. (The status update is inhibited during status read).

A normal "read" command is issued by the CPU with $C/\overline{D}$ = 1 to accomplish this function.

Some of the bits in the Status Read Format have identical meanings to external output pins so that the 8251A can be used in a completely Polled environment or in an interrupt driven environment. TxRDY is an exception.

Note that status update can have a maximum delay of 28 clock periods from the actual event affecting the status.



D_7
D_6
D_5
D_4
D_3
D_2
D_1
D_0
DSR
SYNDET
FE
OE
PE
TxEMPTY
RxRDY
TxRDY
Note 1
SAME DEFINITIONS AS I/O PINS
PARITY ERROR
The PE flag is set when a parity error is detected. It is reset by the ER bit of the Command Instruction PE does not inhibit operation of the 8251A
OVERRUN ERROR
The OE flag is set when the CPU does not read a character before the next one becomes available It is reset by the ER bit of the Command Instruction OE does not inhibit operation of the 8251A however, the previously overrun character is lost
FRAMING ERROR (Async only)
The FE flag is set when a valid Stop bit is not detected at the end of every character It is reset by the ER bit of the Command Instruction FE does not inhibit the operation of the 8251A.
DATA SET READY. Indicates that the DSR is at a zero level

Note 1: TxRDY status bit has different meanings from the TxRDY output pin. The former is not conditioned by $\overline{CTS}$ and TxEN; the latter is conditioned by both $\overline{CTS}$ and TxEN.

i.e. TxRDY status bit = DB Buffer Empty

TxRDY pin out = DB Buffer Empty ·(CTS=0)·(TxEN=1)

Figure 13. Status Read Format

APPLICATIONS OF THE 8251A

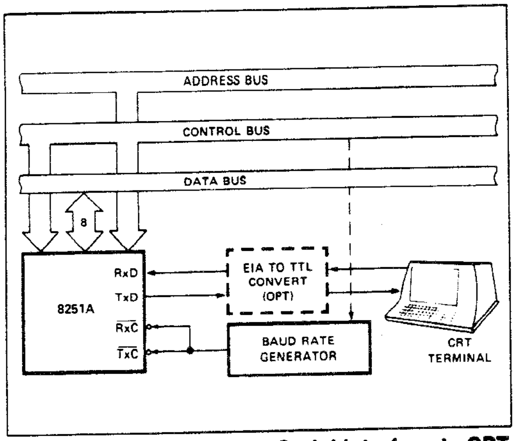


Figure 14. Asynchronous Serial Interface to CRT Terminal, DC—9600 Baud

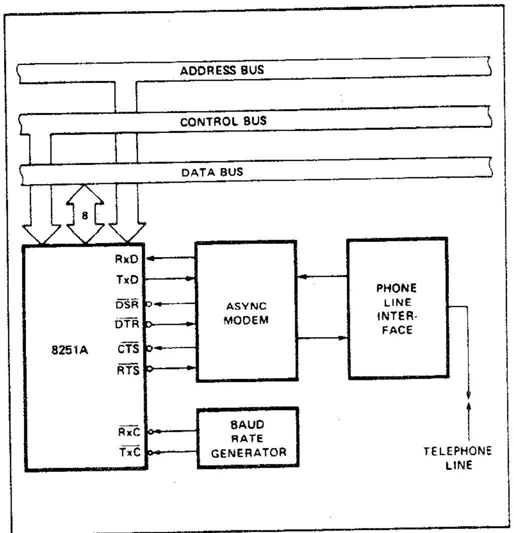


Figure 16. Asynchronous Interface to Telephone Lines

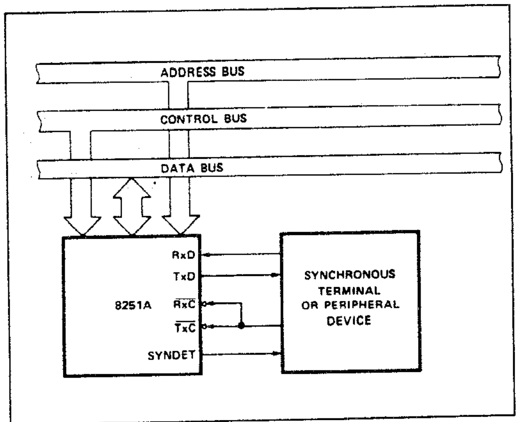


Figure 15. Synchronous Interface to Terminal or Peripheral Device

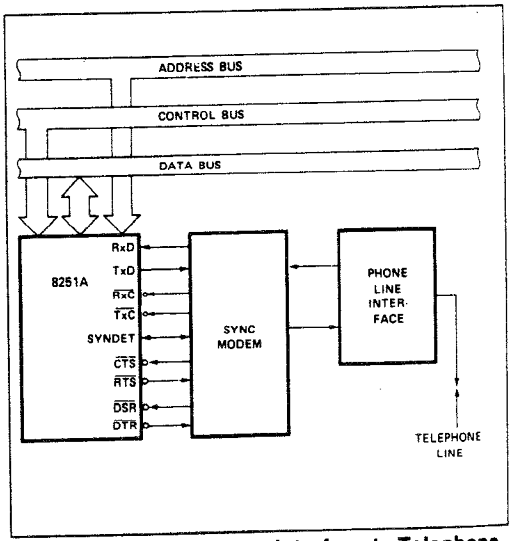


Figure 17. Synchronous Interface to Telephone Lines

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias	0°C to 70°C
Storage Temperature	−65°C to +150°C
Voltage On Any Pin With Respect to Ground	−0.5V to +7V
Power Dissipation	1 Watt

**NOTICE:* *Stresses 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.*

D.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5.0V ±5%, GND = 0V)

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.2	V_{CC}	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -400 μA
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} TO 0.45V
I_{IL}	Input Leakage		±10	μA	V_{IN} = V_{CC} TO 0.45V
I_{CC}	Power Supply Current		100	mA	All Outputs = High

CAPACITANCE (T_A = 25°C, V_{CC} = GND = 0V)

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance		10	pF	fc = 1MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins returned to GND

A.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5.0V ±5%, GND = 0V)

Bus Parameters (Note 1)

READ CYCLE

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AR}	Address Stable Before $\overline{READ}$ ($\overline{CS}$, C/$\overline{D}$)	50		ns	Note 2
t_{RA}	Address Hold Time for $\overline{READ}$ ($\overline{CS}$, C/$\overline{D}$)	50		ns	Note 2
t_{RR}	$\overline{READ}$ Pulse Width	250		ns	
t_{RD}	Data Delay from $\overline{READ}$		250	ns	3, C_L = 150 pF
t_{DF}	$\overline{READ}$ to Data Floating	10	100	ns	

A.C. CHARACTERISTICS (Continued)

WRITE CYCLE

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AW}	Address Stable Before $\overline{\text{WRITE}}$	50		ns	
t_{WA}	Address Hold Time for $\overline{\text{WRITE}}$	50		ns	
t_{WW}	$\overline{\text{WRITE}}$ Pulse Width	250		ns	
t_{DW}	Data Set Up Time for $\overline{\text{WRITE}}$	150		ns	
t_{WD}	Data Hold Time for $\overline{\text{WRITE}}$	50		ns	
t_{RV}	Recovery Time Between WRITES	6		t_{CY}	Note 4

OTHER TIMINGS

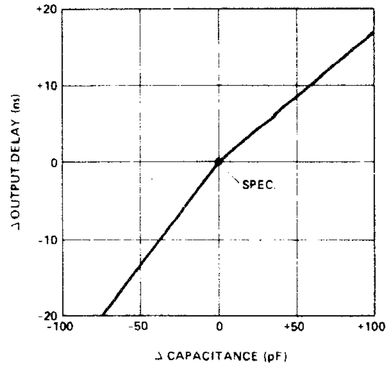
Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{CY}	Clock Period	320	1350	ns	Notes 5, 6
t_{ϕ}	Clock High Pulse Width	140	t_{CY}-90	ns	
$t_{\overline{\phi}}$	Clock Low Pulse Width	90		ns	
t_R, t_F	Clock Rise and Fall Time		20	ns	
t_{DTx}	TxD Delay from Falling Edge of $\overline{\text{TxC}}$		1	µs	
f_{Tx}	Transmitter Input Clock Frequency 1x Baud Rate 16x Baud Rate 64x Baud Rate	 DC DC DC	 64 310 615	 kHz kHz kHz	
t_{TPW}	Transmitter Input Clock Pulse Width 1x Baud Rate 16x and 64x Baud Rate	 12 1		 t_{CY} t_{CY}	
t_{TPD}	Transmitter Input Clock Pulse Delay 1x Baud Rate 16x and 64x Baud Rate	 15 3		 t_{CY} t_{CY}	
f_{Rx}	Receiver Input Clock Frequency 1x Baud Rate 16x Baud Rate 64x Baud Rate	 DC DC DC	 64 310 615	 kHz kHz kHz	
t_{RPW}	Receiver Input Clock Pulse Width 1x Baud Rate 16x and 64x Baud Rate	 12 1		 t_{CY} t_{CY}	
t_{RPD}	Receiver Input Clock Pulse Delay 1x Baud Rate 16x and 64x Baud Rate	 15 3		 t_{CY} t_{CY}	
t_{TxRDY}	TxRDY Pin Delay from Center of last Bit		8	t_{CY}	Note 7
$t_{TxRDY\ CLEAR}$	TxRDY ↓ from Leading Edge of $\overline{\text{WR}}$		6	t_{CY}	Note 7
t_{RxRDY}	RxRDY Pin Delay from Center of last Bit		24	t_{CY}	Note 7
$t_{RxRDY\ CLEAR}$	RxRDY ↓ from Leading Edge of $\overline{\text{RD}}$		6	t_{CY}	Note 7
t_{IS}	Internal SYNDET Delay from Rising Edge of $\overline{\text{RxC}}$		24	t_{CY}	Note 7
t_{ES}	External SYNDET Set-Up Time Before Falling Edge of $\overline{\text{RxC}}$	16		t_{CY}	Note 7
$t_{TxEMPTY}$	TxEMPTY Delay from Center of Last Bit	20		t_{CY}	Note 7
t_{WC}	Control Delay from Rising Edge of WRITE (TxEn, $\overline{\text{DTR}}$, $\overline{\text{RTS}}$)	8		t_{CY}	Note 7
t_{CR}	Control to READ Set-Up Time ($\overline{\text{DSR}}$, $\overline{\text{CTS}}$)	20		t_{CY}	Note 7

A.C. CHARACTERISTICS (Continued)

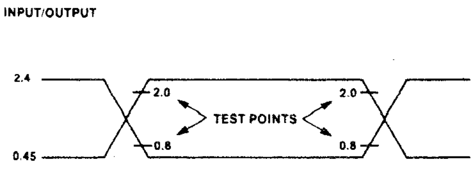
NOTES:

1. AC timings measured V_{OH} = 2.0, V_{OL} = 0.8, and with load circuit of Figure 1.
2. Chip Select (CS) and Command/Data (C/D) are considered as Addresses.
3. Assumes that Address is valid before R_D↓.
4. This recovery time is for Mode Initialization only. Write Data is allowed only when TxRDY = 1. Recovery Time between Writes for Asynchronous Mode is 8 t_{CY} and for Synchronous Mode is 16 t_{CY}.
5. 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 $f_{Rx} \leq 1/(4.5\ t_{CY})$.
6. Reset Pulse Width = 6 t_{CY} minimum; System Clock must be running during Reset.
7. Status update can have a maximum delay of 28 clock periods from the event affecting the status.

TYPICAL Δ OUTPUT DELAY VS. Δ CAPACITANCE (pF)

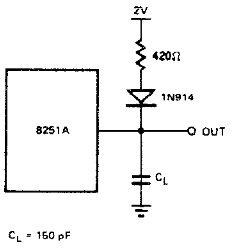


A.C. TESTING INPUT, OUTPUT WAVEFORM



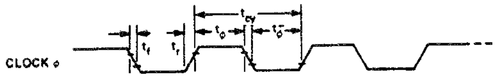
A C TESTING INPUTS ARE DRIVEN AT 2 4V FOR A LOGIC 1 AND 0 45V FOR A LOGIC 0 TIMING MEASUREMENTS ARE MADE AT 2 0V FOR A LOGIC 1 AND 0 8V FOR A LOGIC 0

A.C. TESTING LOAD CIRCUIT

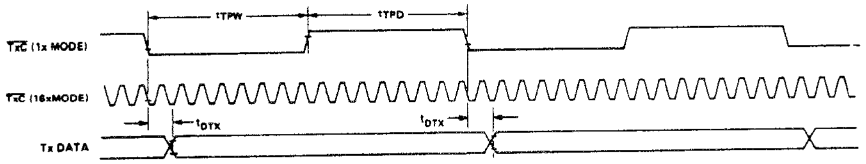


WAVEFORMS

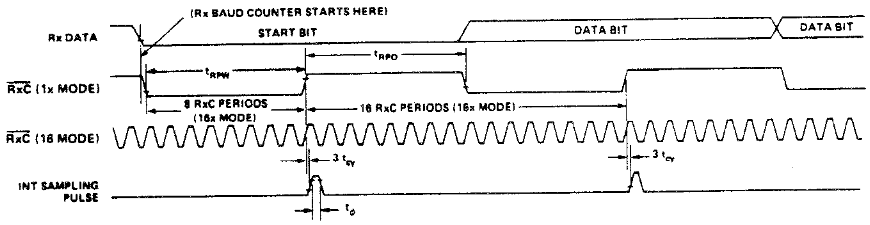
SYSTEM CLOCK INPUT



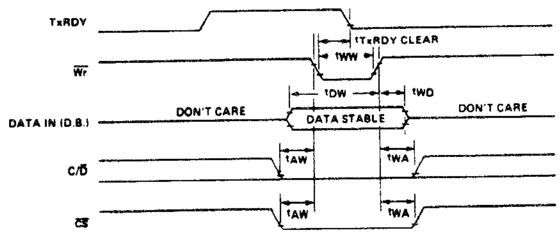
TRANSMITTER CLOCK AND DATA



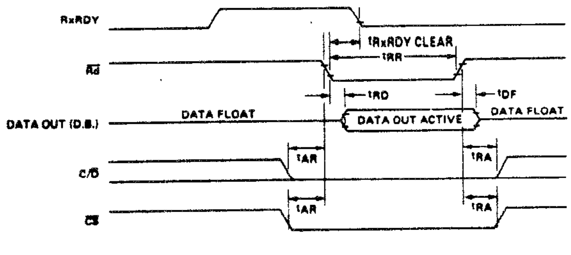
RECEIVER CLOCK AND DATA



WRITE DATA CYCLE (CPU → USART)

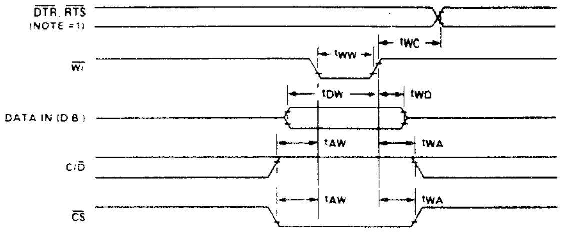


READ DATA CYCLE (CPU ← USART)

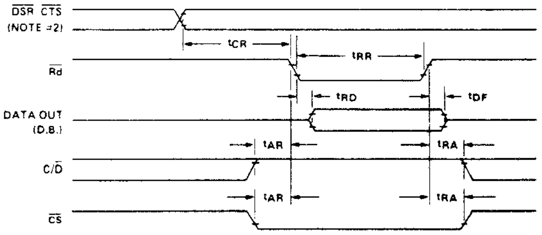


WAVEFORMS (Continued)

WRITE CONTROL OR OUTPUT PORT CYCLE (CPU → USART)

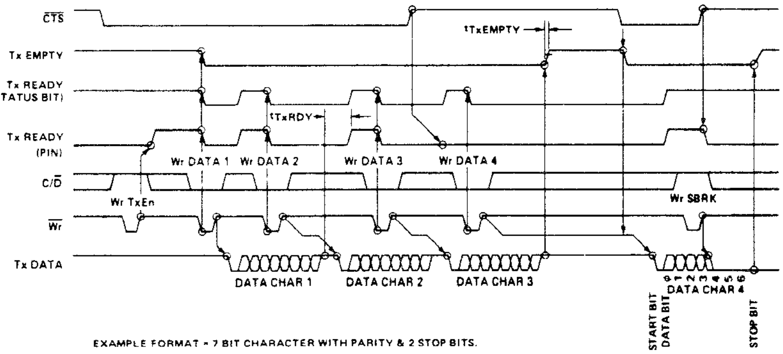


READ CONTROL OR INPUT PORT (CPU ← USART)



NOTE #1: T_{WC} INCLUDES THE RESPONSE TIMING OF A CONTROL BYTE.
NOTE #2: T_{CR} INCLUDES THE EFFECT OF CTS ON THE TxENBL CIRCUITRY.

TRANSMITTER CONTROL AND FLAG TIMING (ASYNC MODE)



EXAMPLE FORMAT = 7 BIT CHARACTER WITH PARITY & 2 STOP BITS.

IO8 MANUAL REGISTRATION FORM

In our effort to continually upgrade our documentation, we would appreciate any feedback you may have concerning this manual. Please mail your comments and suggestions to SSM Customer Service at the address below.

IO8 Serial Number ______________________

NAME ______________________

TITLE ______________________

ADDRESS ______________________

____

Telephone (____) ______________________

SEND TO:

SSM MICROCOMPUTER PRODUCTS, INC.
2190 Paragon Drive
San Jose, CA 95131
Attention: Customer Service Dept.

2142

SSM

SSM MICROCOMPUTER PRODUCTS, INC.
2190 Paragon Drive
San Jose, California 95131

(408) 946-7400 • Telex: 171171
TWX: 910-338-2077