Honeywell

This document and the information contained herein are confidential to and the property of Honeywell Information Systems Inc. and are made available only to Honeywell employees for the sole purpose of maintaining Honeywell's products. This document, any copy thereof and the information contained herein shall be maintained in strictest confidence; shall not be copied in whole or in part except as authorized by the employee's manager; and shall not be disclosed or distributed (a) to persons who are not Honeywell employees, or (b) to Honeywell employees for whom such information is not necessary in connection with their assigned responsibilities. Upon request, or when the employee in possession of this document no longer has need for the document for the authorized Honeywell purpose, this document and any copies thereof shall be returned to the employee's manager. There shall be no exceptions to the terms and conditions set forth herein except as authorized in writing by the responsible Honeywell Vice President.

SERIES 60 (LEVEL 6)

TYPE KCM9101 CONSOLE ADAPTER MANUAL

Doc. No. 71010225-300 Order No. FL17, Rev. 2

Honeywell

This document and the information contained herein are confidential to and the property of Honeywell Information Systems Inc. and are made available only to Honeywell employees for the sole purpose of maintaining Honeywell's products. This document, any copy thereof and the information contained herein shall be maintained in strictest confidence; shall not be copied in whole or in part except as authorized by the employee's manager; and shall not be disclosed or distributed (a) to persons who are not Honeywell employees, or (b) to Honeywell employees for whom such information is not necessary in connection with their assigned responsibilities. Upon request, or when the employee in possession of this document no longer has need for the document for the authorized Honeywell purpose, this document and any copies thereof shall be returned to the employee's manager. There shall be no exceptions to the terms and conditions set forth herein except as authorized in writing by the responsible Honeywell Vice President.

SERIES 60 (LEVEL 6)

TYPE KCM9101 CONSOLE ADAPTER MANUAL

Doc. No. 71010225-300 Order No. FL17, Rev. 2

RECORD OF REVISIONS

REVISION	DATE	AUTHORITY	AFFECTED PAGES
-100	Dec. 1975	--	--
-200	April 1976	--	--
-300	June 1977	BLCO51342 BLCO60147 BLCO60500 BLCO61780	--

Hardware Publications, M&TO, Billerica, MA 01821

Printed in the United States of America
All rights reserved
225/FL17

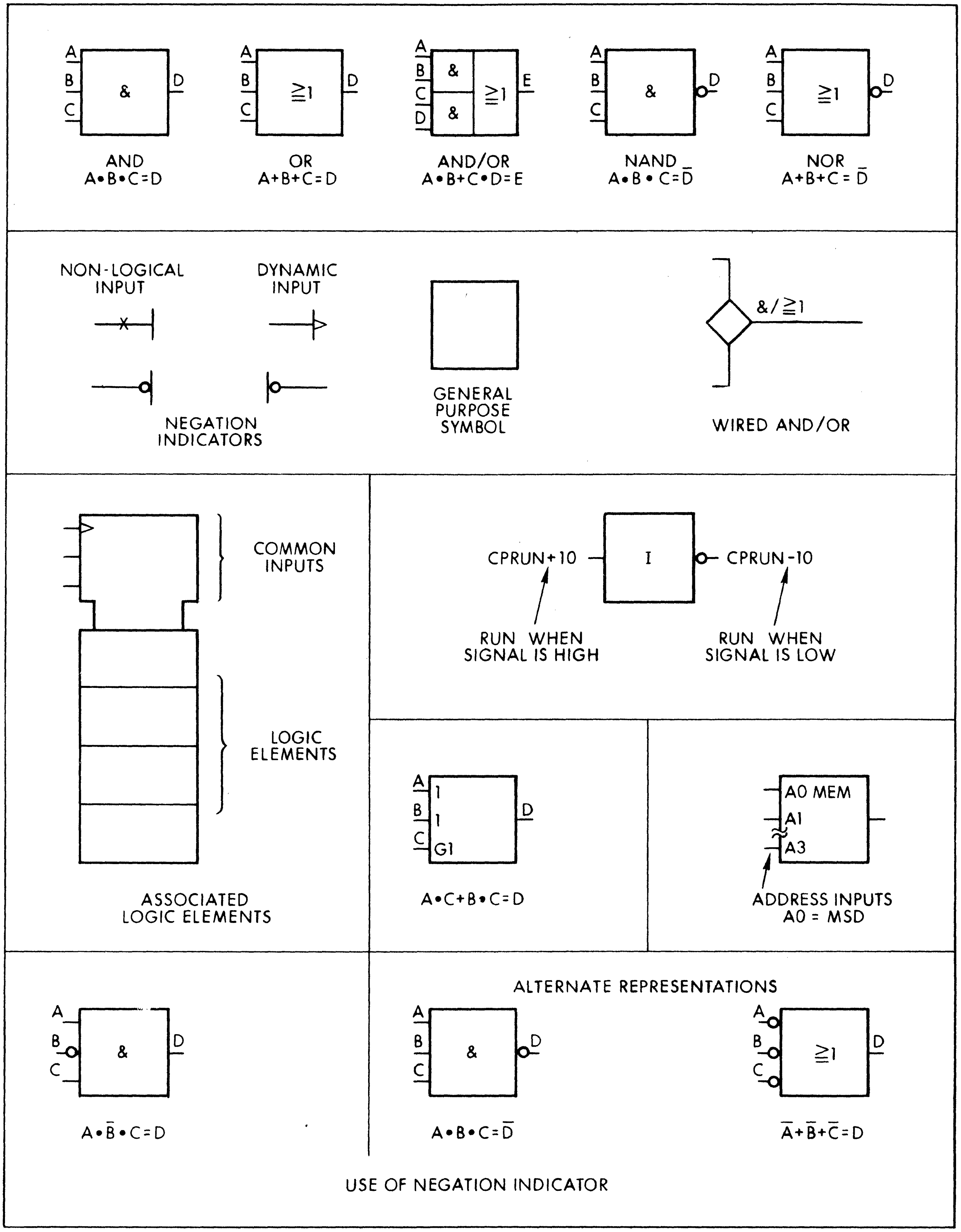
CONTENTS

CONTENTS

ILLUSTRATIONS

TABLES

LOGIC SYMBOLOGY



I
INTRODUCTION

1.1 SCOPE AND PURPOSE OF THIS DOCUMENT

This manual describes the functionality and operation of the Type KCM9101 Console Adapter*. This adapter is designed to be used with a Type MDC9101 Multiple Device Controller (MDC) that is associated with a Series 60 Level 6 Minicomputer System. Information on programming and the minicomputer system functions is presented only as required to make the description of the hardware/firmware of the adapter comprehensible. Operational theory contained in this document is designed to acquaint the reader with the major functional areas of the adapter and to aid in analyzing operation of the adapter in more detail by using the associated reference manual. Information on specific software commands used by the minicomputer to control the MDC is contained in the MDC product manual listed in Table 1-2.

This product manual is composed of three sections and an appendix as follows:

1. Section 1 INTRODUCTION - Briefly describes the scope and purpose of this manual and lists related reference documents. Provides a general description of the adapter and lists representative devices which it can control. Discusses the adapter interfaces to both the MDC and device.

*This manual supports Firmware Listing 77.

2. Section 2 THEORY OF OPERATION - Provides a hardware block diagram with an associated description. Describes the functional operation of the adapter. Provides an intermediate-level description of the hardware with related illustrations. Lists and defines signals from the adapter to the MDC and from the adapter to the device.

3. Section 3 FIRMWARE - Contains information on the portion of the scratch pad memory (SPM) in the MDC which is dedicated to firmware operation of the Console Adapter. Defines the software task and status words which are used by the adapter firmware. Furnishes an overview description of firmware routines with a follow-on intermediate description of each routine and subroutine. Provides a glossary of firmware terms.

4. Appendix INSTALLATION INFORMATION - Provides information on the installation of the devices listed in Table 1-1. This information is useful when changing the device used with the adapter to another device listed in Table 1-1 or to any other device compatible with the adapter.

1.2 GENERAL DESCRIPTION

The Type KCM9101 Console Adapter is a solid state Device-Pac (BD2CSL) used with the Multiple Device Controller (MDC) associated with a Honeywell Series 60 Level 6 Computer. As shown in Figure 1-1, as many as four various compatible adapters can be used on the MDC, and up to four can be Console Adapters. The adapter contains the logic for data handling, control, and interface between the MDC and the device. The device used with the Console Adapter can be any TTY-compatible device or other selected device, including but not restricted to those listed in Table 1-1. The Appendix provides procedures for changing devices in the field.

The ASR and KSR type devices are used as primary input/output devices in many Series 60 Level 6 configurations. A KSR can be used on-line to input data from its keyboard or to type out data received from the adapter. An ASR has the same input and output capability as a KSR, but in addition it may optionally have the capability of reading or punching a paper tape. The keyboard console devices can be used to input data by standard typing procedures and visually display data received from the adapter.

1.3 SYSTEM INTERFACE

The interface between the adapter and the device is full duplex. All input and output data between the adapter and device is in serial-stream form. The stream normally consists of standard communication-type characters. Either 7-bit characters with even parity or 8-bit characters with no parity are used, depending upon software instructions. One or two stop bits can be appended to each character, depending upon the configuration of the adapter. Transmission between the adapter and the device can be in either a current loop mode or a voltage mode specified by Electronic Industries Association Specification RS232C. The selected cable between the adapter and the device determines which mode of transmission is used.

The interface between the adapter and the MDC is half duplex. All input and output signals are carried on lines which use pins on the bottom of the adapter to connect to the MDC. All data is transferred in parallel between the adapter and MDC.

The transmission rate between the adapter and the device can be set at various speeds between 50 and 9600 baud, depending upon the setting of a hexadecimal switch.

1.4 REFERENCE DOCUMENTS

The documents listed in Table 1-2 supplement the information contained in this manual.

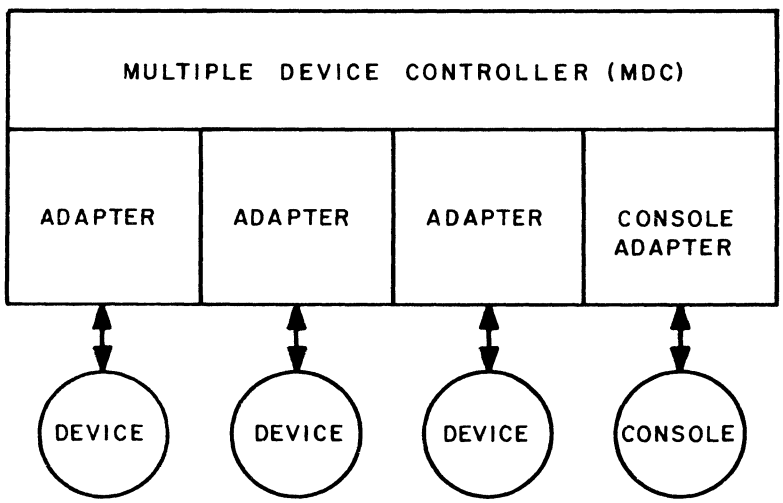


Figure 1-1 Configuration Diagram

Table 1-1 Representative Devices Used With Console Adapter

ITEM NO.	HONEYWELL MARKETING DESIGNATION	SOFTWARE ID NO. (HEX)	DEVICE DESCRIPTION
1	TTU9101	2018	ARS-33 Teletype Console
2	TTU9102	2019	KSR-33 Teletype Console
3	TTU9103	2018	ASR-33 with Auto-Shutdown
4	TTU9104	2019	KSR-33 with Auto-Shutdown
5	DKU9101	201A	CRT (TTY) Keyboard Console-64 Char.
6	DKU9102	201A	CRT (TTY) Keyboard Console-96 Char.
7	TWU9101	201C	30 CPS Keyboard Typewriter Console (KSR) (TTL) W/Keypad
8	TTK9101	-	Automatic Shutdown Kit for TTU9101/TTU9102
9	TWU9104	201C	30 CPS KBD Typewriter Console KSR (MOS) W/O Keypad - 96 Char.
10	TWU9106	201C	120 CPS KBD Typewriter Console KSR (MOS) W/O Keypad - 96 Char.

Table 1-2 Reference Documents

TITLE	DOCUMENT NUMBER	ORDER NUMBER
Model 34/36 System Manual	71010200-201	FL35A
Model 43 System Manual	71010316-100	FN36
Type MDC9101 Multiple Device Controller Manual	71010220-300	FL19
Type MDC9101 Multiple Device Controller Reference Manual (Assy No. 60127882-002)	71010370-100	FL26
Type MDC9101 Multiple Device Controller Reference Manual (Assy No. 60130148-001)	71010378-100	FM42
Type KCM9101 Console Adapter Reference Manual (Assy No. 60127819-001)	71010374-100	FL29
Type KCM9101 Console Adapter Reference Manual (Assy No. 60130145-001)	71010375-100	FM38
Level 6 Minicomputer Handbook	N/A	AS22
Level 6 Control Panel and Peripherals Manual	N/A	AT04
Teleprinter Console Operator Instruction Card	N/A	AT21

II
THEORY OF OPERATION

2.1 CONSOLE ADAPTER FUNCTIONAL OPERATION

The console adapter is designed to operate with devices that use communication-type characters in serial-bit stream for transmitting and receiving. There are separate paths for transmitting and receiving characters between the device and the adapter. Both control and data characters are sent over the same path from the MDC. The device constantly monitors characters it receives and takes appropriate action when it detects a valid control character. Firmware in the MDC can optionally monitor characters received from the device for control characters (CCs) established by software. When a valid CC is detected, the firmware takes various actions depending upon a task word previously placed in the scratch pad memory (SPM) of the MDC by software. The function of software is to format the character stream to make it compatible with the specific device used.

If a key on the keyboard is struck when the adapter is not busy with a programmed input or output operation, the firmware and the software identify the character and take appropriate action. If a key on the keyboard is struck during an output operation, the operation may be stopped or some other action taken, depending on the characters received and the task word in the SPM (see Figure 3-5). During an input operation a character received from the device can be optionally looked at to determine if it is a control character or basic data. When a control character is detected, firmware then causes some appropriate action to be taken (see Figure 3-5).

When the MDC starts a programmed output operation, the firmware first determines the character size (seven bits with parity or eight bits without parity) from the task word in SPM. It then loads this information and stop bit information into the parallel-to-serial encoder (UAR/T) as described in detail in subsection 2.3.2. The firmware then fetches bytes from the computer by direct memory access (DMA) and sends them to the UAR/T in parallel-bit form. The UAR/T then adds a start bit, a parity bit (if parity is used), and either one or two stop bits to the character (see Figure 2-4 for an example of a developed character). The developed character is then sent to the device in serial-bit form. Usually a specified number of bytes are transferred from the computer to the device during a DMA operation. If a key is struck during the output operation, the operation may be stopped or allowed to continue with a notification to firmware at the end of the operation, depending upon the character received and the task word in SPM.

When the MDC starts a programmed input operation, the firmware first determines from the task word the character size (seven bits with parity or eight bits without parity) that is to be input from the device. It then loads this information and stop bit information into the serial-to-parallel decoder (see subsection 2.3.3 for details). If the echo mode is to be used, the echo-back mode flip-flop in the adapter is also set when the UAR/T is control loaded (see subsection 2.3.4 for further details). In the following DMA operation, communication-type characters received by the UAR/T from the device (with stop and start bits) are transformed to either 8-bit or 7-bit bytes and sent to consecutive locations in the computer memory. All bits of 8-bit characters without parity are sent to the computer memory but only the seven data-carrying bits of 8-bit characters received with parity are sent. The number of bytes sent to the computer is determined by firmware when it detects end-of-range, or Control Character 2 (CC2), or Control Character 3 (CC3) from the device. Software presets the range and configures CC2 and CC3 (see Figure 3-5). When the UAR/T receives the character from the device in serial form, it strips off the start and stop bits. It then assembles an 8-bit byte and checks for a parity error (if parity checking is specified), and for a no-stop bit error. If an error is detected, an appropriate signal is generated for use by firmware in developing the status word. The assembled byte is then sent in parallel form to the MDC via firmware control. If parity is specified, the byte consists of seven bits, and if not specified, eight bits. In the firmware a byte received from the adapter may or may not be optionally examined to see if it is a control character. If the byte is examined and a valid control character is detected, the firmware causes some designated action to take place (see Figure 3-5). If the byte is examined and not found to be a control character, it is assumed to be a data character. All characters not optionally examined are also assumed to be data characters. All data characters are sent to the computer memory by using direct memory access (DMA) transfers.

Successive data characters are input from the device to the computer until the firmware detects either a DMA process end-of-range or the examined byte is found to be a Control Character 2 or Control Character 3. If the UAR/T assembles two characters before one is transferred into the MDC, a service rate error signal is generated for use in generating the status word. If a key is struck when the MDC firmware is not processing, an attention interrupt is generated in the MDC firmware. The character of the key struck is saved and, depending upon the task word, may be appended to the next input message from the device to the MDC.

2.2 BLOCK DIAGRAM DESCRIPTION

Figure 2-1 is a block diagram of the major hardware components and interconnections of the console adapter. Primarily, the adapter controls the data flow between the multiple device controller (MDC) and the device. To make the flow compatible between the MDC and the device, the parallel data received from the MDC is encoded to serial data for transfer to the device, and the serial data received from the device is decoded into parallel data for transfer to the MDC.

Several control signals are directly carried between the MDC and adapter via hardware connections between the adapter and the MDC. The device ready logic provides the programmer with a means of checking to see if the device is available and ready to transmit and/or receive. In addition to the hard-wired control connections and the device ready logic, the MDC uses the input multiplexer to input other control information. The ID input to the multiplexer is used by the MDC to identify the type device being serviced (e.g., ASR, KSR, or console). The specific information conveyed by the status 1 and status 2 inputs to the multiplexer is described in subsection 2.3.8. The specific information conveyed by interconnecting lines between the MDC and the adapter and between the adapter and the device is described in subsection 2.3.1.

The echo logic is used to send data received from the device directly back to the device so that it can be printed or displayed for the convenience of the device operator.

The wraparound logic is used when testing. It allows characters sent out from the MDC to be sent directly back to the MDC for comparison functions.

When used with an ASR-33, the timeout logic develops a status signal (TIMOUT) if no input or output activity from or to the device occurs within 37 seconds. This logic makes the adapter and MDC compatible with the automatic turn-off option of the TTY33.

The baud-rate generation logic develops a clock signal which is 16 times the baud rate. This clock signal is required by both the encoder and the decoder; it is also used in the timeout logic. The baud rate can be varied in steps between 50 and 9,600 baud by the setting of a hexadecimal rotary switch.

The EIA logic enables the adapter to use a transmitter and a receiver designed to Specification RS232C of the Electronic Industries Association (EIA). When the EIA interface is used, the data out from the encoder is changed from the TTL level of the adapter to -12 volts for a mark state and +12 volts for a space state prior to being sent out to the device. Conversely, the EIA logic changes the EIA (±12 volts) level of the data received from the device to the required TTL level used in the adapter prior to sending the data to the decoder.

The current mode logic provides the adapter with a means of sending and receiving data to or from the adapter via a 60-milliampere current loop. When transmitting, the data out from the encoder is changed from the TTL level to 60-milliampere flow for a mark state, and to a no-current flow for a space state. Conversely, the current loop logic converts the 60-milliampere or no-flow states received from the device to the TTL level prior to sending the data to the decoder.

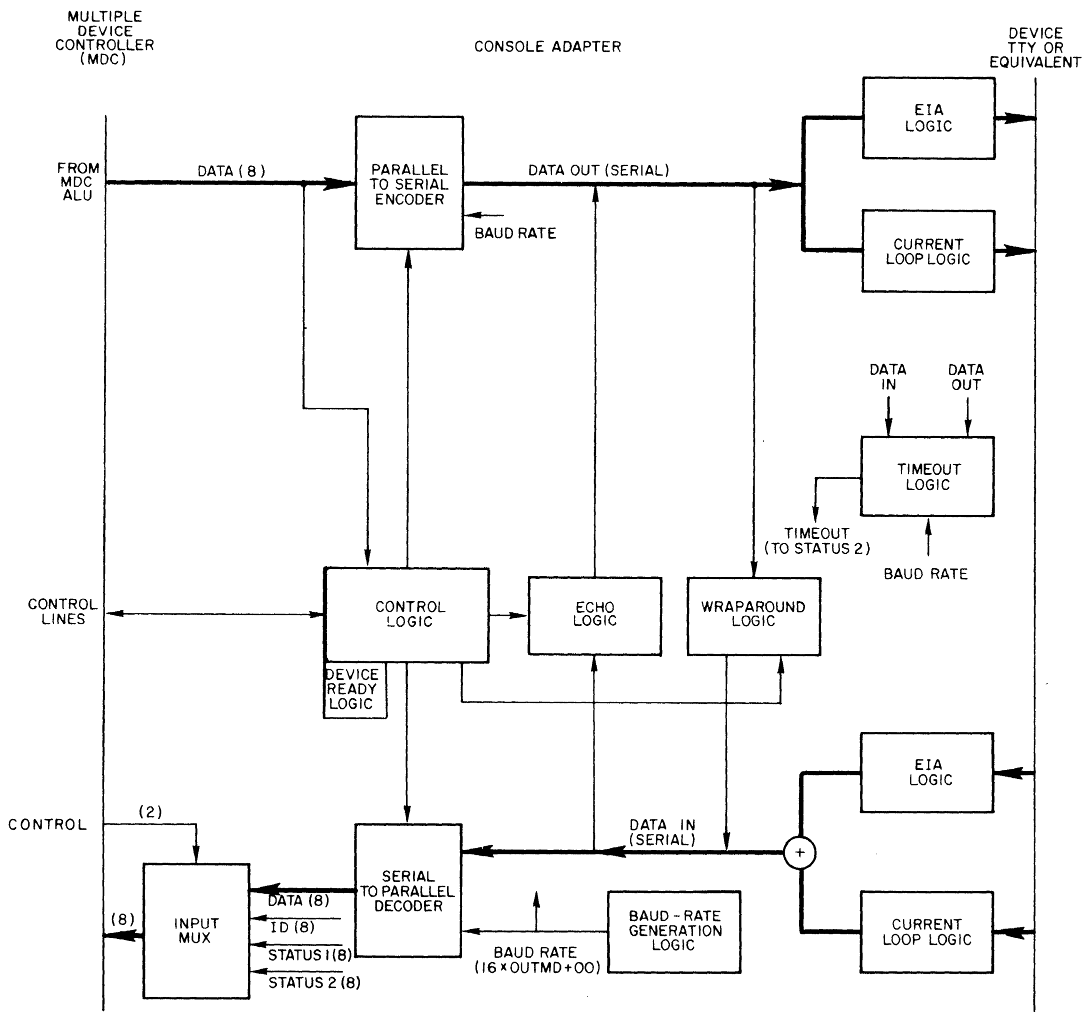


Figure 2-1 Console Adapter Block Diagram

2.3 FUNDAMENTAL HARDWARE DESCRIPTION

2.3.1 Interconnections

Figure 2-2 shows the interconnections between the MDC and the console adapter, and the console adapter and the device. Table 2-1 lists and describes the function of the interconnections between the MDC and the adapter. Table 2-2 lists and describes the function of the interconnections between the console adapter and the device. Note that the adapter provides logic for both a current loop and an EIA interface with the device. However, only one type interface is used with a device. A different model cable is used for each specific type interface and for each specific device.

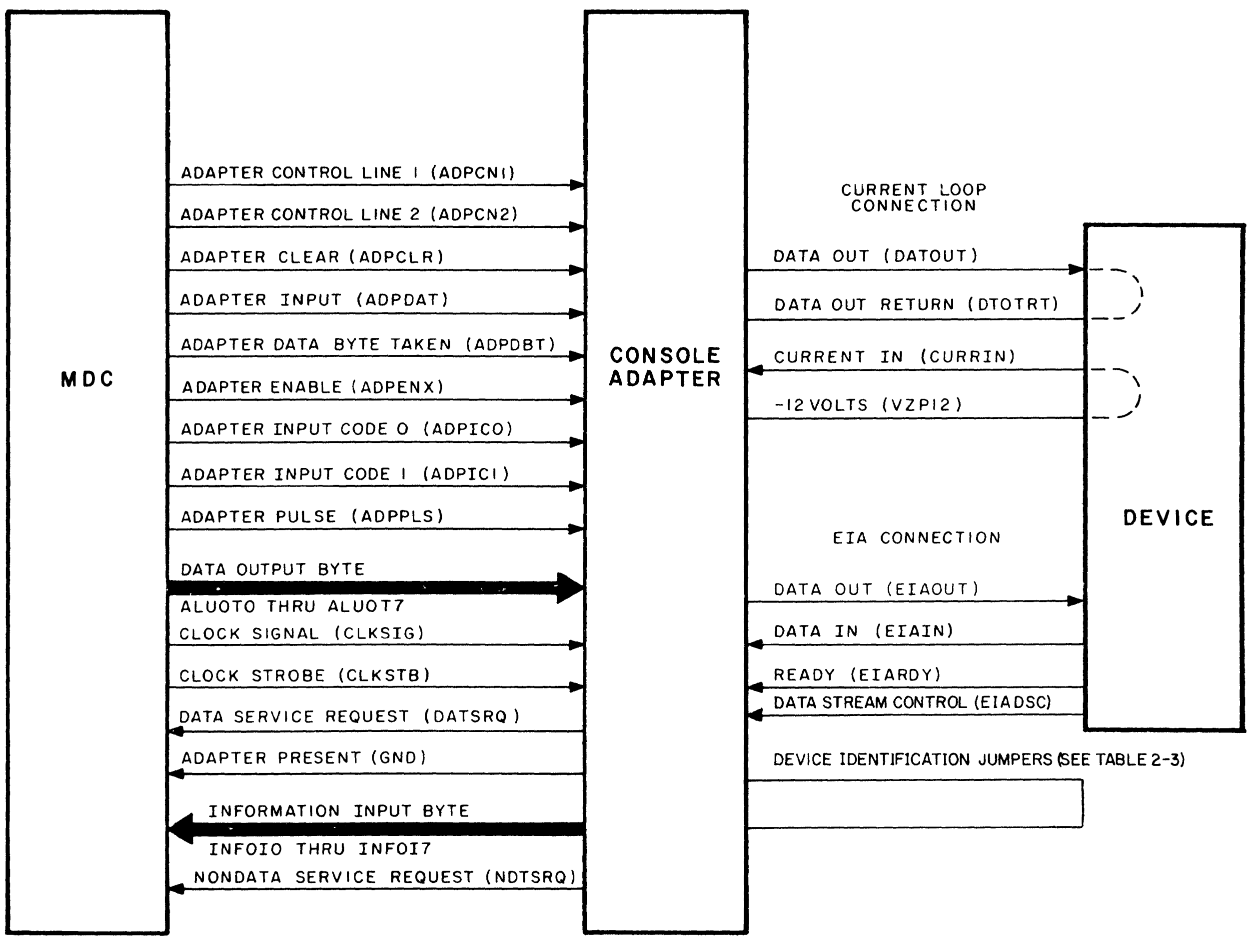


Figure 2-2 Interconnections - MDC to Adapter to Device

Table 2-1 MDC to Console Adapter Interconnections

MNEMONIC	NAME	FUNCTION
ADPCN1	Adapter Control 1 Line	Strobes control inputs into encoder, decoder, echo back, wrap-around, and 8-bit word flip-flops
ADPCN2	Adapter Control 2 Line	Not used
ADPCLR	Adapter Clear	Clears adapter logic
ADPDAT	Adapter Input	Loads data into the parallel-to-serial encoder
ADPDBT	Adapter Data Byte Taken	Notifies adapter that data byte has been taken from the serial-to-parallel decoder
ADPENX	Adapter Enable Signal	Enables adapter for input and output operations
ADPIC0, ADPIC1	Adapter Input Code 0, 1	Selects byte input to the MDC via the input multiplexer
ADPPLS	Adapter Pulse	Not used
ALUOT0 thru ALUOT7	Data Output Byte	Data output byte from the MDC; used to transfer both basic data and control information
CLKSIG	Clock Signal from MDC	A 125-nanosecond duration clock signal at 4 mHz from the MDC
CLKSTB	Clock Strobe from MDC	A 35- to 55-nanosecond duration clock signal at 4 mHz from the MDC
DATSRQ	Data Service Request	Data service request from the adapter to the MDC
GRND	Adapter Present	Informs MDC that adapter is installed
INFOI0 thru INFOI7	Input Byte	Input byte from the input multiplexer of the adapter to the MDC
NDSRQ	Nondata Service Request	Not used

Table 2-2 Console Adapter to Device Connections

MNEMONIC	NAME	FUNCTION
DATOUT	Data Out*	Carries data out from the adapter to the device in serial form
DTOTRT	Data Out Return*	Return for data out
CURRIN	Current In*	Carries data information from the device into the adapter in serial form
VZP12	-12 Volts*	-12 volts required to complete current loop
EIAOUT	Data Out**	Carries data out from the adapter to the device in serial form
EIAIN	Data In**	Carries data in from the device to the adapter in serial form
EIARDY	Device Ready**	Device is operational (see subsection 2.3.9 for details on use of this signal)
EIADSC	Data Stream Control**	Used by certain devices to control when the adapter may transmit data

*Current loop
**EIA connection

NOTE: Details on current loops and EIA connections are contained in subsections 2.3.2 and 2.3.3.

2.3.2 Data Output

Data is sent out to the device as either a 7-bit character with an even parity or an 8-bit character with no parity bit. Either one or two stop bits are used in transmitting characters to or from the device. As shown in Figure 2-3, signal 2STOPB into the UAR/T chip determines whether one or two stop bits are used. This signal is made true (two stop bits used) by jumpering in the device cable. Selection of character size and the use of a parity bit is determined by the setting (or resetting) of signal ALUOT0 (bit 1 of the ALU of the MDC).

When the MDC starts an output operation, the firmware looks at the task word to see if 7-bit (with parity) or 8-bit characters are to be output. The firmware then sets or resets ALUOT0 accordingly. The stop bit and character information is then loaded into the UAR/T by MDC firmware and the load control logic in the adapter. Note that when the UAR/T chip is programmed to transmit 7-bit words with a parity bit that the chip generates an even parity bit for transmission to the device. If the MDC sends out a parity bit under this programming condition, the UAR/T simply ignores it.

When the control signals are loaded into the UAR/T, the MDC firmware also sets the Enable Output Request (ENBORQ) flip-flop which allows a data service request to be sent to the MDC when the Character Ready flip-flop is set. This flip-flop is set either by the Character Done signal out of the UAR/T chip, or by the Data Stream Control signal.

By design of each adapter, only one of these signals is enabled for setting the Character Ready flip-flop. When the Character Done signal is enabled, the Character Ready flip-flop is set when the UAR/T finishes sending out the complete character in serial form. When the Data Stream Control signal is enabled, the device notifies the MDC that it is ready for a new character.

When the control signals are loaded into the UAR/T, if data is to be sent to the device, the Input 8-Bit flip-flop is also set to allow data out to flow to the device via the EIA RS232C transmitter or current loop logic. The Input 8-Bit flip-flop is not set when the MDC sends out characters to the UAR/T to perform timing delays. This prevents the device from using these characters to generate meaningless printouts or displays.

When the device process firmware detects the data service request, it acquires data from the CPU memory and sends it a byte at a time to the UAR/T. Transfer of the bytes to the UAR/T is implemented by the load data logic in the adapter, and the number of bytes transferred is determined by the direct memory access (DMA) logic in the MDC.

In the UAR/T each data byte received from the MDC is made into a compatible character for sending to the device. This consists of:

1. Generating a parity bit for 7-bit characters received from the MDC
2. Adding a start bit
3. Adding either one or two stop bits as specified.

NOTE

TTY-33 devices use two stop bits. This is determined by the configuration jumpers on the device connector card.

Each character is then serially encoded and sent to both the current loop logic and a transmitter specified by RS232C of the Electronic Industries Association (EIA).

In current loop transmission, the quiescent state (mark state) is established by a 60-milliampere current in the line. This is implemented by connecting one end of the output loop to +12 volts through a 200-ohm resistor, and the other end to the data output driver which can sink current when in the low output state. The space state in the current loop is established by the data output driver being turned off.

When using an RS232C transmitter, the mark state is established when the output of the transmitter is at -12 volts, and the space state when the output is at +12 volts.

The data is sent to the device in either the current loop or the EIA RS232C form, depending upon the configuration of the cable connecting the adapter to the device. Figure 2-4 is an example of a character transmitted to a device.

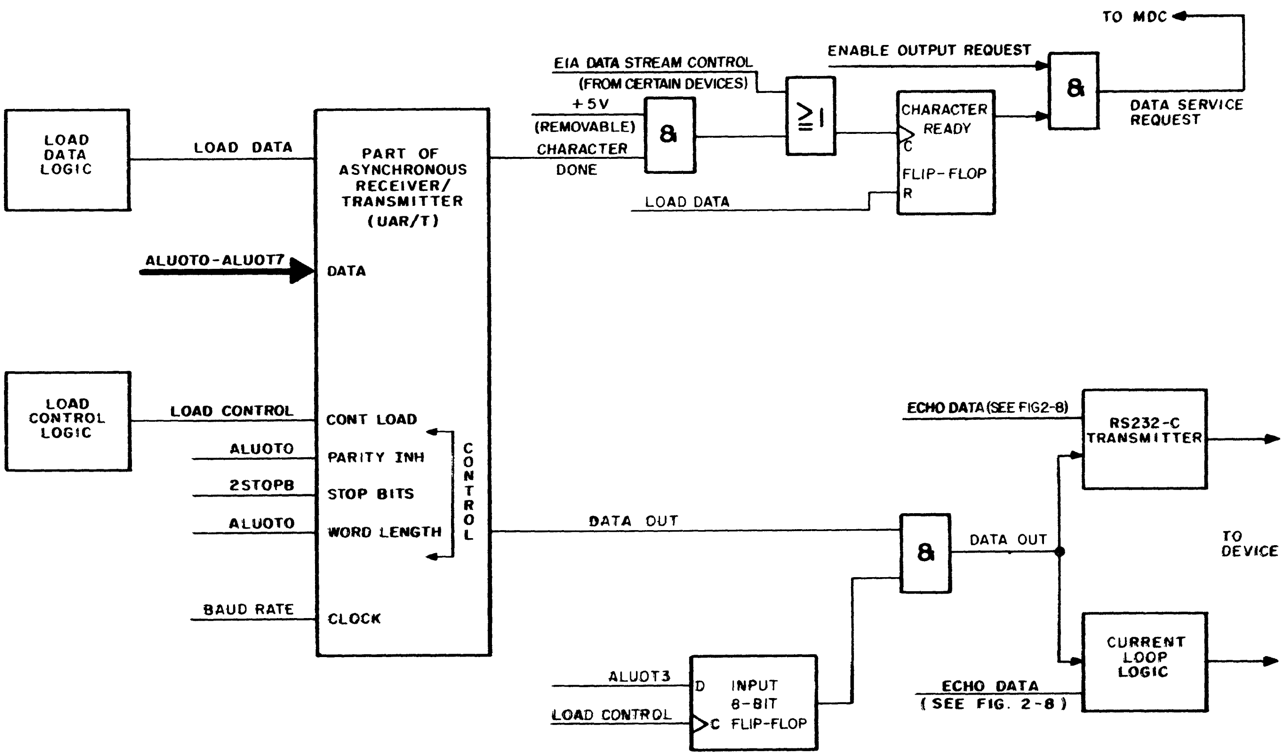


Figure 2-3 Data Output

2.3.3 Data Input

Data is received by the UAR/T chip from the device as either a 7-bit character with an even parity bit or an 8-bit character with no parity bit. Received data characters normally also contain a start bit and either one or two stop bits (see Figure 2-4). Selection of character size, and the use of a parity bit are determined by the setting (or resetting) of signal ALUOT0 (bit 0 of the ALU of the MDC). In the UAR/T the stop bits are stripped from the character before it is sent to the MDC, and if the first stop bit is not detected, an error signal is generated. When 7-bit characters with even parity are specified, the most significant data bit (DATA00) is sent to the MDC as a Zero. The UAR/T, however, uses the bit received from the device to check parity and to generate a parity error signal if an error is detected. (See Figure 2-5.)

When the MDC starts an input operation, the firmware looks at a task word to see if 7-bit characters with parity or 8-bit characters are to be input. If 7-bit characters with parity are to be input, ALUOT0 and ALUOT3 in the MDC are reset. If 8-bit characters are to be input, ALUOT0 and ALUOT3 are set. The stop bit and control information is then loaded into both the UAR/T and the Input 8-Bit flip-flop by the MDC firmware and the load control logic of the adapter. Note that the echo mode can also be set at this time (see subsection 2.3.4 for details).

After the control signals are loaded into the UAR/T and the Input 8-Bit flip-flop, further action does not occur until the UAR/T receives a character from the device via either the current loop logic or a receiver as specified by RS232C of the Electronic Industries Association (EIA).

The current loop logic is basically a 60-milliampere loop as shown in Figure 2-6. The device switches the current loop on when marking and switches it off when spacing. This causes the LM1489 to see approximately -12 volts at its input when the device is marking and approximately +12 volts when it is spacing. The LM1489 then converts the marks to approximately +5 volts and the spaces to approximately 0 volts for use in the TTL logic of the adapter. The 1.5K resistor and the 1-μF capacitor at the input to the LM1489 are used to filter noise spikes out of the circuit.

The receiver specified by RS232C of the EIA translates voltages in the range of +3 volts to +12 volts to a TTL Zero (space) and a -3 volts to -12 volts to a TTL One (mark). Although the current loop and RS232C inputs are shown ORed in Figure 2-5, actually only one type input is connected between the adapter and the device.

When the serial-to-parallel decoder of the UAR/T receives the serial bit data (an 8-bit character with or without parity) from either the current loop or RS232C receiver, it assembles a parallel byte output. If parity was specified when the UAR/T was loaded with control information, it checks for an even parity error signal and generates an error signal if found incorrect. The decoder also checks for the receipt of the first stop bit and generates the Stop Bit Error signal if a discrepancy is noted. After updating the

Parity Error and Stop Bit Error signals, the decoder generates an Input Ready signal, which causes a Data Service Request signal to be sent to the MDC.

In response to the data service request, the MDC inputs the specified 7-bit or 8-bit byte into the MDC via the input multiplexer with the start and stop bits being stripped off by the UAR/T. When a 7-bit byte is transferred, the Input 8-Bit flip-flop is reset and inhibits sending the most significant bit (DATA00) of the assembled character to the MDC. When the MDC inputs the assembled character, its firmware sends out an acknowledge signal which causes the My Data Byte Taken (MYDBTL) signal into the decoder to clear the Input Ready out of the UAR/T.

If the decoder assembles two characters before one is transferred into the MDC, it generates a service rate error signal. Note that all error signals remain set until the following characters are assembled.

In a data input operation, the MDC normally transfers numerous bytes out of the UAR/T into the main memory under direct memory access (DMA) control.

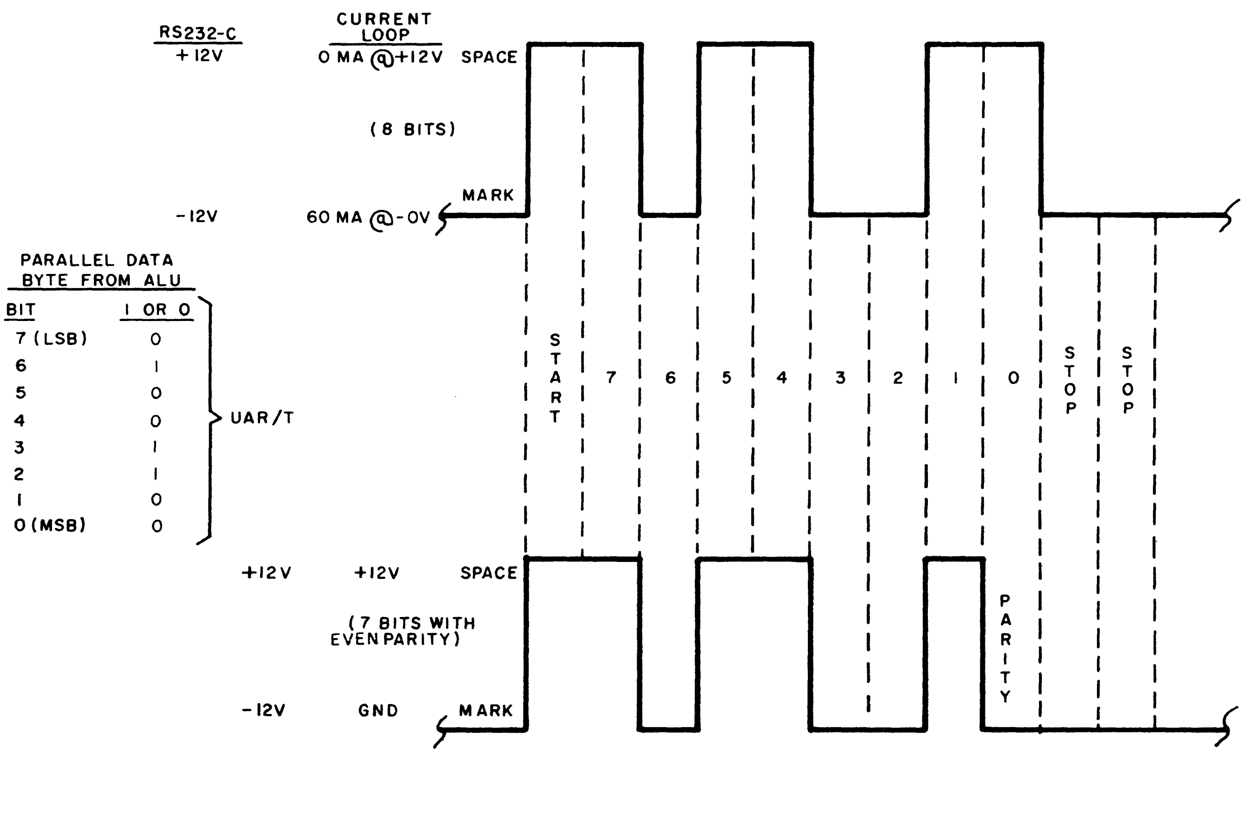


Figure 2-4 Character Example

HONEYWELL PROPRIETARY AND CONFIDENTIAL

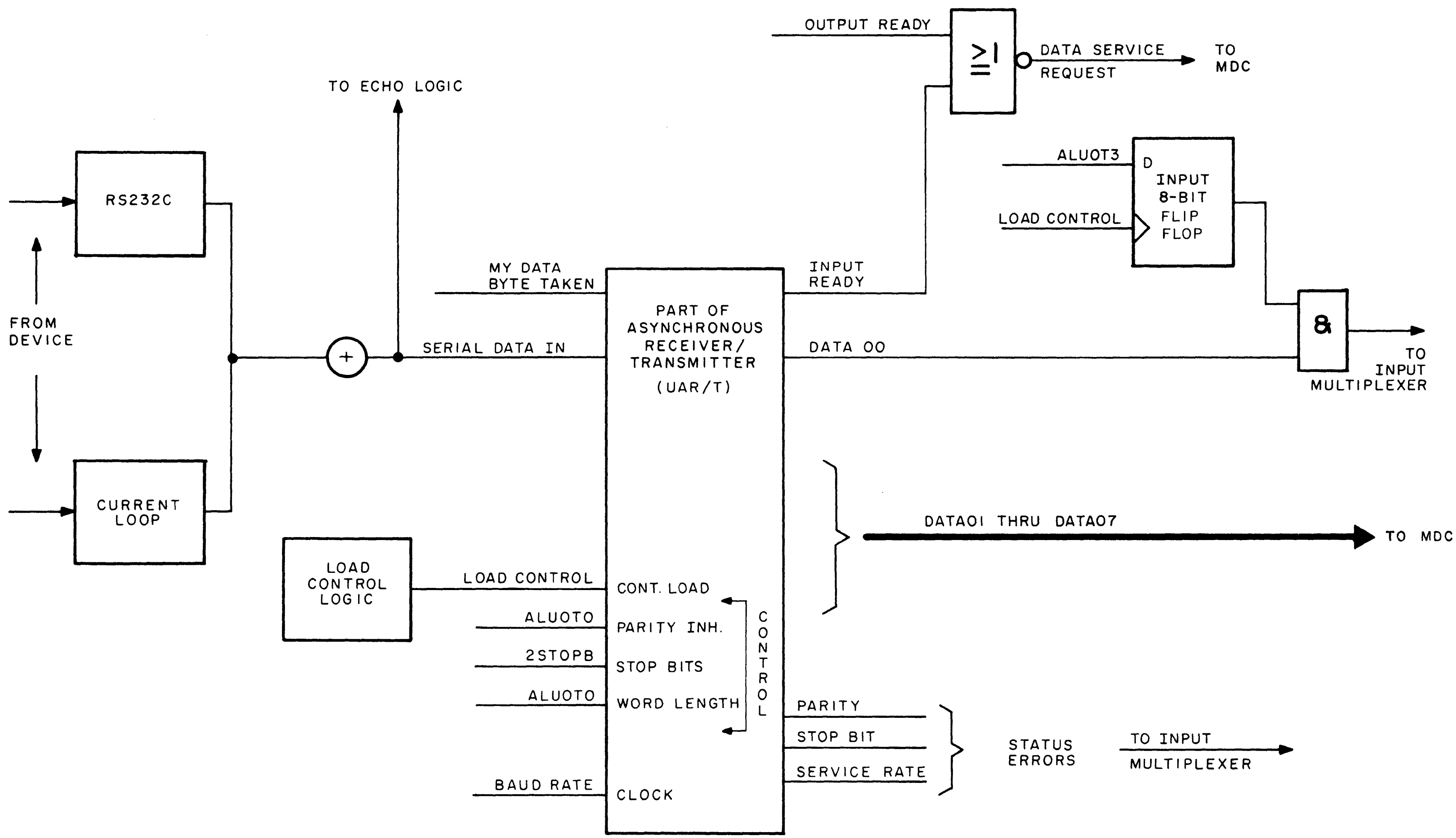


Figure 2-5 Data In

HONEYWELL PROPRIETARY AND CONFIDENTIAL

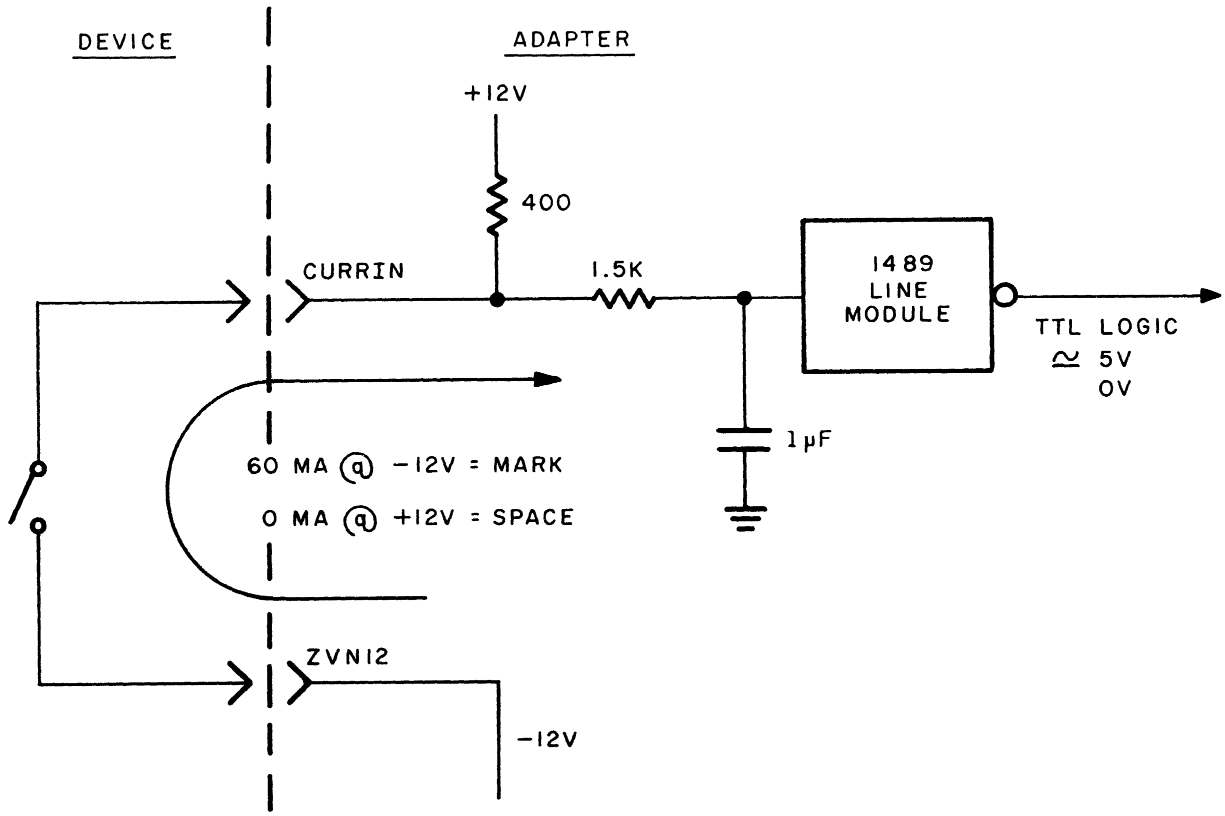


Figure 2-6 Input Current Loop

2.3.4 Echo Logic

The echo logic is used to send data received from the device directly back to the device so that it can be printed or displayed for the convenience of the device operator.

The echo mode is entered when the MDC issues a Load/Control (LOADCN) signal with ALUOT1 (bit 1 of the ALU of the MDC) set. This sets the Echo Back Mode flip-flop (see Figure 2-7). Setting of this flip-flop allows the data input stream from the device to be circulated directly back to the device via the current loop or EIA logic.

2.3.5 Wraparound Logic

The wraparound logic is used by software for diagnostic purposes. This logic allows serial data out of the encoder to be sent back into the decoder. Diagnostic software uses this capability to check the adapter by comparing a character it sends out for similarity when it receives it back. Device on-line or echo operations are precluded when the wraparound logic is enabled.

The wraparound mode is entered when the MDC issues a Load Control (LOADCN) signal with ALUOT5 (bit 5 of the ALU of the MDC) set. This action sets the Wraparound Mode flip-flop (see Figure 2-8). Setting of this flip-flop allows the data output stream from the encoder to be directly sent into the input of the decoder.

2.3.6 Baud-Rate Generation Logic

The output of the baud-rate generation logic (16XOUT) is sent directly to the UAR/T chip, which requires this frequency for both encoding and decoding characters. In addition, the 16XOUT signal is also sent to the timeout logic (see subsection 2.3.7).

The baud-rate generation logic is shown in Figure 2-9. Basically, this logic consists of a free-running crystal oscillator with an output of 7.3728 MHz, a divide-by-eight counter, a baud-rate generator, and a hexadecimal rotary switch.

The output of the crystal oscillator is divided by eight to provide the baud-rate generator with a required 921.6-kHz input. The output of the baud-rate generator is 16 times the selected baud rate. The baud rate is selected by setting the hexadecimal rotary switch in accordance with the table in Figure 2-9.

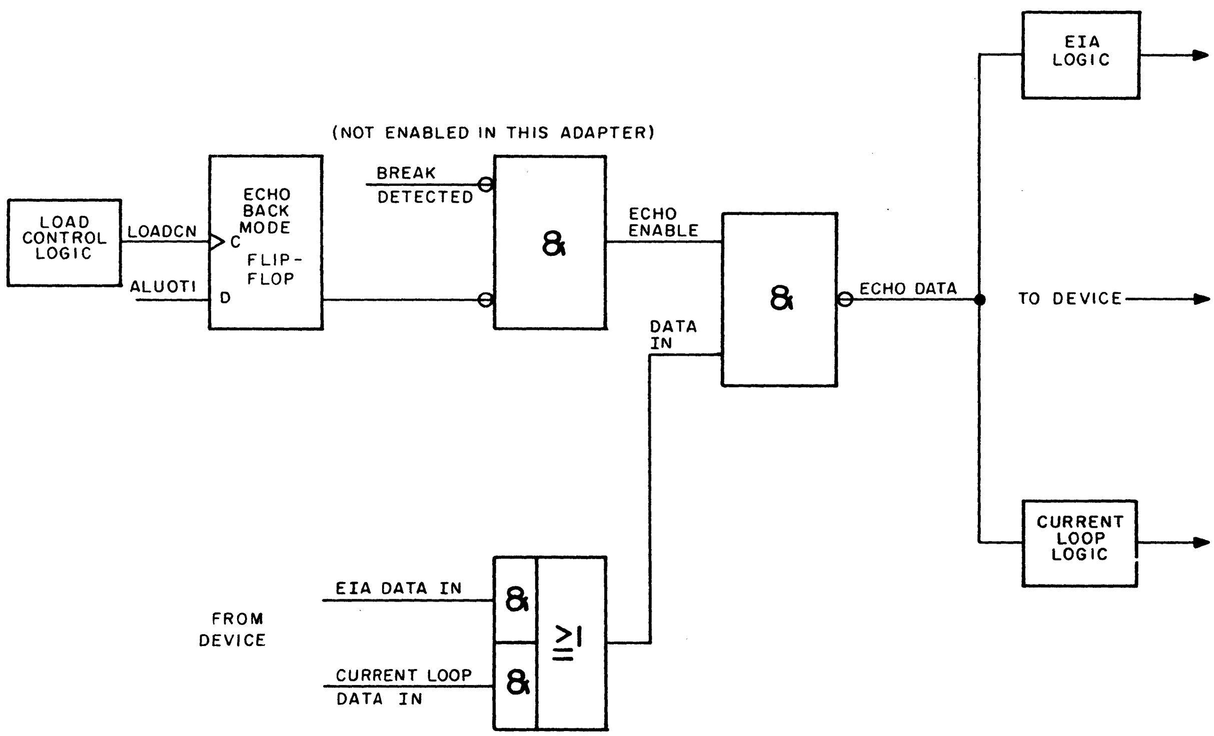


Figure 2-7 Echo Logic

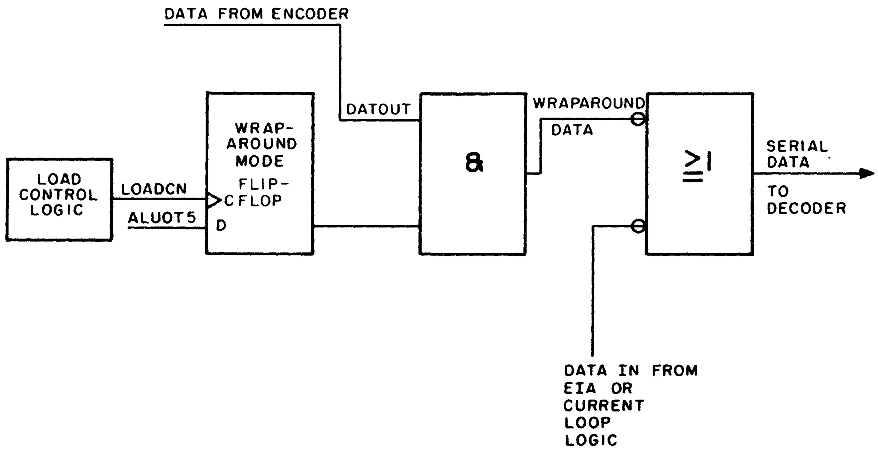
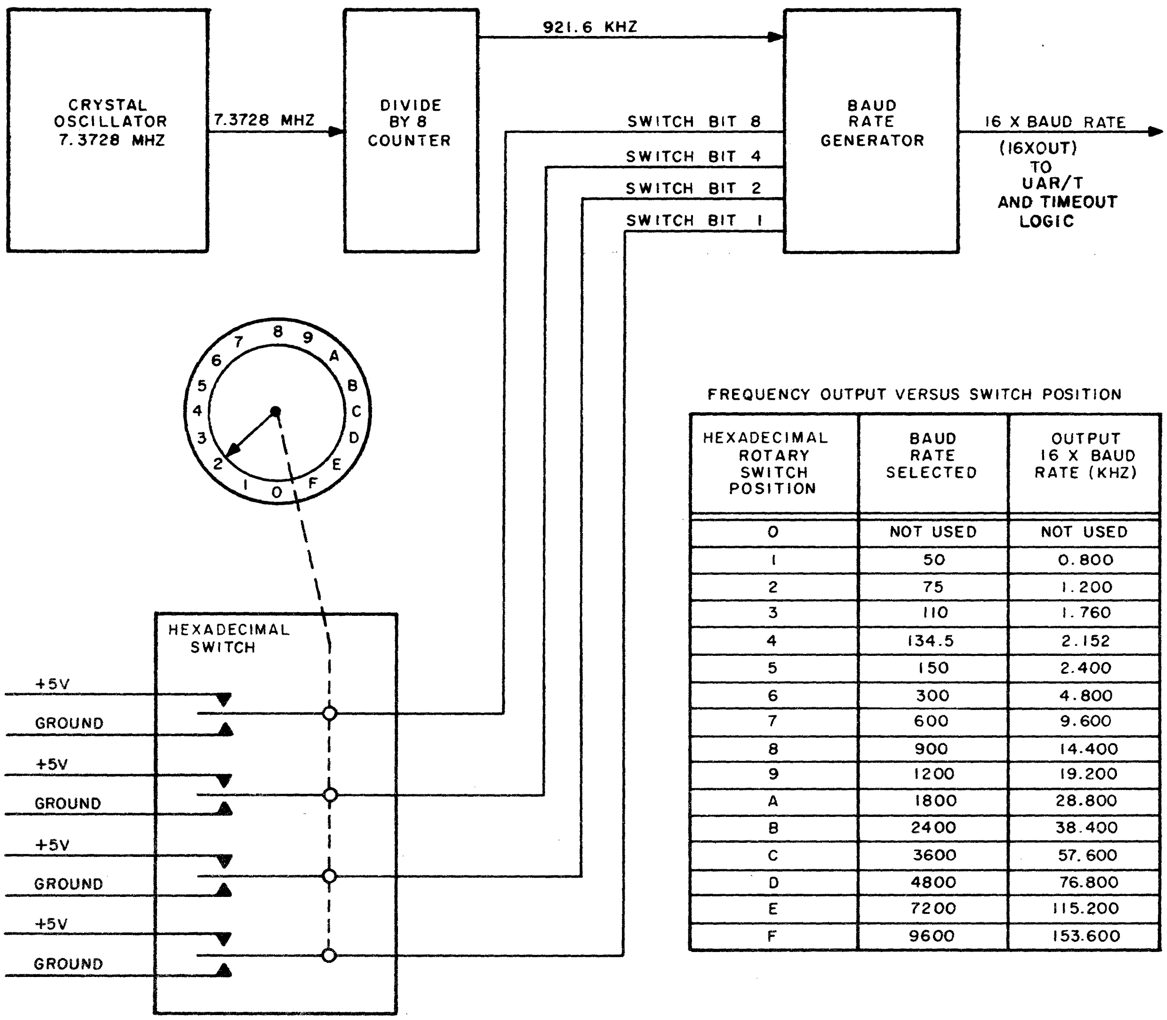


Figure 2-8 Wraparound Logic



FREQUENCY OUTPUT VERSUS SWITCH POSITION

HEXADECIMAL ROTARY SWITCH POSITION	BAUD RATE SELECTED	OUTPUT 16 X BAUD RATE (KHZ)
0	NOT USED	NOT USED
1	50	0.800
2	75	1.200
3	110	1.760
4	134.5	2.152
5	150	2.400
6	300	4.800
7	600	9.600
8	900	14.400
9	1200	19.200
A	1800	28.800
B	2400	38.400
C	3600	57.600
D	4800	76.800
E	7200	115.200
F	9600	153.600

Figure 2-9 Baud Rate Generation

2.3.7 Timeout Logic

The timeout logic is used to notify the firmware in the MDC that there has been no activity, either transmitting or receiving, in the adapter over a specified period of time.

As shown in Figure 2-10, the timeout logic primarily consists of four cascaded counters which divide the output of the baud-rate counter by 65,536, a timeout flip-flop, and clear timer logic. The intent of the timeout logic is to make the adapter and MDC compatible with the automatic motor turn-off feature of a TTY33.

When the adapter is set to operate at 110 baud, the timeout flip-flop is set at approximately 37 seconds if no character is loaded (LOADAT) into the encoder for transmission, or no data bit (MYDBTS) has been acknowledged as taken from the decoder during the 37 seconds. The timeout logic is disabled by an EIA ready signal when the EIA transmission mode is used.

When an output instruction is issued to the adapter, the MDC firmware first checks the timeout status bit, and if set it causes 10 character times to precede the DMA data. Only the first character, a delete character, is actually transmitted to the device. The following nine characters are inhibited from going to the device by clearing the Input 8-Bit flip-flop (see Figure 2-3). The time required to transmit these characters from the encoder to nowhere provides sufficient time for the TTY motor to get up to proper speed.

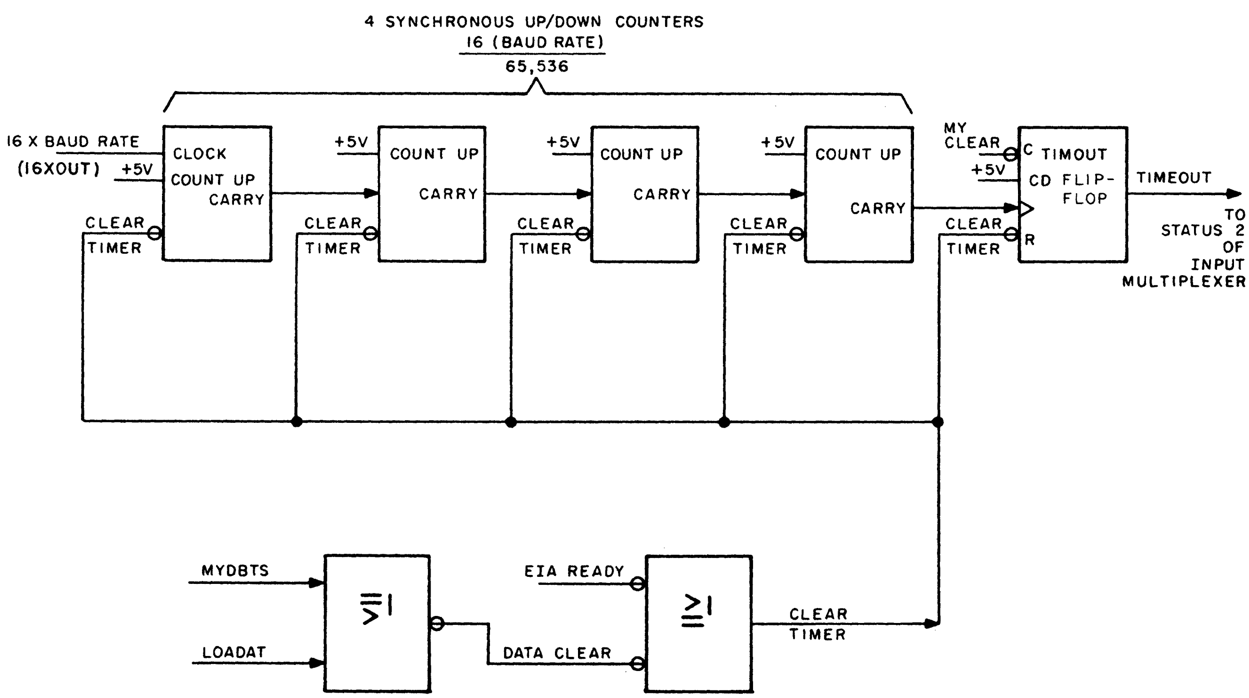


Figure 2-10 Timeout Logic

2.3.8 Input Multiplexer

The input multiplexer (see Figure 2-11) provides a means for the adapter to send data, identification, and status information to the MDC. Depending upon the setting of the code lines from the MDC, one of four bytes of information can be sent to the MDC at a given time. Specifically, these bytes are input data (DATA), identification (ID code), Status 1, and Status 2.

The setting of the code lines (which determine input byte selection), and the Enable line are under control of the MDC firmware. The Enable line is required because the multiplexer is a tristate device. When the multiplexer is not enabled, all its outputs are at infinite impedance (floating), thereby disconnecting all logical inputs from the multiplexer to the MDC. When the multiplexer is enabled, the state of each bit of the selected byte into the MDC is determined by the state of the connected bit into the multiplexer from the adapter. The multiplexer disconnecting capability allows multiplexer inputs from other adapters on the MDC to be in parallel, but only inputs of the enabled adapter are entered into the MDC logic.

The data byte is used to send the MDC data information received from the device. As described in subsection 2.3.3, the Data Service Request line signals the MDC when a new data information byte is valid. MDC firmware then controls transferring the byte into the MDC.

The ID code is used by software to determine what specific type device is associated with the channel. The adapter only supplies the least significant byte of the two-byte software ID code word. The most significant byte is supplied by the MDC; for a console adapter it is always 20_{16}. The byte furnished by the adapter is shown in Table 2-3. The bit configuration for the adapter ID byte is established by selective jumpering in the cable to the device.

The first status byte (Status 1) is used by the MDC to form the most significant byte of a two-byte status word which it forms for software. Firmware of the MDC inputs this byte and modifies it for software at:

1. The end of execution of every output address and range (IOLD) command
2. Master Clear
3. Output control command with either the Initialize or Stop I/O bits set
4. Detection of a keystroke.

The specific inputs to Status 1 of the input multiplexer by the adapter are listed in Table 2-4.

The second status byte (Status 2) into the multiplexer from the adapter provides information to the process firmware of the MDC so that it can determine operational hardware states of the adapter. The process firmware then executes further firmware action (routines), depending upon the operational hardware states found in the adapter.

The specific input to Status 2 of the input multiplexer are listed in Table 2-5.

2.3.9 Device Ready Logic

Device ready logic generates a Device Ready (DEVRDY) signal which can be used by the programmer to determine whether the adapter is available and ready to transmit and/or receive.

As shown in Figure 2-12, when the device is using an EIA RS232-C interface with the adapter, the EIARDY signal causes DEVRDY to go high, providing that the encoder is not in the process of serially transmitting (Character Done). Likewise, when a current loop interface is used, the LINECN signal causes DEVRDY to go high.

Resetting of DEVRDY is only allowed when the adapter is not in the echo back mode, and the encoder is not in the process of serially transmitting.

Note that an EIARDY signal indicates that the device is on-line and the power is on. On the other hand, a LINECN signal simply indicates that a current loop interface is being used.

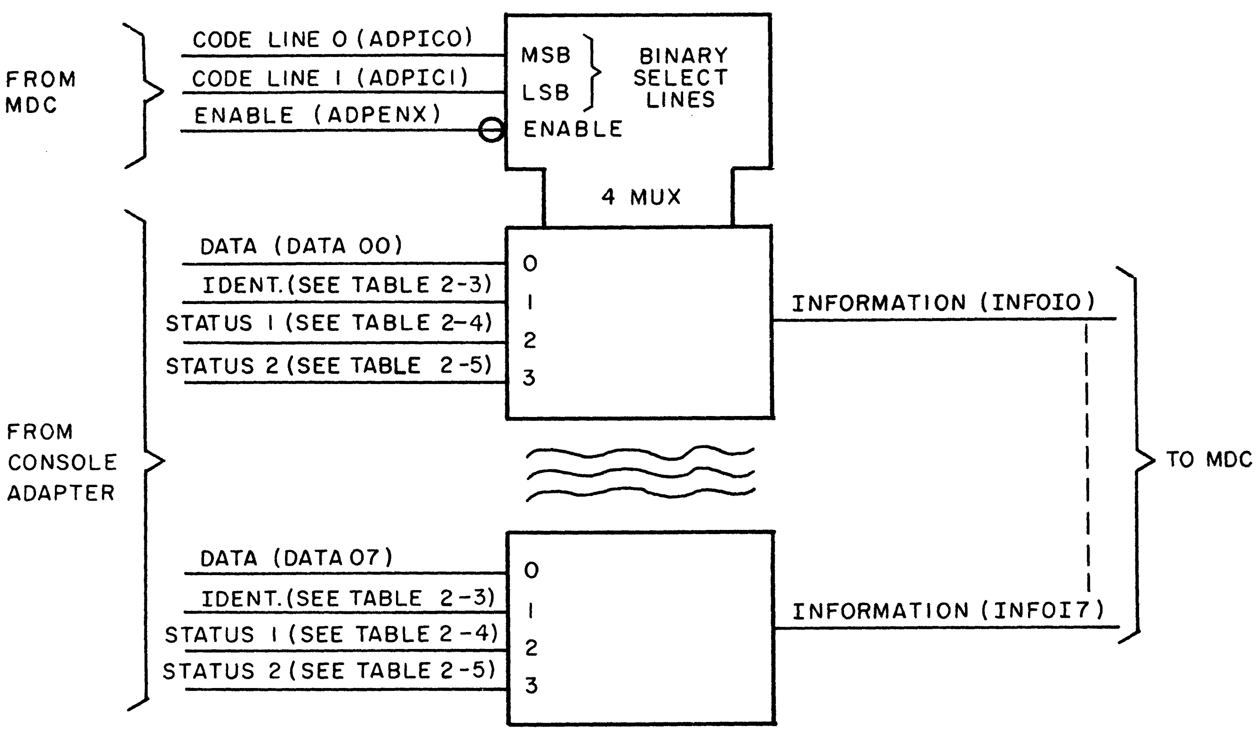


Figure 2-11 Input Multiplexer

Table 2-3 ID Code Development

DEVICE ATTACHED	ID CODE (Hex) MDC	ID CODE (Hex) ADAPTER	ADAPTER ID BYTE 8	9	10	11	12	13	14	15
KSR	20	18	0	0	0	1	1	0	0	0
ASR	20	19	0	0	0	1	1	0	0	1
Display	20	1A	0	0	0	1	1	0	1	0
HISI TYPWRTR CONSOLE	20	1C	0	0	0	1	1	1	0	0
NONE	-	-	0	0	0	1	1	0	1	1

Table 2-4 Status 1 Inputs

BIT	MNEMONIC	DESCRIPTION
0	DEVRDY	Device Ready - The device is ready (EIA) or the device is connected (current loop)
1		Not used
2	SVRTER	Service Rate Error - MDC failed to input at the proper rate
3	EVNPTE	Parity Error - An even parity error was detected
4		Not used
5	NOSTOP	Stop Bit Error - The first stop bit was not detected
6		Not used
7		Not used

Table 2-5 Status 2 Inputs

BIT	MNEMONIC	DESCRIPTION
0	TIMOUT	Timeout - The timer has timed out
1	INPRDY	Input Ready - An input character is ready
2	OUTRQT	Output Data Request - Adapter is ready for an output character
3		Not used
4	NDTSRQ	Not Enabled - Always zero
5		Not used
6		Not used
7	DATOUT	Data Out - Serial data output stream

NOTE

All unused bits should be at Zero.

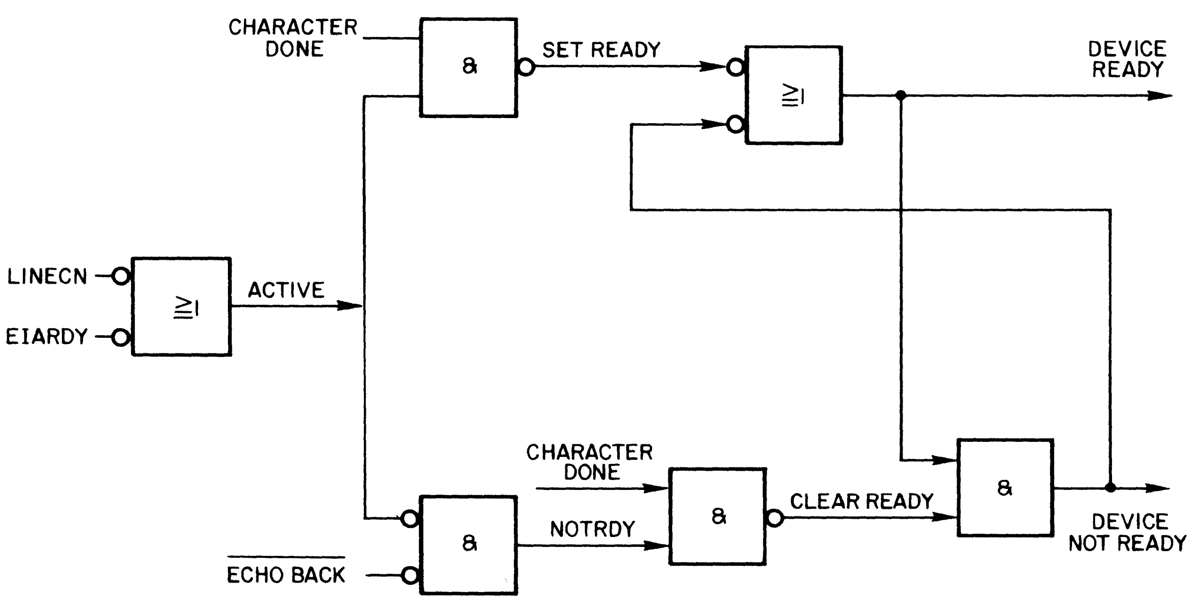


Figure 2-12 Device Ready Logic

III
FIRMWARE

3.1 GENERAL

The firmware associated with the console adapter is physically located in the microprogram control store section of the multiple device controller (MDC). Table 3-1 is a glossary of terms used with the console adapter firmware. Subsection 3.2 describes the section of scratch pad memory (SPM) in the MDC that is dedicated to the console adapter. An overview flowchart and supporting intermediate flowcharts of the firmware are shown in subsection 3.3. Note that several firmware commands are usually required to perform functions called out in these flowcharts. Details on these firmware commands are called out in the firmware listing contained in the MDC reference manual.

Table 3-1 Glossary of Firmware Terms (Sheet 1 of 4)

TERM	EXPLANATION
Abort Termination	The abort termination by the operator during an input operation. This is implemented by a CC2 (see Control Character 2 below).
Attention Character	Any character received from the keyboard when the channel is not busy, or is busy in the output mode.
Attention Flag	Bit 3 of Work Location 2 (WL02) in Scratch Pad Memory (SPM) of the MDC. It is set when an Attention character is detected by the firmware.

Table 3-1 Glossary of Firmware Terms (Sheet 2 of 4)

TERM	EXPLANATION
Break	An Attention character with no stop bits.
Break Enable	Bit 6 of the Output Task Word. If set, a detected Break will be posted and the current operation stopped immediately. If not set, a detected break will be posted, but the current operation will continue.
Channel Not Busy	No programmed activity in progress in the device.
Character Count	Number of characters transferred in the present operation. The character count is used in an input operation to ensure that a CC1 does not delete a character in memory below the starting address of the operation. In an output operation, the character count is used to count the 10 nulls used to time motor start.
Configuration	Bit definition of Control Characters (CCs) used in the firmware.
Configuration Word	Words stored in the Scratch Pad Memory of the MDC to establish Control Characters (CCs) to be used by the adapter.
Control Character 1	Allows the immediately preceding character to be deleted from an input message. This is accomplished by incrementing the DMA range and decrementing the memory address so that the previously received character is overwritten in memory.
Control Character 2	Causes an input operation to be aborted. Sets bit 8 of the Status Word. Software then may allow a retry of the previous operation.
Control Character 3	In the imput mode, this character allows the operator to normally terminate an input operation before the DMA range equals zero. CC3 is unconditionally sent to the device when the operation is terminated by the operator and by design can be used for carriage return (CR) or stop tape, etc. Similarly, a CC3 can be optionally sent to the device at the end of DMA range. In the output mode, software by design can use a CC3 to perform various operational signals to the device; e.g., carriage return, stop a reader, start a reader, etc.
Control Character 4	See Escape Character.
Device Not Ready	Device not ready to operate.

Table 3-1 Glossary of Firmware Terms (Sheet 3 of 4)

TERM	EXPLANATION
Direct Memory Access (DMA)	The procedure of directly accessing consecutive locations in the computer memory through a specific number (range) of locations.
Early Termination	Normal termination of an input operation by the device operator before the DMA range equals zero. This is implemented by Control Character 3 (CC3), and when it occurs it sets bit 9 of the Status Word. See subsection 3.2.6.
Echo	Echo mode. The data stream from the device is sent directly back to the device for display. Bit 7 of the Task Word determines whether or not Echo will be used.
Escape Character	Same as Control Character 4 (CC4). Allows the next character received in the input mode to be sent directly to computer memory. This escape feature allows the operator to send CC1, CC2, CC3, or CC4 to memory without performing any associated control function.
Escape Flag	Bit 0 of Work Location 4 (WL04) of the Scratch Pad Memory (SPM) of the MDC. It is set in the input mode when a received character is similar to configured Control Character 4 (CC4) (the Escape Character).
Motor Start Time	The time taken by the motor on a TTY33 to get up to operating speed after it was turned off by its automatic turn-off feature.
Non-Busy Attention	An attention character detected when the channel is not busy.
No Stop Bit Error	An error signal generated in the hardware of the adapter indicating that it received a character from the device with no appended stop bit. The firmware interprets an Attention character with no stop bits to be a Break character.
Range	The number of consecutive locations in memory still to be accessed during a DMA procedure.

Table 3-1 Glossary of Firmware Tbrms (Sheet 4 of 4)

TERM	EXPLANATION
Service Rate Error	The decoder in the adapter assembled two characters before one was transferred to the MDC. A Service Rate Error is posted in bit 3 of the Status Word. See subsection 3.2.6.
Start-Wait	A firmware routine in the primary firmware of the MDC.
Status 1	A status byte generated by the adapter hardware which is used to form the most significant byte of the Status Word stored in the Scratch Pad Memory of the MDC. See subsection 2.3.8 for details.
Status 2	A status byte generated by the adapter hardware which is used by the firmware to determine operational conditions in the adapter. See subsection 2.3.8 for details.
Software Status Word	A 16-bit word in the Scratch Pad Memory of the MDC which is used by software. See subsection 3.2.6 for details.

3.2 SCRATCH PAD MEMORY

The console adapter firmware is assigned one quadrant of the scratch pad memory (SPM) for storing firmware operational information. The quadrant has 64 addressable locations which store 8-bit bytes of information. Most of the locations store information which is generated by the primary firmware of the MDC. The contents of these locations is described in the firmware section of the MDC manual. The locations which contain information unique to the console adapter are listed in Table 3-2 and are described in the following subsections. Note that when a 16-bit word is used to store pertinent information, two SPM locations are required, and the most significant byte (MSB) equals bits 0 through 7 and the least significant byte (LSB) equals bits 8 through 16.

3.2.1 Task Word

The task word is placed in the SPM by software. The task word for an output operation is different than that for an input operation. The Console Adapter firmware uses this word to make decisions and to take optional procedures. The format for an output operation task is shown in Figure 3-1, and for an input operation in Figure 3-2.

Table 3-2 Console Adapter
Scratch Pad Memory Locations

ADDRESS (Hexadecimal)	MNEMONIC	CONTENTS
06	TSK1	Task Word (LSB)
07	TSK2	Task Word (MSB)
10	CNF1	Control Character 2 (CC2)
11	CNF2	Control Character 1 (CC1)
12	CNF3	Control Character 4 (CC4)
13	CNF4	Control Character 3 (CC3)
18	STS1	Status Word (LSB)
19	STS2	Status Word (MSB)
1A	STS3	Attention Character
26	DID1	Device I.D. (LSB)
27	DID2	Device I.D. (MSB)
2E	WL01	Input - Character Counter; Output - Null Counter
2F	WL02	Attention Flag (bit 3 of byte)
2F	WL02	Attention Flag (bit 3 of byte)
30	WL03	Data Storage During Deference-Suspension
31	WL04	Escape Flag/CC4 Found (bit 0 of byte)
32	WL05	Control Byte for UAR/T Chip in Adapter

HONEYWELL PROPRIETARY AND CONFIDENTIAL

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

BIT 10 - Specifies the character width:

0 = 7-bit with even parity. The adapter will take a byte from memory and alter its high order bit (if necessary) in order to obtain an even number of One bits. The byte is then sent to the device.

1 = 8-bit direct transcription. The adapter takes the byte from main memory and sends it to the device with no modification.

BIT 9 - Specifies whether or not Control Character 3 is to be sent to the device when Range equals zero:

0 = Send CC3

1 = Do not send CC3

BIT 6 - Specifies the required action when a break is detected during an output operation:

0 = Post break detected but continue current operation until its completion.

1 = Post break detected and stop current operation.

Figure 3-1 Output Operation Task Word

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

BIT 10 - Specifies the character width:

0 = 7-bit with even parity. The adapter will ensure that the byte received from the device contains an even number of One bits. In addition, when storing the byte in main memory, it will set the high order bit to Zero.

1 = 8-bit direct transcription. The adapter performs no checks on the byte received from the console and stores all eight bits in the main memory.

BIT 9 - Specifies whether or not Control Character 3 is to be sent to device when Range equals zero:

Range = 0	CC3 Detected
0 = Send CC3	0 = Do not send CC3
1 = Do not send CC3	1 = Send CC3

BIT 8 - Specifies whether or not the checking of all Control Characters (CCs) is to be inhibited:

0 = Not inhibited

1 = Inhibit all CCs

BIT 7 - Specifies whether or not echoing is required:

0 = No echo

1 = Echo

BIT 5 - Specifies the disposition of the previously received Attention character from the keyboard:

0 = Append character to the beginning of this input message.

1 = Discard the Attention character and input the message as received.

Figure 3-2 Input Operation Task Word

3.2.2 Control Character 1 (CC1)

This character is placed in the SPM by software. It is not used in the output mode. It is usually set by software convention to an ASCII ∧, but it can be set to any other convenient bit setting. The firmware, upon detecting this character, increases the range by one and decreases present address by one. (See Table 3-1 and Figure 3-5.)

3.2.3 Control Character 2 (CC2)

This character is placed in the SPM by software. It is not used in the output mode. It is usually set by software convention to an ASCII @, but it can be set to any other convenient bit setting. The firmware, upon detecting this character, will abort an input operation. (See Table 3-1 and Figure 3-5.)

3.2.4 Control Character 3 (CC3)

This character is placed in the SPM by software. It can be used in both the input and output mode. It is usually set by software convention to an ASCII CR, but it can be set to any other convenient bit setting. In the input mode, the firmware uses this character as a signal from the device operator to terminate an input operation before the DMA range equals zero. At the end of an input operation terminated by the operator, the CC3 is unconditionally sent to the device to return the carriage (CR), stop the reader, etc.

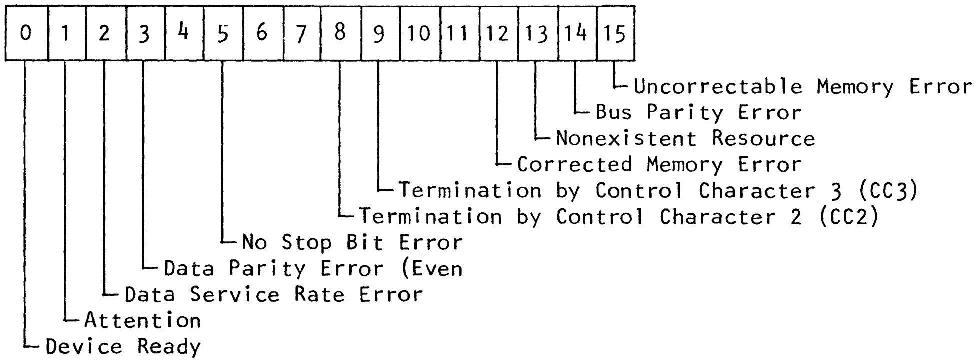
At the end of the DMA range, CC3 may also be optionally sent to the device to perform the miscellaneous device hardware functions. In the output mode, software by design can use a CC3 to perform miscellaneous device hardware functions.

3.2.5 Control Character 4 (CC4)

This character is placed in the SPM by software. CC4 is also called the Escape character and is used in the input mode only. When received, it allows the next character received to be sent directly to memory. This allows the character representing CC1, CC2, CC3, or CC4 to be sent directly to the memory without performing a control function. Note that by firmware design, if the Escape character is all Zeros, the escape feature is nullified.

3.2.6 Software Status Word

The software status word (Figure 3-3) is generated by the adapter hardware and both the console adapter and primary MDC firmware. The setting of bits 0, 2, 3, and 5 is controlled by the adapter hardware (see subsection 2.3.8). The setting of bits 1, 8, and 9 is controlled by console adapter firmware, and bits 12, 13, 14, and 15 by the primary MDC firmware. Figure 3-3 lists the information contained in each bit of the software status word. More details on an Attention character are contained in the following subsection.



All the above status bits are set by the condition indicated and most will remain set until the next IOLD command or until Initialize. An input status order does not clear any of these bits. An exception to this is Device Ready, which always represents the condition of the console (Ready or Not Ready).

BIT	DESCRIPTION
0	Device Ready - In current loop connections, this indicates that the terminal is connected by the presence of the output current when the attachment is not busy (i.e., not sending data to console). In EIA connections, this indicates Data Terminal Ready for direct connections or Carrier Detect for a data set connection.
1	Attention - The operator depressed a key on the console indicating that he wants the attention of the software. Attention is discussed in greater detail in subsection 3.2.7.
2	Data Service Rate Error - On receive, the MDC has failed to take a character from the adapter before the next character arrived (100 milliseconds for TTY 33/35).
3	Data Parity Error - A data parity error was detected by the adapter. This is only applicable if the Task Word has been set to indicate 7-bit mode (bit 10 = 0).
5	No Stop Bit Error - A character was received from the device without a stop bit.
8	Termination by Control Character 2 - Control Character 2 was detected in the input data stream and caused a termination.
9	Termination by Control Character 3 - Control Character 3 was detected in the input data stream and caused a termination.
12	Corrected Memory Error - The data read from memory was accompanied by a signal indicating that an error existed which has been corrected.
13	Nonexistent Resource - A reference was made to a memory address which did not exist.
14	Bus Parity Error - The attachment detected bad bus parity on a transfer toward the MDC.
15	Uncorrectable Memory Error - The data read from memory was accompanied by a signal indicating that an error existed which the memory could not correct.

The reaction of the MDC to errors in the case of the device is as follows:

1. Errors during a data transfer will be indicated by a status report at the end of the record.
2. Errors on I/O commands from CP will set the appropriate status bit and immediately interrupt the CP. The I/O command will be acknowledged normally and stored in the MDC memory, but will not cause any further action. If interrupts are blocked (Level = 0), the command in error will be used as if there were no error.

Figure 3-3 Software Status Word

3.2.7 Attention Character

This location (1A) in the SPM stores the first character that the device operator struck to ask for an Attention interrupt. When the console adapter firmware is not processing an input or output order, any character struck will cause an Attention interrupt. The software can determine that the interrupt was an Attention interrupt by looking at bit 1 of the Status Word (see Figure 3-3). If the Break key is the cause of the Attention interrupt, all the bits of the character will be Zero, and the No Stop Bit error signal will be set.

In cases where more than one key is struck in succession, the console adapter firmware will retain only the first character and discard succeeding characters until an input order or a Stop I/O occurs.

When the console adapter is in the output mode, the character of the first key struck will be stored in this 1A location of the SPM, and bit 1 of the Status Word will be set. The Attention interrupt will be generated and the output stream stopped only if the Break key was struck and bit 6 of the output task word is set to One (see Figure 3-1). Otherwise, the software will become aware of the outstanding Attention interrupt at the normal termination of the output operation.

3.2.8 Device I.D.

Locations 26 and 27 store the device identification. Software uses this word to determine what specific type device is attached to the channel. This word is formed by the MDC and adapter as described in subsection 2.3.8. The I.D. is not used by firmware to operate the console adapter.

3.2.9 Work Location 1 (WL01)

This character is stored in location 2E of the SPM. In the input mode, it is used to count characters transferred from the adapter to computer memory. This count is required to ensure that a Control Character 1 (CC1) does not delete a character in memory below the starting address of the operation.

In the output mode, the character stored in WL01 is a count of the nulls sent to the device to time the starting of a motor.

3.2.10 Work Location 2 (WL02)

Bit 3 of this location is used as the Attention flag for the firmware. It is set when an Attention character is detected by the firmware.

3.2.11 Work Location 3 (WL03)

This location is used to store an input data character during a Deference-Suspension operation. During this operation the Console Adapter pauses during an input operation to allow the MDC to service data requests of higher priority.

3.2.12 Work Location 4 (WL04)

Bit 0 of this location is used as a flag in the input mode to signal that a received character compares with the configured Control Character 4 (CC4). When this flag is set, the next character received (presumably a CC1, CC2, CC3, or CC4) is sent directly to computer memory without performing any associated control function.

3.2.13 Work Location 5 (WL05)

This location stores a byte which is used by the firmware to send control information to the adapter. The specific information contained in the bits of this byte are shown in Figure 3-4.

0	1	2	3	4	5	6	7

- Bit 7 — Not used.
- Bit 6 — Not used.
- Bit 5 — WRAPMD - Specifies whether or not Wraparound mode is to be used:
  - 0 = No wraparound
  - 1 = Wraparound
- Bit 4 — ENBORQ - Enables the adapter to make a request to output a character from the adapter to the MDC:
  - 0 = ENBORQ not enabled
  - 1 = ENBORQ enabled
- Bit 3 — INP8BT
  - During Input:
    - 0 = Force data bit 0 to a Zero when transferring from adapter to main memory
    - 1 = Allow data bit 0 to be sent from adapter to main memory
  - During Output:
    - 0 = Inhibit data from the encoder from going to the transmitters
    - 1 = Allow data from the encoder to go to the transmitters
- Bit 2 — Not used.
- Bit 1 — ECHOBT - Specifies whether or not Echo mode is to be used:
  - 0 = No Echo mode
  - 1 = Echo mode
- Bit 0 — 8BITMODE
  - 0 (Input Mode) = 7-bit with even parity. The adapter will ensure that the byte received from the device contains an even number of One bits. When storing byte in memory, sets high order bit to Zero.
  - 0 (Output Mode) = 7-bit with even parity. The adapter takes byte from memory and alters bit 0 to obtain an even number of One bits (if necessary).
  - 1 (Input Mode) = Adapter performs no checks on 8-bit byte received from the device and stores all eight bits in main memory.
  - 1 (Output Mode) = Adapter takes 8-bit byte from main memory and sends it to the device with no modifications.

Figure 3-4 Adapter Control Byte

3.3 OVERVIEW OF CONSOLE ADAPTER FIRMWARE

Figure 3-5 is an overview flowchart of the firmware associated with the console adapter. The flow is analogous to that in the firmware listing which specifies the precise firmware commands used to execute the required operations. Refer to Table 3-1 for a glossary of the firmware terms used in the firmware flowcharts.

3.3.1 Operational Entry from MDC (See Figure 3-6)

As shown in Figure 3-5, the console adapter firmware starts an operation when the primary MDC firmware responds to a service request for the channel by directing operation to CONSOLE:GO. Then, after a wait to allow channels of higher priority to be serviced first, the operation proceeds. The firmware then tests for several basic conditions to see which way to branch. If the channel is not busy, it assumes that a key was struck by the device operator and goes to CN:KEYSTRK. If the channel is busy and the device is ready, the operation enters either CN:OUTMODE or CN:INPMODE as directed by software.

3.3.2 Keystroke (See Figure 3-7)

The keystroke firmware path is entered at the start of a console operation if the channel is not busy, or during an output operation if the device operator strikes a key. The firmware first determines if this is the first Attention character in this operational sequence by checking the setting of the Attention flag. If the flag is not set, it saves the character and updates the software status word. If the flag is set, it checks to see if this new Attention character is a Break by looking for a No Stop Bit error signal. If it is not a Break character, it disregards the character and returns the firmware flow back to the primary MDC firmware. If it is a Break character, it then updates the software status word. After the software status word is updated, the firmware determines when the key was struck. If the channel is not busy, it assumes that the key was struck when no programmed operation was in progress and it returns the firmware flow back to the primary MDC firmware via CN:DONE. It is then the function of software to take the required follow-on action. If the channel is busy after updating the software status word, the firmware assumes that the key was struck during a programmed output operation and the firmware flows into CN:KEYTEX. If the character is not a Break, the firmware flows back to the primary Start-Wait routine of the MDC. If the character is a Break, the output operation either continues or ceases depending upon the setting of bit 6 of the software task word. If the operation continues, the firmware goes to the primary MDC WAIT-CONTD routine; if the operation ceases, the firmware path is to CN:EXIT2.

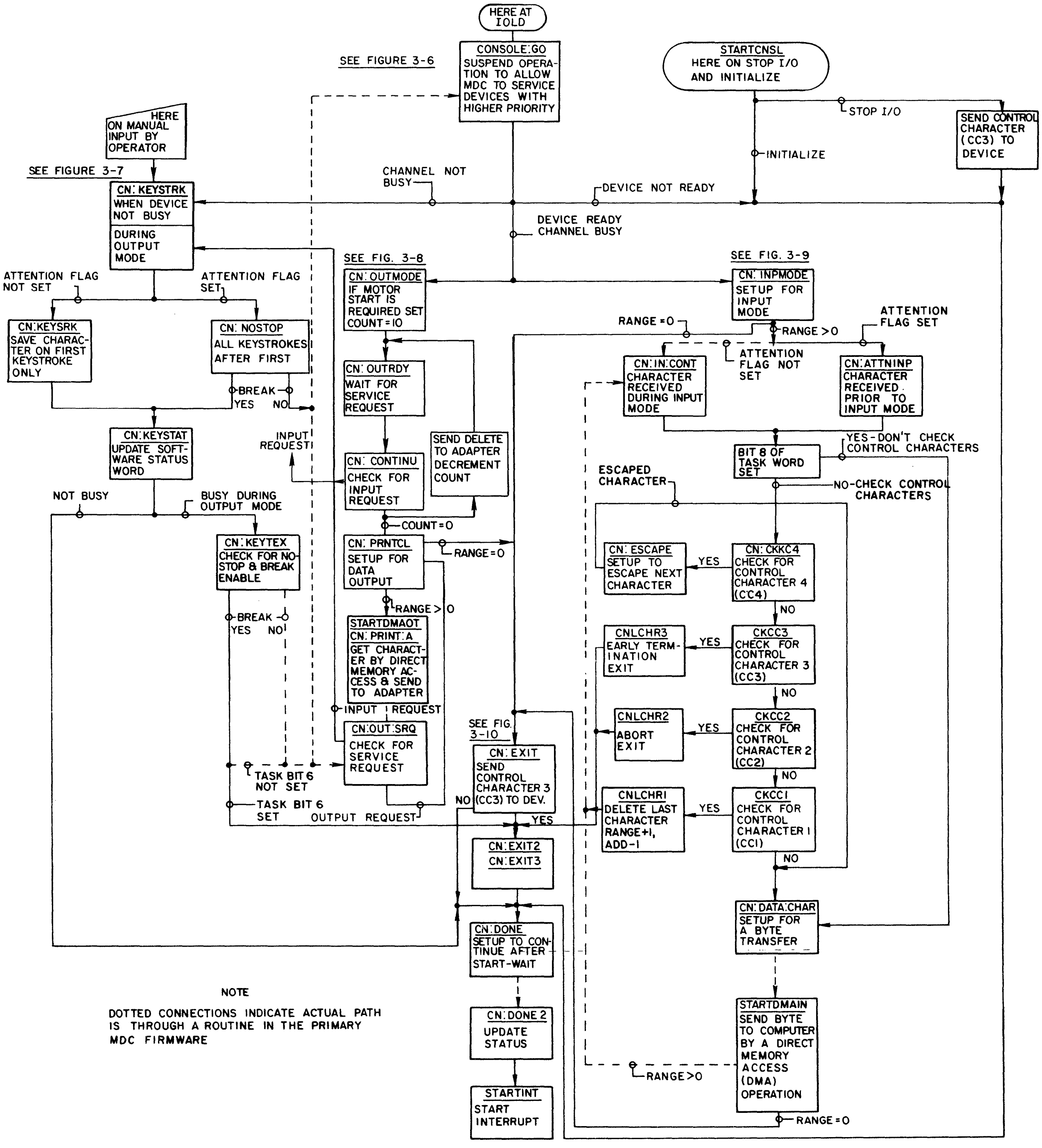


Figure 3-5 Overview of Firmware for Console Adapter

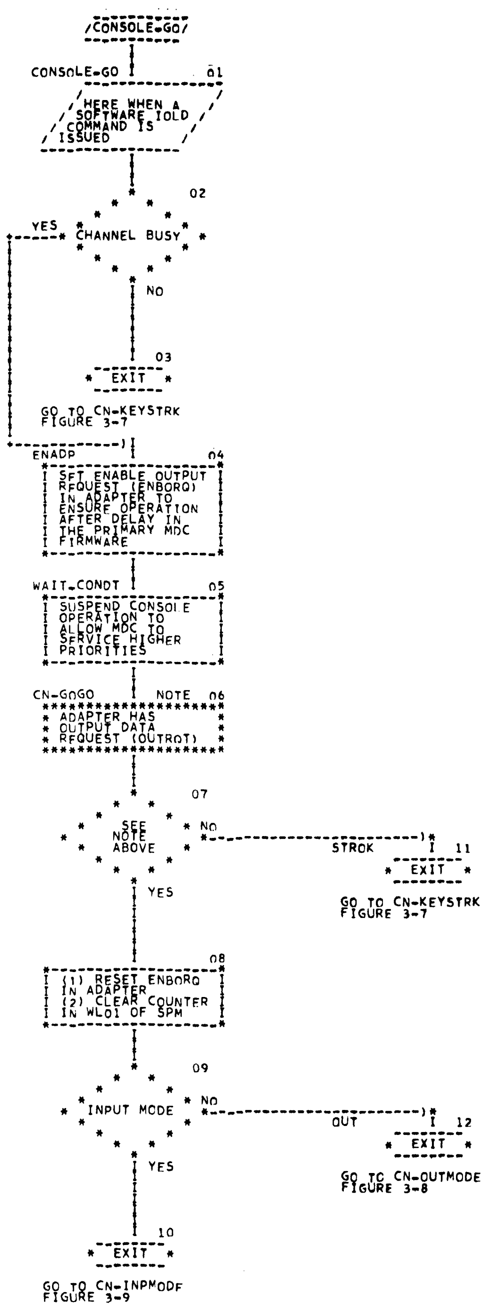


Figure 3-6 Operational Entry From MDC Flowchart

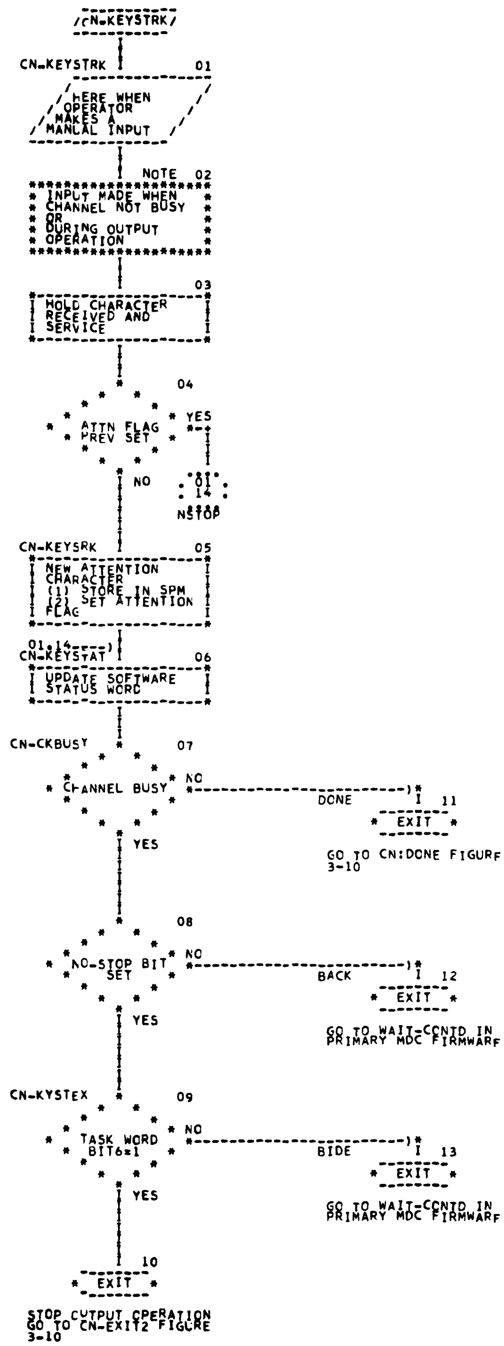


Figure 3-7 Keystroke Flowchart

3.3.3 Output Mode (See Figure 3-8)

When the firmware enters a programmed output operation, it first checks to see if a motor start operation is required, and if it is the counter is set to 10 and the INP8BT flip-flop in the adapter is set. The firmware then waits for a service request from the adapter. If the request is for an input from the adapter, the firmware goes into a Keystroke subroutine as described in subsection 3.3.2. If the request is for an output to the adapter, and a motor start is in progress, the firmware outputs a delete character to the adapter, decrements the count, and clears the INP8BT flip-flop. The flip-flop inhibits sending the delete character to the device on the following passes (see Figure 2-3).

When the firmware detects Count Equal to Zero, it stops looping for a motor start operation and begins the data output transfer. First, it enables the adapter to make output requests by setting signal ENBORQ, and then it sets up for a 7-bit with parity or 8-bit without parity operation. If the range equals zero, the operation is completed, and the firmware flows to the CN:EXIT subroutine. If the range does not equal zero, the firmware then uses the STARTDMAOT subroutine of the primary MDC firmware to DMA a character out of main memory and send it to the adapter. After the character is sent to the adapter, the firmware awaits the next service request from the adapter. If it is an input request, the firmware goes into the Keystroke subroutine as described in subsection 3.2.2 above. If the service request is an output request from the adapter, the firmware then loops back and DMAs another character to the adapter providing the range does not equal zero. When the range equals zero, the operation is completed, and the firmware flow is to the CN:EXIT subroutine.

3.3.4 Input Mode (See Figure 3-9)

When the firmware enters a programmed input operation, it first sets up the adapter to use the echo mode, if ordered, and for 7-bit with parity or 8-bit without parity operation. Presumably, the range does not equal zero at this time; if it does, the firmware stops the operation by flowing to CN:EXIT.

After checking the range, the firmware checks to see if the Attention flag is set. If set, the firmware uses software information to process the Attention character which was received prior to going into the programmed input operation. If bit 5 of the task word is not set, the character is appended to the front of the input message. If bit 5 is set, the Attention character is disregarded. If the character is to be appended and bit 7 (Echo) is set in the task word, the character is then sent to the adapter. The Attention character is then taken from the scratch pad memory and loaded into the bus data register in the MDC where it is processed as described further below.

If the Attention flag is not set when the firmware enters a programmed input operation, the character is taken from the adapter after checking for a Service Rate or a Parity error. The software status word is then updated with this error information, and the character is sent to the bus data register (BDR).

If bit 8 of the task word is set, the character is not checked for being a control character and it is sent directly to main memory via a direct memory access (DMA) operation as described later in this subsection. If bit 8 of the task word is not set when the character is sent to the BDR, the firmware checks it for being a configured control character (CC1, CC2, CC3, or CC4) (see Figure 3-9 for details). In general, the following occurs while the received character is in the BDR.

1. If a CC4 was received on the previous pass through the input mode firmware, the new character (presumably a CC1, CC2, CC3, or CC4) will be sent to main memory without performing the associated control function. This is called the Escape feature.
2. If the configured CC4 is all Zeros, the Escape feature is not allowed.
3. If the new received character is a configured CC3, operation is normally stopped by the device operator, and the operation is terminated via CN:EXIT2. Bit 9 of the software status word is also set to notify software.
4. If the new received character is a configured CC2, operation is aborted by the device operator, and the operation is terminated via CN:EXIT2. Bit 8 of the software status word is also set to notify software.
5. If the new received character is a configured CC1, the last character placed in main memory will be overwritten by the next character received. However, if no previous character was received during this input operation (character count equals zero), overwriting is not allowed.

None of the control characters are sent to the main memory except when they follow a CC4 (the Escape character).

When the firmware does not find a character in the BDR to be a control character, or when no check for a control character was ordered on the character, it is sent to the main memory by a DMA operation. After the character is sent into memory, another range check is made and if range does not equal zero, the firmware loops back to CN-IN-CONT after a stop-wait delay and processes the next character from the adapter as described above (see Figure 3-9). After the character has been sent to memory and range equals zero, the firmware completes the operation via CN:EXIT.

3.3.5 Exits and Dones (See Figure 3-10)

The firmware stops operations in various ways. During any operation, when it tests and finds range equals zero, it first flows through CN:EXIT and sends CC3 to the device if bit 9 of the Task Word is not set. When finishing a Keystroke subroutine which occurred during an output operation, or when an input operation is terminated by the device operator, the exiting flow is through CN:EXIT2 and CN:EXIT3. Here all control information to the adapter except ENBORQ (Enable Output Request) is cleared, and CC3 is unconditionally sent to the console adapter. When finishing a Keystroke subroutine which occurred when no programmed operation was in progress, or when the device was found not ready when starting any operation, the exiting flow starts with CN:DONE. Then, after a start-wait delay, the adapter control byte and CC4 is cleared and the firmware goes into the START-INT subroutine to try to interrupt the MDC.

3.3.6 Start Console

As shown in Figure 3-5, when the primary MDC firmware receives a Stop I.O or Initialize order for the console adapter, it sets the firmware flow at STARTCNSL. If a Stop I/O order is found, a Control Character 3 (CC3) is sent out to the device to cause some designed operation (e.g., carriage return, stop paper tape reader, etc.). The firmware flow for both Initialize and Stop I/O is then to CN:DONE (see Figure 3-10), where clearing and housekeeping functions are performed.

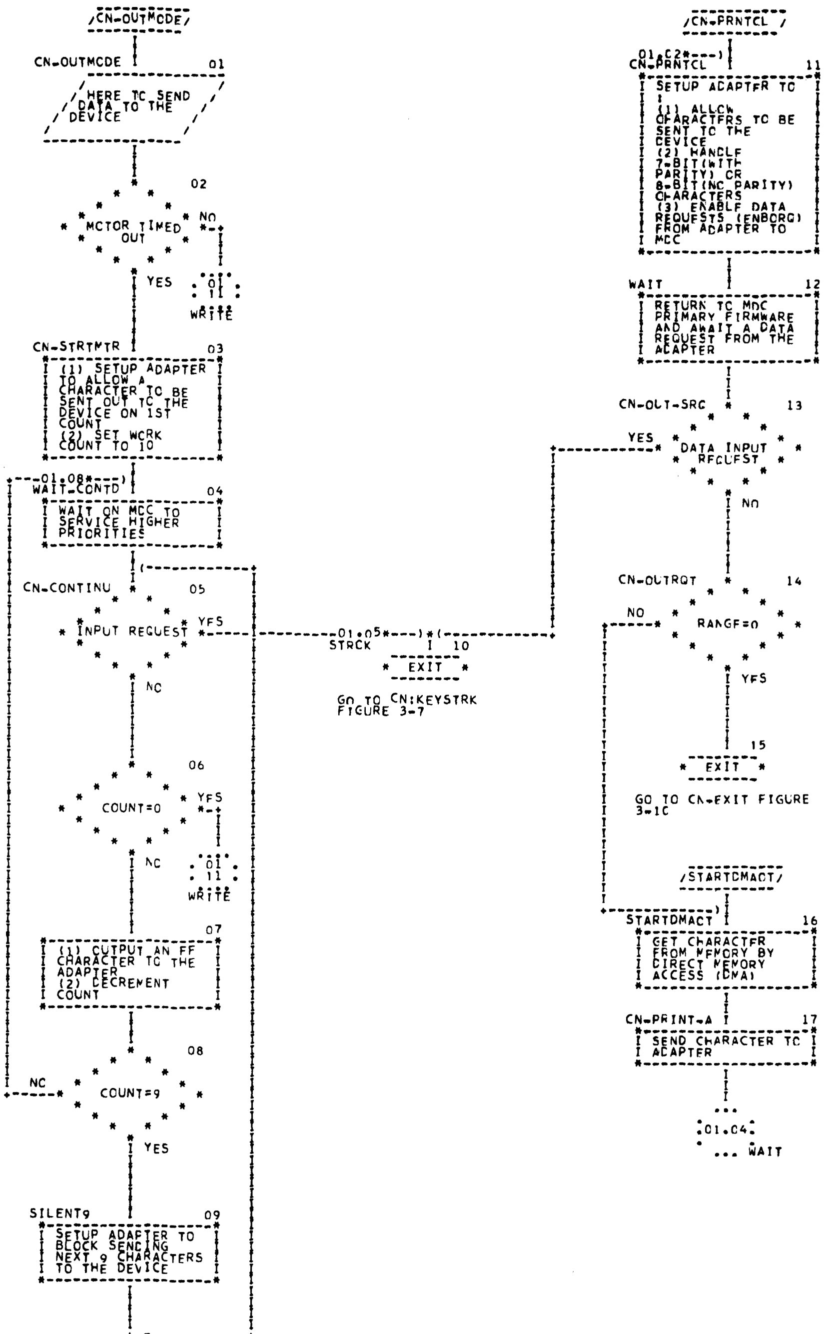


Figure 3-8 Data Output Mode Flowchart

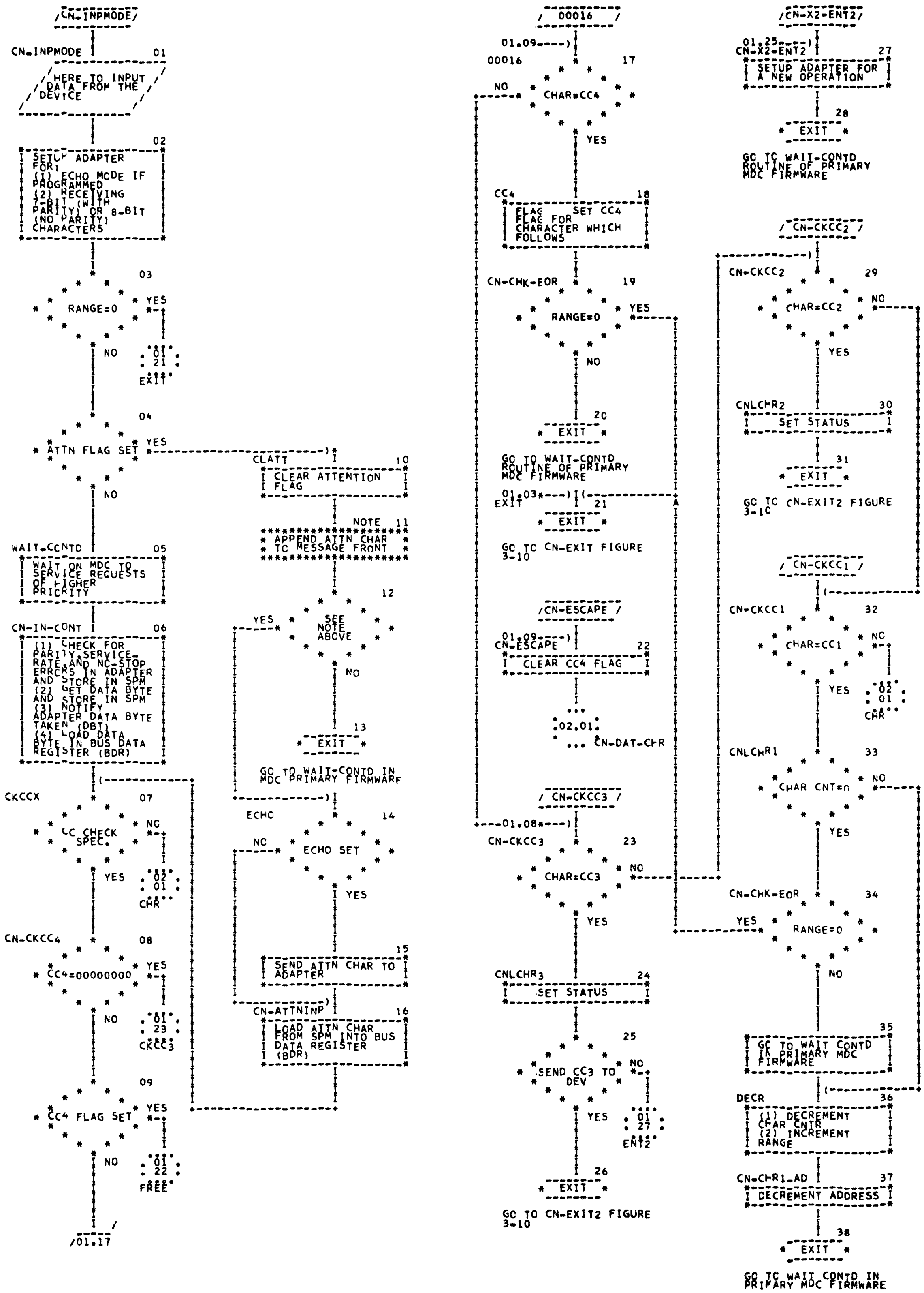


Figure 3-9 Data Input Mode Flowchart (Sheet 1 of 2)

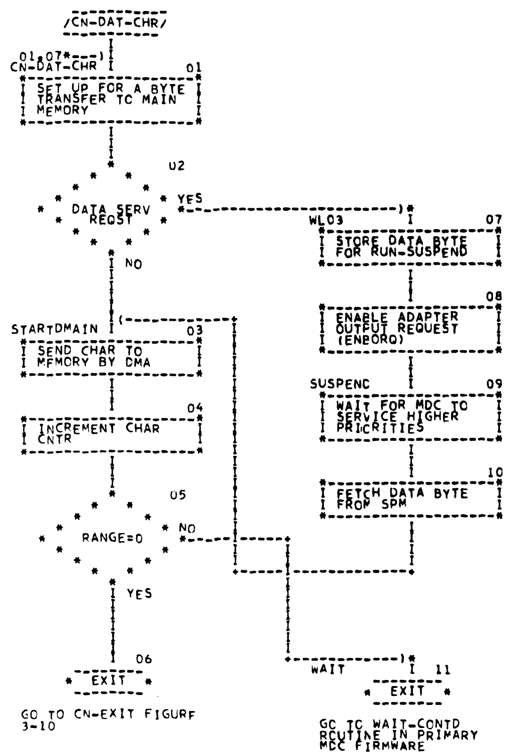


Figure 3-9 Data Input Mode Flowchart (Sheet 2 of 2)

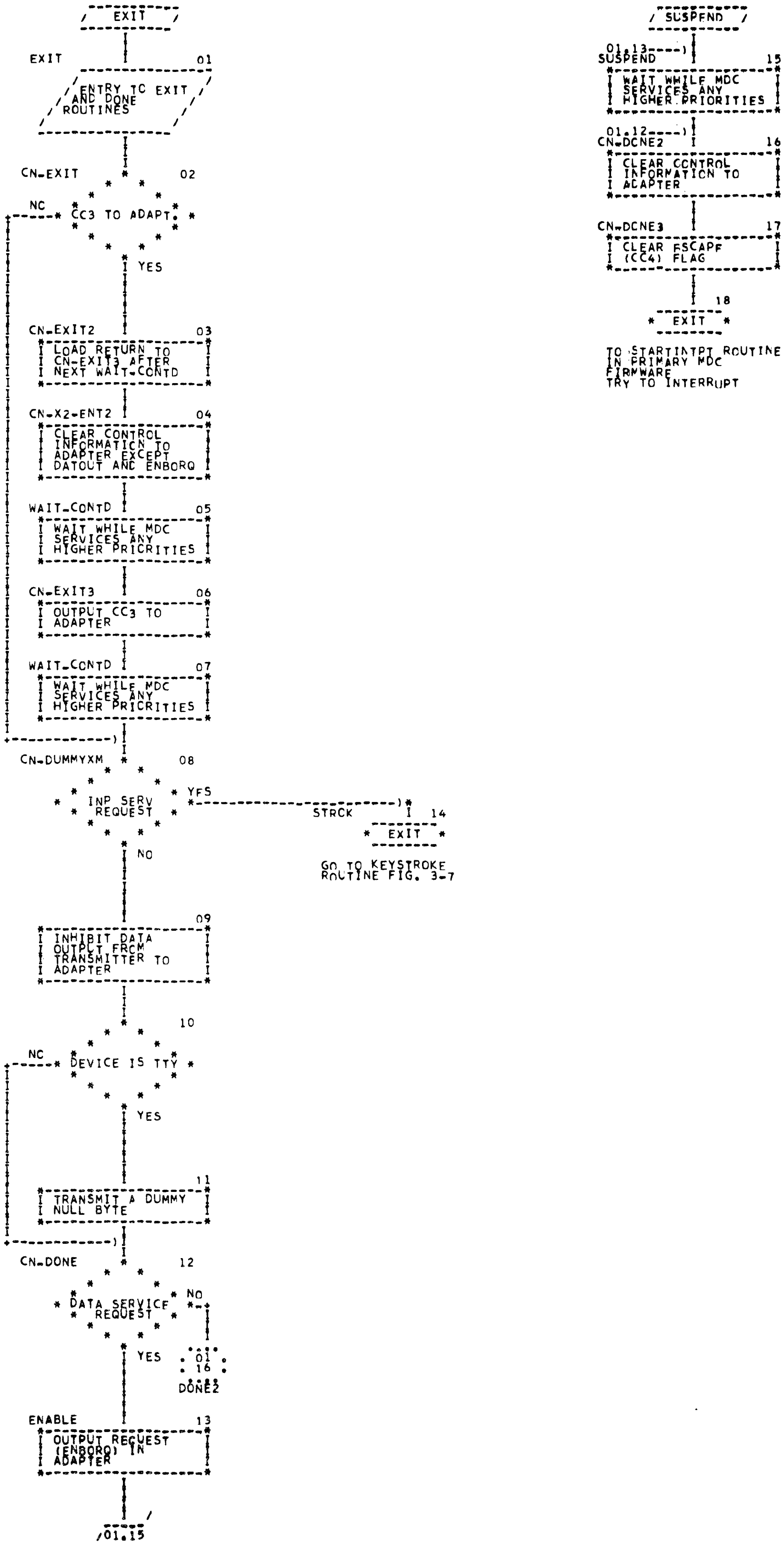


Figure 3-10 Exits and Dones Flowchart

APPENDIX
INSTALLATION INFORMATION

This appendix provides information on the basic differences between the interfaces of different devices and the adapter. This information will enable the changing of devices in the field. Note that only full duplex devices are usable with this adapter. Specifically covered in this appendix are:

- Differences between the EIA and the current loop modes
- Changing the current loop mode from a basic 60 mA to a 20 mA flow.
- Changing device identification
- Changing from one to two (or vice versa) stop bits
- Installing/removing the Device Timeout feature
- Setting the baud rate of the adapter.

A.1 EIA/CURRENT MODE LOOPS

The Console Adapter is designed to operate in either an EIA mode or a current loop mode. The current loop mode is normally used for teletype devices and devices which are used direct connect over distances greater than 30 feet, which is the nominal EIA limit. Although the Console Adapter contains logic for operating in either the EIA or the current loop mode, only one mode is used with a device at a given time. It is possible to change from one mode to the other by changing the cable connection between the adapter and by removing or replacing jumpers on the associated cable paddle (see Figure A-1). Care should be taken to study all steps required before making an EIA/current loop change.

A.1.1 Current Loop Information

The output current loop supplies current to the device for the transmission of output data information. As shown in Figure A-1, +12V is connected to pin 14 of the device cable connector via a 200-ohm resistor. This is the positive side of the output current loop. Pin 2, which is the other end of the current loop, is connected to the open collector of a type 75452 driver. When energized, this driver causes current between +12 and ground through the 200-ohm resistor to provide a nominal 60-milliampere current. Device Ready for a current loop is derived by detecting current in the output loop. Pin 14 must be below +5 volts when connected to the device to report a Device Ready status. To change the adapter to transmit with a nominal 20-milliampere loop, the 200-ohm resistor called out in Figure A-2 must be changed to 800 ohms. When using a current loop over long distances, it may be necessary to change the 200-ohm or 800-ohm resistors slightly to maintain adequate current. High baud rates may not operate with 20-milliampere loops over long distances due to cable band width limitations.

An input current loop supplies current to the device for the transmission of input data information. As shown in Figure A-1, a jumper is normally installed between pin 11 and pin 12 to establish the input current loop. This jumper places +12V on the type 1489A receiver when the line is open at the device (no current flow). When the line is closed at the device, the receiver sees -12V and 60 milliamperes is flowing in the current loop. To change the adapter to receive with a nominal 20 milliampere loop, the 390-ohm or 2 800-ohm resistors shown in Figure A-2 must be changed to 1200-ohm resistors. When using a current loop over long distances, it may be necessary to change the 390-ohm or 800-ohm resistors slightly to maintain adequate current.

A.1.2 EIA Connection Information

As shown in Figure A-1, the EIA output to the device originates at a type 1488 driver and is sent to the device via pin 27 of the connector. Pin 28 is the ground return for this output.

The EIA input from the device is pin 23 of the connector, and the EIA input ground is pin 24.

Note that in the EIA mode a Terminal Ready signal is required from the device. This signal enters the adapter via pin 25 and is fed to a type 1489A receiver. The ground return for this signal is pin 24.

A.2 DEVICE IDENTIFICATION

A device identification (ID) is used by software to identify the specific type controller and device which is attached to a channel. The ID word consists of two 8-bit bytes. The most significant byte is furnished by the controller, and for the MDC it is a constant 20_{16}. Table A-1 lists the general classification of IDs. Table 1-1 lists the specific ID for device options offered by Honeywell. When connecting other device options to the adapter, the device ID should be selected from Table A-1 for the general type device.

The required ID code (Table A-2) is made by connecting jumpers on the cable paddle (Figure A-1). A jumper from pin 17 to pin 18 forces bit 15 of the ID code to Zero, a jumper between pin 19 and pin 20 forces bit 14 to Zero, and a jumper from pin 13 to ground sets bit 13 to One. A jumper from pin 13 to ground also enables the data stream control signal which is required when a HISI Typewriter Console is installed. Also note that an additional line from pin 10 to the device is required. See paragraph 2.3.2 for a description of the use of these additional signals.

A.3 STOP BIT SETTING

As shown in Figure A-1, a jumper between pin 21 and pin 22 on the cable paddle causes only one stop bit to be set out after each character in an output message. Removing the jumper causes two stop bits to be sent out after each character in a message. The setting of the number of stop bits does not affect an incoming message because the adapter only tests for the first stop bit after each character in an incoming message.

A.4 DEVICE TIMEOUT

As shown in Figure A-1, the TTY timeout feature of the adapter can be disabled by placing a jumper between pin 9 and pin 1 on the cable paddle. The theory of operation of the Timeout logic is covered in paragraph 2.3.7 of this manual.

A.5 BAUD RATE

The baud rate at which the adapter sends and receives is determined by the setting of a hexadecimal rotary switch called out in Figure A-2. Table A-3 indicates the baud rate selected by each position of the switch.

A.6 CABLE CONNECTOR IDENTIFICATION

The Honeywell part number for the cable connector which mates the cable to the device with the adapter is 60128160-002. This cable connector is a printed circuit board with a male connector, connector guides, and a retainer handle.

A.7 MDC/DEVICE CABLE OPTIONS

Figure A-3 shows the various cable options which are available for connecting a device to the BD2CSL board in the MDC.

Table A-1 General ID

GENERAL TYPE DEVICE	ID CODE(HEX)
TTY ASR 33	2018
TTY KSR 33	2019
CRT	201A
HISI Type-writer Console (Item 7, 9 and 10 of Table 1-1)	201C

Table A-2 ID Word Formation

DEVICE	ID CODE (Hex)	BYTE FORMED BY CONTROLLER								BYTE FORMED BY ADAPTER							
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TTY ASR33	2018	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0
TTY KSR3	2019	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	1
CRT	201A	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1	0
HISI TYPE-WRTR CONSOLE	201C	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0

Table A-3 Baud Rate Selection

HEXADECIMAL ROTARY SWITCH POSITION	BAUD RATE SELECTED
0	Not Used
1	50
2	75
3	110
4	134.5
5	150
6	300
7	600
8	900
9	1200
A	1800
B	2400
C	3600
D	4800
E	7200
F	9600

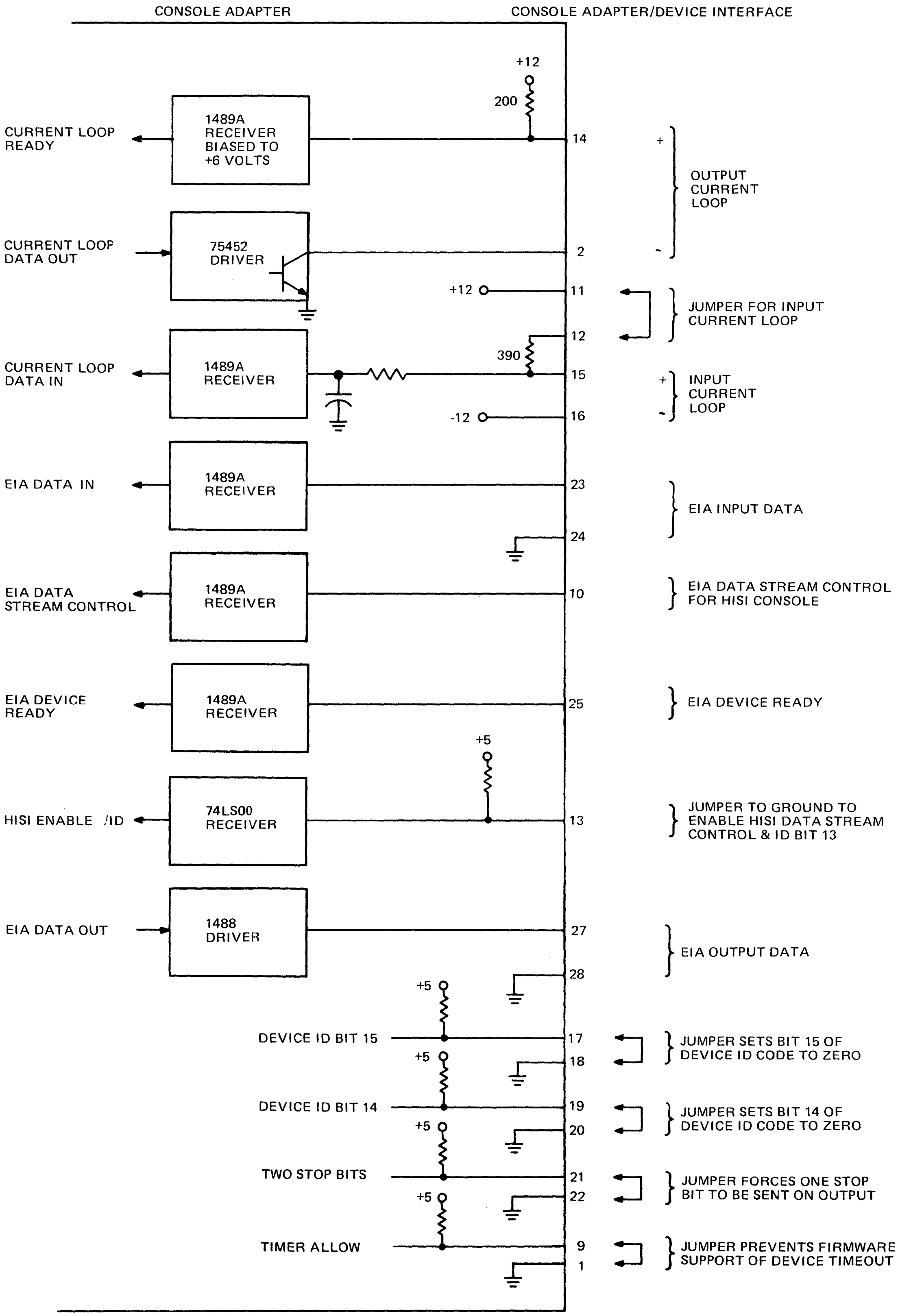


Figure A-1 Console Adapter/Device Interface Information

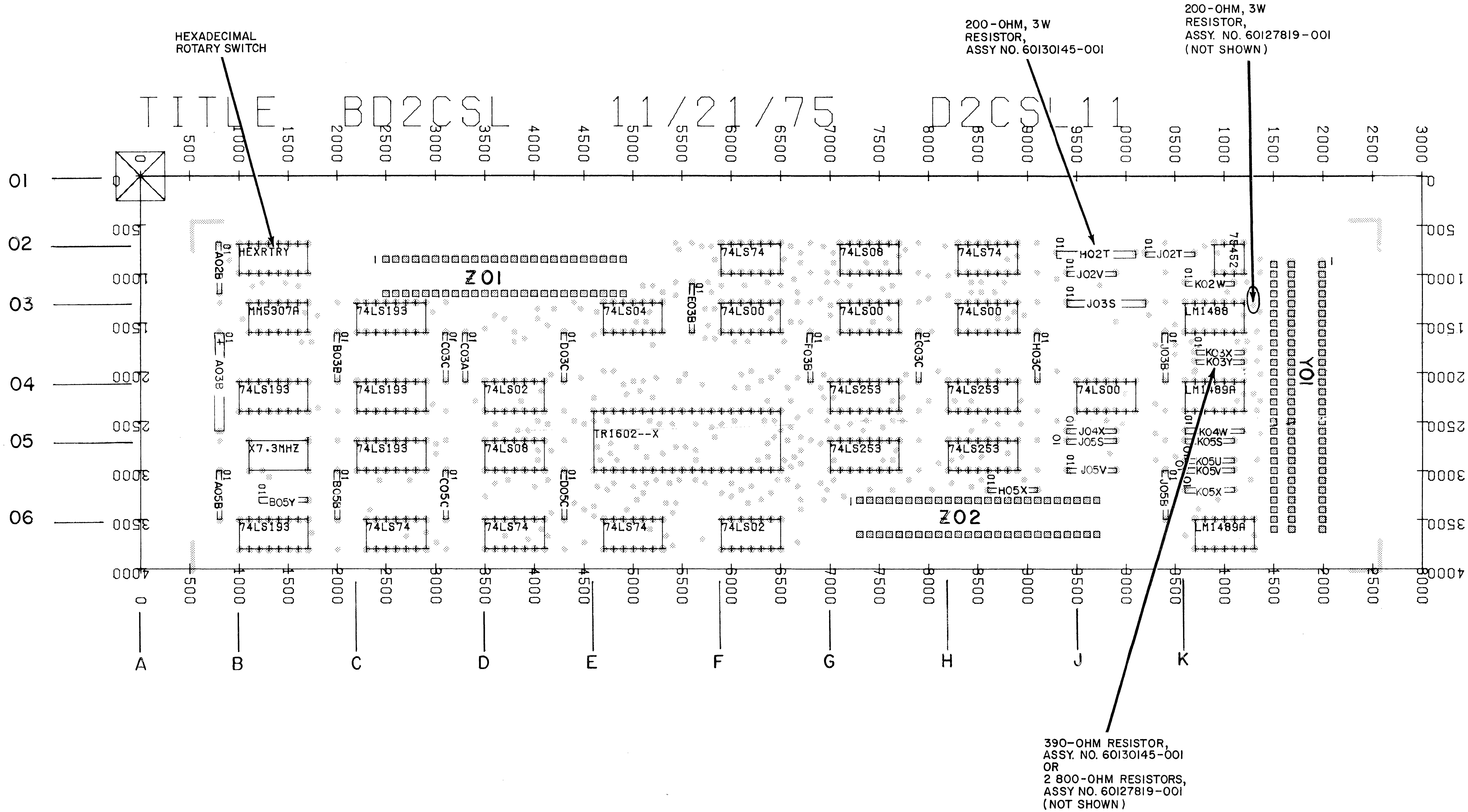


Figure A-2 Console Adapter Board Locations

HONEYWELL PROPRIETARY AND CONFIDENTIAL

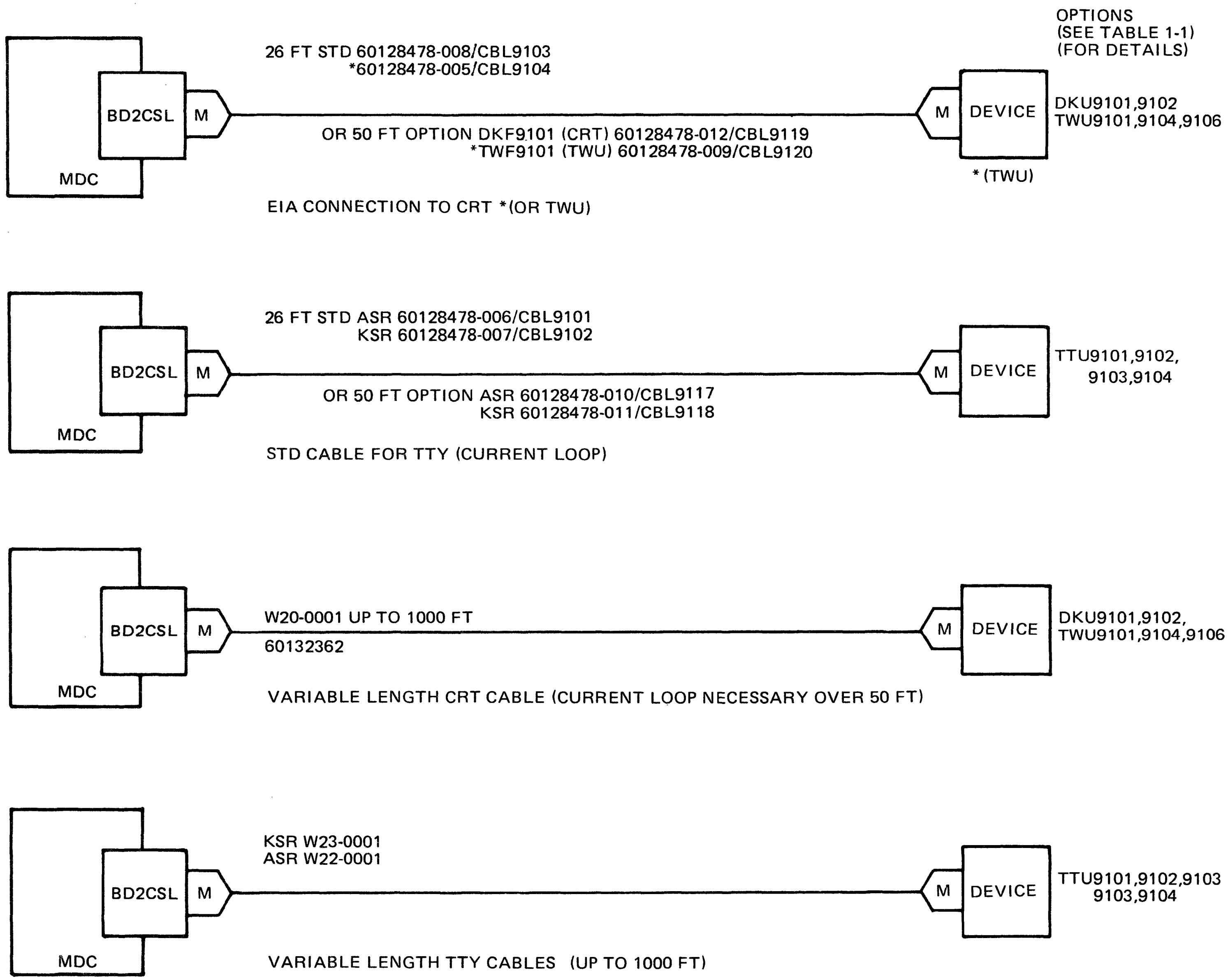


Figure A-3 MDC/Device Cable Options

HONEYWELL PROPRIETARY AND CONFIDENTIAL

PLEASE FOLD AND TAPE –
NOTE: U. S. Postal Service will not deliver stapled forms

M&TO HARDWARE PUBLICATIONS
USER COMMENTS FORM

DOCUMENT TITLE: ______________________________

PART NO.: ______________

ORDER NO.: ______________

ERRORS:

HOW DO YOU USE THIS DOCUMENT?

THEORY ______ ☐

MAINTENANCE ____ ☐

TROUBLESHOOTING ☐

OTHER: ______________________________

____

DOES THIS MANUAL SATISFY YOUR REQUIREMENTS?

YES ☐ NO ☐

IF NOT, PLEASE EXPLAIN ______________________________

____

____

FROM: NAME ______________________ DATE __________

TITLE ______________________________

COMPANY ______________________________

ADDRESS ______________________________

FIRST CLASS
Permit No. 39531
Waltham, Ma.

Business Reply Mail

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

HONEYWELL INFORMATION SYSTEMS INC.
200 SMITH STREET
WALTHAM, MA. 02154

MAIL STATION 872A
HARDWARE PUBLICATIONS, BILLERICA

Honeywell

Honeywell

Honeywell Information Systems
In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario M2J 1W5
In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.

FL17, Rev. 2