Educational Data Systems 2415 Windward Lane, Newport Beach, California 92660

EDS-8

PROGRAMMABLE MULTIPLEXER

USER'S MANUAL

EDS-1015-1

Table of Contents

Appendixes

Figures

0. Introduction

The EDS-8 is a complete I/O Communications System for any Nova type computer. It permits up to 128 devices to communicate concurrently with the mini-computer. These may be any combination of teletypes, typewriters, CRT terminals, printers, card readers, modems, and other RS-232-C or current-loop devices. They may operate in half or full duplex, with or without automatic echo.

The EDS-8 is more than a conventional multiplexer in two fundamental respects. First, it provides program control on a port by port basis over baud rate and most other port parameters. Second, it permits eachport to operate in either character-at-a-time or automatic buffer mode. In automatic buffer mode, each port is assigned a buffer in the core memory of the computer. This may be of any size and have any location in core. The EDS-8 automatically services these buffers, inputting or outputting strings of characters without any interruption of the computer program until the entire string is completed. To accomplish this, it operates through the DMA channel of the computer.

The program gives the EDS-8 its I/O processing commands by storing appropriate control words in certain dedicated core locations. These I/O commands define, for each I/O port:

o Length and location of the string buffer in core from/into which character strings are to be transferred;
o Baud rate for transmission and reception, from among 8 standard rates;
o Character length - 5, 6, 7, or 8 bits;
o Parity mode - even, odd, or none;
o Number of stop bauds - one or two;
o When interrupts are desired - for each character, for certain defined control characters, or only when the end of the specified buffer is reached;
o Whether or not incoming characters are to be automatically echoed.

In addition to these broad programming capabilities, the EDS-8 has exceptional electrical interfacing capabilities. Each port has 4 signal lines: 2 data lines (incoming and outgoing), and 2 auxiliary lines: the outgoing Device Control line and the incoming Device Status line. These auxiliary lines may be programmed for any desired purpose. For example, for a full-duplex modem such as a Bell System 103 Type Data Set, Device Control may be used as the Data Terminal Ready line (to enable automatic hang-up), and Device Status may be the Received Line Signal Detector (permitting automatic log-off). For half-duplex control, Device Control would be the Request-To-Send, and Device Status, Clear-To-Send. For certain line printers, Device Control may be used as the Print Line command, and/or Device Status may indicate the printer's Ready/Busy status.

All four of these lines operate at standard EIA RS-232-C voltage levels. Optionally, the two data lines may also simultaneously operate as current loops, so that standard teletypes may be plugged into the same connectors as EIA terminals.

These features enable the EDS-8 to control up to 128 devices, including combinations of local and remote terminals and peripherals of many different kinds. Further, all these devices may be operated at any standard Baud rate up to 9600 Baud, under software control.

1. General Description

The EDS-8 Multiplexer (hereafter sometimes referred to as EDS-8 and sometimes as Mux) interfaces up to 128 I/O devices to a Nova line computer. The heart of the system is the EDS-300 board which contains all the Common Logic and 8 interface Ports (see Figure 1). Each Port contains the necessary control logic and buffering for full-duplex operation. The Common Logic contains a sequencer which interrogates the ports in turn, and logic to service them when they need it, providing direct access to core memory via the DMA channel and generating interrupts to the program when appropriate. Up to 120 more ports may be added by means of EDS-301 expansion boards. Each 301 board contains 8, 16, or 24 ports. Thus a full 128 port system requires one 300 and five 301 boards.

2. Core Allocation Requirements

To control the EDS-8, the program stores appropriate I/O command words in a certain dedicated area in core, consisting of a control block for each port used. The size of each control block may be 8, 16, or 32 words.

Within each control block, 6 words are used for Mux control, namely words 0, 1, 4, 5, 6, and 7; the remainder may be used for any other purpose. The three even-numbered words control input, the odd-numbered ones control output. In each set of three, two words are pointers to the beginning and end of the desired buffer areas in core. The sizes and locations of these buffers are therefore entirely under software control. Several ports may transmit from the same buffer area at one time, since each keeps track of its own pointers. If the port is operated in single-character mode, the pointer words are not read by the Mux and may be used for any other purpose. The meanings and formats of the six control words are shown in Figure 2 and explained in Section 4.

The EDS-8 is normally delivered wired for 32-word control blocks. This may easily be changed to 16 or 8 by rewiring some jumper wires as described in Appendix A.

The location of the control blocks must be such that each control block starts at an exact multiple of the block size, and such that the first block starts at an exact multiple of eight times the block size. The only other restriction is that the entire control area must not cross a 10000_8-word boundary for 32-word control blocks, or a 4000-word boundary for 16-word blocks, or a 2000-word boundary for 8-word blocks. The desired control block locations are set into the hardware by means of DIP Switches on the 300 and 301 boards. (See Appendix A).

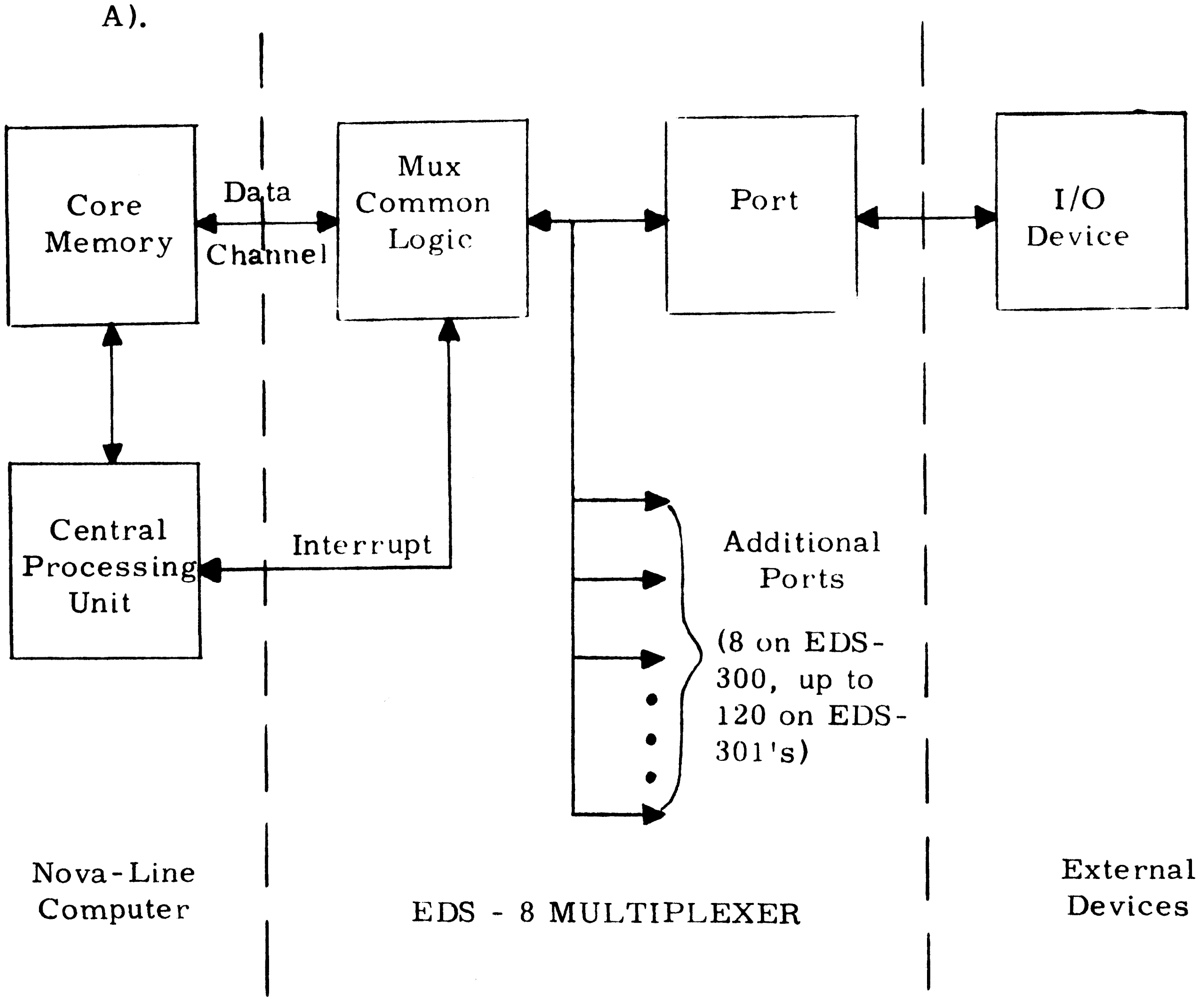
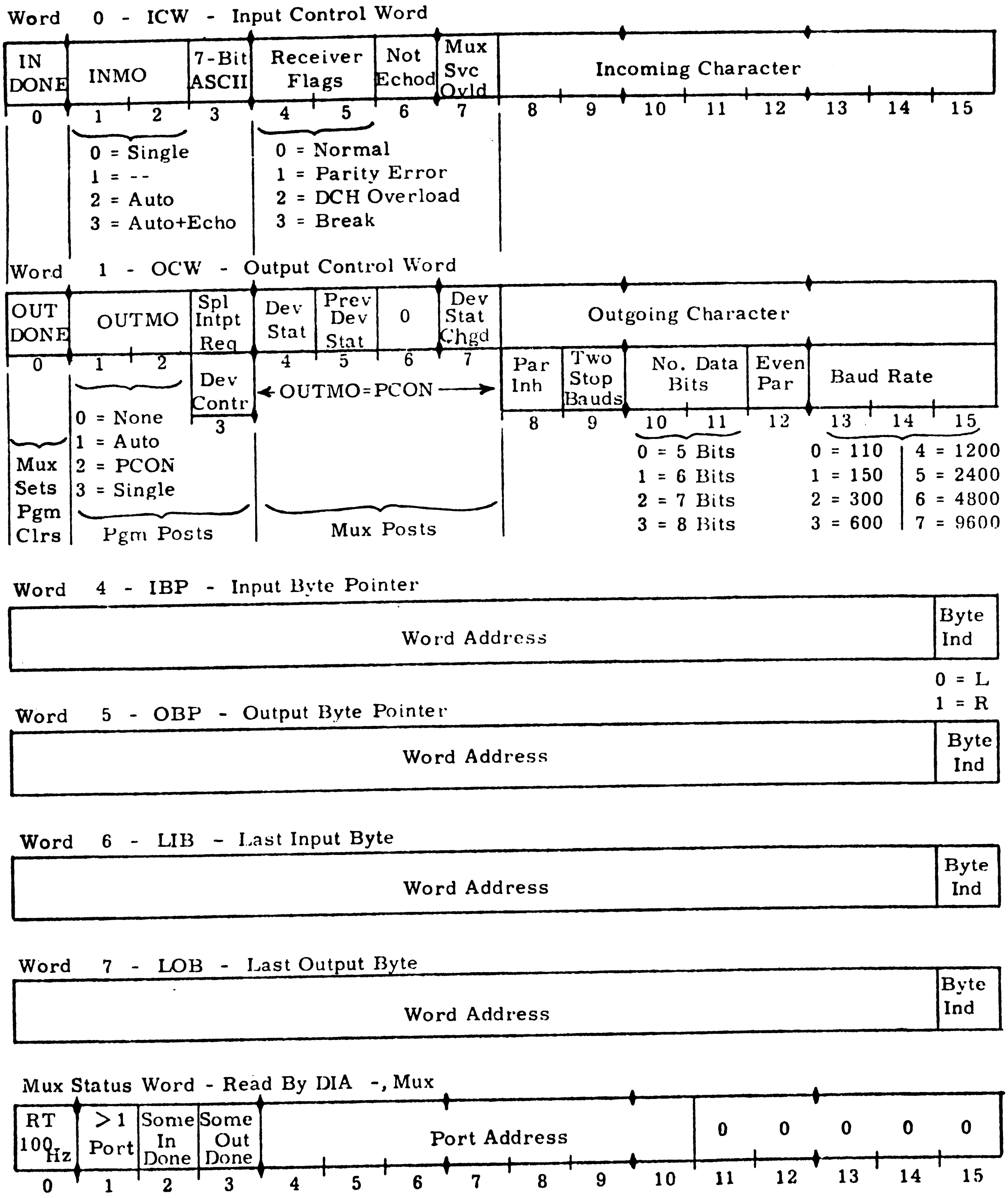


FIGURE 1 - Block Diagram Showing EDS-8 Interfacing External Devices to Nova Computer



Word 0 - ICW - Input Control Word

IN DONE	INMO		7-Bit ASCII	Receiver Flags		Not Echod	Mux Svc Ovld	Incoming Character							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

INMO (bits 1-2):

0 = Single
1 = --
2 = Auto
3 = Auto+Echo

Receiver Flags (bits 4-5):

0 = Normal
1 = Parity Error
2 = DCH Overload
3 = Break

Word 1 - OCW - Output Control Word

OUT DONE	OUTMO		Spl Intpt Req	Dev Stat	Prev Dev Stat	0	Dev Stat Chgd	Outgoing Character							
								Par Inh	Two Stop Bauds	No. Data Bits		Even Par	Baud Rate		
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
			Dev Contr	← OUTMO=PCON →											
Mux Sets Pgm Clrs	Pgm Posts			Mux Posts											

OUTMO (bits 1-2):

0 = None
1 = Auto
2 = PCON
3 = Single

No. Data Bits (bits 10-11):

0 = 5 Bits
1 = 6 Bits
2 = 7 Bits
3 = 8 Bits

Baud Rate (bits 13-15):

0 = 110		4 = 1200	
1 = 150		5 = 2400	
2 = 300		6 = 4800	
3 = 600		7 = 9600	

Word 4 - IBP - Input Byte Pointer

Word Address	Byte Ind

Byte Ind: 0 = L, 1 = R

Word 5 - OBP - Output Byte Pointer

Word Address	Byte Ind

Word 6 - LIB - Last Input Byte

Word Address	Byte Ind

Word 7 - LOB - Last Output Byte

Word Address	Byte Ind

Mux Status Word - Read By DIA -, Mux

RT 100Hz	>1 Port	Some In Done	Some Out Done	Port Address							0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Figure 2 - EDS-8 Control Words

3. Overall Sequence of Operation

In order to start the Mux, it is necessary for the program to give a DOC -,MUX instruction (the data output is immaterial). The Mux then cyclically tests all ports, checking for an Input Request (i.e., a character has been received) or an Output Request (i.e., port is ready to accept next output character). When one of these requests is found, the Mux stops at that port and reads the Input Control Word or Output Control Word belonging to that port. Depending on the instructions contained in this control word, the Mux then carries out the indicated operations, including accessing the automatic buffer if appropriate. When all required actions for the sensed Input or Output Request are completed, the Mux continues on to test the next port, etc. If both Input Request and Output Request are true simultaneously, Input takes precedence and Output is deferred to the next time that that port is inspected. These steps are shown in the Flow Chart in Figure 3.

When an Input or Output process is completed, the Mux sends an Interrupt to the CPU. An Interrupt may also be given by a real-time clock on the EDS-8, once each 10 msec (100 times per second). The program, by giving a DIAS -,MUX, can then clear the Interrupt and simultaneously read the Mux Status Word, which indicates what type of interrupt was given.

4. Definition of Control Words (see Figure 2)

4.1 ICW - Input Control Word (Word 0 of each control block)

Bit 0 - Input Done (INDONE). Set by the Mux under the following conditions:

1) If in Single Character Input mode, when an incoming character is received and stored in Bits 8-15.

2) If in Automatic Input mode (with or without automatic echo), when the assigned input buffer is full; i.e., after an incoming character is placed in the byte specified by LIB.

3) If any of the following three cases is detected by the receiver (see Bits 4 and 5 below).

   Parity error - if Parity Inhfbit is 0.
   Data Channel overload
   Break (framing error)

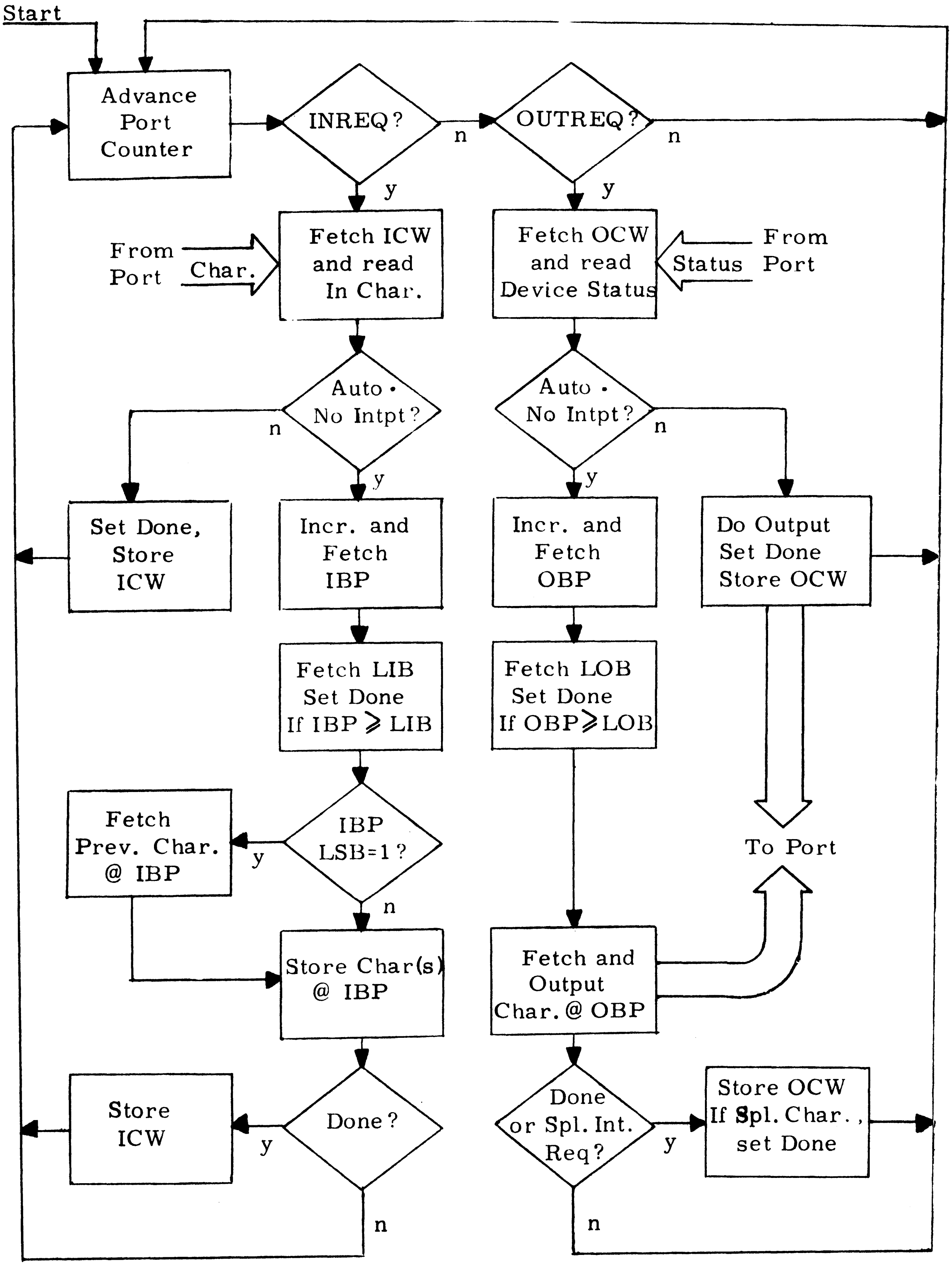


Figure 3 - Flowchart Showing EDS-8 Processing of Inputs and Outputs

4) If in Automatic Input With Echo mode but automatic echo was not accomplished (see Bit 6 below).

5) If in 7-bit ASCII mode, and a "Special" character (i.e., less than octal 40 or greater than octal 173) is received.

At the same time that INDONE is set, an interrupt is also generated.

Input Done must be cleared by the Mux interrupt service program. There is approximately one character time available for this. If another input character is received while Input Done is still set, it will override the character in Bits 8-15 (in this case automatic input is not permitted), and Bit 7 is set indicating that an incoming character has been lost, but a second interrupt is not generated.

Bits 1 and 2 - Input Mode (INMO). Written by the program and read by the Mux to determine what type of input will be done.

1	2		
0	0	=	Single Character Input mode - i.e., each incoming character is placed in Bits 8-15 of ICW, INDONE is set, and an interrupt given.
0	1	=	Illegal.
1	0	=	Automatic Input - i.e., incoming characters are placed in the input buffer defined by IBP and LIB, as long as no interrupt conditions are encountered (see Bit 0 for interrupt conditions).
1	1	=	Automatic Input with Echo - same as Automatic Input except that each character which is placed in the input buffer is also automatically echoed (output).

Bit 3 - 7-Bit ASCII. Written by the program.

0 = Binary - i.e., each incoming character is stored exactly as received, with any unused bits (if Character Length is less than 8) set to 0.

1 = 7-Bit ASCII mode. The most significant bit is always set to 1, the other 7 bits are stored as received. Further, the character is tested to determine if it is a "Special" character. Special characters are defined as those below octal 40

or above octal 173, and are not stored in an automatic buffer but in ICW and produce interrupts. This allows immediate program response to such characters as backspace, carriage return, end-of-message, etc.

Bits 4 and 5 - Receiver Flags. Posted by the Mux each time an input is received.

4	5		
0	0	=	Normal - i.e., none of the three cases below.
0	1	=	Parity Error - if Parity Inhibit in OCW was 0.
1	0	=	Data Channel Overload - i.e., an input character was received before the previous character from that port could be stored away by the Mux. This can only happen if another interface on the computer's data channel, having higher priority than the Mux, has been taking most of the core memory cycles.
1	1	=	Break, or Framing error - i.e., a stop baud was not received. This is used for detecting the Break character, which is a 0 character with no stop baud.

Each of cases 1 through 3 produces an interrupt and sets Input Done. If more than one of these occurs at the same time, only the one with the highest number is recorded.

Bit 6 - Not Echoed. Posted by the Mux if in Automatic Input With Echo mode, and if automatic echo was not possible because the output circuitry was still busy with a previous output. This can only happen if the program started an output while the Mux was in Automatic Input With Echo mode - a condition that should never occur. At the same time that this bit is posted, Input Done is also set and an interrupt generated.

Bit 7 - Mux Service Overload. This bit is set by the Mux if at the time an input character is stored the Input Done flag is still on from a previous input. The length of time available for the Mux service program to service an input interrupt before the next input could come in on the same port as on the average equal to one character transmission time (about 1 msec for 9600 Baud data rate), but could in the worst case be as much as 300 µsec less than that.

Bits 8-15 - Input Character. Every incoming character which produces an Input Done is stored in these bits. In automatic buffer mode characters which are stored in the automatic buffer are not stored in ICW, except that the last character, which produces the buffer-full condition, is stored in both places.

4.2 OCW - Output Control Word (Word 1 of each control block)

Bit 0 - Output Done (OUTDONE). Set by the Mux under the following conditions:

1) If in Single Character or Port Control Output mode, when the output byte is read for transmission to the port. The program then has a minimum of one character time, and typically almost two character times, to reload OCW if uninterrupted output is desired.

2) If in Automatic Buffer Output mode, when the last byte of the automatic buffer is read by the Mux for transmission. Again, if the output control words are reloaded within 1-2 character times, uninterrupted output will result.

3) If in Automatic Output and Special Interrupt Request mode, and if the transmitted character is a Special Control character (<40 or ⩾174 octal). In this event automatic output is terminated.

4) Regardless of output mode, if the Device Status Change bit (Bit 7) becomes a 1.

At the same time that OUTDONE is set, an Interrupt is also generated. Output Done must be cleared by the program before another output can take place on that port. Leaving OUTDONE set inhibits interrupts from that port, even if Device Status changes.

Bits 1 and 2 - Output Mode (OUTMO). Written by the program and read by the Mux to determine what type of output will be done.

1	2		
0	0	=	None - no output.
0	1	=	Automatic buffer output - i.e., Mux will output from automatic buffer defined by OBP and LOB.
1	0	=	Port Control Output (PCON) - i.e., the data in Bits 3 and 8-15 are sent to the port as control data, to do their assigned control functions.
1	1	=	Single character output - i.e., the data byte in Bits 8-15 is sent to the port for transmission to the external device.

Bit 3. The meaning of Bit 3 depends on the Output Mode. If OUTMO=00 (None) or 11 (Single), Bit 3 has no effect. If OUTMO=01 (Auto):

Bit 3 - Special Interrupt Request. Written by the program. If this bit is a 1, the Mux will test each outgoing character to determine if it is a Special Control character (<40 or≥174 octal, 7-bit). If it is, the Mux will still transmit it, but also set OUTDONE, produce an interrupt, and terminate automatic output. This is useful in driving printers which require some special service after a Carriage Return character, such as a Line Feed, or a delay time, or a Print Line command.

If OUTMO=10 (PCON):

Bit 3 - Device Control. Written by the program and transferred by the Mux to the port when the Port Control Output is done. This bit is stored by a flip-flop in the port circuitry and applied as an EIA level on the Device Control Line going to the external device. It may be used for any desired function - it does not interact with the functioning of the port in any other way. For Bell System 103 Type Data Sets, this signal is used as the Data Terminal Ready line (circuit CD). For half-duplex control, this signal is used as the Request to Send (circuit CA).

1 = + 10 volts = EIA Positive
0 = - 10 volts = EIA Negative

Bit 4 - Device Status. Written by the Mux each time OCW is inspected whether any output is done or not.

This is simply the current value of the Device Status line coming from the external device. This line may be used for any desired function - it does not interact with the port in any other way. For 103 Type Data Sets, this line is used for the Received Line Signal Detector (circuit CF).

1 = EIA Positive, i.e. >+3 volts
0 = EIA Negative, i.e. $\leq$ 0 volts, or open

Note: Actually, any voltage above +1.3v will produce a 1, and any voltage below +0.7v will produce a 0; voltages between these limits are indeterminate.

Bit 5 - Previous Status. Written by the Mux each time OCW is inspected whether any output is done or not. The value read from bit 4 is rewritten here. This bit is of no significance to the programmer.

Bit 6 - Unused. Always is zero.

Bit 7 - Device Status Changed. Written by the Mux whenever the device status line does not equal the device status bit (Bit 4) read from the OCW. When this bit is set, Bit 0 (Output Done) is also set and an interrupt is generated. Note that this imposes a requirement on the Mux Service Program, whenever it writes an OCW, to leave Bit 4 the way it was - otherwise the Mux will think Device Status has changed and produce an interrupt.

Bits 8-15 - Output Character. If Output Mode is 11 (Single character), this byte is sent to the port for transmission to the external device. If OUTMO is 00 or 01 (Inactive or Automatic), this byte is ignored. (Exception: In Automatic Output with Special Interrupt Request, this byte must not be a Special Character -- i.e. this byte must be between 40 and 173 or between 240 and 373, inclusive; otherwise the Special Character detection logic will never permit automatic output to get started.) If OUTMO = 10 (Port Control Output), this byte is sent to the port as a control character, governing both output and input. In this case, the meanings of these eight bits are as follows.

Bit 8 - Parity Inhibit.

0 = Parity is generated (transmit) and checked (receive)
1 = No Parity

Bit 9 - Stop Baud Selector.

0 = One Stop Baud
1 = Two Stop Bauds

Note: This governs output only. In input, one stop baud (in fact, just over one-half stop baud) is always adequate.

Bits 10 and 11 - Number of Data Bits.

10	11		
0	0	=	5 Bits
0	1	=	6 Bits
1	0	=	7 Bits
1	1	=	8 Bits

This field governs the data bits only; parity (if not inhibited) is an additional bit. If less than 8 bits are selected, the character will be right justified, and the unused most significant bit(s) will be set to zero on input (exception: 7 Bit ASCII mode - see Input Control Word, Bit 3). For output, the character must be right justified, but the unused most significant bits are ignored.

Bit 12 - Parity Mode (has effect only if Bit 8 = 0)

0 = Odd Parity
1 = Even Parity

Bits 13-15 - Frequency of transmission/reception

13	14	15		
0	0	0	=	110 Baud (standard teletype)
0	0	1	=	150 Baud
0	1	0	=	300 Baud
0	1	1	=	600 Baud
1	0	0	=	1200 Baud
1	0	1	=	2400 Baud
1	1	0	=	4800 Baud
1	1	1	=	9600 Baud

4.3 IBP - Input Byte Pointer. (Word 4 of control block). This word and its partner (LIB) are only used if in Automatic Input mode. IBP must be set up by the program to 1 less than the first byte address of the automatic input buffer. Each time the Mux stores an incoming byte, it will increment IBP and then store the byte at the resulting address. Thus IBP always points to the last input byte stored unless no input bytes have been received yet.

4.4 OBP - Output Byte Pointer. (Word 5 of the control block). This word and its partner (LOB) are used only if in Automatic Output mode. OBP must be set up by the program to 1 less than the byte address of the first byte of the automatic output buffer. Each time the Mux is ready for an output byte, it will increment OBP and then fetch the byte from the resulting address for transmission. Thus OBP always points to the last byte that has been transmitted, if any.

4.5 LIB - Last Input Byte. (Word 6 of control block). Set up by the program to the last byte address of the auto input buffer. The Mux will generate an Input Done Interrupt when a byte is stored at this address. IBP will then be equal to LIB.

4.6 LOB - Last Output Byte. (Word 7 of the control block). Set up by the program to the last byte address of the automatic output buffer. When the byte at that address is picked up for transmission, the Mux will generate an Output Done interrupt (OBP will then be equal to LOB). Note: An automatic buffer may contain as little as one byte. In this case (LOB) = (OBP) +1, initially.

4.7 Mux Status Word (MUXSTWD). This is not one of the control words in each port control block, but it is the word which is read in when the program gives a DIA -,MUX instruction. The Mux Status Word contains information indicating what type of interrupt was given. This word is cleared to zero when a START pulse is given by the program. Therefore, by means of a DIAS -,MUX instruction the program can simultaneously clear DONE following an interrupt, read the Mux Status Word and clear it.

Bit 0-Real-Time (100 Hz). If the Real-Time clock is enabled, once each hundredth of a second the Mux will give an interrupt with this bit on (1). Any additional interrupts will have this bit off (0), provided that the program gave a START pulse following the first interrupt.

Bit 1 - More Than One Port Done. If this bit is a 1 it indicates that since the last time the Mux Status Word was cleared more than one port has produced an Input Done or Output Done interrupt. If this bit is a 0 it indicates that only one or no such interrupt was generated. If a single port has generated both an INDONE and an OUTDONE interrupt, this bit will be a 1.

Bit 2 - Input Done.

0 = No port has generated an Input Done interrupt since the last time MUXSTWD was cleared.

1 = One or more ports have generated INDONE interrupts.

Bit 3 - Output Done.

0 = No port has generated an OUTDONE interrupt since the last time MUXSTWD was cleared.

1 = One or more ports have generated OUTDONE interrupts.

Bits 4-10 - Port Address. If only one port has generated an INDONE or OUTDONE interrupt since the last time MUXSTWD was cleared, this field will store the address of that port. If more than one port has generated interrupts, this field will contain the address of the port which did so last. If no port has generated an interrupt, this field will be zero. The "Address" of a port consists of the 7 bits which identify the 128 possible port control blocks: i.e. bits 4 - 10 of the port's control word addresses if the control block size is 32 words, or bits 5 - 11 for 16-word control blocks, or bits 6 - 12 for 8-word control blocks.

Summary

0	1	2	3	
0	0	0	0	Impossible*
0	0	0	1	One port has Output Done--its address is in Bits 4-10
0	0	1	0	One port has Input Done--its address is in Bits 4-10
0	0	1	1	Impossible*
0	1	0	0	Impossible*
0	1	0	1	More than one port has Output done--no Input
0	1	1	0	More than one port has Input done--no Output
0	1	1	1	At least one Output and one Input done (possibly same port)
1	0	0	0	Real-time interrupt--no Input or Output
1	0	0	1	Real-time plus one Output Done--address in 4-10
1	0	1	0	Real-time plus one Input Done--address in 4-10
1	0	1	1	Impossible*
1	1	0	0	Impossible*
1	1	0	1	Real-time plus more than one Output Done
1	1	1	0	Real-time plus more than one Input Done
1	1	1	1	Real-time plus Input and Output

* Impossible (or hardware fault), provided MUXSTWD is being read after an interrupt and a START pulse was given after the previous interrupt.

5. CPU Control Over Mux

The CPU controls Mux operation by means of the following I/O instructions.

IORST
or CLEAR
(eg. NIOC)

Turns off both Mux and Real-Time Clock. Terminates any output that may have been in process by setting all data output lines to -10 volts, and sets all Device Control lines to +10 volts. Sets baud rate of all ports to 110 baud. Clears DONE and BUSY flags.

IOPLS
(eg. NIOP)

Starts Real-Time Clock. Has no effect on Mux action.

START
(eg. NIOS)

Clears Done flag and Mux Status Word. Thus, a DIAS -,MUX instruction will simultaneously read and clear the Mux Status Word.

DOC -,MUX

Starts the multiplexer proper. Thus, DOCP -,MUX will start both the Mux and the real-time clock. (Only the DOC signal is used--it does not matter what data is output by this instruction.)

DOB -,MUX

Stops all Mux data channel transfers, but does not interrupt any input or output already in process. Because the Mux takes successive data channel cycles when processing a particular port, the DOB instruction will always stop the Mux after completion of one port service and before beginning another. The DOB/DOC pair may be used whenever a control word has to be written, to guard against the possibility that the Mux has just written into that control word. For example, the following sequence may be used in the Mux Output Routine where a new OCW is to be written.

```
LDA      1,C4000    ;mask for bit 4, Device Status
LDA      2,NOCW     ;next OCW desired to be written
INTDS               ;disable interrupts
DOB      0,MUX      ;pause Mux operation
SKPDN    MUX        ;is Mux trying to interrupt?
JMP      .+4        ;  no
DOC      0,MUX      ;  yes - turn Mux back on
INTEN               ;      and let it interrupt
JMP      .-6        ;      try again after the interrupt
LDA      0,OCW      ;pick up OCW
AND      0,1        ;pick out the Device Status bit
ADD      1,2        ;insert device status in new OCW
STA      2,OCW      ;store new OCW
DOC      0,MUX      ;continue Mux operation
INTEN               ;enable interrupts
```

This sequence guards against the possibility of the Device status changing just before the program stores a new OCW. A similar sequence may be used in the Mux Input Service Routine to guard against an incoming character being accidentally overwritten by the program.

DIA -,MUX

Reads in the Mux Status Word (see Section 4.7)

6. <u>Initialization</u> (When power is first turned on)

1) Give an IORST, or NIOC -,MUX.

2) Set up each OCW for initial Port Control Output (see Section 4.2). For example, for a data set ready for automatic answering at 110 Baud with two Stop bauds and using 8 bit characters without parity, OCW would be 50360:

Out Done	OUTMO = PCON		Dev Contr	Dev Stat	Prev Stat	0	Dev Stat Chg	Par Inh	Two Stop Baud	Char. L = 8 bits		Even Par	Freq. = 110 Baud		
0	1	0	1	0	0	0	0	1	1	1	1	0	0	0	0
0	5			0			3			6			0		
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

3) Set up each ICW for the desired input mode. Initially this will generally be Single Character Input. Therefore, ICW would be set to 10000 if 7-Bit ASCII mode is desired or to 0 otherwise.

4) Give a DOC -,MUX. This starts the Mux, setting its BUSY flag. BUSY remains set from here on until an IORST, CLEAR, or DOB -,MUX is given; interrupts cause DONE to come on but do not clear BUSY. The Mux will now do all the indicated port control outputs, setting the Output Done bit for each and also setting the Mux DONE flag.

5) Enable Interrupts. As soon as the Mux DONE flag is set, an interrupt will result. With a typical Interrupt Service Routine it is likely that by the time the DIAS -,MUX is given all eight ports will have been processed by the Mux, so that only one interrupt will be given for all eight. Alternatively, the program can simply wait until the Output Done bit in the last port control block is a 1; when the last one is done, they are all done, because the Mux always starts at Port 0 when it is first turned on.

7. Input and Output

7.1 To Input in Single Character Mode

1) Set ICW to: (see Figure 2)
10000 for 7-bit ASCII mode (msb=1)
0 for normal input (unmodified)

2) When an incoming character arrives, it will be stored in ICW, the INDONE bit will be set, and an interrupt will be given.

7.2 To Output in Single Character Mode

1) Set OCW to: 6x000 plus the desired character, where x = 4 or 0 according as the Device Status bit (Bit 4) is 1 or 0.

2) An interrupt will be given and OUTDONE set when the character is picked up for transmission.

7.3 To Input Automatically into a String Buffer

1) Set IBP to 1 less than the byte address of the first byte in the string. Note: Byte Address equals Word Address shifted one place to the left with the least significant bit = 0 for left byte or 1 for right byte.

2) Set LIB to the byte address of the last byte in the string.

3) Set ICW to:

40000 for no echo and no 7-Bit ASCII interpretation
50000 for no echo but 7-Bit ASCII mode
60000 for automatic echo and no 7-Bit ASCII
70000 for automatic echo and 7-Bit ASCII mode

4) An interrupt will be given and INDONE set if:

Condition	Indication
Buffer full	IBP = LIB
Parity Error	ICW Bits 4, 5 = 0 1
Data Channel Late	ICW Bits 4, 5 = 1 0
Break	ICW Bits 4, 5 = 1 1
Auto echo ordered but not accomplished	ICW Bit 6 = 1
Special Character in 7-Bit ASCII mode	Character is in ICW Bits 8 - 15

7.4 To Output Automatically from a String Buffer

1) Set OBP to 1 less than the byte address of the first byte to be output.

2) Set LOB to the byte address of the last byte to be output.

3) Set OCW to:
20000 for regular automatic output
30240 for interrupts on special control characters

4) An interrupt will be produced and OUTDONE set if:

Condition	Indication
Buffer empty	OBP = LOB
Device Status changed	OCW Bit 7 = 1 (OBP points to last character transmitted)
Special character if special interrupt requested	Character is in OCW Bits 8-15 (OBP also points to it in buffer)

7.5 To Change Port Control

A new Port Control mode may be set up at any time by simply storing the desired Port Control Word (see Section 4.2) in OCW. Thus, for example, Baud rate, or parity mode, or Device Status may be changed whenever the program wishes.

If it is desired to change Port Control (PCON) after completing a transmission, it is necessary to add two blank characters at the end of the desired output string, because of the double buffering in the Port logic. In other words, the OUTDONE interrupt is given when the outgoing character is taken from core and put into the output buffer register of the Port, at which time the previous character is still in the transmit register being transmitted. Thus if a new PCON is given immediately after the OUTDONE interrupt, the last two characters will not yet have been transmitted. The above applies to single character output mode as well as automatic output.

If it is not permissible to transmit blank characters, a corresponding time delay (two character times) may be provided by the software by using the real-time clock.

8. Interrupt Service

1) Give a DIAS -,MUX. This reads in the Mux Status Word (see Figure 2), and clears the Mux Done flag.

2) Test MUXSTWD Bit 0. If 1, do real-time interrupt service. This step is, of course, not needed if the real-time clock has not been turned on.

3) Test Bit 2. If 1, do Input Service for all ports.

4) Test Bit 3. If 1, do Output Service for all ports.

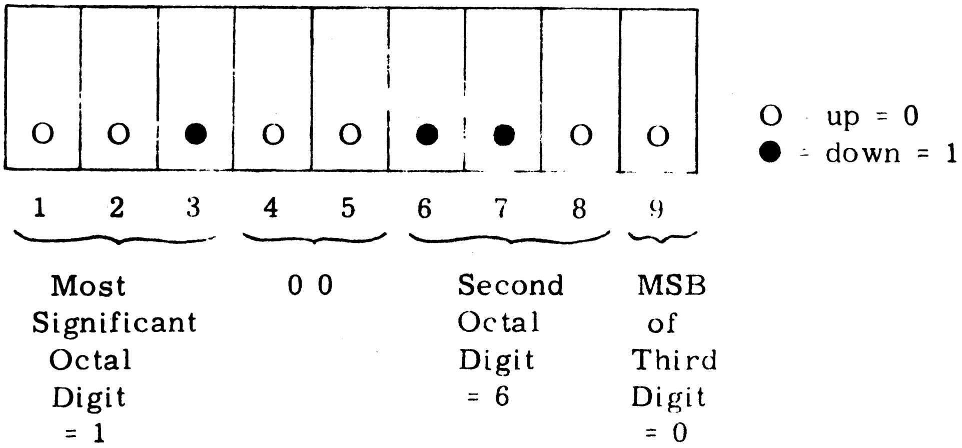
NOTE: This simplified Interrupt Service does not take advantage of the Single Port/Multiple Port Indications in the Mux Status Word, but it is quite satisfactory for most 8-port systems. Those additional indications are primarily for use in systems having considerably more than 8 ports.

Appendix A - Miscellaneous Information

1. How to change Control Block Location

The locations of the Control Blocks for the EDS-8 are easily changed by resetting the DIP switches on the 300 and/or 301 boards, as explained in Figure A-1.

For example, if the first port's Control Block is to start at 16000 (octal), the 9-position DIP Switch on the 300 board must be set as shown below, assuming 32-word Control Block size.



2. How to change Control Block Size

The EDS-8 is normally delivered with 32-word control blocks. This may be changed to 16 or 8-word size by changing the wire-wrapped jumpers at location 12J on the 300 board. Figure A-1 explains how to do this.

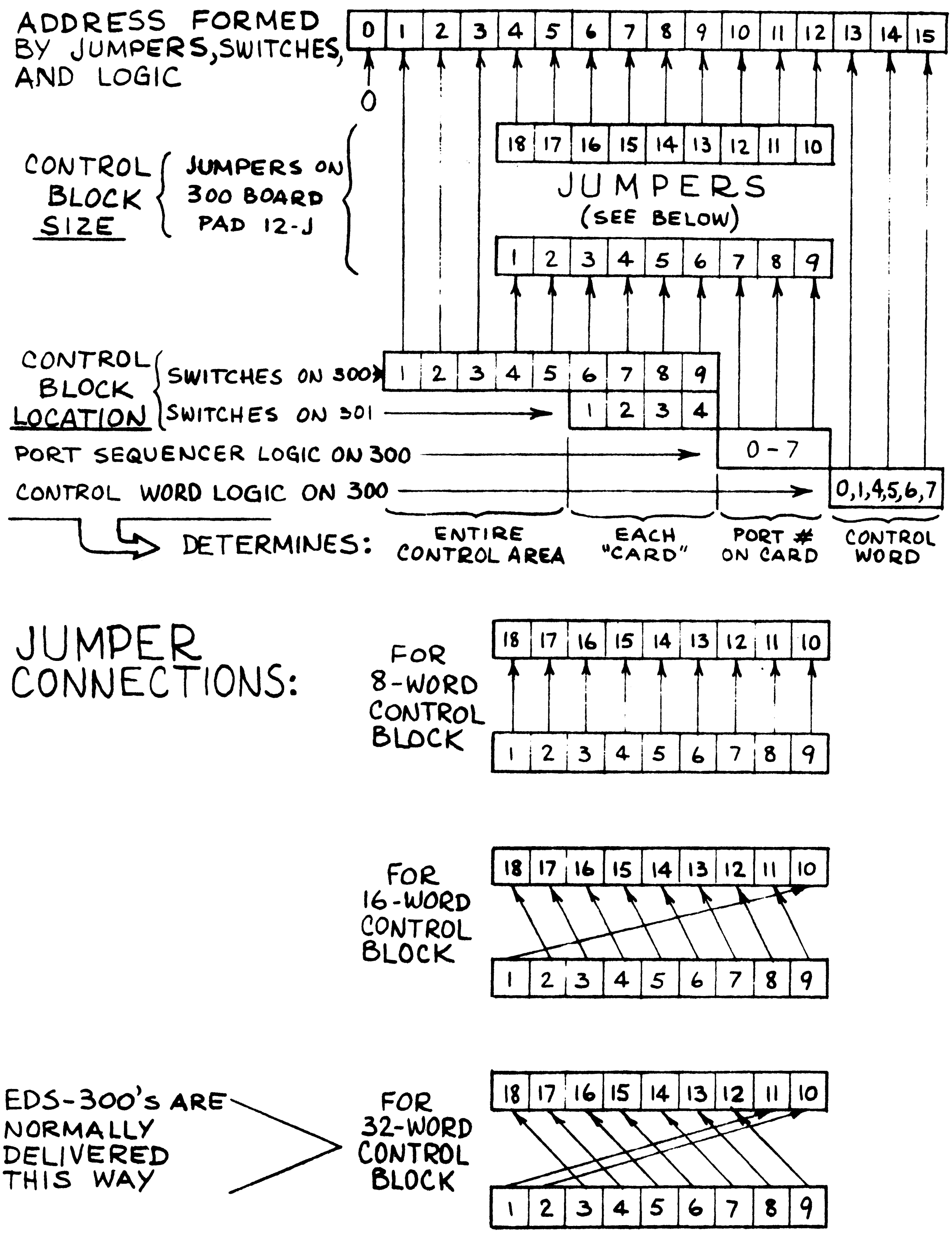
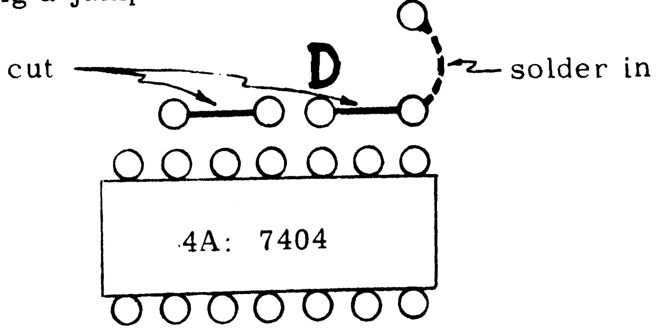


Figure A-1 How to Change EDS-8
Control Block Size and/or Location

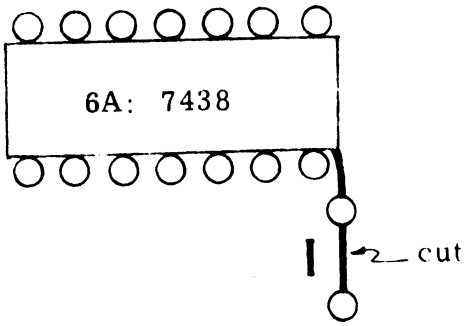
3. How to change Device Code

The EDS-8 is normally delivered as Device Code 25. This can easily be changed to 24 by two simple changes on the 300 board.

a) Change Device Code recognition logic by cutting 2 conductors and soldering a jumper near location 4A as shown below:



b) Change Interrupt Acknowlege logic by cutting one conductor near location 6A as follows:



4. How to change Mask Bit

The EDS-8 is normally delivered with its Interrupt Disable Mask Bit = Bit 5. To change this to another bit, cut the conductor marked M near location 5C and solder in a jumper from its left end to one of the following points:

13A8	for Bit	0
13A10	for Bit	1
13A12	for Bit	2
13A2	for Bit	3
12A10	for Bit	4
12A12	for Bit	5
11A4	for Bit	6
11A6	for Bit	7
12A6	for Bit	8
12A4	for Bit	9
12A2	for Bit	10
12A8	for Bit	11
11A8	for Bit	12
11A10	for Bit	13
11A12	for Bit	14
11A2	for Bit	15

5. Junction Panel Connections (EIA)

EIA Standard RS-232-C defines the interface between a Data Set (e.g., modem) and a Data Terminal (e.g., CRT). The EDS-8 multiplexer can communicate with both data sets and data terminals; however, the 25-pin connectors on the junction panel are wired to make the EDS-8 look like a data set. Therefore, data terminals may be plugged into it directly, but data sets must be connected through a cable which interchanges certain signals. This is explained in the chart below. The connectors on the junction panel are 25-pin Cannon-type "D" female connectors, model DB-25S.

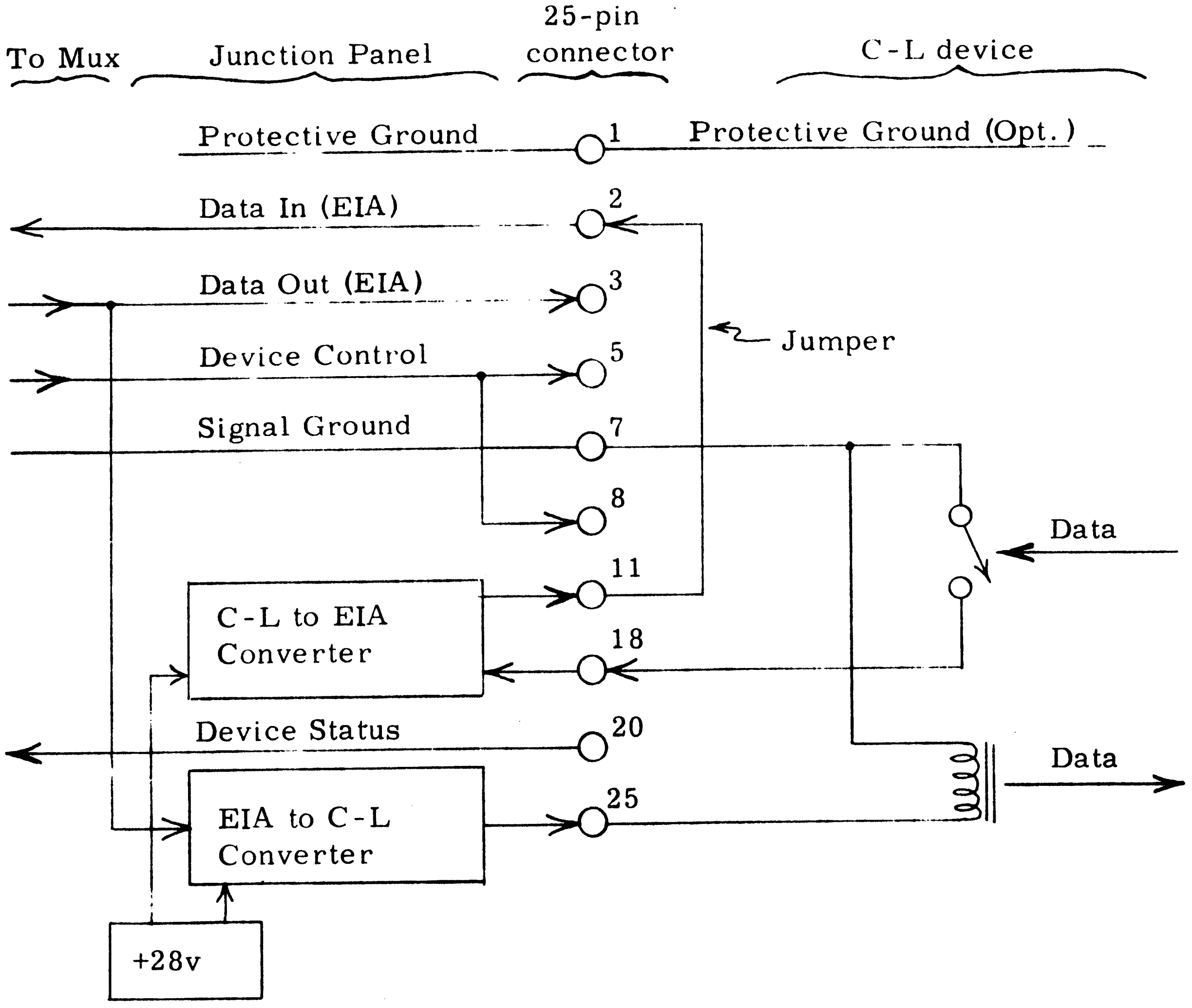
MUX end		Other end					
		Data Terminal		Data Set (Full Duplex)		Data Set (Half Duplex)	
Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
2	Data In (to Mux)	2	Transmitted Data	3	Received Data	3	Received Data
3	Data Out (from Mux)	3	Received Data	2	Transmitted Data	2	Transmitted Data
5	Device Control	5	Clear to Send		--------	4	Request to Send
7	Signal ground	7	Signal ground	7	Signal ground	7	Signal ground
8	Device Control (same as pin 5)	8	Received Line Signal Detector	20	Data Terminal Ready		----*
20	Device Status	20	Data Terminal Ready	8	Received Line Signal Detector	8	Received Line Signal Detector

*In half-duplex systems, pin 20 of the Data Set must be connected to a +5 to 15 volt source, or jumpered to Data Set pin 6 (Data Set Ready).

6. Junction Panel Connections for Current-Loop

The EDS-322-CL Cable Assembly allows the user the versatility of plugging cables requiring either EIA-level or current-loop interfacing into the same 25-pin connector. This is achieved by using pins 11, 18, and 25 (left "unassigned" by EIA spec RS-232-C) for the current-loop (C-L) connections (see Figure below). Pin 25 carries the C-L output from the Mux, and Pin 18 receives the C-L input to the Mux, with Pin 7 (Signal Ground) serving as the return for both. Pin 11 must be jumpered to Pin 2 whenever C-L is used; this is best done inside the 25-pin connector on the C-L cable.

The C-L circuitry on the 322-CL is driven by a +28 volt source, and is designed for 20 ma current in both input and output. This may be increased to 40 or 60 ma by adding two 1.3K (1 watt) or 680 ohm (2 watt) resistors on the 322 board next to each 25-pin connector for which the change is desired.



Copyright (C) 1974
Educational Data Systems

7. Input and Output Timing and Voltage Specifications

The EDS-8's I/O timing specs are determined primarily by the Universal Asynchronous Receiver/Transmitter (UART) chips (Western Digital 1602 B or equivalent) used in each Port and the electrical specs are determined by the EIA/TTL converter chips (Motorola 1488 and 1489 or equivalent). The interrelationship between these circuits is shown in Figure A-2.

The basic timing source is a 6.7584 MHz ±.005% crystal oscillator. This is followed by a frequency divider producing 16 times the selected Baud rate, which in turn drives the UART chip. The UART receiver is so designed as to allow up to almost 47% time distrotion, while the transmitter produces less than 1% distortion. This makes the EDS-8 compatible with virtually all asynchronous modems and terminals in use today.

The voltage levels of the EDS-8's standard I/O circuitry are in accordance with EIA Specification RS-232-C. The 1488 chips are driven from a ±12 volt source, and thus the outgoing levels are:

Data = 0 or Device Control = 1 : +10 volts
Data = 1 or Device Control = 0 : -10 volts

For incoming lines the 1489 chips give the following response:

>+1.3 volts : Data = 0 or Device Status = 1
<+0.7 volts or open : Data = 1 or Device Status = 0
Between +.7v and +1.3v : Indeterminate.

The 1489 offers an input resistance of about 3.5K; the 1488 can drive any load above 2K.

The optional EDS-8 Current-Loop interface is driven from a +28 volt source through a current-limiting resistor of 1.4K for the 20 ma version (or 700 ohm for 40 ma or 470 ohm for 60 ma). These values apply to both the input and output circuits, as shown in Figure A-3.

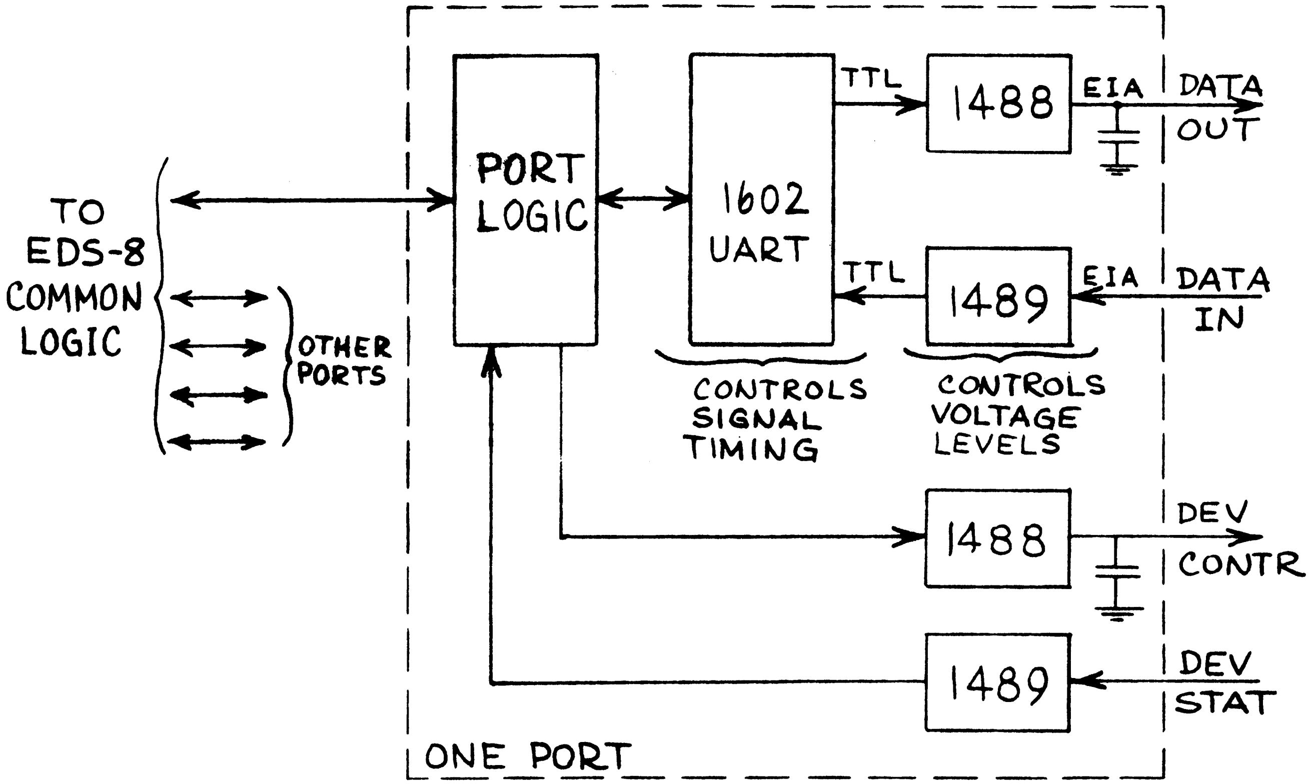


Figure A-2 Block Diagram of EIA Interface

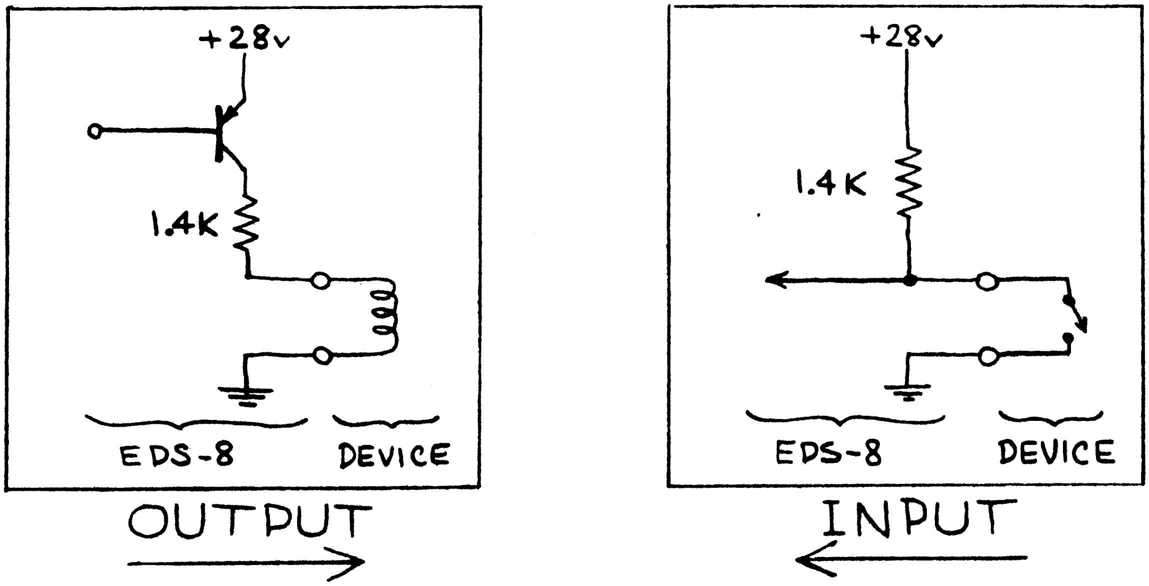


Figure A-3 Current-Loop Circuitry

8. Overall Mux Timing

When no input or output is taking place, the Mux inspects each OCW about once every 50 msec. Therefore, when the software starts an output by setting OCW to an active output mode, there may be a delay of up to 50 msec before the Mux begins transmission. Once transmission has begun on a port, the Mux will reinspect that OCW for the next output command as soon as the output buffer register of that port is empty.

If the Device Status changes while there is no output on that port, there is again a delay of up to 50 msec before the Status Changed bit is posted and an interrupt is produced. When output is in process, however, a Device Status change produces the interrupt as soon as transmission of the current character is completed - i.e., at the time the Mux reexamines OCW to obtain the next character to be output.

When an incoming character is received, the Mux will store it away - and produce an interrupt, if appropriate - within at most 1 msec after the center of the first stop baud.

The relationship between number of ports and maximum data rate per port is shown in Figure A-4. This figure assumes that all ports operate simultaneously at the same data rate. If some of the ports have a lower data rate, a correspondingly larger number of ports may operate simultaneously. Thus, for example, 80 ports at 9600 Baud impose about the same load as 60 ports at 9600 Baud plus 40 ports at 4800 Baud. (This is not an exact relationship, however, and it is wise to allow a reasonable safety margin when making such a conversion.)

When the Mux operates at the maximum total data rate according to Figure A-4, it produces about 80% overhead, since it always takes 4 or 5 successive data channel cycles (see Figure 3 - Flow chart) and then gives up the data channel for one cycle. If the total data rate is less than the maximum, the overhead ratio goes down in direct proportion. Thus, for example, 10 ports at 9600 Baud plus 80 ports at 1200 Baud give about 20% overhead on a Nova 1200. Similarly, 8 ports operating simultaneously at 1200 Baud would produce about 1% overhead on a D-116.

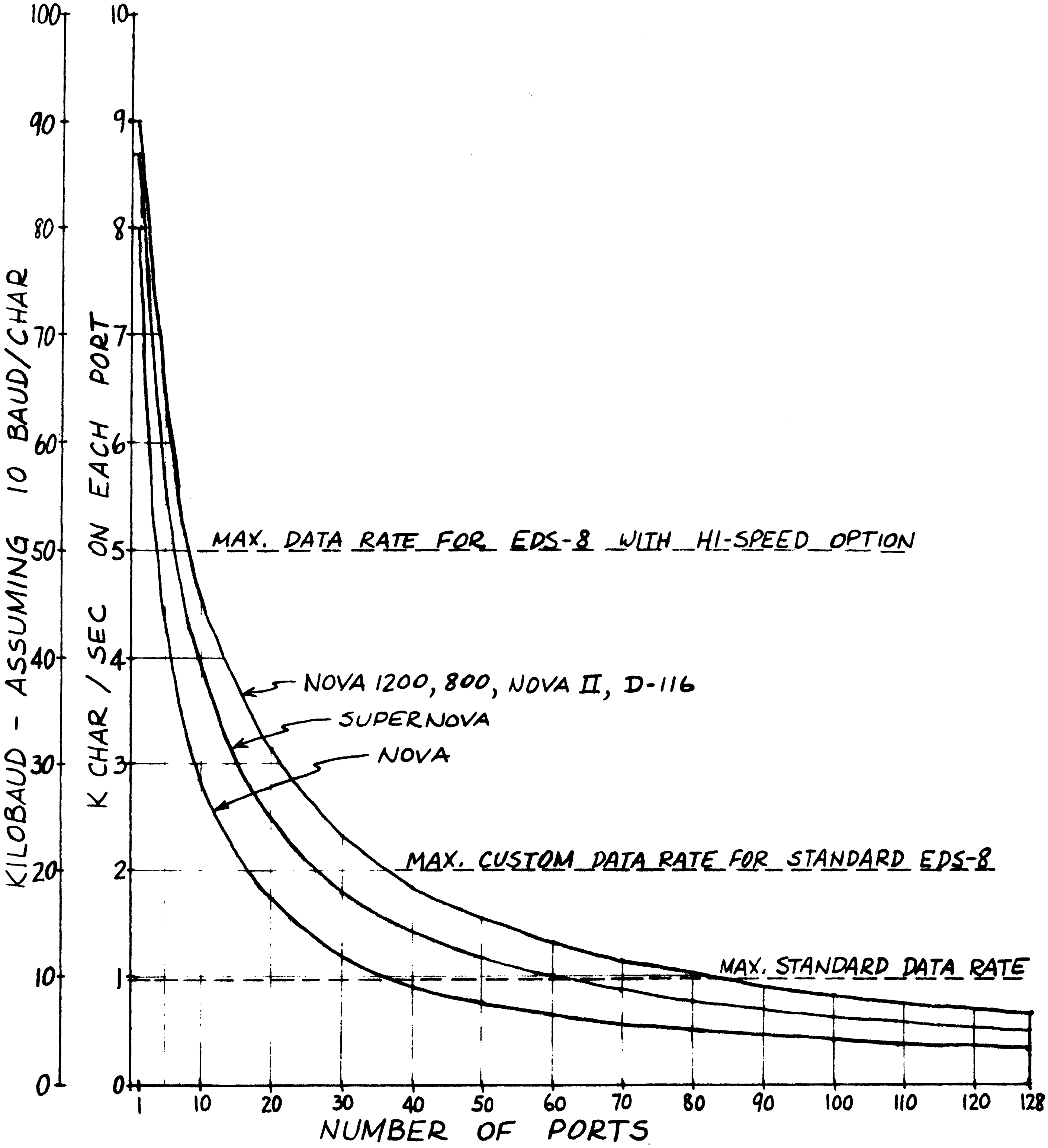


Figure A-4 Relationship Between Number of Ports
and Maximum Data Rate per Port

Appendix B - Installation and Trouble Shooting

1. Installation

a) Set the DIP switches on the 300 board (and 301, if any) to the desired Control Block Locations as explained in Appendix A.

b) With computer power off, insert the 300 board (and 301's, if any) into any slot in the computer chassis.

c) If there are any blank slots in the computer below the 300 board, the Interrupt Priority and Data Channel Priority signals must be jumpered up to the 300 board on the computer's back plane.

d) If any 301 boards are used, they must be connected to the 300 by installing jumpers on the computer's back plane so that the following 41 points on the slot containing the 300 are connected to the corresponding points on any slots containing 301's.

A47
A49
A57
A59
A61
A63
A65
A67
A69
A71
A73
A75
A76
A77
A78
A79
A81
A83
A85
A87
A89
A91

B6
B13
B15
B19
B23
B25
B31
B34
B36
B37
B40
B48
B49
B51
B52
B53
B54
B67
B69

e) Mount the Junction Panel(s) (ribbon cable on top) and Power Supply in a convenient place. Connect the 5-conductor cable(s) from the power supply chassis to the Molex connector on each printed circuit board on the junction panel(s). Attach the free end(s) of the 50 conductor ribbon cable(s) to the connector on the 300 board (and those on the 301's, if any). The ribbon cable may be connected with the cable extending either downward or upward - the only effect of reversing the cable is to reverse the numbering of the eight connectors on the junction panel. Normal left-to-right numbering corresponds to the cable extending upward; this is done so that the cable may be draped over the computer and then to the rear, so that the computer may still be slid in and out of a rack.

f) Plug in the power supply line cord(s), preferably into the rear of the computer so that when the computer is off, no voltage comes to the EDS-8 through the ribbon cables.

The EDS-8 is now ready for use.

2. Trouble Shooting

If the EDS-8 appears not to be working correctly, or if no software to drive it is available, the following elementary test may be used.

Mini - Test Using Computer Front Panel

Set Switches	Press	Comment
	Reset	
Any ICW	Examine	
0	Deposit	
40000	Deposit Next	OCW = Port Control Word
63025	Deposit Next	DOC 0,MUX; turns Mux on
400	Deposit Next	JMP .+0; allow data channel action
ICW + 2	Start	
	Reset	
ICW	Examine	Should still be 0
	Examine Next	OCW should be 140000

If OCW is 146400, it means that the incoming Device Status line has a positive voltage on it and the Mux is working correctly. If ICW is 100777, it is a very strong indication that the -12 volt supply is not getting to the EDS-300 board (especially if the same occurs in all ICW's). This may be checked with a voltmeter or an oscilloscope, at pin 2 of any of the 40-pin UART chips (type 1602). If ICW and OCW are still 0 and 40000, respectively, then the following more extensive test program should be tried.

```
 0:   20416  LDA    0,.+16   ;pick up control word (40000)
 1:   30416  LDA    2,.+16   ;pick up last address
 2:    4416  JSR    .+16     ;pick up first address in A3
 3:   41400  STA    0,0,3    ;store 40000 in all words
 4:  175400  INC    3,3
 5:  172032  SGE    3,2      ;last address reached?
 6:     775  JMP    .-3      ;no, continue storing
 7:   63025  DOC    0,MUX    ;yes, start Mux
10:    4410  JSR    .+10
11:   25400  LDA    1,0,3    ;check all words
12:  106414  SEQ    0,1
13:   63077  HALT            ;word ≠ 40000
14:  175400  INC    3,3
15:     774  JMP    .-4
16:   40000  40000           ;control word stored everywhere
17:   77600  77600           ;last address--may be changed
20:    5400  JSR    0,3
```

This program stores 40000 (PCON output or Auto input) in each word above the program up to the address stored in location 17, then starts the Mux and tests each word to see if it has changed. If so, it halts, with the changed word in A1 and its address in A3.

Press Reset and start at 0.

The program should halt within a second or so, with A3 containing the address of the first OCW, and A1 containing either 140000 or 146400, according as Device Status is 0 or 1. If Continue is then pressed, it should halt at the next OCW, etc., until finally it will halt at the top of core, i.e. with A3 containing the number in cell 17.

If the program halts with an ICW address in A3 it indicates that data (or noise) is coming in, or - if the data is all ones - that the -12v supply is missing. In these cases it may also halt at the corresponding IBP which may have been incremented to 40001.

If the program halts with any other value in A3 it indicates that the address switches on the 300 or 301 boards are not set correctly.

If it halts only at the top of core it may indicate that the address switches are set to a value greater than the amount of core available, or that the Mux board is not plugged all the way in, or that it is not getting a +5 volt supply from the computer.

If the program does not halt at all it indicates that the Mux is never releasing the data channel, the most likely cause being that the Data Channel Priority signal is not getting to the 300 board.