digital

hardware

KL8-JA terminal control/asynchronous data interface technical manual

First Edition, August 1977

Printed in U.S.A.

This document was set on DIGITAL'S DECset-8000 computerized typesetting system.

KL8-JA terminal control/asynchronous data interface technical manual

EK-KL8JA-TM-001

digital equipment corporation • maynard, massachusetts

First Edition, August 1977

Printed in U.S.A.

This document was set on DIGITAL'S DECset-8000 computerized typesetting system.

CONTENTS

FIGURES

TABLE

CHAPTER 1
KL8-JA TERMINAL CONTROL/ASYNCHRONOUS DATA INTERFACE

1.1 GENERAL DESCRIPTION

The KL8-JA is a double-buffered data interface for use between a PDP-8 OMNIBUS (including PDP-8/A and PDP-8/S equipped with DW8-E Bus Converters) and any asynchronous external device with electrically compatible data leads and one of the many serial data formats available with this interface. Its function is to convert parallel data to serial-data format (transmit) while simultaneously converting serial data to parallel-data format (receive).

The KL8-JA transmits at rates of 110, 150, 300, 600, 1200, 2400, 4800, and 9600 baud, providing either 20 mA current loop operation, or EIA/TTL level conversion to the external device. The type of conversion is determined by cable selection. Character lengths can be 5-, 6-, 7-, or 8-bits, with or without either odd or even parity. Stop codes can be either 1 or 2 units, except when used with 5-bit codes, in which cases stop codes are set to 1 or 1.5 units. Line parameters are jumper-selectable. Jumpers are also used for error status word reporting of parity, overrun, and framing errors. Board-mounted switches select baud rate and device codes.

The unit is strapped to generate four filler characters as a result of a line-feed character permitting its operation with the VT05 CRT terminal and other terminals requiring this feature.

The KL8-JA is a prewired system unit consisting of one Quad Module Type M8655 that provides all the logic to perform the functions described. EIA to TTL or 20 mA to TTL conversion is provided on input and output lines. For modem operation, the Data Terminal Ready and the Request To Send lines are held asserted.

KL8-JA character transfers are conducted in full-duplex mode (simultaneous transmit and receive). Paper tape reader control is also provided for use with LT33 DC and DD model Teletypes®.

Operation of this interface with the PDP-8 is via programmed I/O. The skip and interrupt request lines of the OMNIBUS are used and data transfers are made between the PDP-8 accumulator and registers within the M8655 Module.

The KL8-JA is discussed here only to the extent necessary to fully describe the interface operation.

1.2 INSTALLATION AND CHECKOUT

The KL8-JA Terminal Control/Asynchronous Data Interface is installed on-site by DEC Field Service personnel. No attempt should be made by customers to unpack, inspect, install, checkout, or service the option.

The M8655 Module is to be inserted into the PDP-8 OMNIBUS. See Table 2-3, Volume 1 of the *PDP-8/E Maintenance Manual* for information concerning recommended module priorities (the KL8-JA is a 'non-memory option').

®Teletype is a registered trademark of Teletype Corporation.

Proceed as follows:

Step	Procedure
1	Unpack the module and inspect it for any visible damage. Report any damage or omissions immediately to the DEC Field Service Representative.
2	Set the RECEIVE and TRANSMIT switches on the module per the customer-specified device codes. Refer to Sheet 1 of the module circuit schematic.
3	Set the baud rate on the module per the customer's requirements. Refer to Sheet 1 of the circuit schematic.
4	Install jumpers per the customer's requirements for parity, even parity, bits/character, fill characters, TTY, and error status word. Refer to Sheet 1 of the circuit schematic.
5	In the PDP-8/E/F/M, remove power and insert the module in the OMNIBUS slot specified in Table 2-3 of the PDP-8E Maintenance Manual.
6	Load MAINDEC-08-DIKLA-A-PB (Loop Back Test) using normal binary loading procedures.
7	Run the diagnostic per the MAINDEC printout MAINDEC-08-DIKLA-A-D. Run at customer's specified baud rate for one pass in 20 mA mode, and one pass in EIA mode. No errors should occur.

NOTE
For these tests make the following J1 connections:

For 20 mA loop back test
E – H
K – KK
S – AA

For EIA loop back test
E – M
F – J

8a	If the KL8-JA is shipped with a Teletype, load MAINDEC-08-DIKLB-A-PB, using normal loading procedures. Run Program 4 per the description in MAINDEC-08-DIKLB-A-D. No errors are acceptable.
8b	If the KL8-JA is shipped with a VT05, load MAINDEC-08-DGV5A-B-PB, using normal loading procedures. Run the diagnostic for one complete pass per MAINDEC-08-DGV5-B-D. No errors should occur.
8c	If the KL8-JA is shipped with a serial LA30, load MAINDEC-08-DHLAA-A-PB, using normal loading procedures. Run the diagnostic for one complete pass per MAINDEC-08-DHLAA-A-D. No errors should occur.

1.3 FUNCTIONAL DESCRIPTION

Figure 1-1 is a block diagram of the KL8-JA Terminal Control/Asynchronous Data Interface. The functions of this device can be conveniently grouped according to transmitter and receiver functions, as illustrated in the block diagram.

The unit utilizes a Universal Asynchronous Receiver/Transmitter (UART) that interconnects through level converters and a cable to the external terminal. The UART performs the serial-to-parallel and parallel-to-serial conversion of data that is received from or sent to data buffers. These buffers, in turn, interconnect with the OMNIBUS. The buffers are either fed by or feed a second set of buffers within the UART, providing double buffering of data in this interface. Timing for the UART is provided by a timing generator within the KL8-JA.

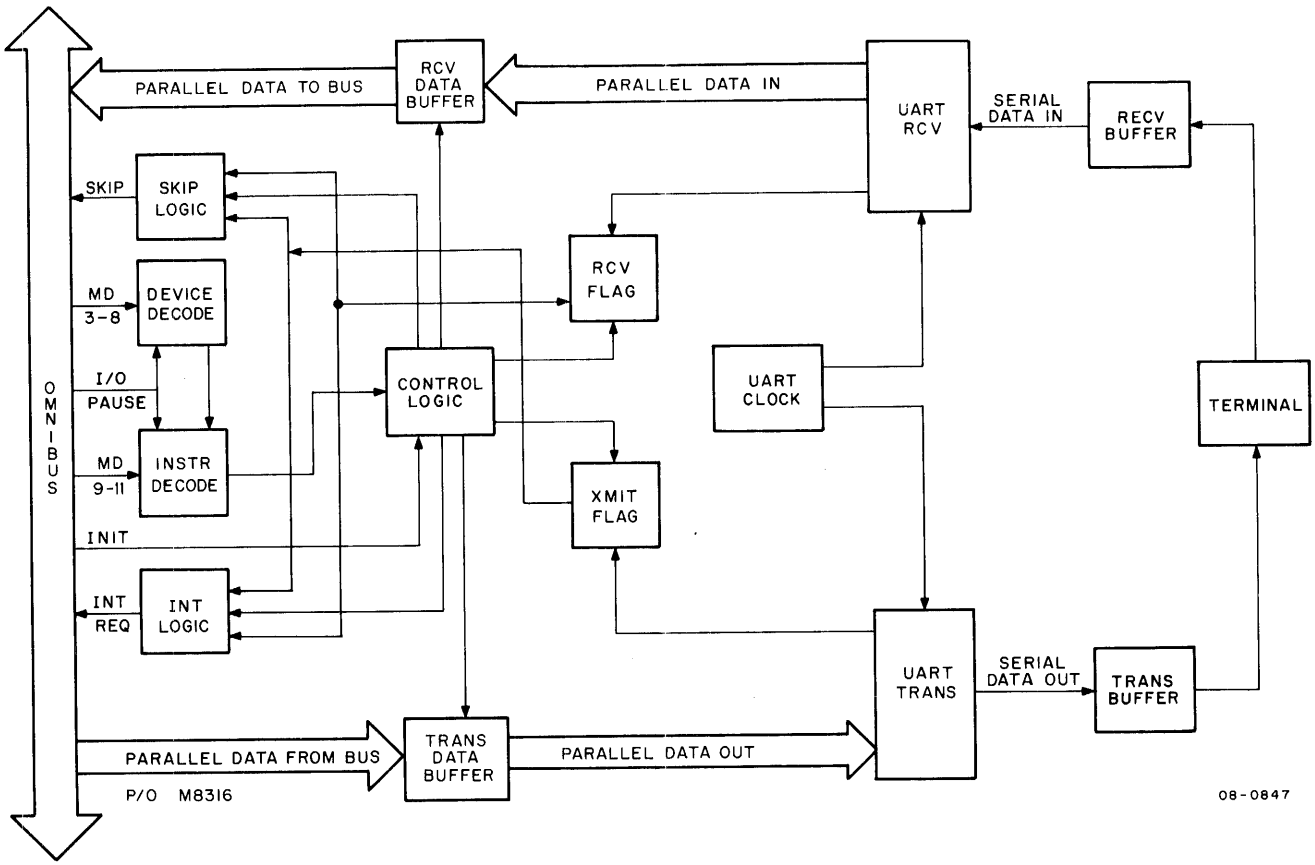


Figure 1-1 KL8-JA Block Diagram

Communication with the PDP-8 Processor via the OMNIBUS is accomplished by instruction decoding networks within the KL8-JA. Upon receipt of the signal I/O PAUSE, the interface is enabled to compare MD03–08 with the "receive" and "transmit" switches. The result of this comparison determines which device code has been issued.

When either of these device codes are decoded, a second level decoder is enabled to interpret the specific instruction for that device. These instructions are decoded from bits 09–11 of the Memory Data (MD). These outputs are fed to a Binary-to-Octal Decoder from which combinations of control signals are distributed to various sections of the interface. Two outputs of this decoder enable either the Receive or Transmit Data Buffer in the interface. It also generates a signal that disables the KA-8E Option (INTERNAL I/O). ('INTERNAL I/O').

In the KL8-JA, two UART signals are used to set the appropriate flags. One sets the Receiver flag and is an indication that the UART has assembled a complete character in its receive buffer. That character can now be transferred in parallel into the Receive Data Buffer. A reset signal is issued to the UART once the transfer between the Receive Data Buffer and the ac is complete.

On the transmit side, another signal is issued that sets the Transmit flag whenever the parallel character in the Transmit Buffer has been shifted out to the device. This signals the interface that another parallel character may be transferred from the Transmitter Data Buffer into the Transmit Buffer within the UART. The flags are cleared from the Binary-to-Octal Decoder when appropriate instructions are received.

The KL8-JA also provides for the generation of four filler characters (nulls) as a result of the sensing of a line feed code. This feature is a jumperable option and is used in conjunction with VT05 Terminals. The KL8-JA will transmit the line feed followed by four null characters. During the transmission of the last null, the transmit flag is set.

The Status Word Enable Option (jumper selectable) permits the loading of error status into AC00–03 when read data instructions are decoded.

The operation of the KL8-JA is similar to that of other Terminal Control Interfaces. INITIALIZE (key clear or CAF 6007 instruction) clears the RCV FLAG, TRANSMIT FLAG and Status Word Enable flip-flop, if applicable. It also sets the Interrupt Enable flip-flop. INITIALIZE does not reset the transmit or receive circuitry; i.e., if the M8655 were in the process of transmitting or receiving a character, the respective flag is set at the appropriate time despite the issuance of INITIALIZE. This circuitry is cleared only when power is first applied to the PDP-8.

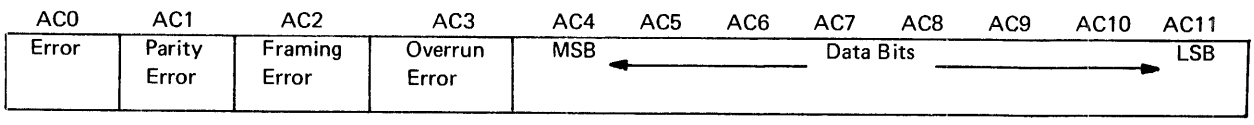
The status word section is operable only when the SWD jumper is installed on the M8655. When this jumper is out, the read status logic is disabled. Error status is read with the data bits when a read IOT is issued (KRS or KRB) if the Status Enable flip-flop was previously set.

AC0 Inclusive OR of the three error conditions, 1=error.

AC1 Parity error (If NP jumper is not installed, this bit will always receive a zero.)

AC2 Framing Error = 1 if a legal stop bit was not detected (a space was detected half way through Stop Bit 1).

AC3 Overrun Error = 1 if the RCV FLAG was not cleared prior to the character now being read (one character transmitted after another by the Teletype without the first being read by the computer).



AC After KRS or DRB Instruction With Status Enabled

1.4 DETAILED LOGIC DISCUSSION

1.4.1 Instruction Decoding and Flags

This logic is shown in detail on Sheet 2 of drawing D-CS-M8655-0-1 and in simplified form in the figures that follow.

1.4.1.1 Device Decoding – The device selection logic (Figure 1-2) decodes MD03 through MD08 of the instruction. The states of the input bits are ANDed with B PAUSE (a function of I/O PAUSE) to feed two sets of XORs, one for transmit decoding and one for receive decoding. Each XOR will yield a low output only when its inputs differ, and a high output when its inputs are the same. Since each switch will ground, or pull low, one of the XOR inputs, a gate, will yield a high output only when its associated switch is set to the same state as the bit to be decoded. Therefore, if a zero is to be decoded, the switch should be open rendering the output XOR pin high; and if a one is to be decoded, the switch should be closed causing the output pin to be pulled low.

NOTE

MD bits are low when a (1) and high when a (0).

By referring to the block schematic, D-CS-M8655-0-1 (Sheet 2), it can be seen that when all switches except the switch associated with bit 06 are open in the Transmit Decoder, and only when all switches except the switches associated with bits 07 and 08 are open in the Receive Decoder, the outputs of each set of XORs will be high only when the proper digit is present at the inputs of the 5384 AND gates. The result is that the line from the common outputs of the Transmit Decoder that feeds pin D2 of the Instruction Decoder will be high when a transmit device code is seen and low when a receive is present. This line determines which set of function outputs are generated by the Instruction Decoder described in Paragraph 1.4.1.2.

When either a receive or a transmit operation is decoded, the signal THIS MODULE SELECTED is generated to condition the instruction decoder logic; and the signal INTERNAL I/O is also generated as the AND of this signal and B PAUSE L to prevent the generation of IOPs during the time the device is selected.

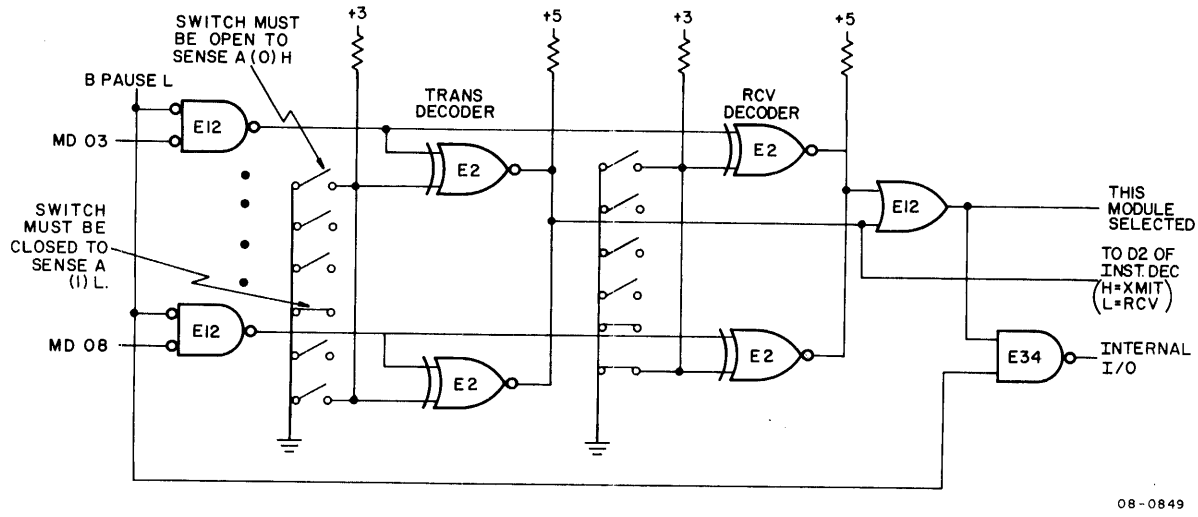


Figure 1-2 KL8-JA Device Decoding

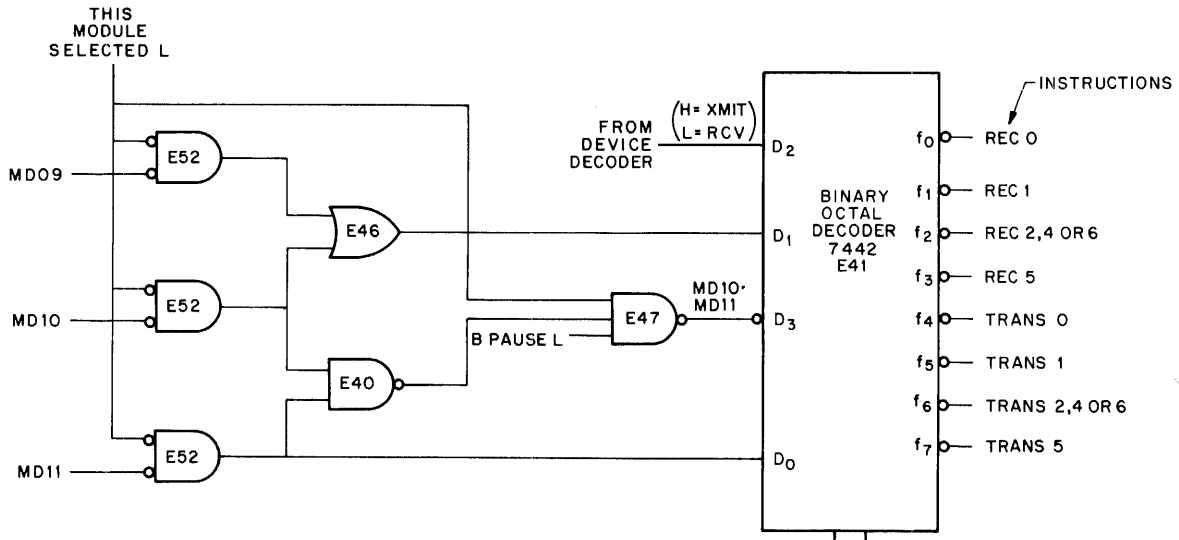
1.4.1.2 Instruction Decoding – Instruction decoding (Figure 1-3) is predicated upon the generation of the signal INTERNAL I/O L generated when the proper device code is sensed as described in Paragraph 1.4.1.1. When this signal is true, the MD09–11 gates are enabled to decode the specific instruction for the device decoded.

The states of MD09–11 condition the combinations at D0–D3 at the 7442 at E41. This chip is a 4-line to 10-line converter but in this application functions as a 3-wire binary-to-octal decoder, yielding the eight functions required (four for the receiver and four for the transmitter). When D2 is high the transmit instructions are decoded and when it is low the receive instructions are decoded. The combination at D3:

(the NAND of [–(MD10•MD11)•I/O PAUSE L•THIS MODULE SELECTED] L)

is used as an enable for the decoder. The state of MD10 is then used to condition two of the operations decoder input gates as described in Paragraph 1.4.1.3.

The operations to be performed are based on the requirements of the instruction set as described in Table 1-1. Figure 1-3 relates the instruction decoded to the outputs of the Instruction Decoder. Note from the truth table that only one output is low at any one time. When the D3 input is driven high, all outputs are high and no decoding takes place.



TRUTH TABLE

INPUTS				OUTPUTS							
D3	D2	D1	D0	F0	F1	F2	F3	F4	F5	F6	F7
0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1
0	1	1	0	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	0

08-0850

Figure 1-3 KL8-JA Instruction Decoding

Table 1-1 KL8-JA Instruction Set

Octal Inst.	Mnem	Decoder Output	Operation
6030	KCF	RECV 0	Receiver flag is cleared without clearing the AC or enabling the reader.
6031	KSF	RECV 1	Skip if Keyboard flag is set. Increments the CPU to one location beyond the next sequential instruction if the Receiver flag is set.
6032	KCC	RECV 2	Clear Keyboard flag and set Reader Run. Clear the Receiver flag, and AC, and enable the reader.
6034	KRS	RECV 4	Read keyboard static. Performs an Inclusive OR of the receiver register and the AC, leaving the result in the AC.
6035 (AC11)	KIE	RECV 5	Set/Clear Interrupt Enable. Loads AC bit 11 into the Interrupt Enable flip-flop. (1)=enable, (0)=disable.
6035 (AC10)	KSE	RECV 5	Set/Clear Status Enable. Loads AC bit 10 into Status Enable flip-flop. (1)=enable, (0)=disable. With SWD jumper installed, the Status Enable flip-flop set causes the status word to be loaded into AC bits 0 through 3 along with the received character with KRS or KRB instructions.
6036	KRB	RECV 6	Read keyboard buffer dynamic. Performs the combined operations of KCC and KRS.
6040	TFL	TRANS 0	Set Teleprinter flag. Sets the Transmit flag.
6041	TSF	TRANS 1	Skip on Teleprinter flag. Increments the CPU to one location beyond the next sequential instruction.
6042	TCV	TRANS 2	Clear Teleprinter flag. Clears the Transmit flag.
6044	TPC	TRANS 4	Load Teleprinter & Print. The least significant bits of the AC are transferred to a data holding register and then transmitted. The Transmit flag is not cleared by this instruction.
6045	SPI	TRANS 5	Skip if Teletype interrupt. The next sequential instruction is skipped if the Transmit or Receive flag is set and the Interrupt Enable flip-flop is set.
6046	TLS	TRANS 6	Print character. Combination of TCF and TPC performed.

1.4.1.3 Operations Decoding – The Operations Decoding (Figures 1-4 through 1-16) is performed by the combinational logic fed by the outputs of the Instruction Decoder described in Paragraph 1.1.2 as conditioned by the states of MD09 and MD10.

The initializing conditions are shown in Figure 1-4. Upon reception of INITIALIZE, select line S0 on the 8266 multiplexer causes its B inputs to appear directly at its f output pins. As can be seen from the figure, this stops the reader, clears the RCV FLAG, prevents the transmission of data, and clears the TRANSMIT FLAG flip-flop. In the processor the AC is also cleared. These are the initializing operations prior to the reception of an instruction.

Figure 1-5 illustrates the logical conditions resulting from the decoding of a RECV 0 instruction (KCF). In this case, select line S1 of the multiplexer is asserted low. This results in disabling of the reader run line, clearing of the Receiver flag; and since the RECV 2, 4, or 6 line is high, the C0 and C1 OMNIBUS control lines are both high, preventing the AC from being cleared.

The logical results of a decoded RECV 1 instruction (KSF) are seen in Figure 1-6. As in the rest of these instructions, S1 is asserted low, enabling the complement of the decoded instruction to appear at the output.

The high that results at E33 will AND with RCV FLAG (1) H (if set by RECEIVED DATA AVAILABLE from the UART) to assert the SKIP line causing the CPU to step to the next sequential instruction.

The RECV 2 (KCC) instruction decoding (Figure 1-7) causes the assertion of the Reader Run line and the clearing of both the AC and the Keyboard flag. The KRS instruction (RECV 4) is illustrated in Figure 1-8 and causes both the assertion of I/O READ DATA H and the inclusive ORing of the AC and Receiver Register, with the result stored in the AC.

With decoding of a RECV 5 instruction (Figure 1-9), two operations are performed. The state of AC bit 11 is loaded into the INTEN flip-flop and AC bit 10 is loaded into the STAT ENBL flip-flop. In both cases, the flip-flop will set or clear depending upon those states. Whenever the status enable is set, if the SWD jumper is installed, the status word is loaded into AC bits 0–3 along with the received character with either a RECV 4 (KRS) or RECV 6 (KRB) instruction.

Figure 1-10 shows the RECV 6 (KRB) decoding. Reader Run is set and the RCV FLAG flip-flop is cleared. The low at MD09 results in asserting both I/O READ DATA and C1. These combined functions of RECV 2 and RECV 4 are described in Table 1-1.

In the transmit instructions, the MD combinations are the same as for receive. Only D2 changes to a low for these instructions. The TRANS 0 instruction (TFL), shown in Figure 1-11, results merely in the setting of the TRANSMIT FLAG flip-flop at TP3.

The TRANS 1 instruction (TSF) produces a low on the SKIP line if the Transmit flag is set. This is simplified in Figure 1-12. All other conditions are negated and the C lines are both held high. As in the receive case, the Skip causes the CPU to skip the next sequential instruction.

When MD10 is low asserted (TRANS 2), the TCF instruction has been decoded. This results in clearing the Transmit flag (Figure 1-13). Figure 1-14 shows the TRANS 4 function (TPC), decoding. When this instruction is received, the least significant bits of the AC are transferred to a data holding register and then transmitted. The Transmit flag is not cleared.

Figure 1-15 illustrates that when a TRANS 5 instruction (SPI) is decoded, if the Interrupt Enable flip-flop is set, the SKIP line is asserted causing the next sequential instruction to be skipped if either the Receive flag or Transmit flag is set.

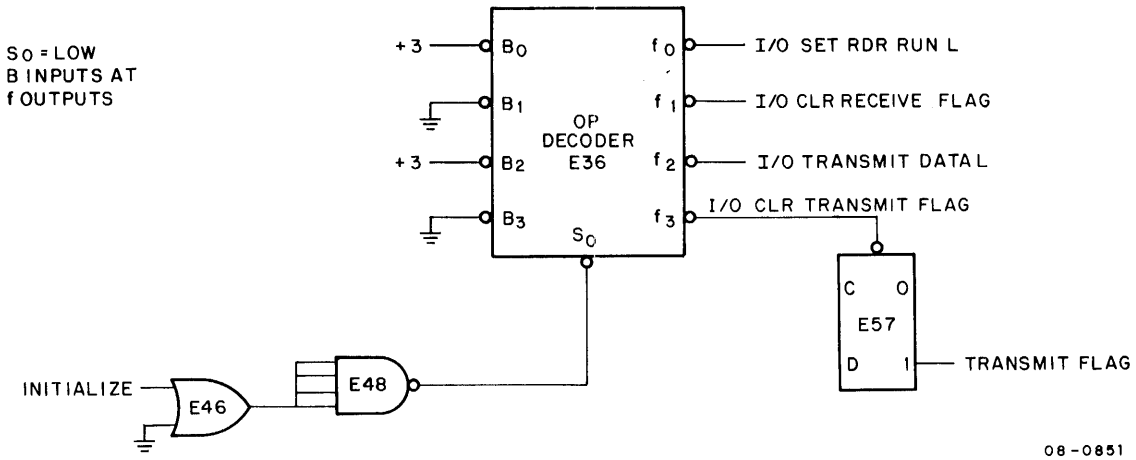


Figure 1-4 Initialize Conditions

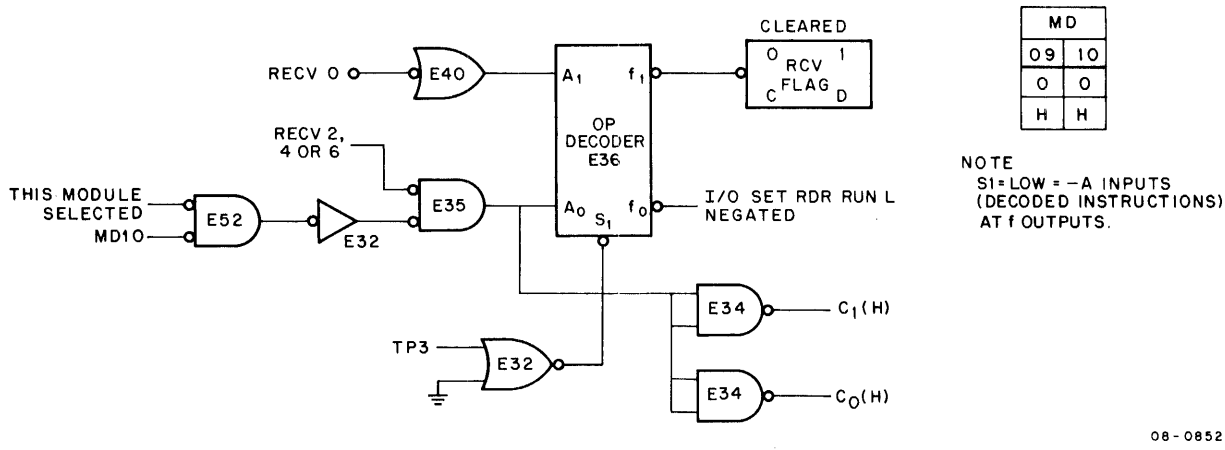


Figure 1-5 RECV 0 Decoding

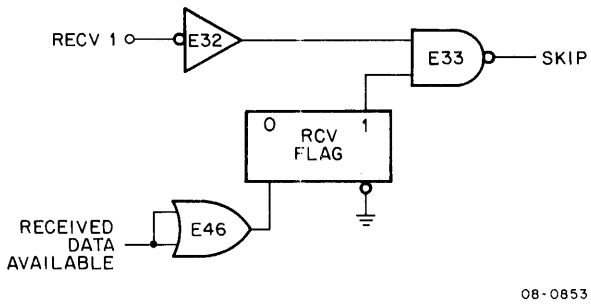


Figure 1-6 RECV 1 Decoding

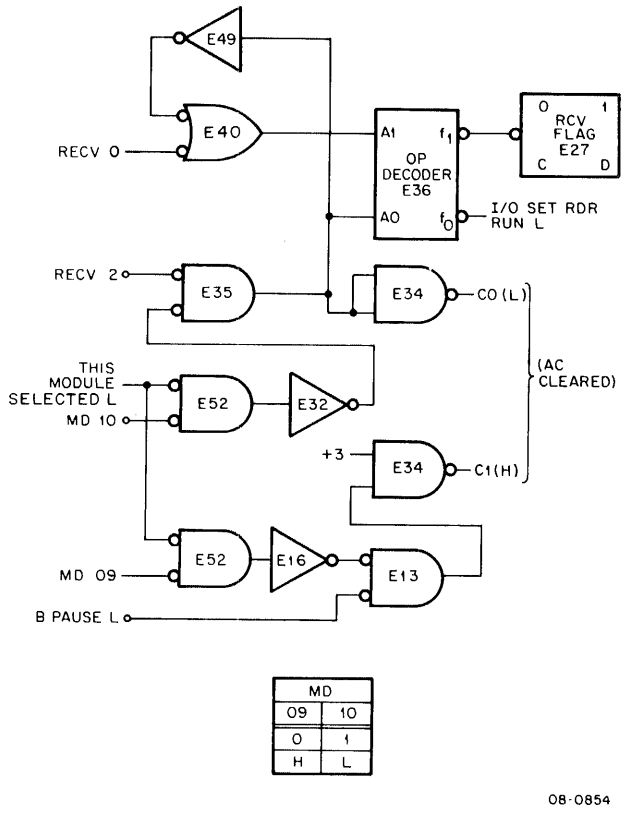


Figure 1-7 RECV 2 Decoding

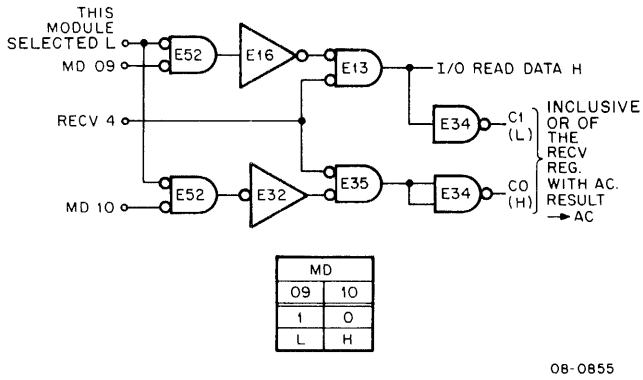


Figure 1-8 RECV 4 Decoding

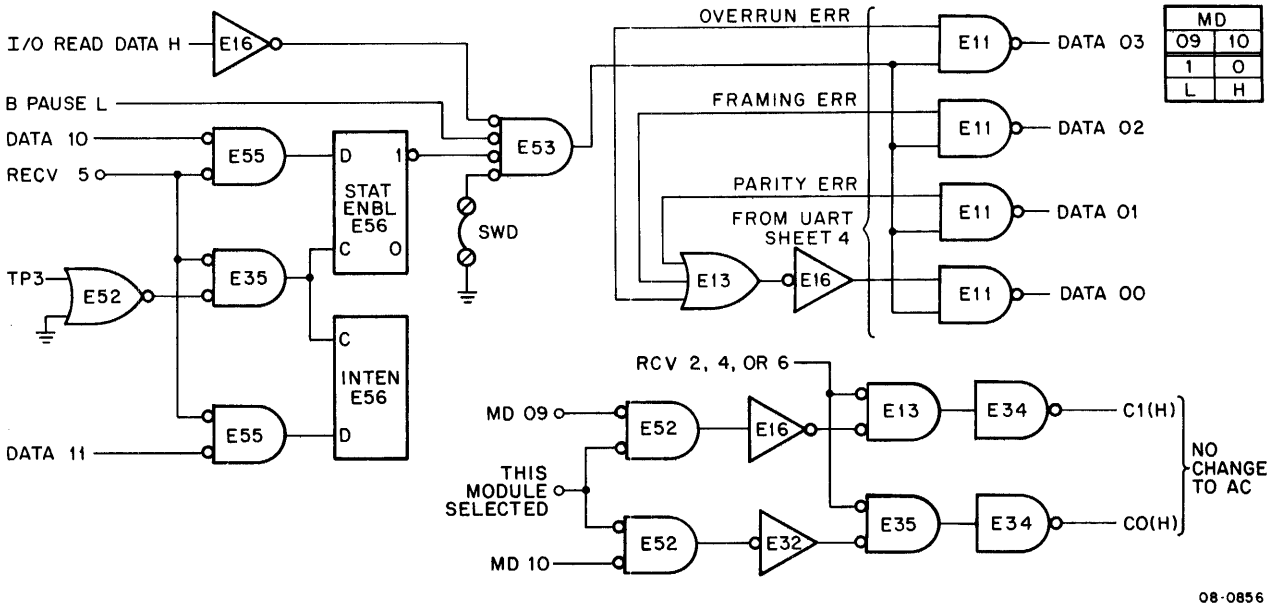


Figure 1-9 RECV 5 Decoding

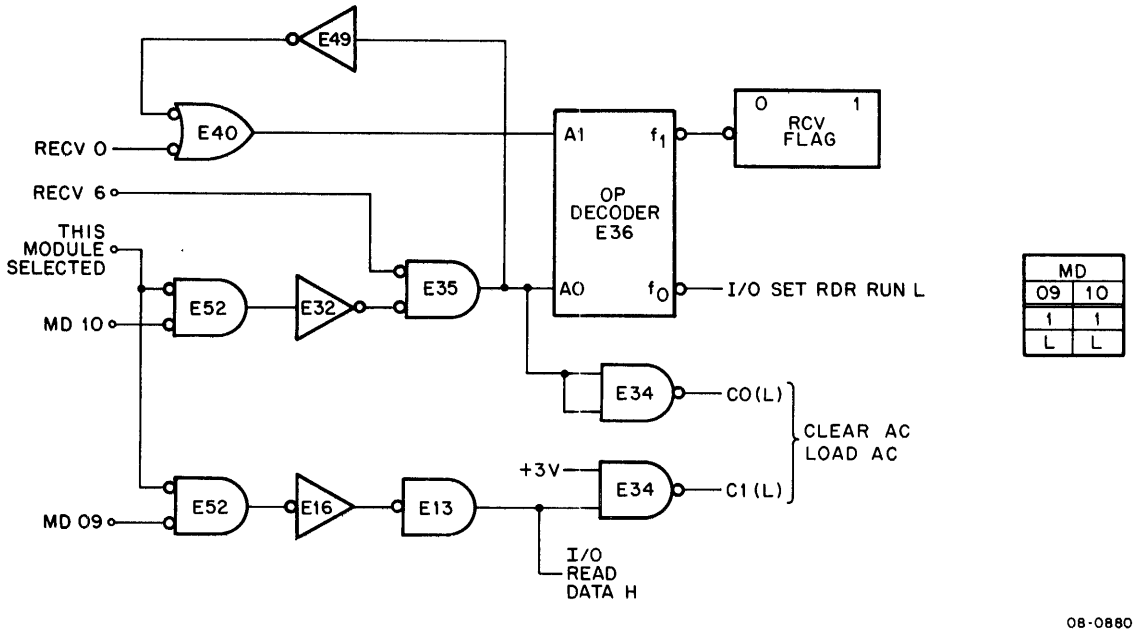


Figure 1-10 RECV 6 Decoding

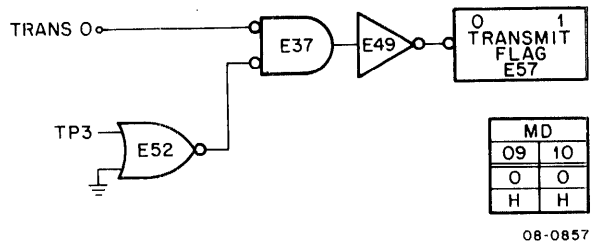


Figure 1-11 TRANS 0 Decoding

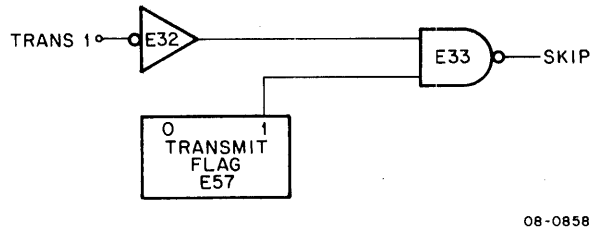


Figure 1-12 TRANS 1 Decoding

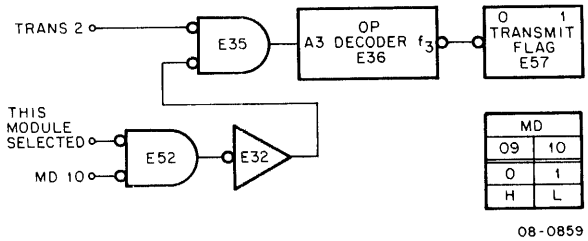


Figure 1-13 TRANS 2 Decoding

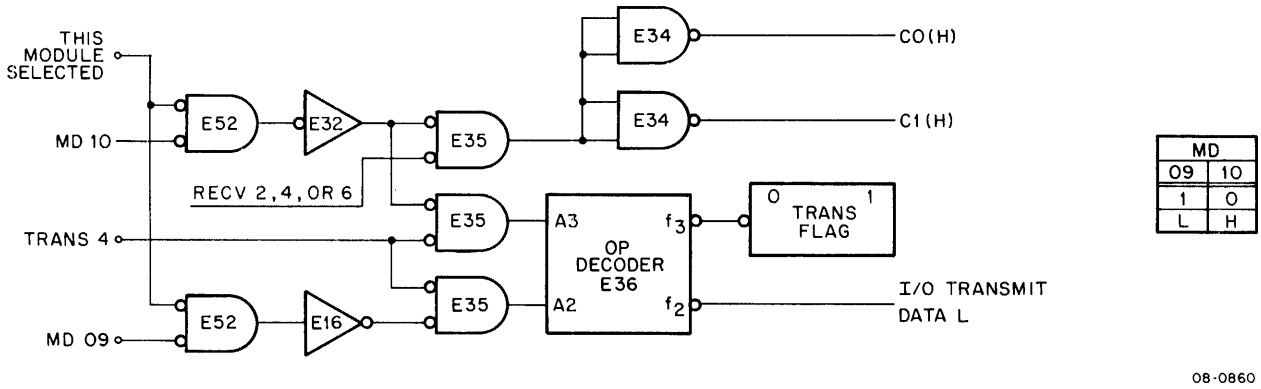


Figure 1-14 TRANS 4 Decoding

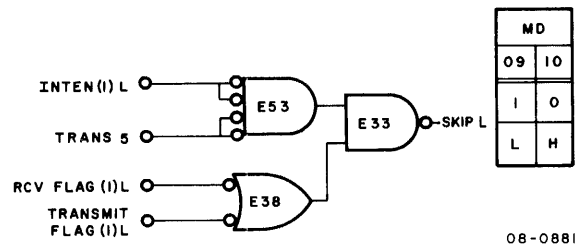


Figure 1-15 TRANS 5 Decoding

In Figure 1-16 the TLS (TRANS 6) instruction is simplified, showing that the results of both TRANS 2 and TRANS 4 are combined, in that both the character is printed (I/O TRANSMIT DATA L) and the Transmit flag is cleared in one operation.

1.4.1.4 Interrupt Control – The interrupt control logic functions to either enable or disable interrupts from the terminal. This circuit is simplified in Figure 1-9 and discussed partially in previous paragraphs. Every time a flag is set (either receiver or transmitter), the INT RQST L signal is asserted when the INTEN flip-flop is set. By changing the state of the INTEN flip-flop, the INT RQST L signal can be either asserted or inhibited. To inhibit INT RQST L, a 0 is first placed into AC11 and applied to the DATA BUS. DATA 11 L is then gated into the INTEN flip-flop by the IOT instruction 6035 decoded by the Operation Decoder. This causes the (1) output to go low and negate gate E33 to inhibit the INT RQST L signal. The other qualifying inputs to E33 are the states of the Receiver or Transmitter flag. If AC11 L is a (1), the INT RQST L signal is then applied to the interrupt control logic via the OMNIBUS as the interface signal that starts the interrupt system sequence of events. The INTEN flip-flop is direct set by INITIALIZE.

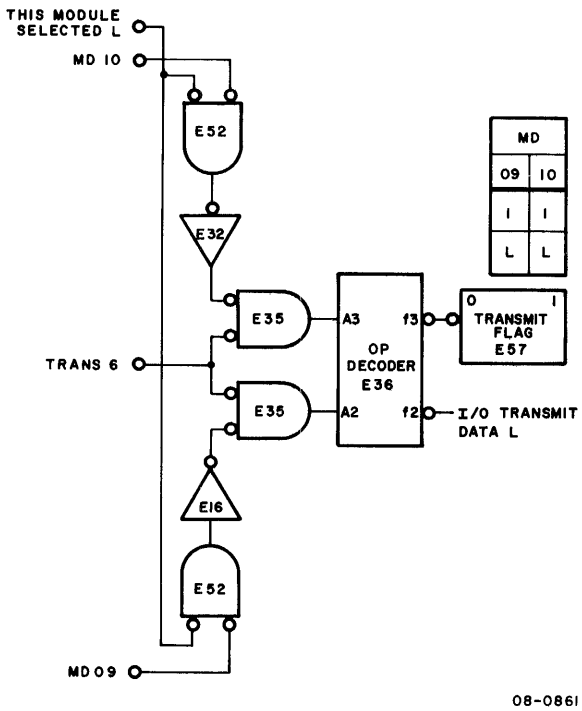


Figure 1-16 TRANS 6 Decoding

1.4.1.5 Skip Control – The skip control logic is shown in Figure 1-17. These portions of logic have been described in part in the preceding paragraphs but are shown here as a single circuit. A skip is generated upon decoding of an IOT instruction 6041 if the Transmit flag is set. This setting would have previously occurred during a 6040 instruction at TP3 if the character was within a line of text (i.e., no LF decoded) and the UART Transmit Buffer was empty.

NOTE

Upon decoding of a LF character at the output of the UART Transmit Buffer, four filler characters are generated causing FILCH to assert low if the "FIL" jumper is installed.

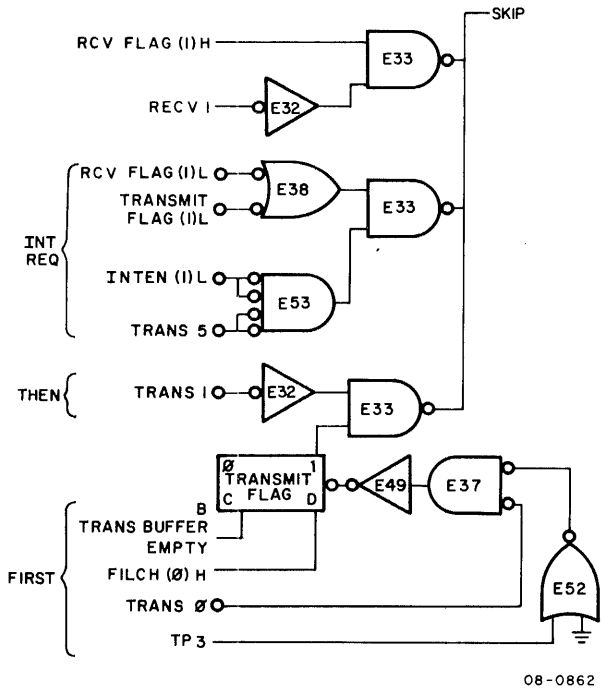


Figure 1-17 Skip Control Logic

A Skip can also be generated when the IOT instruction 6045 is decoded, if the Interrupt Enable flip-flop is set, and if either the Receive or Transmit flag is set.

A third condition for a SKIP signal exists when a 6031 instruction is issued and if the Receive flag is set. A SKIP to the processor will increment the PC causing the next sequential instruction in the program to be bypassed.

1.4.1.6 RCV FLAG – The RCV FLAG flip-flop (Figure 1-18) is tied to the UART Receive Buffer, described in more detail in Paragraph 1.4.3.2. Once the entire character has been assembled within the buffer, the UART issues the signal RECEIVED DATA AVAILABLE that is used to clock the flag against a grounded data input. The flag indicates that the UART Receive Buffer is full and that there is information that can be transferred to the AC register.

RCV Flag (1) H is gated with I/O instruction 6031 to assert the SKIP line. RCV Flag (1) L is ORed with TRANS Flag (1) L to assert INT REQ if the INTEN FF is set.

The flag is cleared by issuance of either a 6032, 6030 or 6036 that decodes to produce I/O CLR RECEIVE FLAG L. The high version of this signal is used to condition a master/slave arrangement at E20 to produce a reset signal to the UART. The TE of the signal I/O CLR RECEIVE FLAG sets the master that, in turn, enables the data input of the slave (Figure 1-19). When the signal P1.18 μs goes low, it clocks the slave, thereby sending RESET DATA AVAILABLE to the reset input of the UART Receive Buffer. The TE of P1.18 μs, a positive transition, then clears the master and its resetting clears the slave. The result is a 1.18 μs reset signal required by the MOS circuitry of the UART. It replaces the one-shot used in earlier designs and results in greater stability.

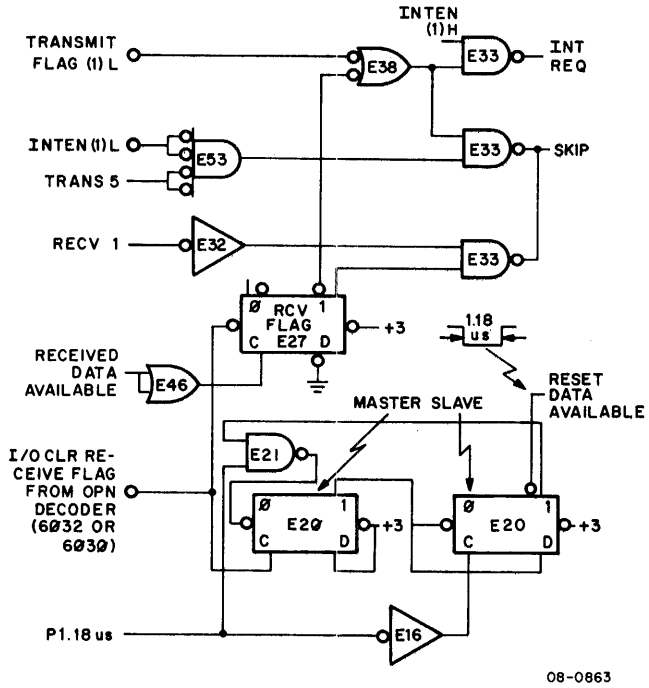


Figure 1-18 RCV Flag

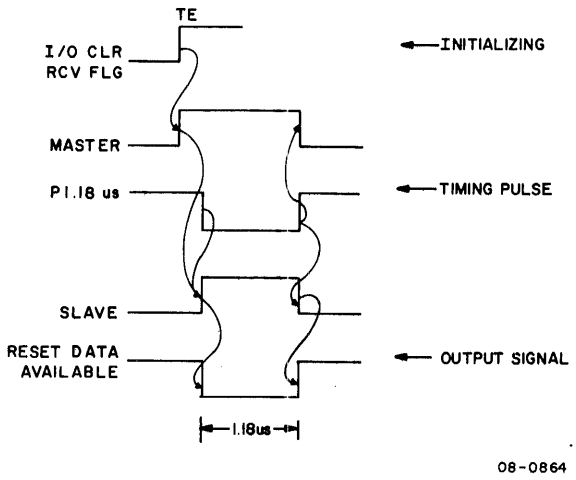


Figure 1-19 Master/Slave Arrangement Timing Diagram

1.4.1.7 Transmit Flag – The Transmit flag (Figure 1-20) is tied to the UART Transmit Buffer. Once the entire character has been shifted out of the buffer, the UART issues the signal TBMT that informs the processor that the UART is ready for another character. This signal is buffered to B TRANS BUFFER EMPTY and applied as a clock to the Transmit flag flip-flop. If the Filler Character flip-flop is cleared [FILCH (0) H], the flag will set (Paragraph 1.4.2.2). When set, it conditions assertion of the SKIP line during a TRANS 1 instruction (TSF).

The flag can also be direct set by TRANS 0 (TFL) on TP3. It is cleared by I/O CLR TRANS FLAG L as decoded from TRANS 2 (TCF) and 6 (TLS). A 6042 instruction should always be used prior to a new Transmit operation.

1.4.2 Timing and Filler Character Generation

This logic is shown on drawing D-CS-M8655, Sheet 3, and in block form in Figure 1-21. The circuit comprises a 5.0688 MHz oscillator that feeds both a Test Synchronization Test flip-flop, used for factory testing, and a Frequency Divider chain. The eight frequencies derived are then fed to a one-of-eight multiplexer that can be switch-set to yield one baud rate for receive and transmit. Means is provided to set the receive baud rate either to the transmit rate or to 150 baud. Two special outputs from the frequency dividers are used for establishing transmit data set time in the UART and to generate filler characters when desired.

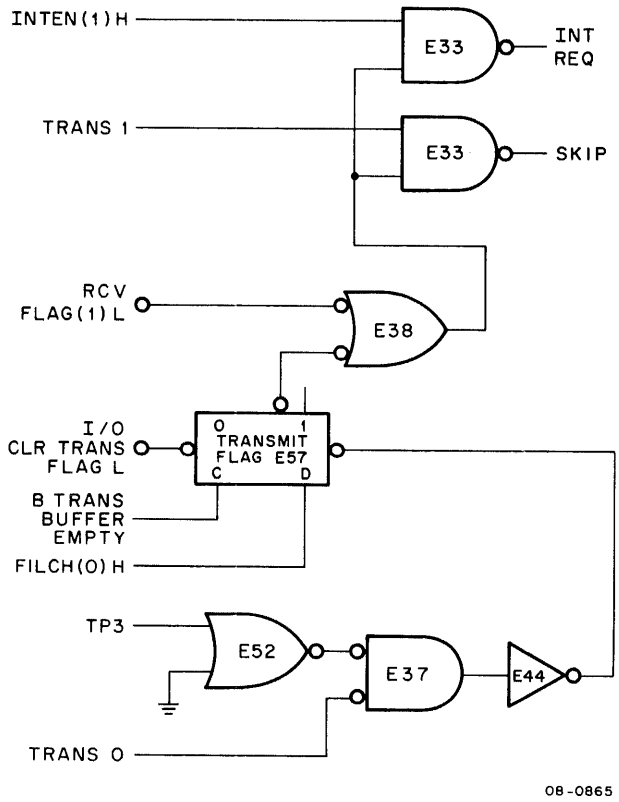


Figure 1-20 TRANSMIT Flag

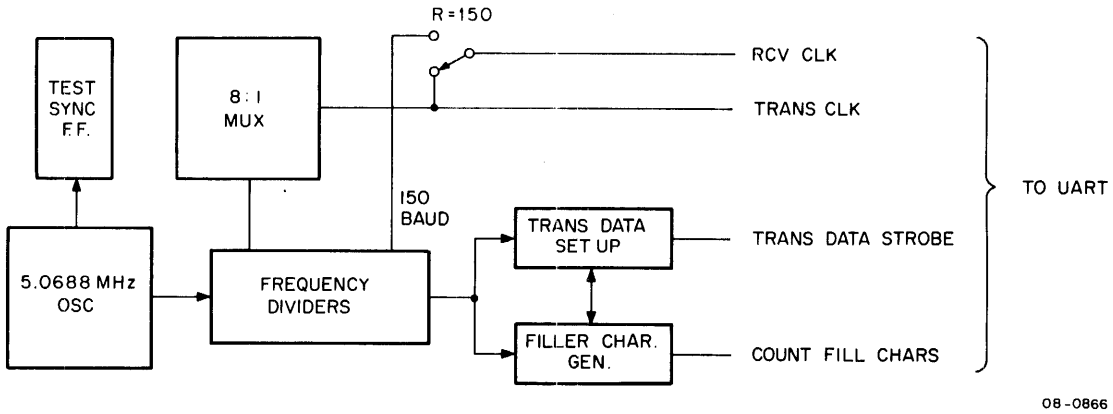


Figure 1-21 Timing and Filler Character Generation, Block Diagram

1.4.2.1 Timing Generator – The KL8-JA timing generator logic is simplified in the block diagram of Figure 1-22. The generator consists of a basic 5.0688-MHz crystal-controlled oscillator and six frequency dividers arranged in two strings of three each. The dividers yield the various baud rates required by the UART. In the block diagram, the outputs of each block are expressed in both pulse width and in frequency. Note that the frequency in kHz is equal to 16 times the baud rate. The baud rates are then applied to an 8-1 multiplexer for speed selection.

Referring to the circuit schematic on Sheet 3 of drawing D-CS-M8655-0-1, the frequency dividers (7492 and 7493) are operated in both a normal mode and a modified mode. The 7492 functions normally as a divide by 12 chip while the 7493 is basically a divide by 16 counter.

In this application, the single 7492 is operated in modified mode by XORing its normal divide by 12 output (E9-pin 8) with the 99-ns output from the 5-MHz oscillator and feeding that result to the A CLOCK input at E9 pin 14. In this way, the divider is caused to divide by 11 instead of 12.

Two of the 7493s (E17 and E18) are operated as divide by sixteens but two (E12 and E15) are modified by XORing to divide by 12. The fifth 7493 (E14) XORs the normal divide by 16 output to divide by 15.

Two derived outputs of the 7493 at E8 are used for special purposes. The 296-ns signal is used to form a 296-ns TRANSMIT DATA STROBE to establish setup time in the UART TBUF, and the 592-ns signal is used in filler character generation.

1.4.2.2 Baud Rate Selector – In the KL8-JA, selection of baud rate is accomplished by a 74151 one-of-eight multiplexer at E22. This is shown in Figure 1-23. The eight rates are tied to D0–D7 and one of these appears at the output f as a function of the select lines S0, S1, and S2 (see truth table in the figure). The rate can then be preset by switches B1,B2, and B3 to yield one TRANSMIT CLK at the output f. Notice that an additional output is provided as a test point, but this output is the inverse of the signal at f.

If the receiver is to operate at the transmitter speed, the switch B=150 is left open. When B=150 is closed, the RECEIVE CLK line receives the 150 baud clock; the receiver operates at 150 baud regardless of transmitter speed.

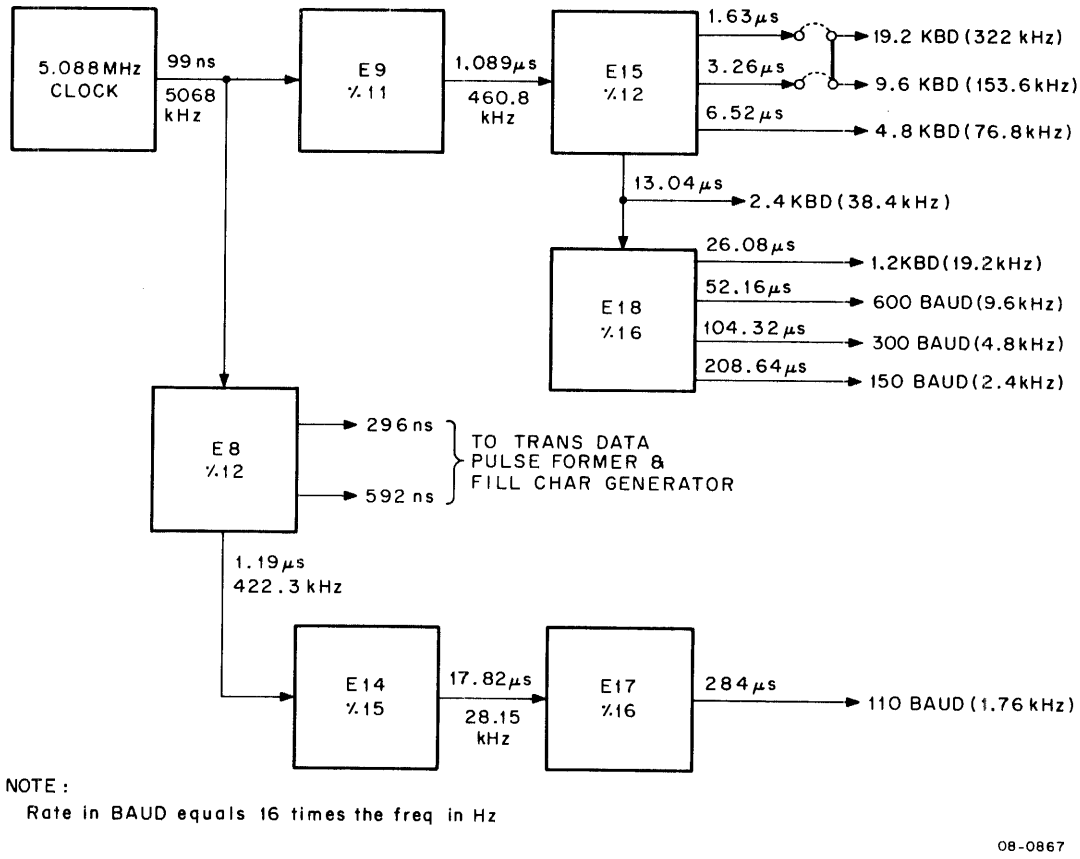


Figure 1-22 KL8-JA Timing Generator, Functional Block Diagram

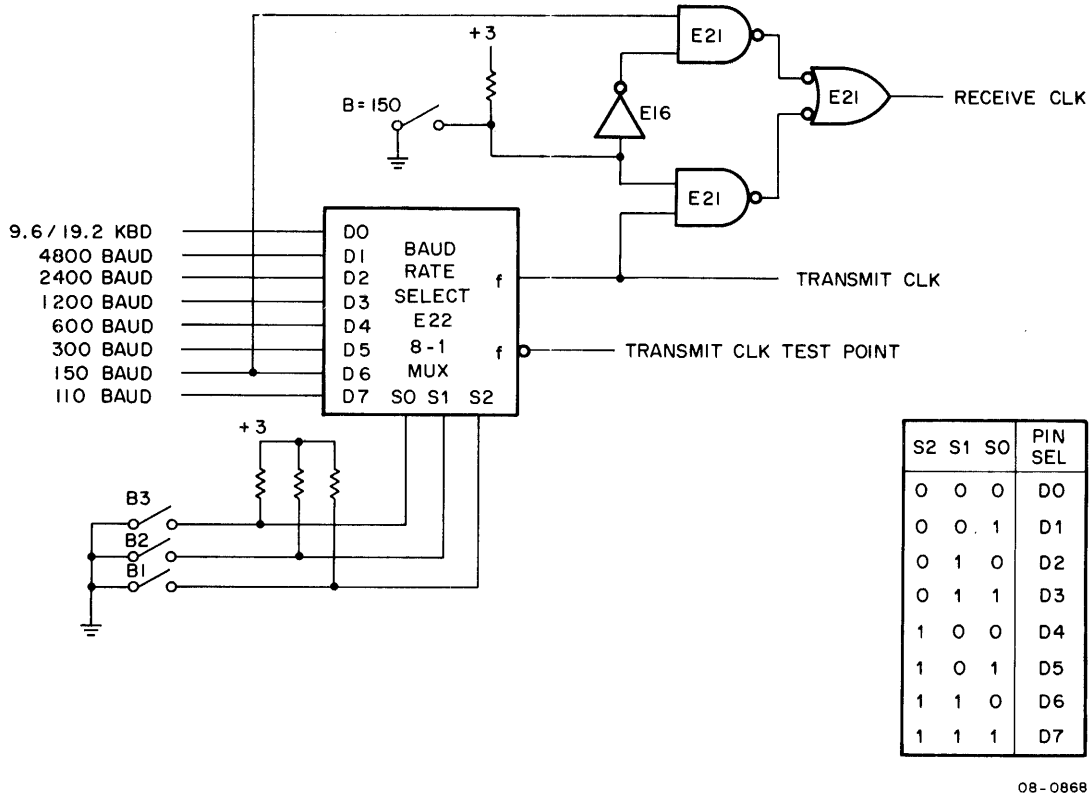
1.4.2.3 Filler Character Generator and TRANSMIT STROBE – The logic shown in Figure 1-24 illustrates the means whereby the KL8-JA generates filler characters, generates a measured TRANSMIT DATA STROBE, and, when filler character generation is enabled, delays setting the Transmit flag until the fourth filler character is generated.

At INITIALIZATION, INIT H clears the Transmit Strobe Master/Slave flip-flops at E42 and INIT L clears the filler character counter (FCCA & FCCB) and the Filler Character flip-flop (FILCH).

Normally, on each character to be transmitted, I/O TRANSMIT DATA L will clock the Master flip-flop at E42 against a grounded data input (Figure 1-25). When it sets, its high output enables its slave which then sets when clocked by the leading edge of P592Ns. When the slave sets, the AND at E54 is enabled, and when P296NS is received, TRANSMIT DATA STROBE goes low to cause the UART to transmit the character in its buffer. When P296NS goes low, the transmit strobe is terminated.

This occurs simultaneously with the TE of P592NS which ANDs at E37 with E42 (0) L to clear the master via the OR at E37. As the master resets, it clears the slave and is ready for the next I/O TRANSMIT DATA L signal.

Normally, within a line of data, the transmit strobe causes each character to be transmitted from the UART and when the UART is empty, the TRANSMIT BUFFER EMPTY signal is used to set the Transmit flag as described in Paragraph 1.4.1.7. This occurs only if FILCH is cleared. If FILCH is set, a flag cannot be set until FILCH is once again cleared. If a Line Feed (012) or (212) code is sent, however, some terminals require extra time to recover before faithful reproduction of the data to follow can be made.



S2	S1	S0	PIN SEL
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	0	0	D4
1	0	1	D5
1	1	0	D6
1	1	1	D7

Figure 1-23 Baud Rate Selector

In these installations, the KL8-JA can be jumpered to enable a Line-Feed Decoder on the output of the Transmit Data Buffer. This decoder is then used to cause four null characters to be sent to the UART before the Transmit flag is again set. The logic for this feature is also shown in Figure 1-24.

The signal TRANSMIT LINE FEED DECODED L, generated on Sheet 4 of D-CS-M8655-0-1, enables FILCH to be set upon receipt of TRANSMIT BUFFER EMPTY from the UART. Since at this time the filler character counter flip-flops FCCA and FCCB are both cleared, they AND with the inverse of this signal at E47 and clock FILCH via E54. This inhibits the Transmit flag.

The TE of TMBT then ANDs with FILCH (0) H at E40 to clock the Master flip-flop at E42, preparatory to generating a TRANSMIT DATA STROBE. When this flip-flop sets, it ANDs with a set FILCH to generate the first of four COUNT FIL CHARS. This signal, in turn, clears the Transmit Data Buffer and when the strobe is generated, loads a null into the UART. At the same time, coincidence at E37 clocks FCCB. This inhibits the gates at E47 and E48 from further clocking FILCH until FCCB and FCCA have counted four TRANSMIT BUFFER EMPTY signals.

During the fourth filler character then, the clocking path to FILCH is once again enabled and, since this character is a null, FILCH resets on the leading edge of TRANSMIT DATA STROBE via E49, thereby enabling the Transmit flag and removing the CLR on the Transmit Data Buffer.

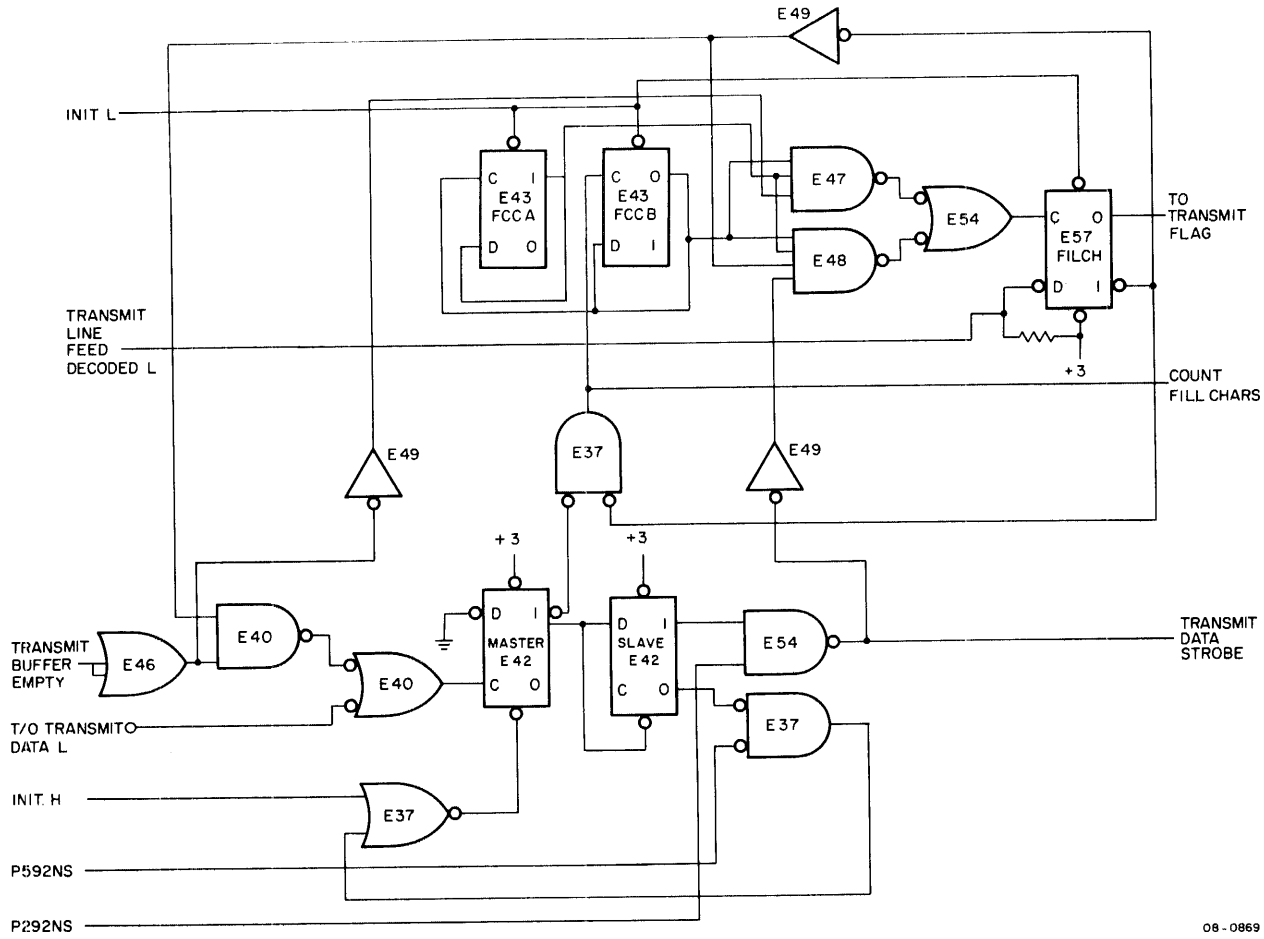


Figure 1-24 Filler Character Generation and Transmit Strobe

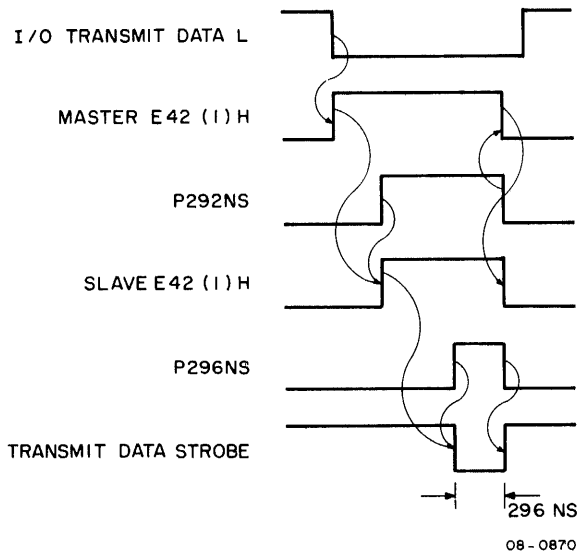


Figure 1-25 Transmit Strobe Timing Diagram

1.4.3 Data In and Data Out

This logical function is shown on drawing D-CS-M8655-0-1, Sheet 4. Figure 1-25 illustrates the logical arrangement in functional form.

In the KL8-JA, data transfers occur between the PDP-8 Accumulator (AC) and registers within the M8655 board.

In some earlier interfaces, the shift registers that communicated with the external device also served as a communications link between the interface and the CPU. In these interfaces when a character was received, the RCV FLAG would be set when a character had been assembled, but that character would remain available *only until* the next character began to be assembled. This forced the program to react very quickly to the flag. In addition, these interfaces created lost time between transmission of characters because the shift register had to be completely empty before the Transmit flag was set. Transmission would not start again then until the program could issue another character.

In the KL8-JA, an additional buffer is utilized between the receiver shift register and the AC. Also a transmit holding register has been inserted between the AC and the transmit shift register.

On the receive side, a character is assembled in the shift register and transferred to the receive holding register. At this time, the Receive flag is set indicating that a character is available. This character remains available to the CPU, in the holding register, until the next character is completely assembled. This lengthens considerably the length of time the program has to react to a Receive flag before reading a character.

On the transmit side, the Transmit flag is raised when the holding register-to-shift register transfer has been made. This eliminates the time lost in earlier arrangements since now, to maintain full-speed transmission of characters, the CPU need only react to the Transmit flag within one character time to refill the holding register.

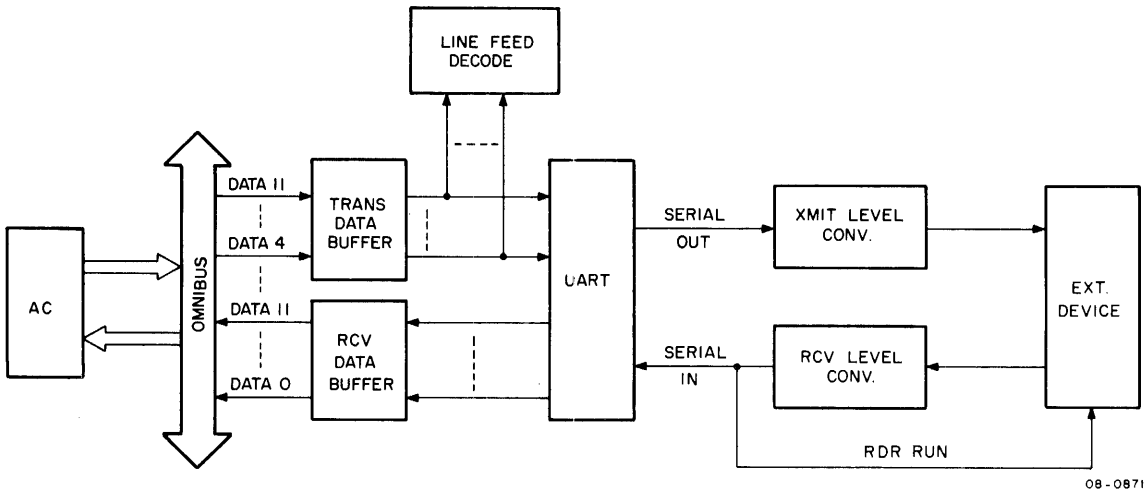


Figure 1-26 Data In/Data Out Block Diagram

1.4.3.1 Data Buffers – The KL8JA Data Buffers are illustrated in Figures 1-27 and 1-28. Data to be transmitted (DATA11–DATA04) is ANDed in parallel with TRANS 2, 4, or 6L and then clocked into the Transmit Buffer (E31 and E25) by I/O TRANSMIT DATA L. From here the high versions of those signals are made available to a transmit holding register within the UART. Appropriate versions of these outputs are also ANDed with a jumperable AND input to decode the presence of a Line Feed code (012 or 212) to yield LINE FEED DECODED. This signal is then used in the generation of filler characters as described in Paragraph 1.4.2.3. That paragraph also describes the source of the signal COUNT FILL CHARS used here to clear the buffer so that nulls can be loaded into the UART during the fill character period.

Received data from the device is transferred from the UART receive holding register (Figure 1-28) in parallel to the AND gates at E23 and E29. These gates are then enabled by I/O READ DATA H if STAT ENBL (1) L is ANDed with B PAUSE L and if the jumper is installed at W3 or SWD. DATA00 is the OR of the three error conditions (Parity, Framing, and Overrun) computed by the UART.

1.4.3.2 Universal Asynchronous Receiver/Transmitter (UART) – The heart of the KL8-JA is its Universal Asynchronous Receiver/Transmitter (UART). The UART is a full-duplex device in which the receiver section accepts asynchronous serial binary characters and converts them to a parallel format for transfer to the PDP-8 AC. The transmitter section accepts parallel binary characters from the OMNIBUS and converts them to a serial asynchronous output with START and STOP bits added.

All serial characters contain a START bit; five to eight DATA bits; one, one and a half, or two STOP bits; and a parity bit, which may be odd, even, or turned off completely. The STOP bits are opposite in polarity to the START bit.

Both the receiver and transmitter are double buffered. The UART internally synchronizes the START bit with the clock input to ensure a full 16-element (clock periods) START bit independent of the time of data loading. Transmitter distortion (assuming perfect clock input) is less than 2.5 percent on any bit up to 10 kbaud. The receiver strobes the input bit within ± 8 percent of the theoretical center of the bit. The receiver also rejects any START bit that lasts less than one-half of a bit time.

Figure 1-29 is a block diagram of the UART receiver. When the receiver is in the idle state, it samples the serial input at the selected clock edges after the first high-to-low transition of the serial input. If the first sample is a high, the receiver returns to the idle state and is ready to detect another high-to-low transition. If, however, the first sample is low, then the receiver enters the data entry state.

If the receiver control logic (jumpers) has not been conditioned to the no parity state, then the receiver checks the parity of the DATA bits plus the PARITY bit following the DATA bits, and compares it with the parity sense on the Parity Select Line. If the parity sense of the received character differs from the parity of the UART control logic, then the Receive Parity Error line goes high and causes the Parity Error bit in the UART status register to set.

If the receiver control logic has been conditioned to the No Parity state, then the receiver takes no action with respect to parity and maintains the Parity Error line in the false (low) state. When the control logic senses a parity error, it generates a PARITY ERR signal. The RECEIVED DATA AVAILABLE signal updates the parity error indicator.

The receiver samples the first STOP bit that occurs either after the PARITY bit, or after the DATA bits if no parity is selected. If a valid (high) STOP bit exists, no further action is taken. If, however, the STOP bit is false (low), indicating an invalid stop code, then the UART control logic provides a FRAMING ERR indication. The status of the framing error bit can also be read from the UART status register.

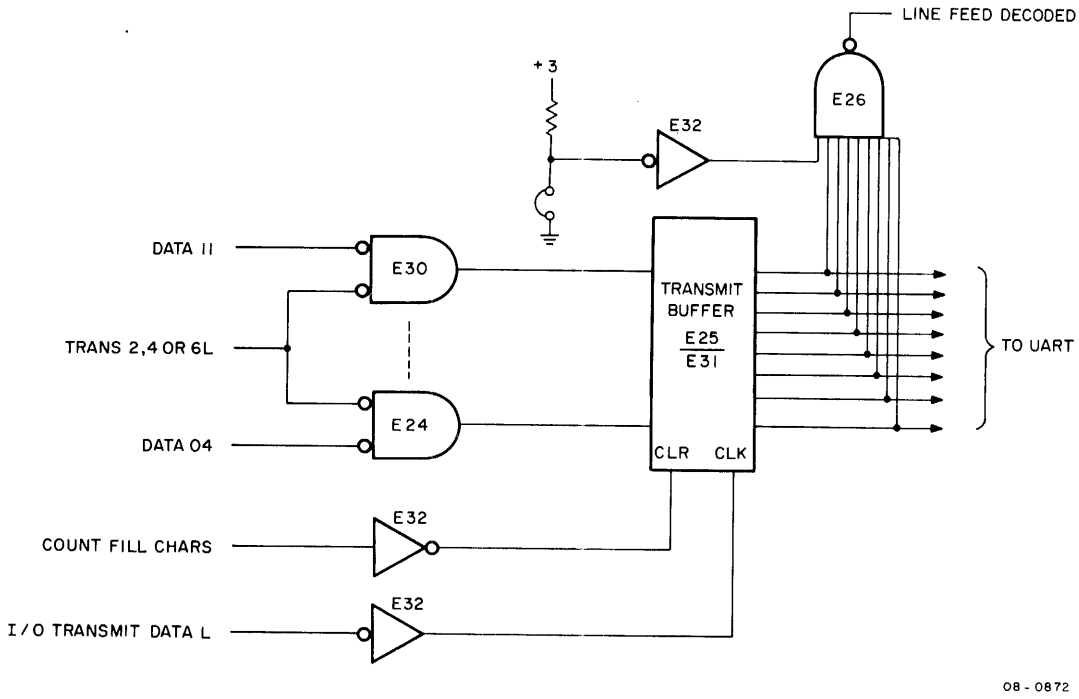


Figure 1-27 Transmit Data Buffer

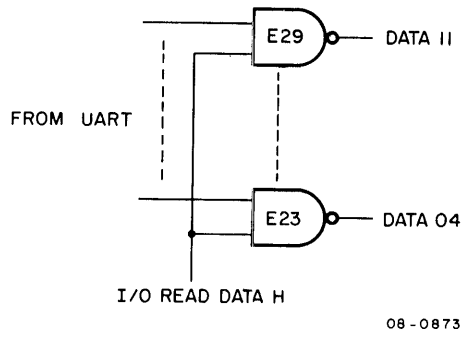


Figure 1-28 Receive Data Buffer

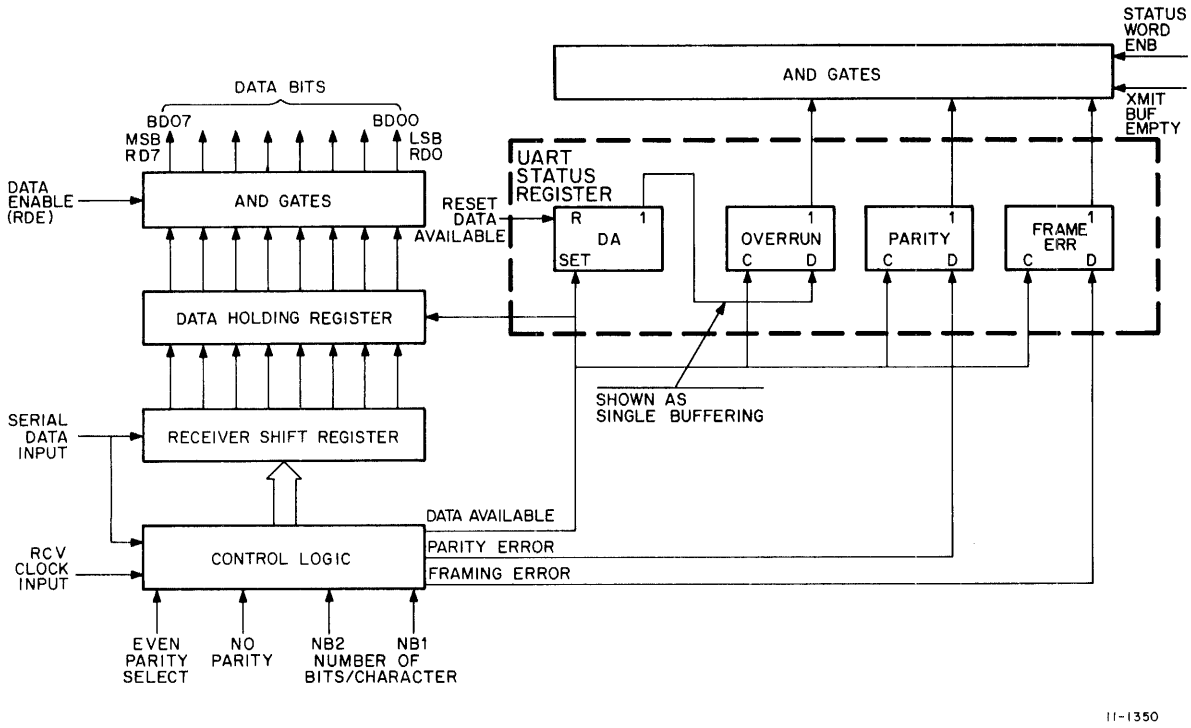


Figure 1-29 UART Receiver Block Diagram

Because the serial input from the device is shifted into the UART a bit at a time, the occurrence of a stop code indicates that the entire data character has been received and shifted into the receiver shift register. After the STOP bit has been sampled, the receiver control logic parallel transfers the contents of the shift register into the receiver data holding register and then sets the RECEIVED DATA AVAILABLE (DA) flag.

The Data Available signal also functions as the clock input to the FRAME ERR, PARITY, and OVERRUN flip-flops in the UART status register. At this point, the DA flip-flop is set, the OVERRUN flip-flop is clear but has a high on the data input because of the output from the DA flip-flop, and the PARITY and FRAME ERR flip-flops are set or cleared depending on the signal (true or false) strobed in from the control logic.

An OVERRUN ERR condition indicates that an additional data character has been sent to the UART holding register before the previous character's Receive flag has been cleared, i.e., reset data available has been generated.

Whenever the serial input line (SI) goes from a high to low and remains at a low level, the receiver shifts in one character, which is all spaces, then sets the FRAME ERR indicator and waits until the input line goes high before shifting in another character.

A block diagram of the transmitter portion of the UART is shown in Figure 1-30. When in the idle state, the transmitter sends out a high on its serial output line (SO). When data is transmitted, a parallel character is strobed into the UART transmitter data buffer by means of the TRANSMIT DATA STROBE signal. The time between the low-to-high transition of the TRANSMIT DATA STROBE and the corresponding high-to-low transition of the serial output line is within one clock cycle (1/16 of a bit time) if the transmitter has been idle. The TRANSMIT DATA STROBE is used to strobe a character from the transmit buffer register in the KL8-JA into the TBUF.

After the data has been loaded into the UART data buffer, it is transferred to the transmitter shift register under control of signals from an encoder that selects the format determined by the control logic. This permits selection of either Parity or No Parity, the type of parity, the number of STOP bits, and the number of DATA bits per character.

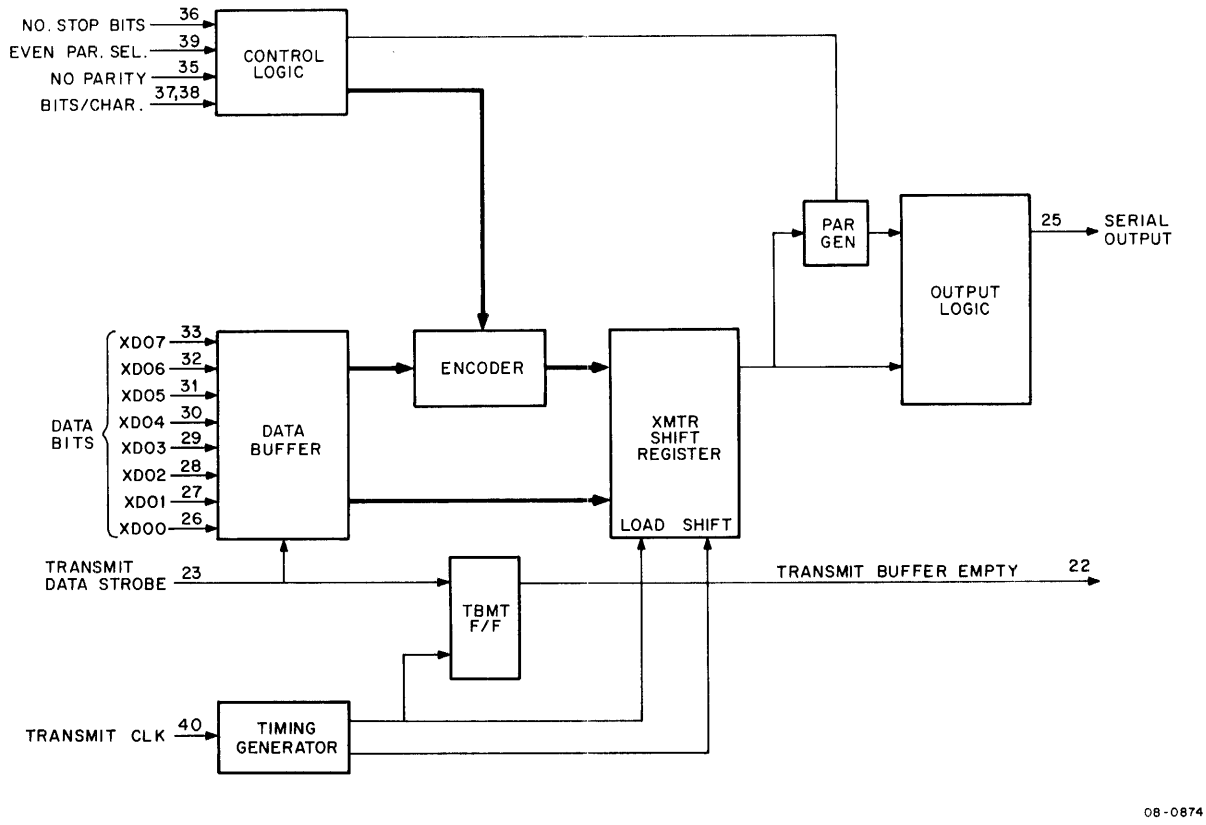


Figure 1-30 Transmitter Block Diagram

The transmitter logic converts the parallel character from the OMNIBUS via the Transmit Buffer into a serial output that is in a format selected by the control logic.

Whenever the Transmitter Data Buffer is loaded while the previous character is being shifted through to the output line, the START bit of the new character immediately follows the last STOP bit of the previous character.

When the TRANSMIT DATA STROBE loads the UART data buffer, the KL8-JA Transmit Buffer is unloaded. Therefore, the condition TRANSMIT BUFFER EMPTY indicates that the transmitter is ready for another character. (It does not indicate the completion of a transmission).

1.4.3.3 Level Converters – In the KL8-JA, level conversion is provided on both the transmit and receive lines between TTL (ground and +3 V) to either EIA levels (±8 V) or 20 mA current loop operation.

The transmit logic is shown in Figure 1-31 and 1-32. The TTL serial data at SO of the UART is gated against ground at E55 and from here is multiplexed to either the 20 mA converter in Figure 1-31 or the EIA converter in Figure 1-32.

In the 20 mA logic, a high to E55 will produce a low at the base of the series transistor establishing a voltage divider from +5 to -15 V. This provides the bias to enable Q6 and Q7. The voltage drop across R42 equals the drop across R43 establishing a bias at the base of Q6 that is 0.7 V more negative than the +5 V source, and a bias at the base of Q7 that is 0.7 V more positive than the -15 V point. This establishes the differential current source conditions to satisfy the 20 mA current loop through the external device.

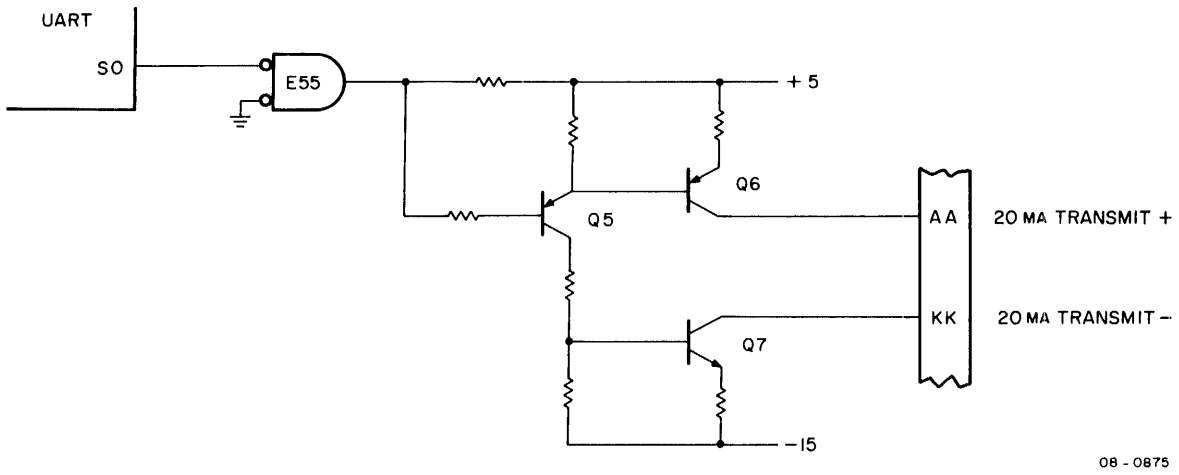


Figure 1-31 Transmit TTL/20 mA Converter

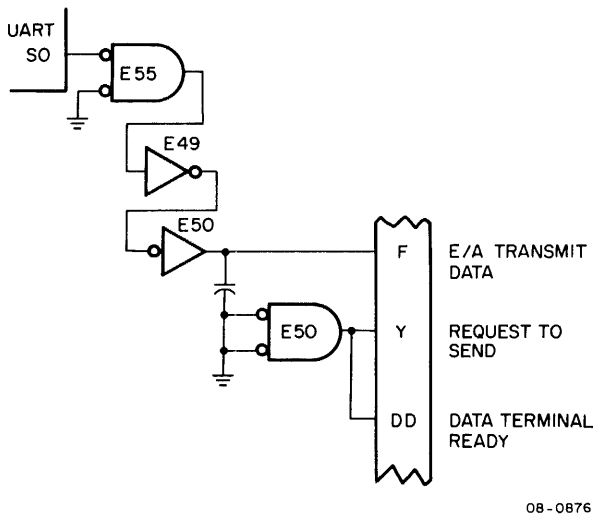


Figure 1-32 Transmit TTL/EIA Converter

The TTL/EIA converter in Figure 1-32 utilizes the same serial out signal to convert ground and +3 at the input to E49 to ±8 V EIA signals at the output of E50. Note that for modem operation the REQUEST TO SEND and DATA TERMINAL READY lines are held asserted.

The receive logic is shown in Figures 1-33 and 1-34. The EIA conversion is rather simple in which EIA DATA IN at pin J of the terminal block is converted to TTL by E45 and then fed out to pin M. A jumper in the cable connects the EIA/TTL RECEIVED signal to pin E that, in turn, feeds the SI input of the UART.

The 20 mA receive logic is given in Figure 1-34. When a high impedance is seen between pins K and S, Q4 and Q2 are off as are Q1 and Q3. Point B is at ±5 V (high) and point C is at 0.0 V (low). When a low impedance is seen between pins K and S, Q1, Q2, Q3, and Q4 are on causing a high at point C and a low at point A.

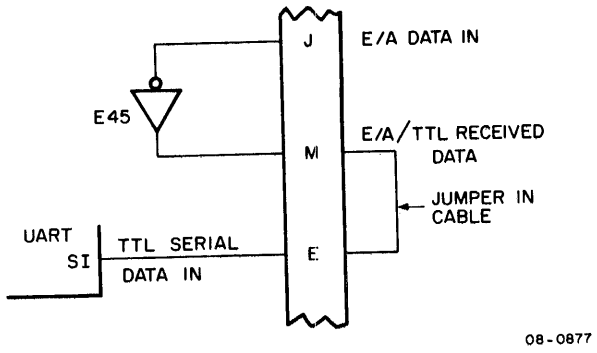


Figure 1-33 Receive EIA/TTL Converter

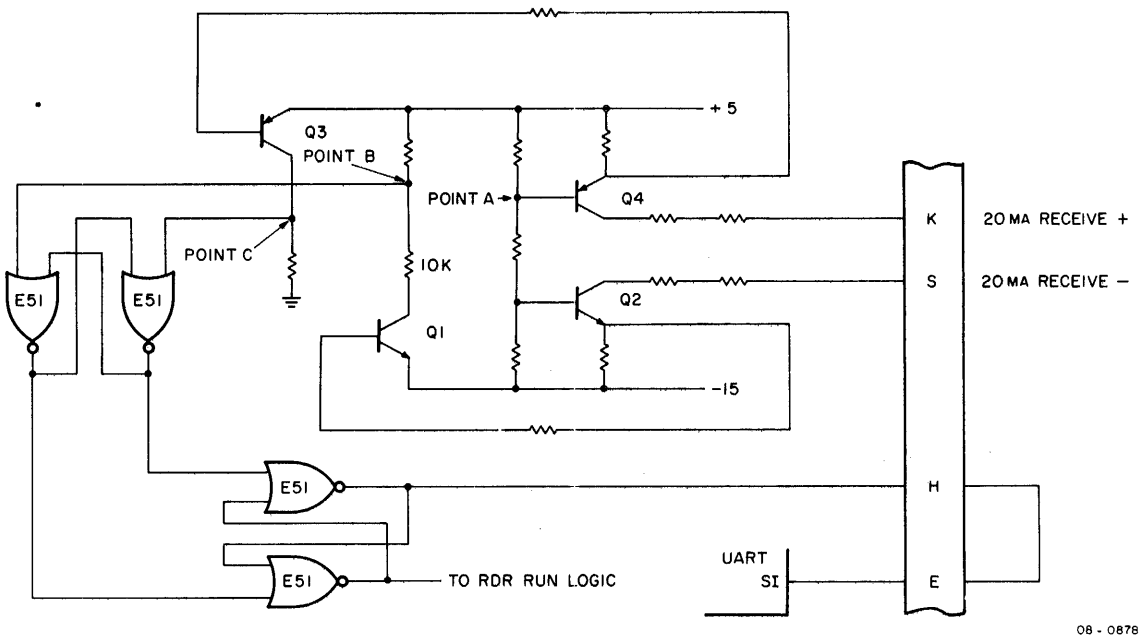


Figure 1-34 Receive 20 mA/TTL Converter

The converted serial input is used to feed the UART SI input via pins H and E of the terminal board, jumpered by the appropriate cable.

The complement output of E51 is also fed to the reader run logic shown in Figure 1-35. The READER RUN is cleared when a high impedance state has been detected for half a baud time (8 clocks). This corresponds to halfway through the start bit of a character.

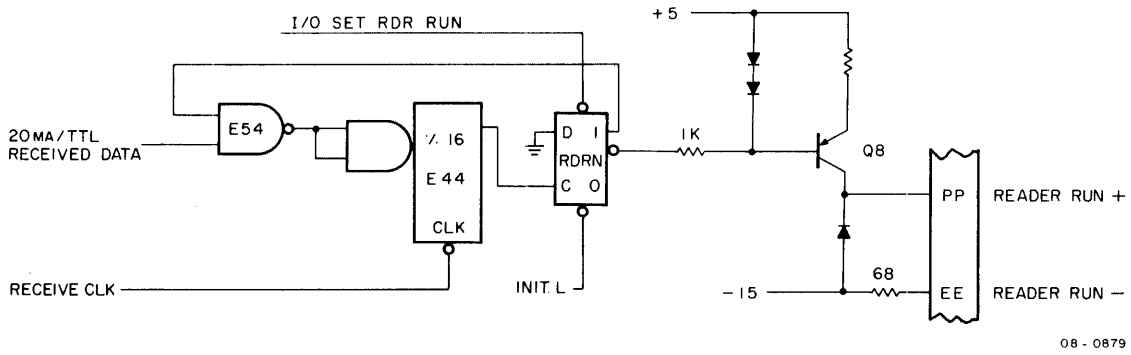


Figure 1-35 Reader Run Control Logic

1.5 MAINTENANCE

NOTE

Module repair may not be accomplished at the user's site.

There are no specific maintenance procedures for the KL8-JA module (M8655) itself. The KL8-JA diagnostic (DEC-08-DIKLA-A-PB) should be run according to the procedure in Chapter 2. Once it has been determined that the KL8-JA module is defective, it should be returned to a Digital repair depot for repair. If possible, cables connecting the KL8-JA and the Printer should be swapped to ensure that a defective cable is not causing the problems encountered during operation.

All defective modules should be shipped to a Digital repair depot as soon as possible for repair.

1.6 SPARE PARTS

Digital recommends that the user keep a spare M8655 module as spare parts for the KL8-JA.

KL8-JA TERMINAL
CONTROL/ASYNCHRONOUS
DATA INTERFACE
TECHNICAL MANUAL
EK-KL8JA-TM-001

Reader's Comments

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc.? Is it easy to use? _____

_____

_____

_____

What features are most useful? _____

_____

_____

_____

What faults do you find with the manual? _____

_____

_____

_____

Does this manual satisfy the need you think it was intended to satisfy? _____

Does it satisfy *your* needs? _____ Why? _____

_____

_____

_____

Would you please indicate any factual errors you have found. _____

_____

_____

_____

Please describe your position. _____

Name _____ Organization _____

Street _____ Department _____

City _____ State _____ Zip or Country _____

— — — — — — — — — — — — Fold Here — — — — — — — — — — —

— — — — — — — — — — Do Not Tear - Fold Here and Staple — — — — — — — — —

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

Digital Equipment Corporation
Technical Documentation Department
Maynard, Massachusetts 01754