M68SVS(D)

BASIC DISPLAY UNIT
USER'S GUIDE

The information in this manual has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, such information does not convey to the purchaser of the devices described any license under the patent rights of Motorola Inc. or others.

Motorola reserves the right to change specifications without notice.

First Edition

TABLE OF CONTENTS

CHAPTER PAGE

1. GENERAL INFORMATION 1-1

1.1 INTRODUCTION 1-1
1.2 FEATURES 1-1
1.2.1 Operational Features 1-1
1.2.2 Hardware Features 1-3
1.2.3 Software Features 1-3
1.3 SPECIFICATIONS 1-4
1.4 GENERAL INFORMATION 1-4
1.4.1 Video Monitor 1-4
1.4.2 CRT Controller Board 1-6
1.4.2.1 Microprocessor Control Circuit 1-6
1.4.2.2 Video Generator Circuit 1-9
1.4.2.3 Display Memory Circuit 1-9
1.4.2.4 Program Memory Circuit 1-9
1.4.3 CRT Configuration Board 1-9
1.4.4 Chassis/Housing and Power Supply 1-9
1.4.5 Resident Executive Firmware 1-10
1.5 EQUIPMENT SUPPLIED 1-12
1.6 OPTIONAL EQUIPMENT 1-12
1.6.1 Extended Motherboard Option 1-12
1.6.2 Keyboard Option 1-12
1.6.3 Extended Communications Feature Option 1-14
1.6.4 Extended Display Feature Option 1-14
1.6.5 Interconnecting Cable Assembly and Connector Options 1-14

2. INSTALLATION INSTRUCTIONS AND INTERCONNECTION CONSIDERATIONS 2-1

2.1 INTRODUCTION 2-1
2.2 UNPACKING INSTRUCTIONS 2-1
2.3 INSPECTION 2-1
2.4 INSTALLATION INSTRUCTIONS 2-1
2.4.1 External Device Interconnections 2-1
2.4.2 Extended Motherboard Installation 2-3
2.5 BASIC DISPLAY UNIT INTERCONNECTIONS 2-3

3. OPERATING INSTRUCTIONS 3-1

3.1 INTRODUCTION 3-1
3.2 OPERATING CONTROLS 3-1
3.2.2 Front Panel Controls 3-1
3.2.3 Rear Panel Controls 3-1
3.2.4 Optional Keyboard Assembly Controls 3-3
3.3 GENERAL OPERATING INFORMATION 3-3
3.3.1 Initialization 3-5
3.3.2 Operator Test Procedure 3-6
3.3.2.1 Power On Check 3-6
3.3.2.2 Off Line Checks 3-6

CHAPTER PAGE

3. OPERATING INSTRUCTIONS (cont'd)

3.4 COMMAND DESCRIPTION 3-7
3.4.1 Communications Interface Commands 3-7
3.4.2 Keyboard Interface Commands 3-8
3.4.3 Extended Display Feature Option Commands 3-9
3.5 PROGRAMMING INFORMATION 3-9
3.5.1 Programming Considerations 3-9/3-28
3.5.1.1 Monitor Control Software Process Module Functions 3-29
3.5.1.2 Keyboard Control Software Process Module Functions 3-35
3.5.1.3 Terminal Control Software Process Module Functions 3-37
3.5.1.4 Communications Control Software Process Module Functions 3-37
3.5.1.5 Display Control Software Process Module Functions 3-38
3.5.2 Organization of Application Software 3-38
3.5.2.1 Initialization Process 3-40
3.5.2.2 Subroutine Interface 3-40
3.5.2.3 Keyboard Interpreter Interface 3-40
3.5.2.4 Display Control Software Process Communications Input Interface 3-40
3.5.2.5 Display Control Software Process Communications Output Interface 3-44
3.5.2.6 Spare Switches 3-44

4. DIAGRAMS 4-1

4.1 SCHEMATIC AND INTERCONNECTION DIAGRAMS 4-1

APPENDIX A: GLOSSARY OF TERMS A-1 thru A-5
APPENDIX B: ASCII CHARACTER SET B-1 thru B-5

LIST OF ILLUSTRATIONS

Figure		Page
1-1	M68SVS10100 or M68SVS10200 Basic Display Unit	1-2
1-2	Basic Display Unit Block Diagram	1-7
1-3	Basic Display Unit Memory Map	1-8
2-1	Current Loop to Basic Display Unit Interconnection	2-2
2-2	RS-232C to Basic Display Unit Interconnection	2-2
3-1	Basic Display Unit, Front and Rear Panel Controls	3-4
3-2	Basic Display Unit System Flow Chart	3-10
3-3	Software Interrupt Handler Flow Chart	3-30
3-4	Internal Interrupt Handler Flow Chart	3-34
3-5	Keyboard Routine Flow Chart	3-36
3-6	Example of Code to Define an Application Subroutine and its Associated Initialization Process	3-41
3-7	Application Subroutine Interface with Resident Input/Output Modules	3-42
3-8	Example of User Application Subroutine Logic	3-43
4-1	CRT Terminal Interconnection Diagram	4-3
4-2	CRT Controller Board Schematic Diagram (Sheet 1 of 3)	4-5
4-2	CRT Controller Board Schematic Diagram (Sheet 2 of 3)	4-7
4-2	CRT Controller Board Schematic Diagram (Sheet 3 of 3)	4-9
4-3	Configuration Board Schematic Diagram	4-11
4-4	Power Supply Assembly Schematic Diagram	4-13
4-5	Video Monitor Assembly Schematic Diagram	4-15

LIST OF TABLES

Table		Page
1-1	Basic Display Unit Specifications	1-5
1-2	Display Codes and Characters	1-11
1-3	Optional Equipment	1-13
2-1	Extended Motherboard Bus Interface Signals (J1)	2-3 thru 2-7
2-2	Keyboard Interface Signals (J2)	2-7/2-8
2-3	Current Loop Interface Signals	2-8/2-9
2-4	RS-232C Interface Signals	2-9/2-10
3-1	Rear Panel Controls	3-1/3-2/3-3
3-2	Basic Display Unit Communications Interface Commands	3-11 thru 3-13
3-3	Cursor Position Code	3-14
3-4	Extended Display Feature Communication Interface Commands	3-15 thru 3-27

CHAPTER 1

GENERAL INFORMATION

1.1 INTRODUCTION

The Basic Display Unit is an extremely flexible CRT-based terminal capable of being configured by the user to perform a variety of functions. These functions may be implemented either independently or in groups by including various hardware and software options with the M68SVS10100 or M68SVS10200 Basic Display Unit. This manual describes only the Basic Display Unit and its operation. Supplements to this manual describe each of the functions provided by individual options. The Basic Display Unit is shown in Figure 1-1. A glossary of terms is provided in Appendix A, defining the various terms and acronyms used throughout this manual that may be unfamiliar to the user.

1.2 FEATURES

The features of the Basic Display Unit are divided into three categories: Operational Features, Hardware Features, and Software Features.

1.2.1 Operational Features

The following list of operational features are inherent in the design of the Basic Display Unit, even though optional hardware and/or software may be required for full implementation (as indicated in the list).

- Modular design concept based on the MC6800 Microprocessor Unit (MPU) that readily permits additional functions to be incorporated via optional hardware and/or software
- Multiple display mode capability
    - Scroll mode (Standard)
    - Page/Edit/Protect mode (Optional)
- Multiple data entry capability
    - Remote entry to either internal or external device
        - Serial asynchronous communications interface (Standard)
        - Control commands for 103 and 202 modems (Optional)
    - Keyboard entry (Standard) (keyboard optional)
- All commands are assigned a unique 8-bit value capable of being represented by an escape sequence to facilitate 7-bit transmission
- Normally transparent terminal commands can be displayed without execution to aid in program debugging and fault isolation

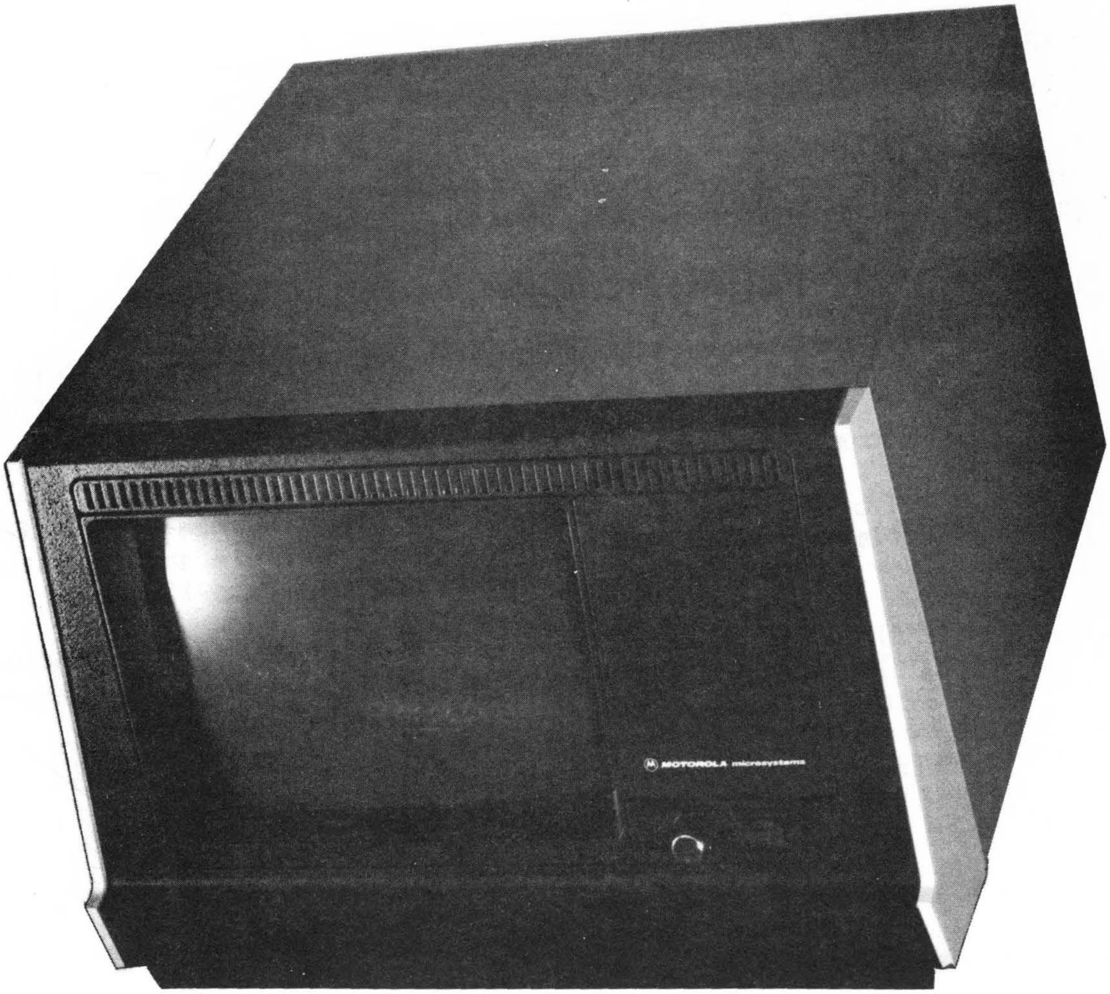


FIGURE 1-1. M68SVS10100 or M68SVS10200 Basic Display Unit

1.2.2 Hardware Features

The following hardware features are standard with the Basic Display Unit:

- Video monitor with 12-inch diagonal display screen capable of displaying a maximum of 1920 characters, organized into 24 lines with 80 characters per line
- Changeable character generator capable of producing a set of 128 characters within a 7 X 9 dot matrix: 96 upper and lower case ASCII characters, 24 lower case Greek characters, 2 upper case Greek characters (Omega and Sigma), and 6 special characters
- Shared memory between the video control and microprocessor system
- Choice of shared or non-shared bus extended motherboard
- Display controls for brightness and full screen inversion
- Operator switches that permit Field Attribute Codes (FAC's) and normally transparent terminal commands to be displayed
- Switch selectable asynchronous serial data interfaces compatible with both EIA RS-232C and 20/60 mA current loops (with source current)
- Switch selectable transmission word length of 7 or 8 bits
- Switch selectable baud rates of 110, 150 300, 600, 1200, 1800, 2400, 4800, and 9600
- Switch selectable communications modes: half/full duplex, even/odd/no parity, and 1 or 2 stop bits
- Switch selection of optional modem controls (103 or 202)
- Spare switches available for interpretation by user software
- Blinking inverted video cursor
- Audible alarm
- Provides expansion for 1 to 8 EXORciser or Micromodule compatible boards

1.2.3 Software Features

The following software features are standard with the Basic Display Unit.

- Embedded Field Attribute Codes (FACs) for controlling blink, intensity (half-bright), video inversion, underlining, and field protection (these operations are optionally available via firmware)
- Incremental and absolute cursor positioning

- Display Control software process module used to position the cursor, execute generic display attribute commands, control scrolling of the display area, and execute screen splitting commands

- Keyboard Control software process module used to accept encoded characters from the keyboard and queue the key value for further processing

- Communication Control software process module used to accept data for transmission and to queue input data from the communications link for the Display Control software process module

- Terminal Control software process module used to read the control switches and to queue commands for the Display Control and Communications Control software process modules

- Monitor Control software process module used to execute Monitor Operator Calls (MOCs) to facilitate queue manipulations and to control execution of all other software process modules on a priority basis

1.3 SPECIFICATIONS

The specifications for the Basic Display Unit are identified in Table 1-1.

1.4 GENERAL INFORMATION

The Basic Display Unit incorporates an M6800 microprocessor system to control the display of alphanumeric data generated by a keyboard or host computer. The Basic Display Unit is an integral part of the EPIC 68 Display Unit and is capable of being programmed by the user to perform a variety of terminal tasks. Due to its flexible hardware and software design, the Basic Display Unit can also be used as the basis for many other applications (small business computers, etc.). The Basic Display Unit consists of a Motorola M3000 Video Monitor for display, a CRT Controller Board containing all of the necessary control electronics and firmware operating routines, a CRT Configuration Board providing the means to manually select basic operating criteria, a Chassis/Housing with Power Supply, and the necessary Resident Executive Firmware to control the display and communications link and any user-programmed functions. The following paragraphs provide a general description of each major assembly within the Basic Display Unit. Figure 1-2 provides a block diagram of the Basic Display Unit.

1.4.1 Video Monitor

The Motorola model M3000 Video Monitor is a fully transistorized (except CRT), self-contained display unit with a 12-inch diagonal CRT. This monitor has a video response range of 10 Hz to 22 MHz (within -3dB), thus enabling it to display up to 80 characters per line within a 9 X 11 dot matrix (character produced using only a 7 X 9 dot matrix). Data displayed by the Video Monitor is controlled by the CRT Controller Board and Display Control software process module through separate vertical/horizontal sync and video inputs to the monitor.

The CRT used is a magnetic deflection type with integral implosion protection. The display screen (field) is coated with P4 phosphor (white) and is covered with a plastic anti-glare shield. An operating voltage of +70 Vdc is supplied by the internal regulated power supply which incorporates a

TABLE 1-1. Basic Display Unit Specifications

CHARACTERISTICS	SPECIFICATIONS
Power Requirements	110 Vac, 60 Hz @ 4 A (M68SVS10100) 220 Vac, 50 Hz @ 2 A (M68SVS10200)
Word Size	
Data	8 Bits
Address	16 Bits
Instructions	8, 16, or 24 Bits
Instructions	72 variable length instructions
Addressing Modes	Seven addressing Modes: Direct, Relative, Immediate, Indexed, Extended, Implied, and Accumulator.
Memory Requirements	
Used By System	
ROM	2K Bytes (Executive) 4K Bytes (Display/Communications)
RAM	2K Bytes (Display) 1K Byte (Scratchpad)
Available To User	Up to 32K depending upon software options
Current Loop Interface Signals	20 to 60 milliamperes
RS-232C Interface Signals	
Input Signals	EIA RS-232C compatible serial data/control input lines (±30 Vdc signal range)
Output Signals	EIA RS-232C compatible serial data/control output lines (±10 mA current limited)
Physical Characteristics	
Width X Height X Depth	18.54 in.(W) X 13.13 in. (H) X 20.29 in. (D)
Weight	61 lbs.
Input/Output Connectors	
Current Loop (6 pin)	AMP 1-350241-9 with pins 350037-1 or equivalent
RS-232C (25 pin)	CINCH DB-25P or equivalent

universal power transformer; thus, permitting the supply to operate from either 115 or 230 Vac, 50/60 Hz.

In addition to the power supply, two other printed circuit boards are used in the Video Monitor: a vertical/video control board and a horizontal control board. Circuitry on these boards consist of two stages for video amplification, five stages for vertical sync and deflection processing, and five stages for horizontal sync and deflection processing. Dynamic focusing is also incorporated in the monitor. All standard operating and adjustment controls are located on these circuit boards, except for brightness, which is located on the front panel of the Basic Display Unit.

1.4.2 CRT Controller Board

The CRT Controller Board is a single, 13½ X 16 inch printed circuit board located in the base of the Basic Display Unit Chassis/Housing assembly. It incorporates the circuitry needed to control all of the functions of the Basic Display Unit. The circuits contained on this board are separated into four functional groups. The function of each group is briefly described in the following paragraphs with a memory map provided in Figure 1-3.

1.4.2.1 Microprocessor Control Circuit. The Microprocessor Control circuit contains the MC6800 Microprocessing Unit (MPU), three MC6820 Peripheral Interface Adapters (PIAs), one MC6850 Asynchronous Communications Interface Adapter (ACIA), 1K byte of static RAM memory, and the necessary clock and control circuits needed to operate the MPU. The MPU executes internally stored firmware programs to control all of the Basic Display Unit functions. These programs are more fully described in paragraph 1.4.5.

The PIA provides parallel input/output interface to the MPU data bus. In this system, one PIA is used to input data from an optional external keyboard; another PIA is used to input switch position status from the CRT Configuration Board; the third PIA is used to control cursor positioning on the display screen. All three PIAs are controlled by the resident software.

The ACIA provides asynchronous serial input/output interface to the MPU data bus. Only one ACIA is used in this system, providing it with the means of communicating with a host computer or other serial device. The ACIA and associated interface circuitry is compatible with RS-232C or 20/60 mA current loop devices. In addition, by selecting the Extended Communications Feature option, this serial data port will operate with either 103 or 202 modem devices or their equivalents.

The microprocessor bus (consisting of the address, data, and control busses) may be expanded by connecting an optional 8-card motherboard into connector J1, located on the reverse side of the CRT Controller Board. Two versions of the motherboard are available. One version (M68SVS50100) extends the complete microprocessor bus, permitting additional memory or other functional modules (either EXORciser or Micromodule) to be readily added to the Basic Display Unit. The other version (M68SVS50200) extends only the power bus, permitting another complete microcomputer system to be installed within the Basic Display Unit. Only one of the motherboards can be used at any one time. However, due to the ease with which the motherboard can be replaced, motherboards could be interchanged to permit reconfiguration of the Basic Display Unit for test and/or development purposes. If neither motherboard is used, one additional module can be added to the Basic Display Unit by inserting it into connector J1.

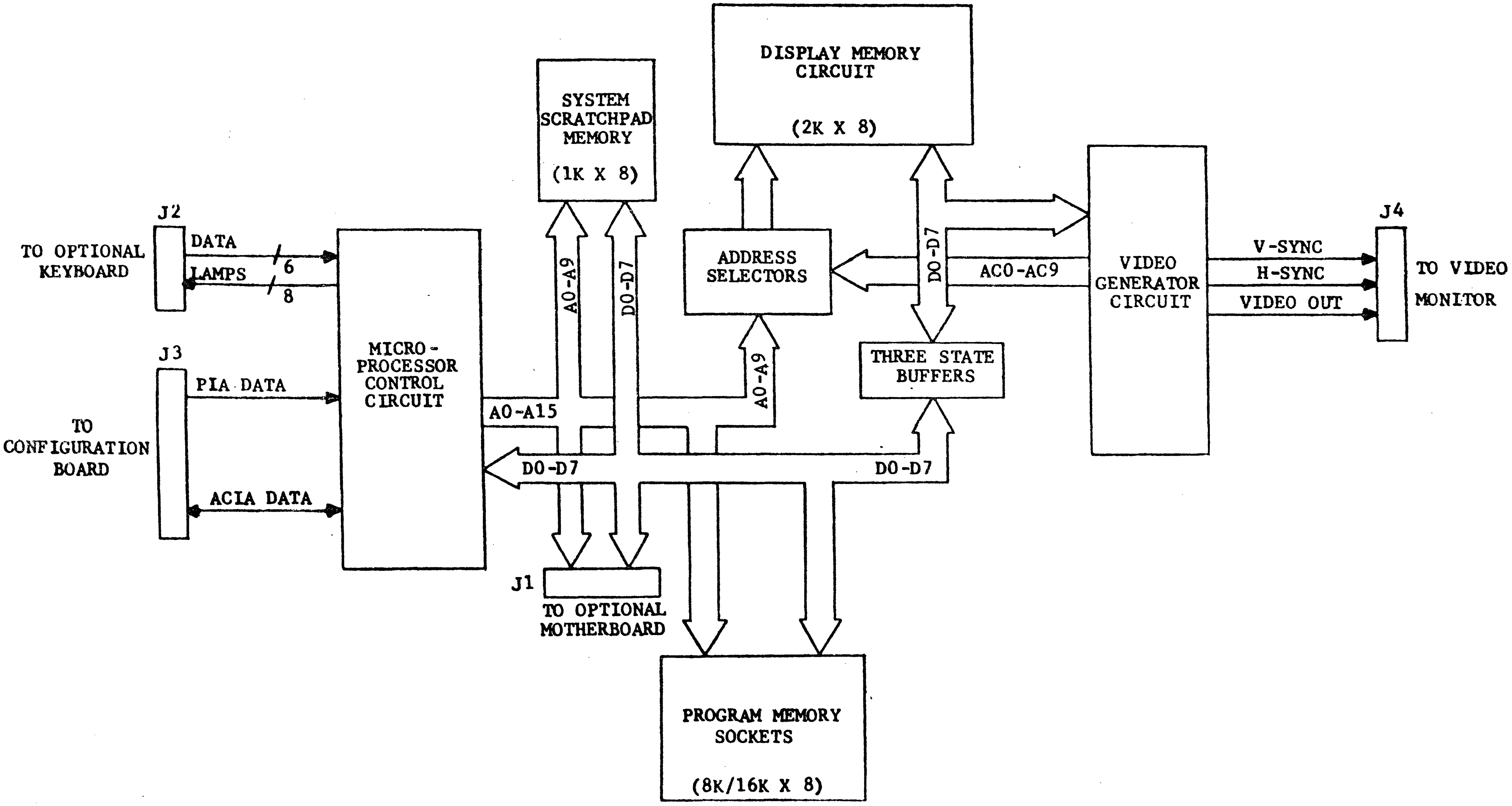
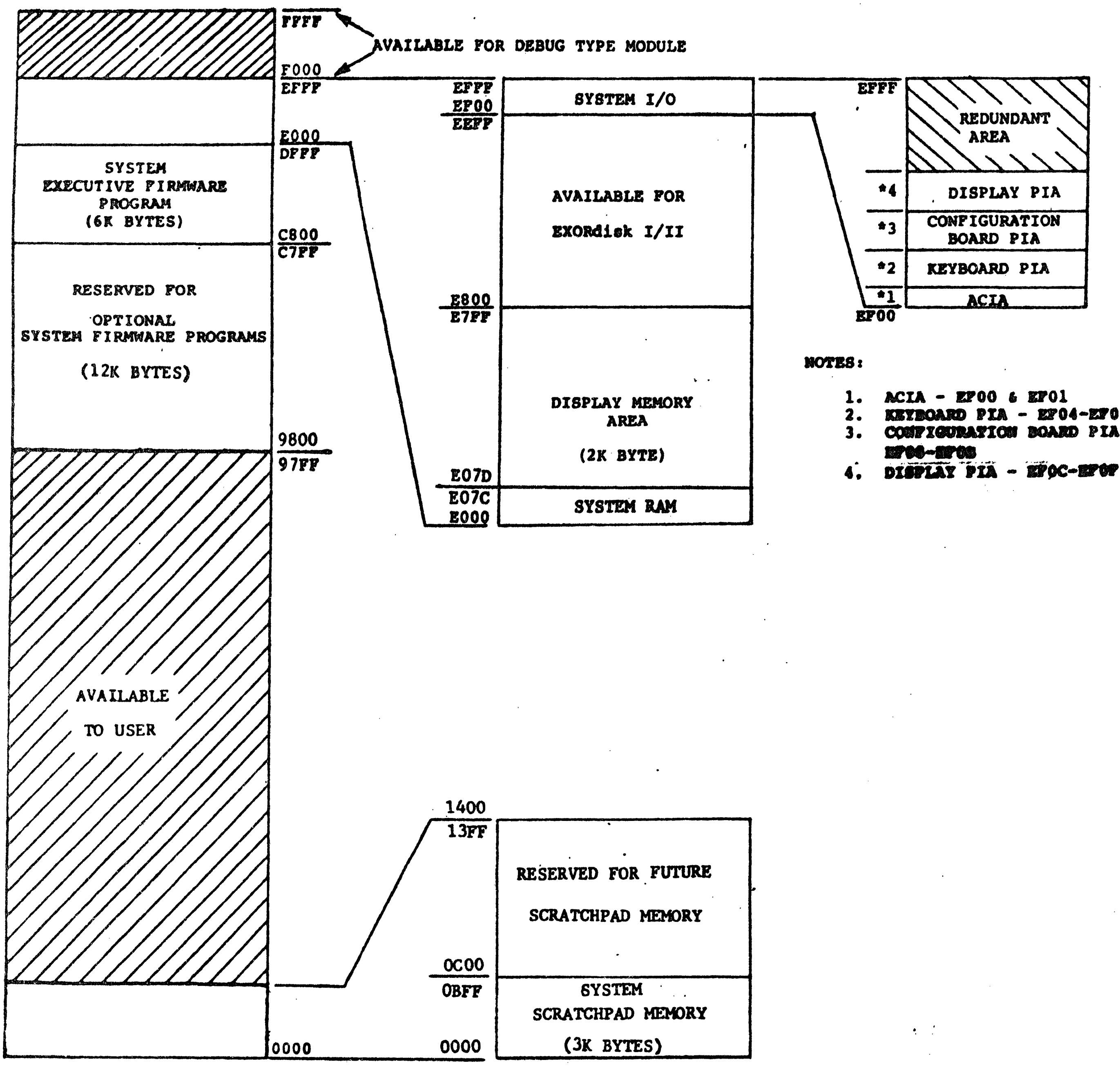


FIGURE 1-2. Basic Display Unit Block Diagram


FFFF
AVAILABLE FOR DEBUG TYPE MODULE
F000
EFFF
E000
DFFF
SYSTEM EXECUTIVE FIRMWARE PROGRAM (6K BYTES)
C800
C7FF
RESERVED FOR OPTIONAL SYSTEM FIRMWARE PROGRAMS (12K BYTES)
9800
97FF
AVAILABLE TO USER
0000
EFFF
EF00
EEFF
SYSTEM I/O
AVAILABLE FOR EXORdisk I/II
E800
E7FF
DISPLAY MEMORY AREA (2K BYTE)
E07D
E07C
E000
SYSTEM RAM
EFFF
REDUNDANT AREA
*4
DISPLAY PIA
*3
CONFIGURATION BOARD PIA
*2
KEYBOARD PIA
*1
ACIA
EF00
1400
13FF
RESERVED FOR FUTURE SCRATCHPAD MEMORY
0C00
0BFF
SYSTEM SCRATCHPAD MEMORY (3K BYTES)
0000

NOTES:

1. ACIA - EF00 & EF01
2. KEYBOARD PIA - EF04-EF07
3. CONFIGURATION BOARD PIA - EF08-EF0B
4. DISPLAY PIA - EF0C-EF0F

FIGURE 1-3.. Basic Display Unit Memory Map

1.4.2.2 Video Generator Circuit. The Video Generator Circuit produces the characters displayed on the CRT screen and controls the positioning of each character. Display characters are stored in the form of 8-bit (1 byte) words within the Display Memory Circuit. The memory circuit is sequentially accessed by the Video Generator Circuit to retrieve each display word. A character generator then converts each word into the proper dot pattern (within a 9 X 11 dot matrix) to produce the desired character on the screen.

In addition to generating the displayed character, the Video Generator Circuit also produces the vertical and horizontal sync signals supplied to the Video Monitor for positioning the characters on the screen. The sync signals and the sequentially accessed memory circuits are synchronized together to display data at the proper location on the screen.

1.4.2.3 Display Memory Circuit. The Display Memory Circuit consists of sixteen, 1K X 1 static memory devices, organized into 2048 8-bit bytes. This circuit is used to store data displayed on the CRT screen and is shared between the Microprocessor Control Circuit and the Video Generator Circuit. This shared-memory technique permits the MPU to directly store data for display, while permitting the Video Generator Circuit to control display refresh. This unburdens the MPU, permitting it to process more functions. To implement shared memory, address selectors and three-state buffers are also included within this circuit.

1.4.2.4 Program Memory Circuit. The Program Memory circuit contains the firmware programs used to control the operation of the Basic Display Unit. Up to 16K bytes of firmware may be used: eight 1K byte AROM/ROM/PROM devices or eight 2K byte mask programmable ROM devices. As delivered, the complete display/executive program is contained within three 2K ROMs. These programs control all of the display and standard communications functions (refer to paragraph 1.4.5 for more detail). The remaining five program sockets are reserved for other optional programs.

1.4.3 CRT Configuration Board

The CRT Configuration Board is located on the upper portion of the Basic Display Unit back panel and contains the switches used for manual operation selections. The connectors used to interconnect the Basic Display Unit with a host computer or other device are also located on this board. In addition, this board contains the 20/60 mA current loop interface circuitry, the RS-232C interface circuitry, the baud rate generator and selection circuitry, and the audio alarm.

1.4.4 Chassis/Housing and Power Supply

The Chassis/Housing consists of a plastic base, bezel, and back panel with a metal shroud and bottom plate. The base mounts the CRT Controller Board, Video Monitor, Power Supply, and the bezel and back panel. The metal bottom plate is attached to the bottom of the base with screws. The metal shroud slips over the Basic Display Unit and attaches on the sides near the lower edge of the base.

The Power Supply is mounted at the rear of the Basic Display Unit, with a cooling fan mounted on the back panel. The Power Supply provides +5 Vdc, +12 Vdc, and -12 Vdc power for the CRT Controller Board and any additional modules connected to the MPU bus via the optional extended motherboard. The +5 Vdc is used for the logic circuits and the +12 Vdc and -12 Vdc are used for the interface circuits. The +5 Vdc power supply voltage is also provided to the optional keyboard assembly.

An integral card cage, located in the right-rear corner of the Basic Display Unit, permits up to 8 modules to be added to the display unit if one of the optional extended motherboards has been installed. Access to the card cage is obtained by removing a small metal door mounted on the back panel.

1.4.5 Resident Executive Firmware

The Resident Executive Firmware controls the operation of the Basic Display Unit. This executive program has been designed with a high degree of modularity, permitting it to be expanded to 16 different process modules; all of which share the MPU on a priority basis. Five process modules have been implemented in the Resident Executive Firmware, allowing 11 additional process modules to be incorporated by the user; either by including various optional firmware modules or by preparing specific process modules for the intended application.

The Monitor Control software process module controls execution priority of all other software process modules. This process module is also used to execute Monitor Operator Calls (MOCs): MOCs permit a software process module or an interrupt handler to request that a function be performed on its behalf by the monitor.

The Terminal Control software process module queues commands to the Display Control and Communications Control software process modules. This process module is also used to read the positions of the control switches located on the CRT Configuration Board.

The Display Control software process module accepts data and commands from external devices via the interface and interprets or executes these inputs to affect the data displayed on the screen. This process module is used to position the cursor, execute generic display attribute commands (such as video invert, blink, etc.), control scrolling of the display area, and execute screen splitting commands.

The Communications Control software process module controls the flow of serial data between the Basic Display Unit and the host computer or other device. This software process module works in conjunction with the Display Control software process module.

The Keyboard Control software process module accepts encoded characters (normally generated by the optional keyboard) and passes the decoded results to the Display Control software process module for display.

The actual data displayed within the 80 character positions of all 24 lines is controlled by a 7-bit word stored at the corresponding address within the 1920 byte positions reserved for display storage within the Display Memory Circuit on the CRT Controller Board. These words contain either a 7-bit ASCII character or a 7-bit Field Attribute Code (FAC). All 7-bit characters are displayed within a 7 X 9 dot matrix on the display screen. The 128 characters capable of being displayed are shown in Table 1-2. The FAC code is normally displayed as a space.

TABLE 1-2. Display Codes and Characters

ASCII CODE (HEX)	CHARACTER	ASCII CODE (HEX)	CHARACTER	ASCII CODE (HEX)	CHARACTER	ASCII CODE (HEX)	CHARACTER
00	α	20	(blank)	40	@	60	`
01	β	21	!	41	A	61	a
02	γ	22	"	42	B	62	b
03	δ	23	#	43	C	63	c
04	ε	24	$	44	D	64	d
05	ζ	25	%	45	E	65	e
06	η	26	&	46	F	66	f
07	θ	27	'	47	G	67	g
08	ι	28	(	48	H	68	h
09	κ	29	)	49	I	69	i
0A	λ	2A	*	4A	J	6A	j
0B	μ	2B	+	4B	K	6B	k
0C	ν	2C	,	4C	L	6C	l
0D	ξ	2D	-	4D	M	6D	m
0E	ο	2E	.	4E	N	6E	n
0F	π	2F	/	4F	O	6F	o
10	ρ	30	Ø	50	P	70	p
11	σ	31	1	51	Q	71	q
12	τ	32	2	52	R	72	r
13	υ	33	3	53	S	73	s
14	φ	34	4	54	T	74	t
15	χ	35	5	55	U	75	u
16	ψ	36	6	56	V	76	v
17	ω	37	7	57	W	77	w
18	Ω	38	8	58	X	78	x
19	√	39	9	59	Y	79	y
1A	¢	3A	:	5A	Z	7A	z
1B	|	3B	;	5B	[	7B	{
1C	¬	3C	<	5C	\	7C	¦
1D	÷	3D	=	5D	]	7D	}
1E	Σ	3E	>	5E	^	7E	~
1F	≈	3F	?	5F	_	7F	▨

FAC codes are used to control display attributes (or characteristics) by either selecting or deselecting blink, half-bright intensity, video inversion, underline, non-display, and field protect. Commands used to select or deselect these attributes are available only when optional features have been added to the Basic Display Unit. When incorporated, these full screen attributes can also be manually controlled by the DISPL FAC CODE and VIDEO INV switches located on the CRT Configuration Board. The position of the cursor is also addressable, permitting it to be moved to any of the 1920 character positions on the screen.

When used in conjunction with an optional keyboard assembly, The Basic Display Unit can be operated in the Local mode for testing both the keyboard and display operation using the Keyboard Control and Display Control software process modules. When operated in this mode, the Communications Control software process module is disabled, preventing data exchange between the Basic Display Unit and the host computer.

1.5 EQUIPMENT SUPPLIED

The Basic Display Unit is shipped with this manual included. All optional items ordered with the Basic Display Unit will be included within the shipping carton, along with the respective supplements to this manual.

1.6 OPTIONAL EQUIPMENT

Various options are available for the Basic Display Unit to enhance its capabilities. In addition, any of the EXORciser or Micromodule Family modules can be added directly to the Basic Display Unit system bus. If one of the optional extended motherboards are used, up to eight additional modules can be incorporated in the system. The options available specifically for the Basic Display Unit are listed in Table 1-3.

1.6.1 Extended Motherboard Option

Two different extended motherboards are offered as options. The M68SVS50100 Extended Motherboard (with shared bus) extends the complete system bus to 8 additional edge of card connectors, permitting up to eight additional EXORciser or Micromodule modules to be connected directly to the system bus. The M68SVS50200 Extended Motherboard (Not-Shared) extends only the power bus to 8 edge of card connectors, permitting a second complete microcomputer system to be installed within the Basic Display Unit. All of the connectors used on these motherboards are bus compatible with EXORciser and Micromodule Family modules.

NOTE

When the M68SVS50100 Extended Motherboard option is used, a second MPU-based module must not be installed into the extended card cage, since this would connect two MPU systems onto the same bus.

1.6.2 Keyboard Option

Several different keyboards are offered as optional equipment. Regardless of the keyboard selected, it is attached to the Basic Display Unit via an interconnect cable assembly. For further information regarding the keyboard options, refer to the Keyboard Assembly Supplement.

TABLE 1-3. Optional Equipment

OPTION	PART NUMBER
Basic Display Unit	
1) 110 Vac, 60 Hz	M68SVS10100
2) 220 Vac, 50 Hz	M68SVS10200
Extended Motherboards	
1) Shared Bus	M68SVS50100
2) Non-Shared Bus	M68SVS50200
Keyboards	
1) Standard TTY Keys + Control Keys	M68SVS30100
2) Same as Keyboard 1 + 12 Function Keys	M68SVS30200
3) Same as Keyboard 1 + 16 Function Keys, Editing Keys, Cursor Control Keys, and Page Mode Control Keys	M68SVS30300
4) Same as Keyboard 3 + Auxiliary Control Keys	M68SVS303A0
5) Same as Keyboard 3 + Numeric Keypad	M68SVS30400
6) Same as Keyboard 5 + Auxiliary Control Keys	M68SVS304A0
7) Same as Keyboard 2 + Editing Keys, Cursor Control Keys, and Page Mode Control Keys	M68SVS30500
Extended Communications Feature (for Modem Control)	M68SVS20100
Extended Display Feature (for Page/Edit/Protect mode)	M68SVS20200
RS-232 To Modem Interconnect Cable Assembly (10 feet)	M68SVS40100
Keyboard Interconnect Cable Assemblies	
1) Flat Ribbon Cable with Connector (18 inch)	M68SVS40200
2) Round Cable with Connector (3 feet)	M68SVS40300
3) Round Cable with Connector (6 feet)	M68SVS40400
Connector Kit (includes: 2 - EIA RS-232C 25-pin connectors and 1 - 6-pin current loop connector)	M68SVS40500

1.6.3 Extended Communications Feature Option

The Extended Communications Feature option provides the necessary control for operating the Basic Display Unit with a Type 103 or 202 (or equivalent) modem device. For further information regarding this option, refer to the Extended Communications Feature Supplement.

1.6.4 Extended Display Feature Option

The Extended Display Feature option provides control of Page mode, Edit mode, and Protect mode operation. For further information regarding this option, refer to the Extended Display Feature Supplement.

1.6.5 Interconnecting Cable Assembly and Connector Options

Several Keyboard to Basic Display Unit and RS-232 to Modem interconnecting cable assemblies are available as options. These options provide a ready-made solution to the problem of interconnects. For users desiring to make their own Basic Display Unit to external device interconnect cables, the various connectors required are also provided as an option. For further information regarding the keyboard interconnecting cable assembly, refer to the Keyboard Assembly Supplement.

CHAPTER 2

INSTALLATION INSTRUCTIONS
AND
INTERCONNECTION CONSIDERATIONS

2.1 INTRODUCTION

This chapter provides the unpacking, inspection, installation, and interconnection instructions for the Basic Display Unit. This chapter also discusses the module's interconnection signals.

2.2 UNPACKING INSTRUCTIONS

Unpack the Basic Display Unit from its shipping carton and, referring to the packing list, verify that all of the items are present, including any of the options that may have been ordered. Save the packing materials for storing or reshipping of the display. If the shipping carton is damaged upon receipt, request that the carrier's agent be present while the display is being unpacked and inspected.

2.3 INSPECTION

The Basic Display Unit should be inspected upon receipt for broken, damaged, or missing parts or any other physical or electrical damage.

2.4 INSTALLATION INSTRUCTIONS

The following paragraphs discuss connecting the Basic Display Unit to a host computer or other device and installing the optional extended motherboard. Instructions for installing all other options are provided in the respective supplement to this manual.

2.4.1 External Device Interconnections

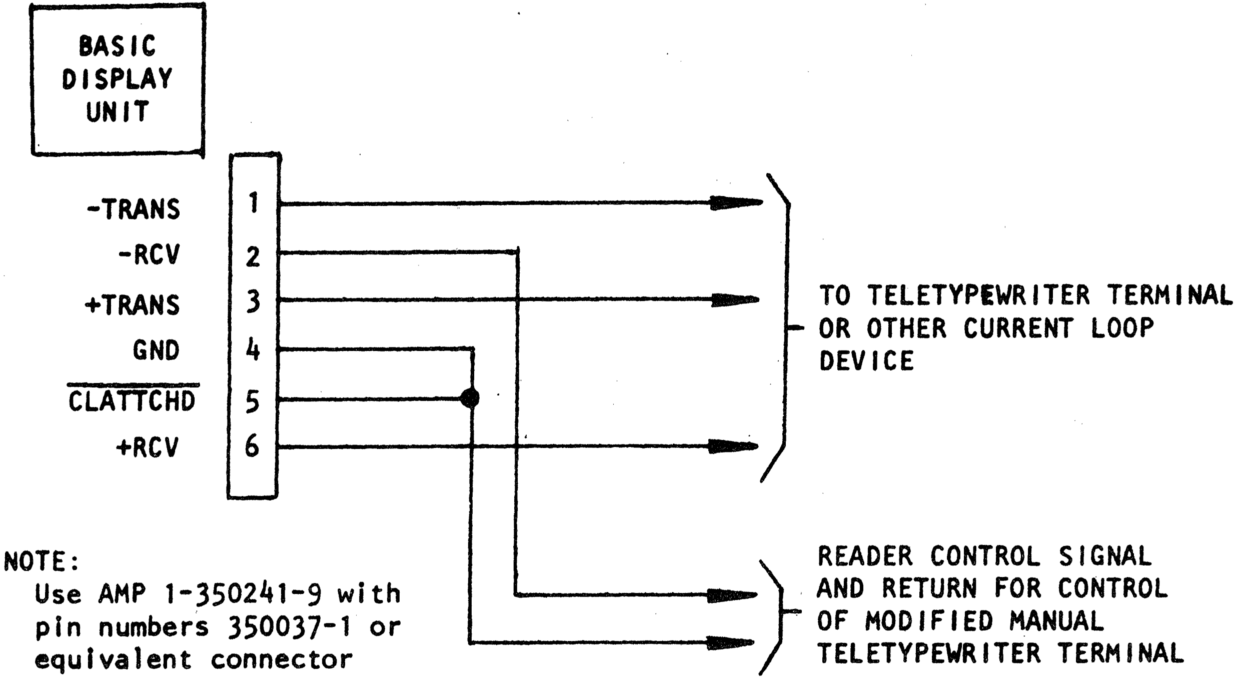
The user has the option of connecting the Basic Display Unit to any current loop or RS-232C compatible external device. Position the switches on the CRT Configuration Board in accordance with the instructions provided in paragraph 2.5.

The current loop interconnection is accomodated by a 6-pin connector located on the back panel of the Basic Display Unit. Refer to Figure 2-1 for current loop interconnection information.

CAUTION

> THE CURRENT LOOP IS DESIGNED TO OPERATE WITH 20 TO 60 mA CURRENT LOOPS. THE CURRENT LOOP MUST NEVER EXCEED 60 mA.

The RS-232C interconnection is provided by a 25-pin connector located on the back panel of the Basic Display Unit. Refer to Figure 2-2 for interconnection information.


BASIC DISPLAY UNIT
-TRANS
-RCV
+TRANS
GND
CLATTCHD
+RCV
1
2
3
4
5
6
TO TELETYPEWRITER TERMINAL OR OTHER CURRENT LOOP DEVICE
READER CONTROL SIGNAL AND RETURN FOR CONTROL OF MODIFIED MANUAL TELETYPEWRITER TERMINAL

NOTE:
Use AMP 1-350241-9 with pin numbers 350037-1 or equivalent connector

FIGURE 2-1. Current Loop To Basic Display Unit Interconnection

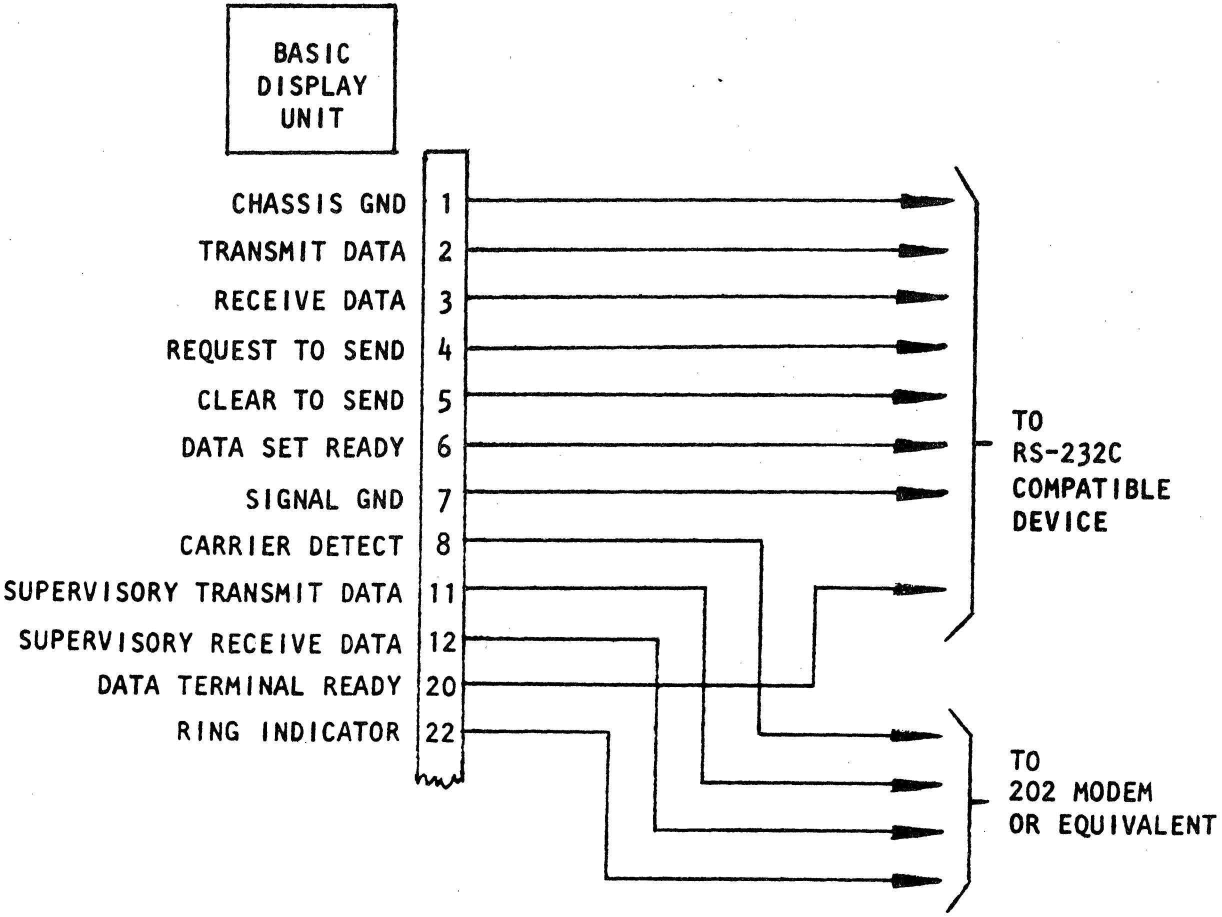


FIGURE 2-2. RS-232C To Basic Display Unit Interconnection

2.4.2 Extended Motherboard Installation

The optional extended motherboard may be easily installed following the procedure described below.

a. Remove system power.

b. Carefully remove the metal shroud from the Basic Display Unit by removing four retaining screws (two screws located on each side near the bottom).

c. Insert the extended motherboard into the connector provided at the right-hand side of the display unit base. Note that the connectors on the motherboard should be oriented to the rear of the chassis when inserted into the connector (an index key is used in the connector to prevent incorrect connection).

d. Additional modules may be installed into the card cage by removing the small metal door located on the rear panel. When installing additional modules, the component side of the module should be facing up (an index key is used in each socket on the extended motherboard to prevent improper installation). The metal door must be kept in place during operation to ensure proper air flow and cooling.

2.5 BASIC DISPLAY UNIT INTERCONNECTIONS

Interconnections to the Basic Display Unit can be made via four different connectors: two internal to the display and two external. The internal connectors (J1 and J2) are used for extending the bus and connecting the optional keyboard, respectively. The external connectors (current loop connector and RS-232C interface connector) are used to connect external devices (such as the host computer) to the display. The signals provided on these connectors are identified in the following tables. These tables list each pin connection, signal mnemonic, and signal characteristic.

TABLE 2-1. Extended Motherboard Bus Interface Signals (J1)

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
A,B,C	+5V	+5 Vdc - Used for the display's logic circuits.
D	$\overline{\text{IRQ}}$	INTERRUPT REQUEST - This low level signal is used to request that the MPU initiate an interrupt sequence. The MPU will wait until it completes the instruction being executed before it recognizes the request. At that time, if the interrupt mask bit in the MPU condition code register is not set, the MPU will begin the interrupt sequence.
E	$\overline{\text{NMI}}$	NON-MASKABLE INTERRUPT - This low level signal is used to request that a non-maskable interrupt be generated within the MPU. The MPU will wait until it completes the instruction being executed before it recognizes the $\overline{\text{NMI}}$ signal. At that time, regardless

TABLE 2-1. Extended Motherboard Bus Interface Signals (J1)(Contd)

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
	$\overline{\text{NMI}}$ (Contd)	of the setting of the interrupt mask bit in the MPU's condition code register, the MPU will begin the interrupt sequence.
F		Not Used
H	GND	GROUND
J	Ø2	PHASE 2 - MPU clock signal.
K	GND	GROUND
L	MEM-CLK	MEMORY CLOCK - An ungated TTL level signal used to refresh external memory modules.
M	-12V	-12 Vdc - Used to generate the -5 Vdc.
N	TSC	THREE-STATE CONTROL - This input signal line is used to hold the address bus, the R/W signal line, and the VUA signal line all in the high impedance state whenever TSC is held high. This prevents the MPU from accessing the system. A jumper is located on the CRT Controller Board to hold this signal low, preventing it from being used. This jumper must be removed if three-state control is desired.
P	BA	BUS AVAILABLE - This signal is normally held low. When activated, it will go high to indicate that the MPU has stopped and the address and data busses are available for accessing by the rest of the system. This will occur when the $\overline{\text{HALT}}$ line is a low level. At such time, all the address bus, data bus, R/W signal, and VUA signal bus drivers will go to their high impedance (off) states.
R	MEM-RDY	MEMORY READY - This signal permits the Basic Display Unit to operate with slow memory modules. When this command is a low level signal, the Ø1 and Ø2 clock signals are stretched, with the Ø2 clock signal held high. The Ø2 clock signal remains at a high level until the initiating module completes its memory operation. When completed, the initiating module returns MEM-RDY to a high level, enabling the clock to again generate the Ø1 and Ø2 clock signals.
S		Not Used
T	+12V	+12 Vdc - Available for use by external memory devices.

TABLE 2-1. Extended Motherboard Bus Interface Signals (J1)(Contd)

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
U to W		Not Used
X,Y,Z	GND	GROUND
$\overline{A}$ to $\overline{F}$		Not Used
$\overline{H}$	$\overline{D3}$	DATA bus (bit 3) - One of 8 bidirectional data lines used to transfer data between the MPU and all other devices connected to the bus.
$\overline{J}$	$\overline{D7}$	DATA bus (bit 7) - Same as $\overline{D3}$ on P1-$\overline{H}$.
$\overline{K}$	$\overline{D2}$	DATA bus (bit 2) - Same as $\overline{D3}$ on P1-$\overline{H}$.
$\overline{L}$	$\overline{D6}$	DATA bus (bit 6) - Same as $\overline{D3}$ on P1-$\overline{H}$.
$\overline{M}$	A14	ADDRESS bus (bit 14) - One of 16 address lines used to select any location within the system.
$\overline{N}$	A13	ADDRESS bus (bit 13) - Same as A14 on P1-$\overline{M}$.
$\overline{P}$	A10	ADDRESS bus (bit 10) Same as A14 on P1-$\overline{M}$.
$\overline{R}$	A9	ADDRESS bus (bit 9) - Same as A14 on P1-$\overline{M}$.
$\overline{S}$	A6	ADDRESS bus (bit 6) - Same as A14 on P1-$\overline{M}$.
$\overline{T}$	A5	ADDRESS bus (bit 5) - Same as A14 on P1-$\overline{M}$.
$\overline{U}$	A2	ADDRESS bus (bit 2) - Same as A14 on P1-$\overline{M}$.
$\overline{V}$	A1	ADDRESS bus (bit 1) - Same as A14 on P1-$\overline{M}$.
$\overline{W}$,$\overline{X}$,$\overline{Y}$	GND	GROUND
1,2,3	+5V	+5 Vdc - Used for the display's logic circuits.
4	HALT	HALT - When this input is in the high state, the MPU will fetch the instruction addressed by the program counter, will start instruction execution, and will notify all other modules that the bus is busy by holding the BA signal low. When low, all activity in the MPU will be halted and the bus is made available to the other modules by holding the BA signal high.
5	RST	RESET - This input/output signal, when used as an output, provides the low level reset circuit output to any external device requiring a Reset signal. An external switch closure to ground may be applied to this pin to reset the Basic Display Unit.

TABLE 2-1. Extended Motherboard Bus Interface Signals (J1)(Contd)

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
6	R/W	READ/WRITE - This three-state output line is controlled by the R/W output of the MPU and indicates to any plug-in module whether the MPU is performing a memory read (high) or write (low) operation.
7		Not Used
8,9	GND	GROUND
10	VUA	VALID USER'S ADDRESS - This high level signal is used to indicate that the address on the address bus is valid.
11	-12V	-12 Vdc - Used to generate the -5 Vdc.
12	REF-REQ	REFRESH REQUEST - This s-gnal, when low, initiates a refresh of the dynamic memory modules. During the refresh operation, the clock circuit is inhibited from generating its Ø1 and Ø2 clock signals; but still generates the MEM-CLK signal.
13	REF-GNT	REFRESH GRANT - This signal, when high, instructs the dynamic memory modules to refresh.
14,15		Not Used
16	+12V	+12 Vdc - Available for use by external memory devices.
17 to 19		Not Used
20,21,22	GND	GROUND
23 to 28		Not Used
29	$\overline{\text{D1}}$	DATA bus (bit 1) - Same as $\overline{\text{D3}}$ on P1-$\overline{\text{H}}$.
30	$\overline{\text{D5}}$	DATA bus (bit 5) - Same as $\overline{\text{D3}}$ on P1-$\overline{\text{H}}$.
31	$\overline{\text{D0}}$	DATA bus (bit 0) - Same as $\overline{\text{D3}}$ on P1-$\overline{\text{H}}$.
32	$\overline{\text{D4}}$	DATA bus (bit 4) - Same as $\overline{\text{D3}}$ on P1-$\overline{\text{H}}$.
33	A15	ADDRESS bus (bit 15) - Same as A14 on P1-$\overline{\text{M}}$.
34	A12	ADDRESS bus (bit 12) - Same as A14 on P1-$\overline{\text{M}}$.
35	A11	ADDRESS bus (bit 11) - Same as A14 on P1-$\overline{\text{M}}$.
36	A8	ADDRESS bus (bit 8) - Same as A14 on P1-$\overline{\text{M}}$.

TABLE 2-1. Extended Motherboard Bus Interface Signals (J1)(Contd)

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
37	A7	ADDRESS bus (bit 7) - Same as A14 on P1-$\overline{\text{M}}$.
38	A4	ADDRESS bus (bit 4) - Same as A14 on P1-$\overline{\text{M}}$.
39	A3	ADDRESS bus (bit 3) - Same as A14 on P1-$\overline{\text{M}}$.
40	A0	ADDRESS bus (bit 0) - Same as A14 on P1-$\overline{\text{M}}$.
41,42,43	GND	GROUND

TABLE 2-2. Keyboard Interface Signals (J2)

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
1	KB-DØ	KEYBOARD data (bit 0) - One of 7 data lines used to input ASCII encoded characters from the keyboard.
2	KB-D1	KEYBOARD data (bit 1) - Same as KB-DØ on P2-1.
3	KB-D2	KEYBOARD data (bit 2) - Same as KB-DØ on P2-1.
4	KB-D3	KEYBOARD data (bit 3) - Same as KB-DØ on P2-1.
5	KB-D4	KEYBOARD data (bit 4) - Same as KB-DØ on P2-1.
6	KB-D5	KEYBOARD data (bit 5) - Same as KB-DØ on P2-1.
7	KB-D6	KEYBOARD data (bit 6) - Same as KB-DØ on P2-1.
8	KB-D7	KEYBOARD data (bit 7) - The data line used to define whether the input ASCII character is to be interpreted as a character (low) or control (high) word.
9	$\overline{\text{KBSTB}}$	KEYBOARD STROBE - The low-going pulse generated by depressing any key on the keyboard, and used to initiate an interrupt request.
10	$\overline{\text{KBATT}}$	KEYBOARD ATTACHED - A low level input signal used to indicate that the keyboard is attached to the Basic Display Unit.
11	$\overline{\text{ON LN DR}}$	ON LINE DRIVER - A low level output signal used to illuminate the ON LINE key target light when in the On Line mode.

TABLE 2-2. Keyboard Interface Signals (J2)(Contd)

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
12	$\overline{\text{AUT LF DR}}$	AUTO LINE FEED DRIVER - A low level output signal used to illuminate the AUTO LF key target light when in the Auto Line Feed mode. (Optional key)
13	$\overline{\text{PAG M DR}}$	PAGE MODE DRIVER - A low level output signal used to illuminate the PAGE MODE key target light when in the Page mode. (Optional key)
14	$\overline{\text{AUX LN DR}}$	AUXILIARY LINE DRIVER - A low level output signal used to illuminate the AUX LINE key target light. (Optional key)
15	$\overline{\text{AUX EN DR}}$	AUXILIARY ENABLE DRIVER - A low level output signal used to illuminate the AUX ENBL key target light. (Optional key)
16	$\overline{\text{INS CH DR}}$	INSERT CHARACTER DRIVER - A low level output signal used to illuminate the INS CHAR key target light. (Optional key)
17	+5V	+5 Vdc - Supplied to keyboard for logic circuits.
18		Not Used
19,20	GND	GROUND

TABLE 2-3. Current Loop Interface Signals

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
1	-TRANS	- TRANSMIT - This -12 Vdc line provides the signal return for a teletypewriter terminal or other device using a current loop interface.
2	-RCV	- RECEIVE - This signal provides the control to operate the paper tape reader on a modified teletypewriter terminal under control of the Basic Display Unit.
3	+TRANS	+ TRANSMIT - This line accepts the input from a teletypewriter terminal or other current loop device.
4	GND	GROUND - This line is connected to the Basic Display Unit ground. This pin is also connected to pin 5 via a jumper connection in the cable.

TABLE 2-3. Current Loop Interface Signals (Contd)

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
5	$\overline{\text{CLATTCHD}}$	CURRENT LOOP ATTACHED - This low level input signal indicates that a current loop device is attached to the Basic Display Unit.
6	+RCV	+ RECEIVE - This line transfers data to a teletypewriter or other current loop device.

TABLE 2-4. RS-232C Interface Signals

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
1	CHASSIS GND	CHASSIS GROUND - Common for the -12 Vdc source. This line provides a safety ground connection directly to the power supply for RS-232C compatible devices.
2	TRANSMIT DATA	TRANSMIT DATA - This line transfers data to an RS-232C compatible device.
3	RECEIVE DATA	RECEIVE DATA - This line accepts input data from an RS-232C compatible device.
4	REQUEST TO SEND	REQUEST TO SEND - This output line is used to request permission from the RS-232C device to begin sending data.
5	CLEAR TO SEND	CLEAR TO SEND - This line is a high level when the RS-232C device is ready to receive data.
6	DATA SET READY	DATA SET READY - This line is a high level when an RS-232C device is connected to the Basic Display Unit and the device is operating.
7	SIGNAL GND	SIGNAL GROUND - This line provides a common signal connection to the RS-232C device.
8	CARRIER DETECT	CARRIER DETECT - This line is a high level when the RS-232C device has detected the carrier signal.
9,10		Not Used

TABLE 2-4. RS-232C Interface Signals (Contd)

PIN NUMBER	SIGNAL MNEMONICS	SIGNAL NAME AND DESCRIPTION
11	SUPERVISORY TRANSMIT DATA	SUPERVISORY TRANSMIT DATA - This line transfers data to a 202 Modem device (or equivalent) for transmission.
12	SUPERVISORY RECEIVE DATA	SUPERVISORY RECEIVE DATA - This line accepts received data from a 202 Modem device (or equivalent).
13 to 19		Not Used
20	DATA TERMINAL READY	DATA TERMINAL READY - This output line indicates to the RS-232C device that the Basic Display Unit is ready.
21		Not Used
22	RING INDICATOR	RING INDICATOR - This line indicates that the 103 or 202 Modem has been called (received a ring from an external source).
23 to 25		Not Used

CHAPTER 3
OPERATING INSTRUCTIONS

3.1 INTRODUCTION

Information in this chapter is intended to familiarize the user with the location and basic functions of the controls on the Basic Display Unit. This chapter also contains information needed to program the Basic Display Unit.

3.2 OPERATING CONTROLS

The operating controls of the Basic Display Unit are comprised of three functional groups: front panel controls, rear panel controls, and the optional Keyboard Assembly.

3.2.2 Front Panel Controls

The front panel controls turn the Basic Display Unit on and off and adjust the brightness of the CRT display. These controls are shown in Figure 3-1.

3.2.3 Rear Panel Controls

The rear panel controls consist of three-9 SPST switch DIP arrays, one momentary contact pushbutton switch, and a volume control. The three-9 switch, single-pole, single-throw dip arrays are located on the top left-hand side of the panel, and are used to manually establish various operating characteristics. All the remaining controls are located on the top right-hand side of the panel. The function of each control is described in Table 3-1. The rear panel controls are shown in Figure 3-1.

TABLE 3-1. Rear Panel Controls

CONTROL	FUNCTION
FAC CODE	With the exception of the Set and Reset Field Protect FAC codes (Field Attribute Code), all FAC codes that have been received and stored in the Display Memory can be enabled and/or displayed by setting the appropriate switch. If the OFF/ENABLE switch is positioned to ENABLE, FAC codes are permitted to perform their designated function. As such, FAC code commands set or reset the attributes for the cursor prompt position. Contiguous attribute commands will set or reset those attributes for a single prompt position. If the OFF/DISPL switch is in the DISPL position and the OFF/ENABLE switch is in the OFF position, all FAC codes will be displayed. If the OFF/ENABLE switch is in the ENABLE position, the OFF/DISPL switch is in the DISPLAY position, and a SET NON-DISPLAY FAC code is received, the FAC codes following the SET NON-DISPLAY code will not be displayed. The FAC code operations are functional only when the optional Extended Display Feature is provided.
TRANS MODE	The TRANS (Transparent) MODE switch permits the user to display all characters received without recognition of commands. This mode is intended to aid in debugging programs.

TABLE 3-1. Rear Panel Controls (cont'd)

CONTROL	FUNCTION
VIDEO INV	The VIDEO INV (Video Invert) switch permits the user to select either a black background with white lettering (OFF) or a white background with black lettering (ON).
A, B, and C	These switches are spares and are available for interpretation by user software.
SPEC CHAR	This switch permits all the special characters to be displayed on the screen when in the DISPL position. The special characters are represented by the hexadecimal codes 00 through 1F shown in Table 1-2. ASCII control commands that also fall within the range of 00 to 1F will not be executed.
LINE FREQ	This switch permits selection of a 50 Hz or 60 Hz line frequency.
DUPLEX	The DUPLEX switch permits selection of a HALF or FULL duplex mode of operation. In FULL duplex, data will be displayed only if the computer or external device connected to the communication interface echoes back the data transmitted from a keyboard device. In HALF duplex, data is routed from the keyboard device to the display so that the user can see what is actually being transmitted on the communication interface.
PARITY	These switches permit the user to include a parity bit in the transmitted and received data stream. If PARITY is selected, then the user must select either EVEN or ODD parity.
XMIT WORD	This switch permits the user to select seven or eight data bits in the transmitted and received data streams.
STOP BITS	This switch permits the use of one or two stop bits in the transmitted and received data streams.
CONNECTION	This switch permits the use of a 103 or 202 MODEM or DIRECT connection via the RS-232 or current loop communication interface. MODEM selection can be used only when the optional Extended Communication Feature is provided. If the Extended Communication Feature is not provided and an optional keyboard is attached, the keyboard's target lighted keys are blinked at half-second intervals.
MODEM TYPE	When the CONNECTION switch is in the MODEM position, the MODEM TYPE switch permits selection of the 103 or 202 modem. The 103 modem can be configured in the HALF or FULL duplex mode. The 202 modem is configured in the HALF duplex mode and the host system must send to the terminal first. The terminal can be forced into the transmit mode by entering the selected code (ETX or EOT) on the keyboard, after first going into the ON-LINE mode.
TURN AROUND	When the 202 is selected as the modem type, the user has the option to select the operation which determines the receive or transmit mode. With the switch in the S-CHAN position, the transmit or receive operation is controlled by control lines connected directly to the modem device. In the CODE position, the same lines are used to control the transmit or receive operation; however, a code can be used to select the operation (refer to CODE switch description).

TABLE 3-1. Rear Panel Controls (cont'd)

CONTROL	FUNCTION
CODE	The CODE switch permits the user to specify the code (EOT or ETX) which selects the receive or transmit operation. When transmitted data equals the selected code, the Basic Display Unit will switch to receive. When received data equals the selected code, the unit will switch to transmit.
BAUD RATE	These nine switches permit the communications baud rate to be established. Either 110, 150, 300, 600, 1200, 1800, 2400, 4800, or 9600 baud can be selected.
RESET	This switch permits the terminal to be manually set and initialized. The RESET switch must be depressed whenever a rear panel configuration switch is changed.
VOLUME INCREASE	This control permits the audio level of the bell to be adjusted.

3.2.4 Optional Keyboard Assembly Controls

Several different keyboards are offered as optional equipment for the Basic Display Unit. Refer to the Keyboard Assembly Supplement for information pertinent to its controls.

3.3 GENERAL OPERATING INFORMATION

The following general rules apply to the Basic Display Unit.

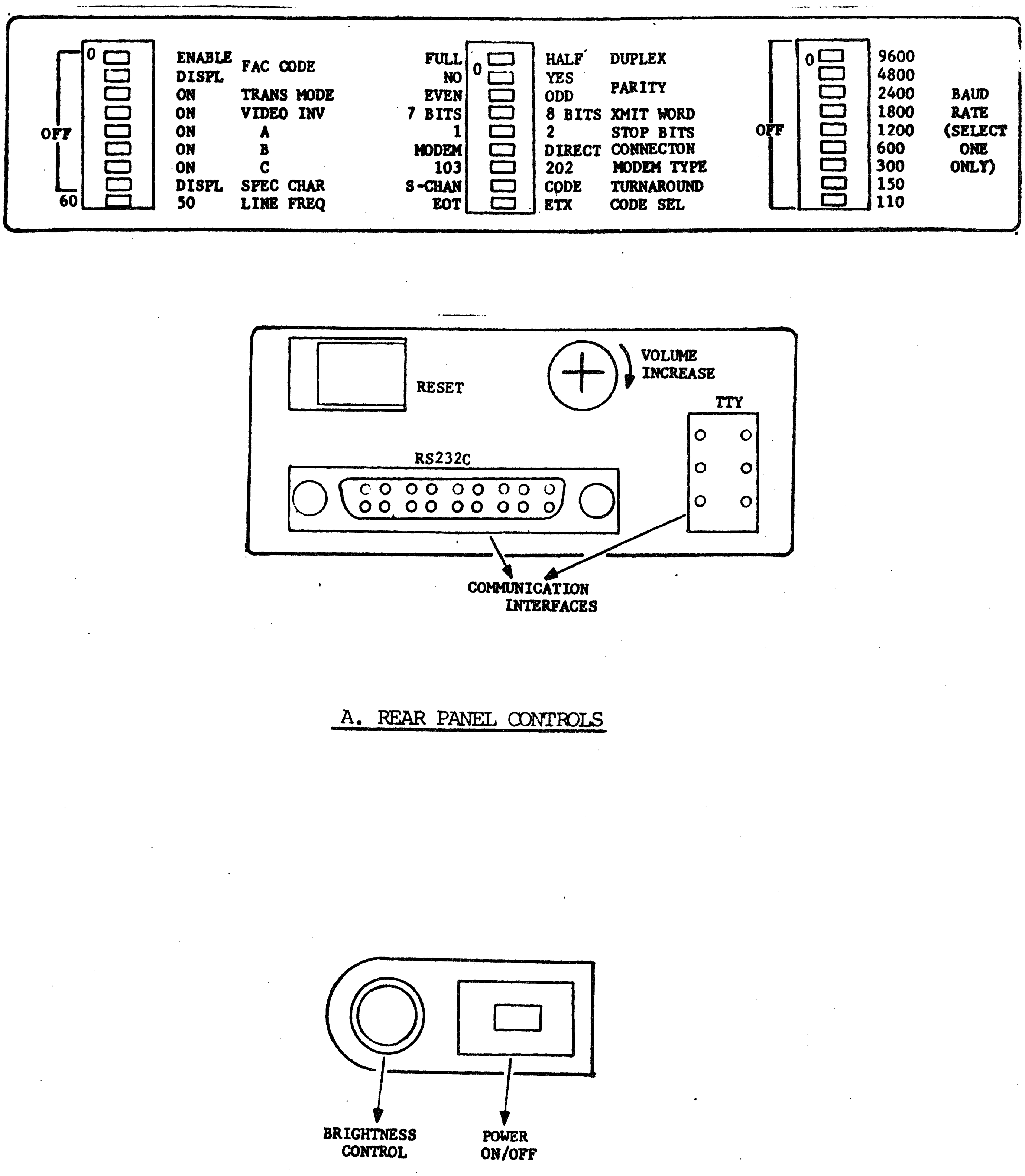
- The Basic Display Unit can operate in one of three display modes: Scroll, Page, and Protect. In Scroll mode, the screen display area serves as a scroll area. In this mode, a line feed command causes an upward scroll of all lines if the cursor is in the last line. The top line is scrolled out of the display area and the bottom line is left blank with the cursor in the current column.

  In the optional Page mode, the screen display area represents a page of information. The cursor moves about within this page and is controlled by commands. The cursor provides a "prompt" position to indicate where the next character will be displayed.

  The optional Protect mode is a more constrained PAGE mode; that is, data is displayed as a page of information, but any area may be protected. When protected, the area cannot be changed unless the mode is reset.

- All commands (functions) may originate from a keyboard or remotely via the communication interface.

- All commands have a unique 8-bit value which may be represented by an escape sequence to facilitate 7-bit transmission. Some commands are also ASCII control characters.



A. REAR PANEL CONTROLS

B. FRONT PANEL CONTROLS

FIGURE 3-1. Basic Display Unit, Front and Rear Panel Controls

- A screen display area may be defined, via commands, to represent any rectangle within the absolute display area. Most commands are constrained to function within the screen display area. There are special commands to read and write data anywhere in the absolute display area.

- Communication configuration switches on the rear panel of the Basic Display Unit are read only during power on reset or terminal reset. Therefore, whenever the position of these switches is changed, a reset must be initiated.

- Transparent mode forces the terminal to display all characters as received without recognition of commands. This mode is intended to aid program debug.

- The cursor always serves as a prompt and indicates where the next character will be displayed. As the characters are displayed, the cursor advances under control of certain mode settings. At all times, the cursor is restricted to the screen display area.

  In Page mode, the cursor's position changes according to mode settings and its relative position in the screen display area. When Protect mode is set, the cursor always prompts at a non-protected position. As a character is displayed, the cursor advances to the right to the next non-protected position.

3.3.1 Initialization

The first step performed when initializing the Basic Display Unit is to verify that the LINE FREQ switch, located on the rear panel, is set to the proper line frequency (50 Hz or 60 Hz). The next step is to set the remaining switches located on the left-hand side of the rear panel for the desired operation.

When the operation of the unit has been established, power can be applied by pressing the power on switch located on the front panel. The Basic Display Unit's software and hardware are brought to the following state when power is applied to the unit:

a. Screen display area equals 1 through 24 lines and columns 1 through 80

b. Display is "blanked"

c. Cursor is in the Home position (upper left-hand corner of screen)

d. Scroll mode is set

e. Transparent mode is reset

f. Normal video is set

g. On Line mode is set

h. Auto Line Feed mode is reset

After a one-minute warmup, adjust the Brightness control to produce a well-defined cursor. The Basic Display Unit is now ready for data transmission.

NOTE

The rear panel single-pole, single-throw switch DIP arrays used to establish the Basic Display Unit's operation are read only when power is applied to the terminal or the RESET switch is set.

3.3.2 Operator Test Procedure

The basic operating functions of the Basic Display Unit can be validated (exclusive of the serial communications interface) by an operator-monitored series of checks when the unit is first powered up. These checks can also be made if a unit malfunction is suspected during daily operation. Note that all of the logic functions of the "Power on Check" can be simulated using the RESET switch on the rear panel.

The following checks are made, assuming the user has the optional Keyboard Assembly attached to the unit. If a user supplied keyboard device is used, any references to the target lighted keys of optional Keyboard Assembly should be ignored.

3.3.2.1 Power On Check - When the Basic Display Unit is powered on, the ON LINE target lighted key should not be illuminated. After about one-quarter to one-half second, the ON LINE key should be illuminated. An exception to this would be if a rear panel modem switch setting has been made and the optional Extended Communications Feature is not provided with the unit. In this case, all keyboard target lighted keys will blink on or off at half-second intervals. This will continue until the modem option is disabled by changing the associated rear panel switches and resetting the unit.

After a one-minute warmup, the screen should be blank with the cursor block blinking in the upper left-hand corner of the screen. No other characters should be visible on the screen at this point. If the cursor is not visible on the screen, turn the Brightness control until a raster is faintly visible on the screen. The cursor should now be visible. Adjust the brightness and contrast controls to produce a well-defined cursor.

3.3.2.2 Off Line Checks - When the Power On Check has been successfully completed, the following off line checks can be made. Place the unit in off line mode by depressing the ON LINE key.

Character Entry Checks

a. Depress the 'a' key and enter a number of the characters in memory and on the display.

b. Depress the SHIFT key and enter more characters. Note that any letters entered are now capitalized.

c. Depress the RETURN key and notice that the cursor moves to the first location of that line.

d. Depress the LINE FEED key and note that the cursor moves to the same column in the line directly below its previous location.

e. Depress the "ESCAPE" followed by "@" keys and note that the cursor returns to the Home position (the first column on the first line).

f. Depress the LINE FEED key several times followed by several "a" keys. This should move the cursor toward the center of the screen.

g. Depress the "ESCAPE" key followed by a lower case "q". Note that the display completely erases and the cursor returns to the Home position.

Cursor Control Checks

Before performing the checkout, depress the "ESCAPE" key followed by a lower case "q" to reset the unit. Insure that the unit is in a loopback mode (Local or Half Duplex), the cursor is in the Home position and the display is blank. Perform the checkout as follows:

a. Depress "ESCAPE" followed by upper case "B" keys, repeating this sequence until the cursor reaches the bottom line of the display.

b. Depress the "ESCAPE" followed by upper case "D" keys, repeating this sequence until the cursor nears the last position on the line. Release the key at this time.

c. Depress the "ESCAPE" followed by upper case "C" keys, repeating this sequence to cause the cursor to wrap around to the last position of the line above.

d. Depress the "ESCAPE" followed by "@" keys to move the cursor to the first character position of the first line.

This completes the operator self-test of the Basic Display Unit.

3.4 COMMAND DESCRIPTION

The Basic Display Unit accepts data and commands via the communications interface or a keyboard device and interprets them to effect what is presented on the display. The Basic Display Unit reacts to the full ASCII character set, receiving and storing both upper and lower case codes. For this reason, the Basic Display Unit can be used without software modification in applications now using teletypewriter compatible equipment. The effect of each character capable of being displayed is shown in Table 1-3.

3.4.1 Communications Interface Commands

The Communication Interface is used to transfer serial data between the Basic Display Unit and the host system. The interface consists of two parts. The first receives data and commands from the host system. The second part transmits data and commands to the host system. Hence, the interface is a two-way simultaneous communication mechanism. The interface is implemented as dual in-memory circular queues.

An 8-bit byte which totally and uniquely defines each character, which may be upper or lower case, or a control function, is the basic element that is communicated on the interface. The byte contains seven bits, equivalent to the code of a character, and an eighth bit defining whether the code is to be interpreted as a display character or as a control function.

The set of 8-bit bytes recognized by the Basic Display Unit is restricted to the hexadecimal values 00 through FF. A byte whose value is within the range of 00 to 7F represents a character code. The code includes the 128 ASCII upper and lower case letters, numerals, and general purpose symbols (refer to Appendix B). In addition to the ASCII characters, the codes that normally represent the ASCII control characters (NUL, SOH, ETX, etc.) can be used to represent special characters. In the Special Character mode, a code received in the range of 00 through 1F is recognized as a lower case Greek character, an upper case Greek character (Omega or Sigma), or a special symbol character. The Special Character mode can be entered by setting the associated rear panel switch or by command.

A byte whose value is within the range of 80 to FF potentially represents a command. Restricting these commands to this range provides a means of representing command bytes as escape sequences without making use of the ASCII control characters, thus avoiding conflicts with existing communications protocols which place special significance on certain ASCII control characters.

An escape sequence consists of an ASCII Escape character (1B) followed by a byte whose value is within the range of 20 to 7F. The escape character is used to represent the left-most bit of the following byte. Hence, the escape sequence allows command bytes to be communicated on the interface in a strict 7-bit format. Two escape characters in sequence are used to represent a character to be displayed.

Table 3-2 summarizes the commands available for the Basic Display Unit. It provides the value, display area, action/effect, and byte sequence for each command. The virtual display area is organized into 24 lines with 80 characters per line. The absolute display area is defined as being the full screen area. The byte sequence is the total number of bytes needed to achieve the action/effect.

Table 3-3 lists the ASCII codes used to position the cursor in the virtual display area. To position the cursor, a command followed by the ASCII position code(s) must be received on the interface. The commands that are used to perform this function are: LOAD CURSOR, BOTTOM DISPLAY LINE, TOP DISPLAY LINE[1], LEFT DISPLAY COLUMN[1], and RIGHT DISPLAY COLUMN[1].

3.4.2 Keyboard Interface Commands

The keyboard interface provides a flexible method of connecting any byte-oriented keyboard to the Basic Display Unit. Data flows from the keyboard via an 8-bit data bus. The interface is designed to accept the full complement of codes that can be communicated on the communications interface. For further information in regard to the keyboards offered as optional equipment to the Basic Display Unit, refer to the Keyboard Assembly Supplement.

[1] Optional Extended Display Feature Commands

3.4.3 Extended Display Feature Option Commands

In addition to the standard commands discussed in paragraphs 3.4.1 and 3.4.2, the Extended Display Feature option provides controls to operate the Basic Display Unit in Page mode, Protect mode, and Edit mode. Table 3-4 lists each command that is supported by the Extended Display Feature that can be received via the communications interface or keyboard device.

3.5 PROGRAMMING INFORMATION

In order to provide the user with maximum flexibility in implementing the Basic Display Unit, a software interface has been designed allowing the user to introduce a subroutine into the system that may intercept data and command characters in three logical areas:

- Keyboard input to Keyboard Control Software Process Module
- Communications output from Display Control software process module
- Communications input to Display Control software process module

In each case the user may interpret the character(s) received and, through the user's translation program, replace the original character value(s) with a new value or values.

The user may then "queue" the data and activate a process that is either user coded or one of the resident processes. In lieu of either of the former options, control may be returned to the calling program to continue normal processing.

The remainder of this chapter explains how the user may interface with the various resident modules to implement an "application" process and/or subroutine. Details of the system software are shown in the Memory Map (Figure 1-3) and the Flow Chart (Figure 3-2).

3.5.1 Programming Considerations

Before the user can write application programs for the Basic Display Unit, he must be familiar with the intricacies of the software. All software modules abide by a set of rules governing the execution under management of the Monitor Control software process module. These rules are implicitly agreed upon since there are no hardware mechanisms to enforce them.

- Each process has its own stack whose size is sufficient to avoid stack overflow.
- When a process has nothing to do, it suspends itself.
- When a process or interrupt handler places data in a circular queue, it activates the process which services that queue.
- A process or interrupt handler will write into only those RAM areas assigned to it.

The Basic Display Unit may have up to 16 processes which share the processor on a priority basis. At all times, the active process with the highest priority is executing. When it suspends, the active process with the

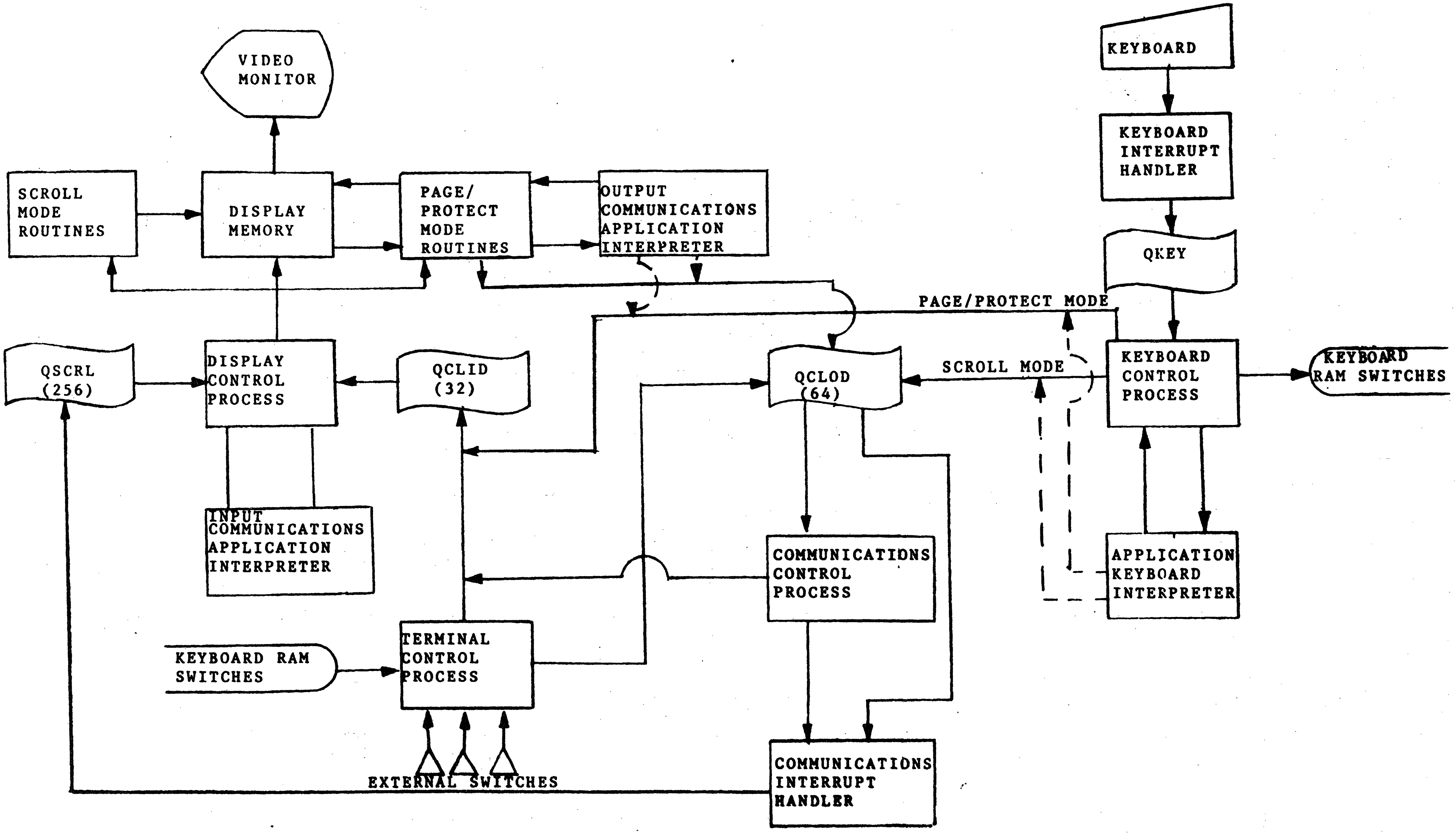


FIGURE 3-2. Basic Display Unit System Flow Chart

Table 3-2. Basic Display Unit Communications Interface Commands

COMMAND	INTERFACE VALUE		DISPLAY AREA	ACTION/EFFECT	BYTE SEQUENCE		
					NO. OF BYTES		DESCRIPTION
	NON-ESC.	ESC SEQ.			NON-ESC.	ESC.SEQ.	
HOME	C0	ESC,@	VIRTUAL	CURSOR MOVES TO LEFT-MOST COLUMN IN TOP LINE.	1	2	COMMAND BYTE(S)
CURSOR DOWN ONE LINE (↓)	C2	ESC,B	VIRTUAL	CURSOR MOVES DOWN ONE LINE IN SAME COLUMN; IF ALREADY IN LAST LINE, SCROLL IS EXECUTED & LAST LINE IS BLANKED.	1	2	COMMAND BYTE(S)
CURSOR LEFT ONE COLUMN (←)	C3	ESC,C	VIRTUAL	CURSOR MOVES LEFT ONE COLUMN WITHOUT LINE ROLLOVER.	1	2	COMMAND BYTE(S)
CURSOR RIGHT ONE COLUMN (→)	C4	ESC,D	VIRTUAL	CURSOR MOVES RIGHT ONE COLUMN WITHOUT LINE ROLLOVER.	1	2	COMMAND BYTE(S)
LOAD CURSOR	C5	ESC,E	VIRTUAL	IF CURSOR IS OUTSIDE VIRTUAL DISPLAY AREA, THE COMMAND IS IGNORED; OTHERWISE, CURSOR MOVES TO DESIGNATED POSITION.	3	4	COMMAND BYTE(S) PLUS TWO BYTES DESIGNATING CURSOR POSITION (see Table 3-3)
ENABLE KEYBOARD	D2	ESC,R	N.A.	CAUSES KEYBOARD TO BE ENABLED (DATA CAN BE ENTERED FROM KEYBOARD).	1	2	COMMAND BYTE(S)
DISABLE KEYBOARD	D3	ESC,S	N.A.	CAUSES KEYBOARD TO BE DISABLED (DATA CANNOT BE ENTERED FROM THE KEYBOARD). KEYBOARD CAN BE ENABLED AGAIN ONLY THROUGH RECEIPT OF A KEYBOARD ENABLE COMMAND RECEIVED THROUGH THE COMMUNICATIONS INTERFACE.	1	2	COMMAND BYTE(S)
BOTTOM DISPLAY LINE	CA	ESC,J	ABSOLUTE	THE VIRTUAL DISPLAY AREA'S BOTTOM LINE IS MADE TO CORRESPOND WITH LINE DESIGNATED. (See Note 3).	2	3	COMMAND BYTE(S) PLUS ONE BYTE DESIGNATING THE LINE (see Table 3-3)
TAB	DA	ESC,Z	VIRTUAL	CURSOR MOVES RIGHT TO NEXT TAB POSITION WITHOUT LINE ROLLOVER.	1	2	COMMAND BYTE(S)
BACK TAB	DB	ESC,[	VIRTUAL	CURSOR MOVES LEFT TO PREVIOUS TAB POSITION WITHOUT LINE ROLLOVER.	1	2	COMMAND BYTE(S)

Table 3-2. Basic Display Unit Communications Interface Commands (cont'd)

COMMAND	INTERFACE VALUE		DISPLAY AREA	ACTION/EFFECT	BYTE SEQUENCE		
	NON-ESC.	ESC SEQ.			NO. OF BYTES		DESCRIPTION
					NON-ESC.	ESC.SEQ.	
SET TABS	DC	ESC,\	VIRTUAL	THE BLOCK OF DATA FOLLOWING THE COMMAND BYTE(S) DEFINES THE TAB POSITIONS. TAB POSITIONS ARE INDICATED BY THE HTS (09) TO SET A TAB, OR THE CANCEL (18) TO CLEAR A TAB, AND ARE SEPARATED BY SPACE (20) CHARACTERS TO INDICATE TAB SPACING (SETTING). THE BLOCK IS TERMINATED BY A LINE FEED CHARACTER. THE COMMAND BYTE(S) FOLLOWED BY LF WITH NO SPACES, CANCELS(18) OR HTS (09) WILL CAUSE ALL TABS TO BE RESET.	VARIABLE	VARIABLE	COMMAND BYTE(S), SPACE BYTES, HTS (09) OR CANCEL (18), AND A LF
SET TRANSPARENT MODE	EC	ESC,l	ABSOLUTE	SETS TRANSPARENT MODE OPERATION; CURSOR DOES NOT CHANGE POSITION.	1	2	COMMAND BYTE(S)
RESET TRANSPARENT MODE	ED	ESC,m	ABSOLUTE	RESETS TRANSPARENT MODE; CURSOR DOES NOT CHANGE POSITION.	1	2	COMMAND BYTE(S)
SET VIDEO INVERT	EE	ESC,n	ABSOLUTE	SETS VIDEO INVERT OPERATION; CURSOR DOES NOT CHANGE POSITION.	1	2	COMMAND BYTE(S)
RESET VIDEO INVERT	EF	ESC,o	ABSOLUTE	RESETS VIDEO INVERT OPERATION; CURSOR DOES NOT CHANGE POSITION.	1	2	COMMAND BYTE(S)
SET DISPLAY SPECIAL CHARACTERS	FC	ESC,:	N.A.	PERMITS SPECIAL CHARACTERS TO BE DISPLAYED.	1	2	COMMAND BYTE(S)
RESET DISPLAY SPECIAL CHARACTERS	FD	ESC,}	N.A.	SPECIAL CHARACTERS CANNOT BE DISPLAYED.	1	2	COMMAND BYTE(S)
TERMINAL RESET	F1	ESC,q	ABSOLUTE	CAUSES THE BASIC DISPLAY UNIT TO BE RESET AS IF THE "RESET" SWITCH HAD BEEN ACTIVATED.	1	2	COMMAND BYTE(S)
END-OF-LINE FAC	FE	ESC,9	N.A.	AN OUTPUT COMMAND USED TO MARK UNUSED SPACE IN LINE. LINE LENGTH DETERMINED BY RIGHT-MOST DATA BYTE POSITION. REMAINING UNUSED, UNFILLED POSITIONS TO 80TH COLUMN ARE NOT CONSIDERED PART OF LINE.	1	2	COMMAND BYTE(S)

Table 3-2. Basic Display Unit Communications Interface Commands (cont'd)

COMMAND	INTERFACE VALUE		DISPLAY AREA	ACTION/EFFECT	BYTE SEQUENCE		
					NO. OF BYTES		DESCRIPTION
	NON-ESC.	ESC SEQ.			NON-ESC.	ESC.SEQ.	
DISPLAY STATUS ON	F2	ESC,r	ABSOLUTE	CERTAIN STATUS COMMANDS CAN BE RECEIVED ON THE COMMUNICATIONS INTERFACE AND DISPLAYED. THE LAST THREE POSITIONS IN THE ABSOLUTE DISPLAY AREA ARE USED TO SHOW A BLINKING STATUS INDICATOR. THE INDICATORS AND THEIR ASSOCIATE CODES ARE: INDICATOR CODE FUNCTION PE 98 PARITY ERROR RO 9A RECEIVE OVERRUN FE 9B FRAMING ERROR FRAMING AND PARITY ERROR COMMANDS ARE FOLLOWED BY THE BYTE SUFFERING THE ERROR CONDITION.	1	2	COMMAND BYTE(S)
DISPLAY STATUS OFF	F3	ESC,s	ABSOLUTE	RECEIVED STATUS INDICATORS PE, RO, AND FE CANNOT BE DISPLAYED.	1	2	COMMAND BYTE(S)
ASCII BEL (BELL)	07	NONE	N.A.	CAUSES THE BELL TO ISSUE TONE FOR A PERIOD OF ONE-HALF SECOND.	1		COMMAND BYTE
ASCII CAN (CANCEL)	18	NONE	N.A.	CURSOR MOVES TO LEFT MARGIN, LINE BLANKED WITH SPACE (20) CHARACTER.	1		COMMAND BYTE
ASCII BS (BACKSPACE)	08	NONE	VIRTUAL	CURSOR MOVES LEFT ONE COLUMN WITHOUT LINE ROLLOVER.	1		COMMAND BYTE
ASCII HT (HORIZONTAL TAB)	09	NONE	VIRTUAL	CURSOR MOVES RIGHT TO NEXT TAB POSITION (WITHOUT LINE ROLLOVER) AS DEFINED BY THE SET TABS COMMAND.	1		COMMAND BYTE
ASCII LF (LINE FEED)	0A	NONE	VIRTUAL	CURSOR MOVES DOWN ONE LINE IN SAME COLUMN. IF ALREADY IN LAST LINE, SCROLL IS EXECUTED AND LAST LINE IS BLANKED.	1		COMMAND BYTE
ASCII FF (FORMFEED)	0C	NONE	VIRTUAL	CURSOR MOVES RIGHT ONE COLUMN WITHOUT LINE ROLLOVER.	1		COMMAND BYTE
ASCII CR (CARRIAGE RETURN)	0D	NONE	VIRTUAL	CURSOR MOVES TO LEFT MARGIN.	1		COMMAND BYTE

NOTES:
1. All non-escape interface values are in hexadecimal
2. All escape sequence interface values are ASCII characters
3. After executing this command, the cursor is checked for being within the virtual display. If it is not, it is moved to the right until it is. If Protect mode is set, it seeks a non-protected column to the right.

TABLE 3-3. Cursor Position Code

LINE/COLUMN	ASCII CODE	LINE/COLUMN	ASCII CODE	LINE/COLUMN	ASCII CODE
1	SPACE	28	;	55	V
2	!	29	<	56	W
3	"	30	=	57	X
4	#	31	>	58	Y
5	$	32	?	59	Z
6	%	33	@	60	[
7	&	34	A	61	\
8	'	35	B	62	]
9	(	36	C	63	^
10	)	37	D	64	_
11	*	38	E	65	`
12	+	39	F	66	a
13	,	40	G	67	b
14	-	41	H	68	c
15	.	42	I	69	d
16	/	43	J	70	e
17	0	44	K	71	f
18	1	45	L	72	g
19	2	46	M	73	h
20	3	47	N	74	i
21	4	48	O	75	j
22	5	49	P	76	k
23	6	50	Q	77	l
24	7	51	R	78	m
25	8	52	S	79	n
26	9	53	T	80	o
27	:	54	U		

NOTE: CURSOR POSITION = COMMAND BYTE(S), LINE CODE, COLUMN CODE

TABLE 3-4. Extended Display Feature Communication Interface Commands

COMMAND	INTERFACE VALUE		DISPLAY AREA	MODE	ACTION/EFFECT	BYTE SEQUENCE		
						NO. OF BYTES		DESCRIPTION
	NON-ESC.	ESC SEQ.				NON-ESC.	ESC.SEQ.	
HOME	CO	ESC,@	VIRTUAL	SCROLL	CURSOR MOVES TO LEFT-MOST COLUMN IN TOP LINE.	1	2	COMMAND BYTE(S)
				PAGE	CURSOR MOVES TO LEFT-MOST COLUMN IN TOP LINE.			
				PROTECT	CURSOR MOVES TO LEFT-MOST COLUMN IN TOP LINE, SEEKS NON-PROTECTED COLUMN TO RIGHT; IF NONE EXISTS, CURSOR REMAINS IN LEFT-MOST COLUMN OF TOP LINE.			
CURSOR UP ONE LINE (↑)	C1	ESC,A	VIRTUAL	SCROLL	IGNORED	1	2	COMMAND BYTE(S)
				PAGE	CURSOR MOVES UP ONE LINE IN SAME COLUMN WITH PAGE ROLLOVER.			
				PROTECT	CURSOR MOVES UP ONE LINE IN SAME COLUMN WITH PAGE ROLLOVER, SEEKS NON-PROTECTED COLUMN TO THE LEFT.			
CURSOR DOWN ONE LINE (↓)	C2	ESC,B	VIRTUAL	SCROLL	CURSOR MOVES DOWN ONE LINE IN SAME COLUMN. IF ALREADY IN LAST LINE, SCROLL IS EXECUTED AND LAST LINE IS BLANKED.	1	2	COMMAND BYTE(S)
				PAGE	CURSOR MOVES DOWN ONE LINE IN SAME COLUMN WITH PAGE ROLLOVER.			
				PROTECT	CURSOR MOVES DOWN ONE LINE IN SAME COLUMN WITH PAGE ROLLOVER, SEEKS NON-PROTECTED COLUMN TO THE RIGHT.			
CURSOR LEFT ONE COLUMN (←)	C3	ESC,C	VIRTUAL	SCROLL	CURSOR MOVES LEFT ONE COLUMN WITHOUT LINE ROLLOVER.	1	2	COMMAND BYTE(S)
				PAGE	CURSOR MOVES LEFT ONE COLUMN WITH LINE AND PAGE ROLLOVER.			
				PROTECT	CURSOR MOVES LEFT ONE COLUMN WITH LINE AND PAGE ROLLOVER. SEEKS NON-PROTECTED COLUMN TO THE LEFT.			

TABLE 3-4. Extended Display Feature Communication Interface Commands (cont'd)

COMMAND	INTERFACE VALUE		DISPLAY AREA	MODE	ACTION/EFFECT	BYTE SEQUENCE		
						NO. OF BYTES		DESCRIPTION
	NON-ESC.	ESC SEQ.				NON-ESC.	ESC.SEQ.	
CURSOR RIGHT ONE COLUMN (→)	C4	ESC,D	VIRTUAL	SCROLL	CURSOR MOVES RIGHT ONE COLUMN WITHOUT LINE ROLLOVER.	1	2	COMMAND BYTE(S)
				PAGE	CURSOR MOVES RIGHT ONE COLUMN WITH LINE AND PAGE ROLLOVER.			
				PROTECT	CURSOR MOVES RIGHT ONE COLUMN WITH LINE AND PAGE ROLLOVER. SEEKS NON-PROTECTED COLUMN TO THE RIGHT.			
LOAD CURSOR	C5	ESC,E	VIRTUAL	SCROLL, PAGE	IF POSITION IS OUTSIDE VIRTUAL DISPLAY, COMMAND IS IGNORED; OTHERWISE, CURSOR MOVES TO DESIGNATED POSITION.	3	4	COMMAND BYTE(S) PLUS TWO BYTES DESIGNATING CURSOR POSITION (See TABLE 3-2)
				PROTECT	IF POSITION IS OUTSIDE VIRTUAL DISPLAY, COMMAND IS IGNORED; OTHERWISE, CURSOR MOVES TO DESIGNATED POSITION. IF POSITION CONTAINS A PROTECT FAC OR IS PART OF A PROTECTED FIELD, CURSOR IS MOVED RIGHT TO NON-PROTECTED COLUMN. IF POSITION CONTAINS A NON-PROTECT FAC, CURSOR REMAINS AT DESIGNATED POSITION; HOWEVER, ONLY FACS MAY BE INSERTED AT THIS POSITION AND A CURSOR RIGHT COMMAND SHOULD FOLLOW TO ENSURE CURSOR POSITIONING AT A NON-PROTECTED COLUMN.			
SET SCROLL MODE	C8	ESC,H	VIRTUAL	SCROLL	IGNORED	1	2	COMMAND BYTE(S)
				PAGE, PROTECT	PAGE AND PROTECT MODES ARE RESET, SCROLL MODE IS SET, CURSOR RETURNS TO HOME POSITION.			
ENABLE KEYBOARD	D2	ESC,R	N.A.	SCROLL, PAGE, PROTECT	CAUSES KEYBOARD TO BE ENABLED (DATA CAN BE ENTERED FROM KEYBOARD).	1	2	COMMAND BYTE(S)

TABLE 3-4. Extended Display Feature Communication Interface Commands (cont'd)

COMMAND	INTERFACE VALUE		DISPLAY AREA	MODE	ACTION/EFFECT	BYTE SEQUENCE		
	NON-ESC.	ESC SEQ.				NO. OF BYTES		DESCRIPTION
						NON-ESC.	ESC.SEQ.	
DISABLE KEYBOARD	D2	ESC,S	N.A.	SCROLL, PAGE, PROTECT	CAUSES KEYBOARD TO BE DISABLED (DATA CANNOT BE ENTERED FROM THE KEYBOARD).	1	2	COMMAND BYTE(S)
BOTTOM DISPLAY LINE	CA	ESC,J	ABSOLUTE	SCROLL, PAGE, PROTECT	THE VIRTUAL DISPLAY AREA'S LAST LINE IS MADE TO CORRESPOND WITH THE LINE DESIGNATED (SEE NOTE 3).	2	3	COMMAND BYTE(S), ONE BYTE DESIGNATING THE LINE (SEE TABLE 3-3).
TAB	DA	ESC,Z	VIRTUAL	SCROLL	CURSOR MOVES RIGHT TO NEXT TAB POSITION (WITHOUT LINE ROLLOVER).	1	2	COMMAND BYTE(S)
				PAGE	CURSOR MOVES RIGHT TO NEXT TAB POSITION (WITH LINE AND PAGE ROLLOVER) AS DEFINED BY THE SET TABS COMMAND.			
				PROTECT	CURSOR MOVES RIGHT TO LEFT-MOST COLUMN OF NEXT NON-PROTECTED FIELD (WITH LINE AND PAGE ROLLOVER).			
BACK TAB	DB	ESC,[	VIRTUAL	SCROLL	CURSOR MOVES LEFT TO PREVIOUS TAB POSITION (WITHOUT LINE ROLLOVER).	1	2	COMMAND BYTE(S)
				PAGE	CURSOR MOVES LEFT TO PREVIOUS TAB POSITION (WITH LINE AND PAGE ROLLOVER).			
				PROTECT	CURSOR MOVES LEFT TO LEFT-MOST COLUMN OF PREVIOUS NON-PROTECTED FIELD (WITH LINE AND PAGE ROLLOVER).			
SET TABS	DC	ESC,	VIRTUAL	PAGE, SCROLL, PROTECT	THE BLOCK OF DATA FOLLOWING THE COMMAND BYTE DEFINES THE TAB POSITIONS. THE TAB POSITIONS ARE INDICATED BY HTS (09) TO SET A TAB OR CANCEL (18) TO CLEAR A TAB, AND ARE SEPARATED BY SPACE (20) CHARACTERS TO INDICATE TAB SPACE (SETTING). THE BLOCK IS TERMINATED BY A LINE FEED CHARACTER. THE COMMAND BYTE FOLLOWED BY LF WITH NO SPACES, CANCELS OR HTS WILL CAUSE ALL TABS TO BE RESET.	VARIABLE	VARIABLE	COMMAND BYTE(S) SPACES, HTS (09), OR CANCEL (18), LF

TABLE 3-4. Extended Display Feature Communication Interface Commands (cont'd)

COMMAND	INTERFACE VALUE		DISPLAY AREA	MODE	ACTION/EFFECT	BYTE SEQUENCE		
						NO. OF BYTES		DESCRIPTION
	NON-ESC.	ESC SEQ.				NON-ESC.	ESC.SEQ.	
SET TRANSPARENT MODE	EC	ESC,l	ABSOLUTE	SCROLL, PAGE, PROTECT	SETS TRANSPARENT MODE OPERATION; CURSOR DOES NOT CHANGE POSITION.	1	2	COMMAND BYTE(S)
RESET TRANSPARENT MODE	ED	ESC,m	ABSOLUTE	SCROLL, PAGE, PROTECT	RESETS TRANSPARENT MODE OPERATION; CURSOR DOES NOT CHANGE POSITION.	1	2	COMMAND BYTE(S)
SET VIDEO INVERT	EE	ESC,n	ABSOLUTE	SCROLL, PAGE, PROTECT	SETS VIDEO INVERT OPERATION; CURSOR DOES NOT CHANGE POSITION.	1	2	COMMAND BYTE(S)
RESET VIDEO INVERT	EF	ESC,o	ABSOLUTE	SCROLL, PAGE, PROTECT	RESETS VIDEO INVERT OPERATION; CURSOR DOES NOT CHANGE POSITION.	1	2	COMMAND BYTE(S)
SET DISPLAY SPECIAL CHARACTERS	FC	ESC,:	N.A.	SCROLL, PAGE, PROTECT	PERMITS SPECIAL CHARACTER TO BE DISPLAYED.	1	2	COMMAND BYTE(S)
RESET DISPLAY SPECIAL CHARACTERS	FD	ESC,}	N.A.	SCROLL, PAGE, PROTECT	SPECIAL CHARACTERS CANNOT BE DISPLAYED.			
END-OF-LINE FAC	FE	ESC,~	N.A.	SCROLL, PAGE, PROTECT	AN OUTPUT COMMAND USED TO MARK UNUSED SPACE IN LINE. LINE LENGTH DETERMINED BY RIGHT-MOST DATA BYTE POSITION. REMAINING UNUSED, UNFILLED POSITIONS TO 80TH COLUMN ARE NOT CONSIDERED PART OF LINE.	1	2	COMMAND BYTE(S)
TERMINAL RESET	F1	ESC,q	ABSOLUTE	SCROLL, PAGE, PROTECT	CAUSES THE BASIC DISPLAY UNIT TO BE RESET AS IF THE "RESET" SWITCH HAD BEEN ACTIVATED.	1	2	COMMAND BYTE(S)

TABLE 3-4. Extended Display Feature Communication Interface Commands (cont'd)

COMMAND	INTERFACE VALUE		DISPLAY AREA	MODE	ACTION/EFFECT	BYTE SEQUENCE		DESCRIPTION
						NO. OF BYTES		
	NON-ESC.	ESC SEQ.				NON-ESC.	ESC.SEQ.	
DISPLAY STATUS ON	F2	ESC,r	ABSOLUTE	SCROLL, PAGE, PROTECT	CERTAIN STATUS COMMANDS CAN BE RECEIVED ON THE COMMUNICATIONS INTERFACE AND DISPLAYED. THE LAST THREE POSITIONS IN THE ABSOLUTE DISPLAY AREA ARE USED TO SHOW A BLINKING STATUS INDICATOR. THE INDICATORS AND THEIR ASSOCIATE CODES ARE: INDICATOR CODE FUNCTION PE 98 PARITY ERROR RO 9A RECEIVE OVERRUN FE 9B FRAMING ERROR FRAMING AND PARITY ERROR COMMANDS ARE FOLLOWED BY THE BYTE SUFFERING THE ERROR CONDITION.	1	2	COMMAND BYTE(S)
DISPLAY STATUS OFF	F3	ESC,s	ABSOLUTE	SCROLL, PAGE, PROTECT	RECEIVED STATUS INDICATORS PE, RO, and FE CANNOT BE DISPLAYED.	1	2	COMMAND BYTE(S)
ASCII BEL (BELL)	07	NONE	N.A.	SCROLL, PAGE, PROTECT	CAUSES THE BELL TO ISSUE TONE FOR A PERIOD OF ONE-HALF SECOND.	1		COMMAND BYTE
ASCII BS (BACKSPACE)	08	NONE	VIRTUAL	SCROLL	CURSOR MOVES LEFT ONE COLUMN WITHOUT LINE ROLLOVER.	1		COMMAND BYTE
				PAGE	CURSOR MOVES LEFT ONE COLUMN WITH LINE AND PAGE ROLLOVER.			
				PROTECT	CURSOR MOVES LEFT ONE COLUMN WITH LINE AND PAGE ROLLOVER. SEEKS NON-PROTECTED COLUMN TO THE LEFT.			

TABLE 3-4. Extended Display Feature Communication Interface Commands (cont'd)

COMMAND	INTERFACE VALUE		DISPLAY AREA	MODE	ACTION/EFFECT	BYTE SEQUENCE		
						NO. OF BYTES		DESCRIPTION
	NON-ESC.	ESC SEQ.				NON-ESC.	ESC.SEQ.	
ASCII HT (HORIZONTAL TAB)	09	NONE	VIRTUAL	SCROLL	CURSOR MOVES RIGHT TO NEXT TAB POSITION (WITHOUT LINE ROLLOVER) AS DEFINED BY THE SET TABS COMMAND.	1		COMMAND BYTE
				PAGE	CURSOR MOVES RIGHT TO NEXT TAB POSITION (WITH LINE AND PAGE ROLLOVER) AS DEFINED BY THE SET TABS COMMAND.			
				PROTECT	CURSOR MOVES RIGHT TO THE LEFT-MOST COLUMN OF THE NEXT NON-PROTECTED FIELD (WITH LINE AND PAGE ROLLOVER).			
SET PAGE MODE	C7	ESC,G	VIRTUAL	SCROLL	SCROLL MODE IS RESET, PAGE MODE IS SET, CURSOR RETURNS TO HOME POSITION.	1	2	COMMAND BYTE(S)
				PAGE	IGNORED			
				PROTECT	PROTECT MODE IS RESET, PAGE MODE IS SET, CURSOR RETURNS TO HOME POSITION.			
TOP DISPLAY LINE	C9	ESC,I	ABSOLUTE	SCROLL, PAGE, PROTECT	THE VIRTUAL DISPLAY AREA'S TOP LINE IS MADE TO CORRESPOND WITH THE LINE DESIGNATED (SEE NOTE 3)	2	3	COMMAND BYTE(S), ONE BYTE DESIGNATING THE LINE (SEE TABLE 3-3)
READ CURSOR	C6	ESC,F	VIRTUAL	SCROLL, PAGE, PROTECT	4 BYTES ARE SENT ON THE INTERFACE - START DATA COMMAND, TWO BYTE CURSOR POSITION, END DATA COMMAND.	1	2	COMMAND BYTE(S)
ASCII LF (LINE FEED)	0A	NONE	VIRTUAL	SCROLL	CURSOR MOVES DOWN ONE LINE IN SAME COLUMN. IF ALREADY IN LAST LINE, SCROLL IS EXECUTED AND LAST LINE IS BLANKED.	1		COMMAND BYTE
				PAGE	CURSOR MOVES DOWN ONE LINE IN SAME COLUMN, WITH PAGE ROLLOVER.			
				PROTECT	CURSOR MOVES DOWN ONE LINE IN SAME COLUMN WITH PAGE ROLLOVER. SEEKS [illegible]ROTECTED COLUMN TO THE R[illegible]			

TABLE 3-4. Extended Display Feature Communication Interface Commands (cont'd)

COMMAND	INTERFACE VALUE		DISPLAY AREA	MODE	ACTION/EFFECT	BYTE SEQUENCE		
	NON-ESC.	ESC SEQ.				NO. OF BYTES		DESCRIPTION
						NON-ESC.	ESC.SEQ.	
ASCII VT (VERTICAL TAB)	0B	NONE	VIRTUAL	SCROLL	IGNORED	1		COMMAND BYTE
				PAGE	CURSOR MOVES UP ONE LINE IN SAME COLUMN WITH PAGE ROLLOVER.			
				PROTECT	CURSOR MOVES UP ONE LINE IN SAME COLUMN WITH PAGE ROLLOVER. SEEKS NON-PROTECTED COLUMN TO THE LEFT.			
ASCII FF (FORM FEED)	0C	NONE	VIRTUAL	SCROLL	CURSOR MOVES RIGHT ONE COLUMN WITHOUT LINE ROLLOVER.	1		COMMAND BYTE
				PAGE	CURSOR MOVES RIGHT ONE COLUMN WITH LINE AND PAGE ROLLOVER.			
				PROTECT	CURSOR MOVES RIGHT ONE COLUMN WITH LINE AND PAGE ROLLOVER. SEEKS NON-PROTECTED COLUMN TO THE RIGHT.			
ASCII CR (CARRIAGE RETURN)	0D	NONE	VIRTUAL	SCROLL, PAGE	CURSOR MOVES TO LEFT MARGIN.	1		COMMAND BYTE
				PROTECT	CURSOR MOVES TO LEFT MARGIN. SEEKS NON-PROTECTED COLUMN TO THE RIGHT.			
LEFT DISPLAY COLUMN	CB	ESC,K	ABSOLUTE	SCROLL, PAGE, PROTECT.	THE VIRTUAL DISPLAY AREA'S LEFT COLUMN IS MADE TO CORRESPOND WITH THE COLUMN DESIGNATED (SEE NOTE 3).	2	3	COMMAND BYTE(S), ONE BYTE COLUMN (SEE TABLE 3-3)
RIGHT DISPLAY COLUMN	CC	ESC,L	ABSOLUTE	SCROLL, PAGE, PROTECT	THE VIRTUAL DISPLAY AREA'S RIGHT COLUMN IS MADE TO CORRESPOND WITH THE COLUMN DESIGNATED (SEE NOTE 3).	2	3	COMMAND BYTE(S), ONE BYTE COLUMN (SEE TABLE 3-3)
SET PROTECT MODE	CD	ESC,M	VIRTUAL	SCROLL	SCROLL MODE IS RESET, PROTECT MODE IS SET.	1	2	COMMAND BYTE(S)
				PAGE	PAGE MODE IS RESET, PROTECT MODE IS SET.			
				PROTECT	IGNORED - CURSOR SEEKS FIRST NON-PROTECTED COLUMN ON SCREEN. SIMULATES "HOM " COMMAND.			

TABLE 3-4. Extended Display Feature Communication Interface Commands (cont'd)

COMMAND	INTERFACE VALUE		DISPLAY AREA	MODE	ACTION/EFFECT	BYTE SEQUENCE		
	NON-ESC.	ESC SEQ.				NO. OF BYTES		DESCRIPTION
						NON-ESC.	ESC.SEQ.	
WRITE ABSOLUTE	CE	ESC,N	ABSOLUTE	SCROLL, PAGE, PROTECT	THE BLOCK OF DATA OR COMMAND BYTES ARE WRITTEN INTO THE ABSOLUTE DISPLAY AREA STARTING AT THE ABSOLUTE CURSOR POSITION DESIGNATED. THE COMMAND BYTES ARE RESTRICTED TO THE SET OF FIELD ATTRIBUTE COMMANDS. THE CURSOR DOES NOT CHANGE POSITION.	VARIABLE	VARIABLE	COMMAND BYTE(S), START DATA COMMAND, 2 BYTE CURSOR POSITION, VARIABLE DATA OR COMMAND BYTES, END DATA COMMAND
READ ABSOLUTE	CF	ESC,O	ABSOLUTE	SCROLL, PAGE, PROTECT	THE ABSOLUTE DISPLAY POSITIONS, STARTING WITH THE 1ST CURSOR POSITION IN THE COMMAND SEQUENCE AND ENDING WITH THE 2ND CURSOR POSITION ARE SENT OUT ON THE INTERFACE. THIS BLOCK OF DATA IS FRAMED WITH THE START DATA AND END DATA COMMANDS. EACH END OF LINE FROM THE DISPLAY AREA IS FLAGGED BY AN END-OF-DATA (F9;ESC,y) CHARACTER.	5	6	COMMAND BYTE(S), 2 BYTE CURSOR POSITION - "START", 2 BYTE CURSOR POSITION - "END"
CHARACTER INSERT	DO	ESC,P	VIRTUAL	SCROLL	IGNORED	3	5	COMMAND BYTE(S), DATA BYTES(S), COMMAND BYTE(S) (SEE NOTE 4)
				PAGE	ALL COLUMNS IN LINE, STARTING WITH CURSOR COLUMN, ARE MOVED RIGHT ONE COLUMN. THE BYTE IN THE RIGHT-MOST COLUMN IS DISCARDED. THE DATA BYTE IS PLACED AT THE CURSOR AND THE CURSOR IS ADVANCED ONE COLUMN TO THE RIGHT WITH LINE ROLLOVER.			
				PROTECT	SAME AS PAGE, EXCEPT MOVEMENT RESTRICTED TO SEQUENCE OF NON-PROTECTED COLUMNS TO THE RIGHT OF CURSOR POSITION. (CONTIGUOUS NON-PROTECTED FIELD).			

TABLE 3-4. Extended Display Feature Communication Interface Commands (cont'd)

COMMAND	INTERFACE VALUE		DISPLAY AREA	MODE	ACTION/EFFECT	BYTE SEQUENCE		
						NO. OF BYTES		DESCRIPTION
	NON-ESC.	ESC SEQ.				NON-ESC.	ESC.SEQ.	
CHARACTER DELETE	D1	ESC,Q	VIRTUAL	SCROLL	IGNORED	1	2	COMMAND BYTE(S)
				PAGE	ALL COLUMNS IN LINE, STARTING WITH CURSOR COLUMN +1 ARE MOVED TO THE LEFT ONE COLUMN. THE RIGHT-MOST COLUMN IS FILLED WITH A SPACE (20) CHARACTER. CURSOR DOES NOT CHANGE POSITION.			
				PROTECT	SAME AS PAGE, EXCEPT THE NON-PROTECTED COLUMNS UNDER AND CONTIGUOUS TO THE RIGHT WITHIN THE NON-PROTECTED FIELD ARE MOVED.			
PAGE ERASE	D4	ESC,T	VIRTUAL	SCROLL, PAGE	ERASES ALL CHARACRERS FROM CURSOR TO END OF DISPLAY AND REPLACES WITH SPACE (20) CHARACTER. CURSOR DOES NOT CHANGE POSITION.	1	2	COMMAND BYTE(S)
				PROTECT	ERASES ALL UNPROTECTED CHARACTER POSITIONS FROM THE CURSOR TO THE END OF DISPLAY AND REPLACES WITH SPACE (20) CHARACTER. CURSOR DOES NOT CHANGE POSITION.			
LINE ERASE	D5	ESC,U	VIRTUAL	SCROLL, PAGE,	STARTING WITH CURSOR POSITION, LINE IS BLANKED TO THE RIGHT WITH SPACE (20) CHARACTER. CURSOR DOES NOT CHANGE POSITION.	1	2	COMMAND BYTE(S)
				PROTECT	ALL COLUMNS IN NON-PROTECTED FIELD UNDER AND CONTIGUOUS RIGHT OF CURSOR ARE BLANKED. CURSOR NOT CHANGE POSITION.			
LINE INSERT	D6	ESC,V	VIRTUAL	SCROLL, PROTECT	IGNORED	1	2	COMMAND BYTE(S)
				PAGE	ALL LINES, STARTING WITH CURRENT LINE, ARE MOVED DOWN ONE LINE. LAST LINE IS DISCARDED. CURRENT LINE IS BLANKED WITH SPACE (20) CHARACTER. CURSOR RESTS IN FIRST POSITION OF NEW BLANK LINE.			

TABLE 3-4. Extended Display Feature Communication Interface Commands (cont'd)

COMMAND	INTERFACE VALUE		DISPLAY AREA	MODE	ACTION/EFFECT	BYTE SEQUENCE		DESCRIPTION
						NO. OF BYTES		
	NON-ESC.	ESC SEQ.				NON-ESC.	ESC.SEQ.	
SEND PAGE	D9	ESC,Y	VIRTUAL	SCROLL	IGNORED	1	2	COMMAND BYTE(S)
				PAGE	ALL POSITIONS FROM HOME THROUGH CURSOR POSITION, OR END OF LINE FAC ON THE LINE ON WHICH CURSOR RESTS (WHICHEVER IS FOUND FIRST), ARE SENT ON THE INTERFACE AS A BLOCK OF DATA. THIS BLOCK IS FRAMED BY THE START DATA COMMAND AND THE END DATA COMMAND. EACH END OF LINE IS FLAGGED BY AN END-OF DATA (F9;ESC,y) CHARACTER. CURSOR DOES NOT CHANGE POSITION. NOTE: ANY PROTECTED FIELDS THAT MAY BE SENT IN THIS MODE WILL BE BRACKETED BY FIELD ATTRIBUTE CODES TO INDICATE START AND END OF PROTECTED FIELDS.			
				PROTECT	ALL NON-PROTECTED POSITIONS FROM HOME THROUGH END OF PAGE ARE SENT ON THE INTERFACE AS A BLOCK OF DATA. THIS BLOCK IS FRAMED AS IN PAGE MODE. EACH END OF FIELD IS FLAGGED BY AN END-OF-DATA (F9;ESC,y) CHARACTER. CURSOR DOES NOT CHANGE POSITION.			
LINE DELETE	D7	ESC,W	VIRTUAL	SCROLL, PROTECT	IGNORED	1	2	COMMAND BYTE(S)
				PAGE	ALL LINES, STARTING WITH LINE DIRECTLY BELOW CURRENT LINE, ARE MOVED UP ONE LINE. LAST LINE IS BLANKED. CURSOR MOVES TO FIRST CHARACTER POSITION OF LINE IN WHICH IT INITIALLY RESTED.			

TABLE 3-4. Extended Display Feature Communication Interface Commands (cont'd)

COMMAND	INTERFACE VALUE		DISPLAY AREA	MODE	ACTION/EFFECT	BYTE SEQUENCE		
						NO. OF BYTES		DESCRIPTION
	NON-ESC.	ESC SEQ.				NON-ESC.	ESC.SEQ.	
CLEAR	D8	ESC,X	VIRTUAL	SCROLL, PAGE	ALL POSITIONS IN DISPLAY AREA ARE FILLED WITH SPACES (20). CURSOR MOVES TO HOME POSITION.	1	2	COMMAND BYTE(S)
				PROTECT	ALL NON-PROTECTED POSITIONS IN DISPLAY AREA ARE FILLED WITH SPACES (20). CURSOR MOVES TO LEFT-MOST POSITION OF FIRST NON-PROTECTED FIELD OR, IF NO NON-PROTECTED FIELD, MOVES TO HOME POSITION.			
ENABLE LOAD	FA	ESC,z	N.A.	SCROLL, PAGE, PROTECT	THIS COMMAND CAUSES THE NEXT OCCURRENCE OF AN ASCII CONTROL CHARACTER "DC2" (12) TO SIGNAL THE START OF LOAD RECORDS AS DESCRIBED FOR THE EXBUG MEMORY LOAD FUNCTION. (See: M6800 EXORciser User's Guide Figure 4-1.) ALSO CAUSES OCCURRENCE OF AN ASCII CONTROL CHARACTER "DC4" (14) TO SIGNAL THE END OF LOAD RECORDS. (SEE NOTE 5).	1	2	COMMAND BYTE(S)
DISABLE LOAD	FB	ESC,{	N.A.	SCROLL, PAGE, PROTECT	THIS COMMAND CAUSES THE ASCII CONTROL CHARACTERS "DC2" AND "DC4" TO BE TREATED IN THE NORMAL MANNER.	1	2	COMMAND BYTE(S)
START DATA	DD	ESC,]	VIRTUAL, ABSOLUTE	SCROLL, PROTECT, PAGE	THIS COMMAND BYTE IS USED TO FRAME BLOCKS OF DATA AT THE BEGINNING AND PRECEDES THE BYTE.	1	2	COMMAND BYTE(S)
END DATA	DE	ESC,^	VIRTUAL, ABSOLUTE	SCROLL, PAGE, PROTECT	THIS COMMAND BYTE IS USED TO FRAME BLOCKS OF DATA AT THE END AND FOLLOWS THE LAST DATA BYTE.	1	2	COMMAND BYTE(S)
END-OF-DATA INDICATOR	F9	ESC,y	N.A.	PAGE, PROTECT	OUTPUT CHARACTER(S) IDENTIFYING END OF LINE (PAGE MODE) OR END OF UNPROTECTED FIELD (PROTECT MODE). USED ONLY IN SEND PAGE, SEND LINE, AND READ ABSOLUTE COMMANDS.			

TABLE 3-4. Extended Display Feature Communication Interface Commands (cont'd)

COMMAND	INTERFACE VALUE		DISPLAY AREA	MODE	ACTION/EFFECT	BYTE SEQUENCE		
	NON-ESC.	ESC SEQ.				NO. OF BYTES		DESCRIPTION
						NON-ESC.	ESC.SEQ.	
SEND LINE	DF	ESC,_	VIRTUAL	SCROLL	IGNORED	1	2	COMMAND BYTE(S)
				PAGE	ALL POSITIONS FROM BEGINNING OF THE LINE ON WHICH CURSOR RESTS THROUGH END OF LINE ARE SENT ON THE INTERFACE AS A BLOCK OF DATA. THIS BLOCK IS FRAMED BY THE START DATA COMMAND AND THE END DATA COMMAND. CURSOR DOES NOT CHANGE POSITION. ANY PROTECTED FIELDS THAT MAY BE SENT IN THIS MODE WILL BE BRACKETED BY FIELD ATTRIBUTE CODES TO INDICATE START AND END OF PROTECTED FIELDS.			
				PROTECT	ALL NON-PROTECTED POSITIONS FROM BEGINNING OF LINE ON WHICH CURSOR RESTS THROUGH END OF THE LINE ARE SENT ON THE INTERFACE AS A BLOCK OF DATA. THIS BLOCK IS FRAMED AS IN PAGE MODE. EACH END OF FIELD IS FLAGGED BY AN END-OF-DATA (F9;ESC,y) CHARACTER. CURSOR DOES NOT CHANGE POSITION.			

TABLE 3-4. Extended Display Feature Communication Interface Commands (cont'd)

COMMAND	INTERFACE VALUE		DISPLAY AREA	MODE	ACTION/EFFECT	BYTE SEQUENCE		
						NO. OF BYTES		DESCRIPTION
	NON-ESC.	ESC SEQ.				NON-ESC.	ESC.SEQ.	
FIELD ATTRIBUTES -			VIRTUAL ABSOLUTE	SCROLL, PROTECT, PAGE	THESE COMMANDS ARE USED TO DEFINE ATTRIBUTES IN THE DISPLAY AREA. AS SUCH, THEY SET OR RESET THE ATTRIBUTES FOR THE CURSOR PROMPT POSITION. CONTIGUOUS ATTRIBUTE COMMANDS WILL SET (OR RESET) THOSE ATTRIBUTES FOR A SINGLE PROMPT POSITION. CURSOR POSITION IS NOT CHANGED BY A FIELD ATTRIBUTE COMMAND.	1	2	COMMAND BYTE(S)
SET BLINK	E0	ESC,`						
RESET BLINK	E1	ESC,a						
SET VIDEO INVERT	E2	ESC,b						
RESET VIDEO INVERT	E3	ESC,c						
SET HALF BRIGHT	E4	ESC,d						
RESET HALF BRIGHT	E5	ESC,e						
SET UNDERLINE	E6	ESC,f						
RESET UNDERLINE	E7	ESC,g						
SET NON-DISPLAY	E8	ESC,h						
SET DISPLAY	E9	ESC,i						
SET FIELD PROTECT	EA	ESC,j						
RESET FIELD PROTECT	EB	ESC,k						

NOTES:

1. All non-escape interface values are in hexadecimal.
2. All escape interface values are ASCII characters.
3. After executing each of the virtual display boundary commands (C9, CA, CB, CC), the cursor is checked for being within the virtual display. If it is not, it is moved to the right until it is and if Protect mode is set, it seeks a non-protected column to the right.
4. The second character insert command byte terminates the Character Insert mode. Any other command will also terminate this mode, but will not be executed.
5. Following a "DC2" control character, an improperly formatted load record or the presence of a command byte will cause this command to be aborted. The occurrence of any character other than a NULL, Line Feed, or Carriage Return embedded between load records will cause this command to be aborted.

next highest priority will execute. The lowest priority process, IDLE, executes only when there is nothing else to do, and it never suspends. The following resident process modules have been implemented in the Basic Display Unit:

Display Control Software Process

- Executes commands to position cursor
- Executes generic display attribute commands
- Scrolls display area
- Executes screen splitting commands

Keyboard Control Software Process

- Accepts encoded characters from keyboard
- Allows application module interpretation of key value (optional)
- Queues key value for further processing

Communication Control Software Process

- Accepts data for transmission
- Queues input data from communications interface to Display Control Software Process Module

Terminal Control Software Process

- Reads Control Switches
- Queues commands to Display Control software process and Communications software process modules

Monitor Control Software Process

- Executes Monitor Operator Calls (MOCS) to facilitate queue manipulation, etc.
- Controls execution of all other modules on a priority basis

Idle Software Process

- Executes only when there is nothing else to do, and it never suspends

System Initialization and Tone Process

- Defaults the Basic Display Unit to its initial state when powered on or if a reset occurs
- Executes the audio alarm command (BELL)

Four interrupt handlers are supported by the Basic Display Unit:

- Power On - causes system initialization to be executed.
- Non-maskable interrupts - only returns from interrupts, currently does nothing.
- Software Interrupts - Monitor performs designated function and, if necessary, reschedules active processes.
- Internal Interrupts - from PIAs and ACIA. Each interrupt handler is given an opportunity to ascertain whether they have something to do on this interrupt occasion.

- The initial state of the Basic Display Unit is as defined in paragraph 3.3.1.
- The initialized Queues and Queue descriptors are:

Queue	Size	Name
Communication Link Input	32 Bytes	QCLID
Communication Link Output	64 Bytes	QCLOD
Scroll Input	256 Bytes	QSCRL
Keyboard Output	16 Bytes	QKEY

3.5.1.1 Monitor Control Software Process Module Functions

The following information describes the Monitor Control Process Module:

- System Initialization. Following a power on interrupt or system reset, the Monitor executes a system initialization module which operates as a portion of the System Initialization and Tone process. At the time the monitor leaves its initialization routine, all standard processes have been established and their stacks initialized. All PIAs and ACIAs are configured with properly formatted data direction and control registers.
- Service All Software Interrupts (Monitor Operator Calls). The general MOCS interface requires that a process or interrupt handler load specific values in its register(s) and execute a simulated Software Interrupt Routine (JSR SIMSWI), which is followed directly by the MOCS number (see Figure 3-3). Upon return from the Monitor, the process or interrupt handler should examine the carry indicator (if used by the MOCS) to ascertain what action took place or if any results are in its register. The following list contains the MOCS number, its function and code, and what is input and output from the queue. In the event that the user desires to create his own MOCS, the address of the processing routine must be stored into location "UMOCS". To be recognized as a user MOCS, the byte following the JSR SIMSWI instruction must be greater than decimal 16 (HEX 10), less than decimal 15 (HEX 4B), or negative. If the user processes the MOC in his routine, exiting should be done by an RTI instruction. If the user doesn't wish to process the MOC, he should exit by using an RTS instruction. The user should avoid introducing MOC routines into the system that may interfere with normal MONITOR functions (i.e., activate or suspend processes). The results will be unpredictable.

1. Get Byte from Queue (MOCGBQ)

   Inputs - X Register = Address of Queue Descriptor

   Outputs - A Register = Data Byte if Carry Bit Set
   Carry Bit Clear if Queue is empty

2. Place Byte and Lock (MOCPBL)

   Inputs - X Register = Address of Queue Descriptor

   A Register - Data Byte

   Outputs- Carry Bit Set if good placement and Queue is locked
   Carry Bit Clear if Queue Over-run of queue already locked.

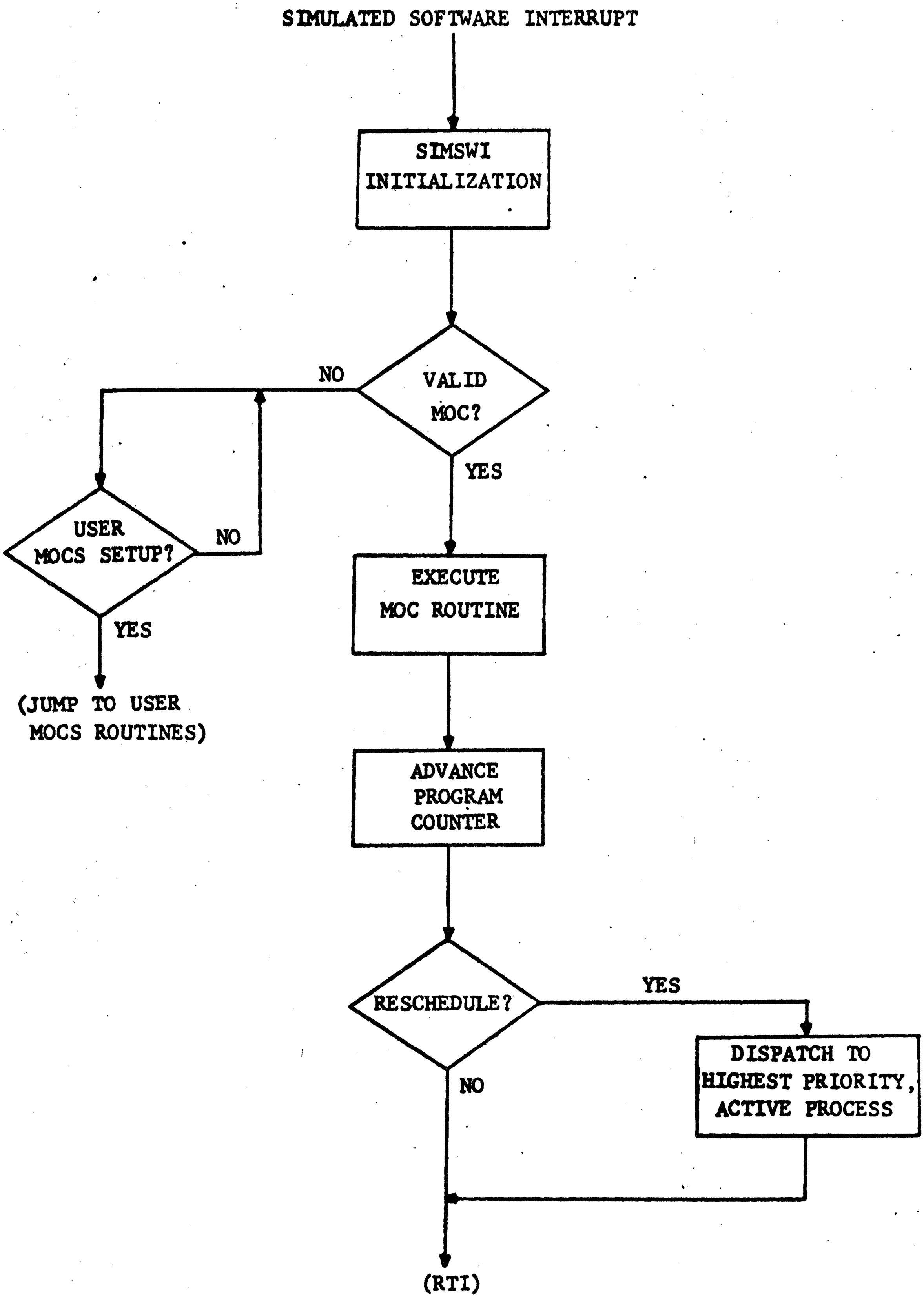


FIGURE 3-3. Software Interrupt Handler Flow Chart

3. Place Byte in Queue (MOCPBQ)

   Inputs - X Register = Address of Queue Descriptor

   A Register = Data Byte

   Outputs- Carry Bit Set if good placement
   Carry Bit Clear if Queue Over-run or queue locked

4. Place Byte and Unlock (MOCPBU)

   Inputs - X Register = Address of Queue Descriptor

   A Register = Data Byte

   Outputs- Carry Bit Set if good placement and Queue is unlocked
   Carry Bit Clear if Queue Over-run

5. Place Byte and Activate (MOCPBA)

   Inputs - X Register = Address of Queue Descriptor

   A Register = Data Byte

   Outputs- Carry Bit Set if good placement and Service Process has been activated.
   Carry Bit Clear if queue over-run

6. Get Byte, Activate if Empty (MOCGBE)

   Inputs - X Register = Address of Queue Descriptor

   Outputs- A Register = Data Byte if Carry Bit is Set.
   Carry Bit Clear and Queue filling routine activated if Queue is empty

7. Activate Process N (MOCACT)

   Inputs - A Register = Process Number

   Outputs- Not Applicable

8. Suspend Process N (MOCSUS)

   Inputs - A Register = Process Number

   Outputs- Not Applicable

9. Activate Process N, Suspend ME (MOCACM)

   Inputs - A Register = Process Number

   Outputs- Not Applicable

10. Delay Me for N*27.5 Milliseconds (MOCDLY)

    Inputs - A Register = N

    Outputs- Carry Bit Set if Time ran out
    Carry bit clear if activated for some other reason.

11. Setup Interrupt Handler (MOCIIH)

    Inputs - X Register = Address of the Interrupt Handler

    Outputs - Carry Bit Set if Interrupt Handler Established
    Carry Bit Clear if Interrupt Handler not Established

12. Reset My Timer (MOCRST)

Inputs - None

Outputs - None

13. Activate Co-Process N (MOCACP)

Inputs - X Register = Address of Register Vector

A Register = N

Outputs- Carry Bit Set if Process Activated
Carry Bit Clear if Process Not Activated

Where X contains the address of the following structure:

Byte #	Meaning
1-2	New Stack Pointer Address for Process N
3	New Condition Code for Process N
4	New B Register value
5	New A Register value
6-7	New X Register value
8-9	New Program Counter

Upon re-entering execution, Process N will have these values in its Registers.

14. Suspend Me (MOCSUM)

Inputs - None

Outputs - Carry Bit Set if Process Suspended
Carry Bit Cleared if Process Not Suspended

15. Move/Fill N Bytes (MOCMOV)

Inputs - X Register = Address of Move Description

A Register = Constant Byte (If Used)

Outputs- Carry Bit Set if Move Completed
Carry Bit Cleared if Move Not Completed

Move Description

Byte #	Meaning
1-2	From Address, 2/=Ø, then contents of register A is used to fill Destination Area
3-4	Destination Address
5	Number of Bytes to Move or Fill

16. Terminal Reset (MOCRES)

Inputs - None

Outputs - None

- Activate and suspend processes. The Monitor reschedules the active process list so that the highest priority process will commence execution when the Monitor returns from the Software Interrupt (RTI). This means a process can pre-empt its own execution by activating a higher priority process.

- Maintain system timer with 27.5 millisecond granularity. A process can delay its own execution for N*27.5 milliseconds.

- Maintain circular queue descriptors. The format of each descriptor is as follows:

Byte #	Meaning
1-2	Address of first Byte in Queue Space - Right-most N Bits of Byte 2 must be Ø, where N is given by Byte 3.
3	An N bit mask value that gives the size of the queue as a Modulo to bytes 1 and 2. That is, a 256 byte queue has a mask value of $(FF)_{16}$ and byte 2 = Ø, whereas a 16 byte queue has a mask value of $(ØF)_{16}$ and byte 2 = $(XØ)_{16}$.
4	Relative Position of Last Byte Retrieved.
5	Relative Position of Next Byte to Hold New Element.
6	Queue Serving (Take from) Process Number
7	Queue filling Process Number
8-9	Next Function Address*
1Ø	Queue Lock Byte - Ø = Unlocked, 1 = Locked*

* The "next function" address is used by the resident firmware in process-multiple byte commands. NOTE: The contents of the queue descriptors should <u>not</u> be modified by an external (user) process.

- Maintain Interrupt Vectors. The interrupt transfer vectors are controlled and initialized by the Monitor. When an internal interrupt (PIA or ACIA) occurs (see Figure 3-4), the Monitor services its own PIAs and then executes a JSR to the other interrupt handlers, thus allowing them an opportunity to examine their PIAs or ACIAs. They return to the Monitor via an RTS. The starting address of the interrupt vectors available to the user for application programs are as follows:

Hex Address	Label	Function
0126	TKEYI	Keyboard Interpreter
0128	OCAI	Output Communications Application Interpreter
012A	ICAI	Input Communications Application Interpreter
012C	UMOCS	User MOC Routines
037A	QDCLOD	Communications Interface Output Queue Descriptor
0384	QDCLID	Communications Interface Input Queue Descriptor
DFF5	SIMSWI	Entry to Monitor Call (MOC) Processing
EFOA	PIA2BD	Peripheral Interface Adapter for spare switches A, B, and C located on the rear panel.

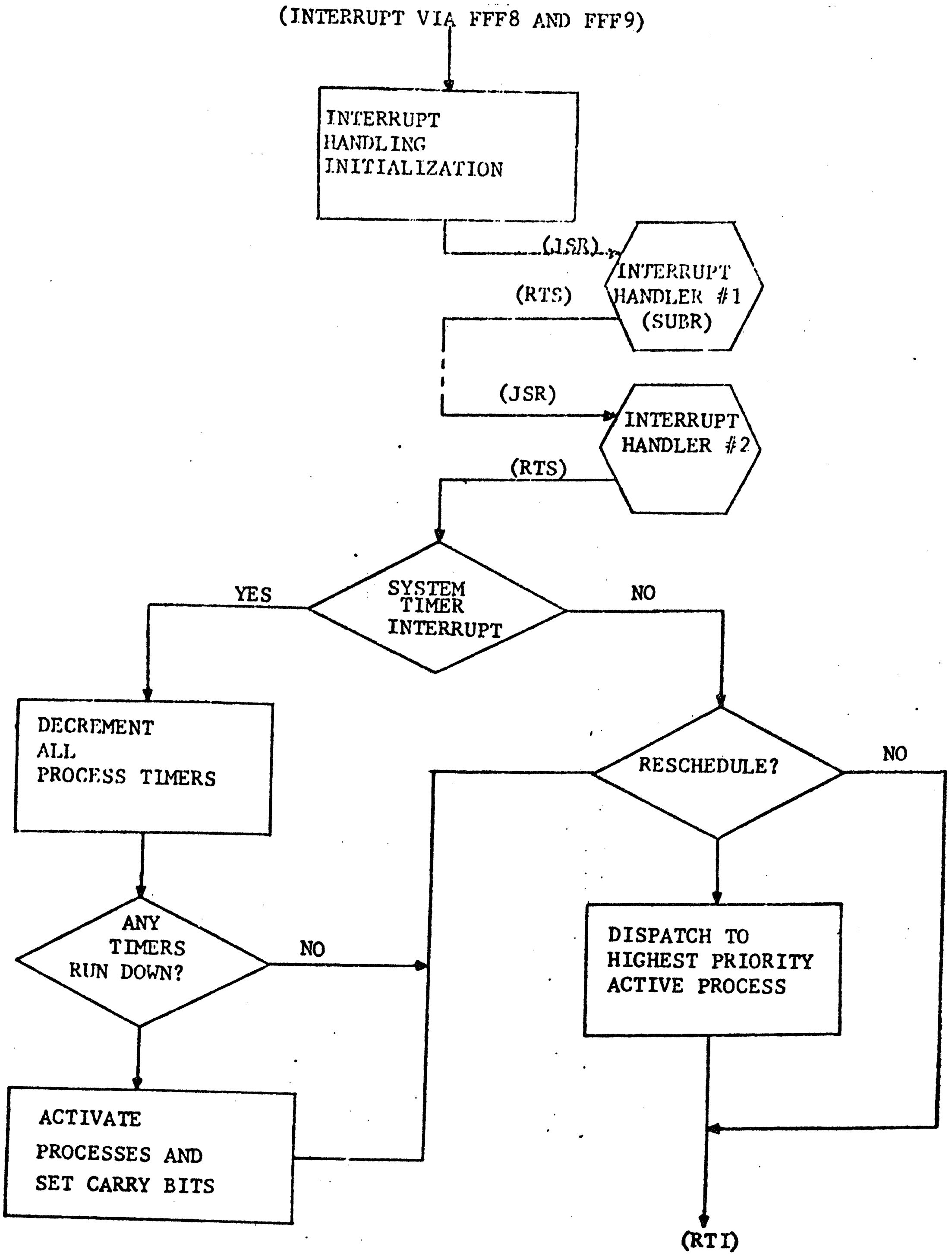

(INTERRUPT VIA FFF8 AND FFF9)
INTERRUPT HANDLING INITIALIZATION
(JSR)
INTERRUPT HANDLER #1 (SUBR)
(RTS)
(JSR)
INTERRUPT HANDLER #2
(RTS)
SYSTEM TIMER INTERRUPT
YES
NO
DECREMENT ALL PROCESS TIMERS
ANY TIMERS RUN DOWN?
NO
ACTIVATE PROCESSES AND SET CARRY BITS
RESCHEDULE?
NO
DISPATCH TO HIGHEST PRIORITY ACTIVE PROCESS
(RTI)

FIGURE 3-4. Internal Interrupt Handler Flow Chart

- The Initialization routines terminate by entering a Tone Process routine that immediately suspends. This routine is activated (as Process #1) when it is desirable to turn on the audible alarm for one-half second.

- All processes are established as active. When the Initialization/ Tone Process suspends, they are entered at their initial entry points, all in turn in order of priority, so that they may perform any local initialization that is desired.

3.5.1.2 Keyboard Control Software Process Module Functions

When an interrupt is received from a keyboard, the key value is read by the keyboard interrupt handler and queued in QKEY (see Figure 3-5). The keyboard process is then activated.

For each key value taken from QKEY, the application keyboard subroutine is called (if it is present). Upon return from this subroutine, the state of the carry condition is used to:

- Treat key value in usual manner - carry set.
- Ignore key stroke - carry cleared.

If the keyboard has been disabled, the key value is ignored.

The following commands are looked for and cause RAM switches to be set/reset:

- Auto Line Feed
- On-Line

The Break-shifted and non-shifted values are always queued to the communication process via QCLOD, regardless of mode setting (Scroll, Page, or Project).

A Page mode command from the keyboard will cause the following conditions if the Extended Display Feature is present. Otherwise, the key stroke is ignored.

a. If Protect mode is set, the mode will change to Page mode and the key will remain lit.

b. If Page mode is set, the mode will be changed to Scroll mode and the key lit will go out.

c. If Scroll mode is set, the mode will change to Page mode and the key will be lit.

In Scroll mode, the following translation is performed:

UP ARROW TO VERTICAL TAB
DOWN ARROW TO LINE FEED
LEFT ARROW TO BACK SPACE
RIGHT ARROW TO FORM FEED

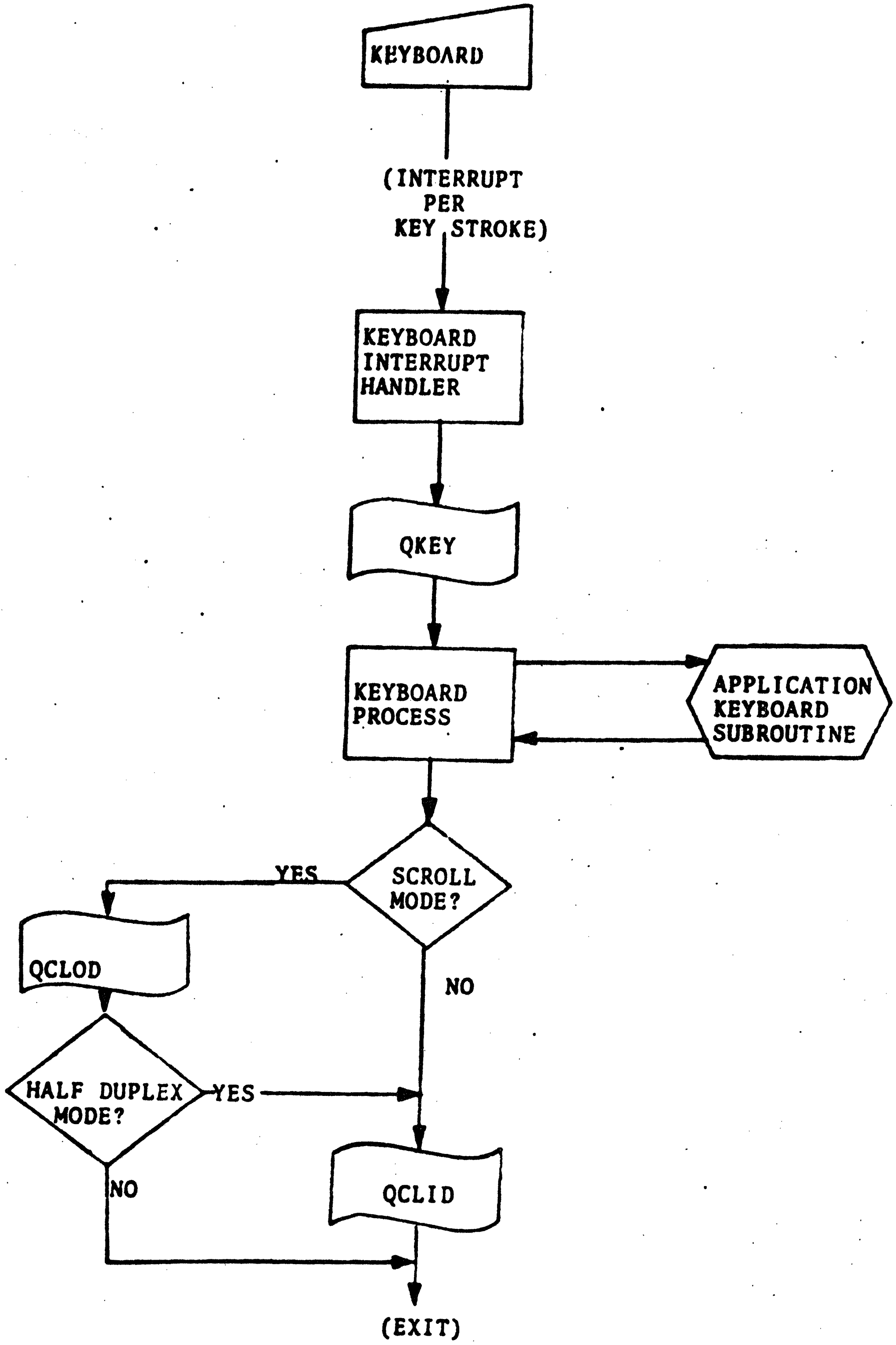

KEYBOARD
(INTERRUPT PER KEY STROKE)
KEYBOARD INTERRUPT HANDLER
QKEY
KEYBOARD PROCESS
APPLICATION KEYBOARD SUBROUTINE
SCROLL MODE?
YES
NO
QCLOD
HALF DUPLEX MODE?
YES
NO
QCLID
(EXIT)

FIGURE 3-5. Keyboard Routine Flow Chart

In Page mode, escape sequences are looked for and queued to the display process via QCLID. The keyboard process makes sure these two byte sequences are contiguous in QCLID.

In half duplex mode, any key value queued to the communications process is also queued to the display process.

When auto LF is selected, a carriage return key is automatically followed by a LF key value. This is true whenever the CR is queued to the communication process or the display process.

3.5.1.3 Terminal Control Software Process Module Functions

The Terminal Control Software Process looks for changes in switch settings. When any switch is changed, the process suspends for a short period to de-bounce the switches and then reads the switches and stores their content in designated RAM locations.

If the video invert or transparent mode switches are changed, the display process is notified by a command queued to QCLID.

For communication interfaces using modems, when on-line, the process checks for the data set ready to be on. If it is not, the communications process is activated.

For 202 modem usage, when the terminal is on-line and in XMT mode, the process checks for secondary channel receive to be on. If it is not, a command to switch to RCV mode is queued to the communications process via QCLOD.

3.5.1.4 Communications Control Software Process Module Functions

All data queued to QCLOD is removed by the communications process and queued to the display process when the terminal is in off-line mode.

When in the on-line mode, data bytes taken from QCLOD are sent on the communications line.

The framing and parity used on the line are determined by the setting of the configuration switches.

The following commands are recognized and executed when taken from QCLOD:

BREAK (8C)
Disconnect (9C)
Switch to RCV mode (9D) - 202 modems only
Switch to XMT mode (9E) - 202 modems only

When in on-line mode, the following interface signals must be present:

Data set ready
Data carrier detect (except when in transmit mode with 202 modems)

The data terminal ready signal is always on when the terminal has been powered on. The exception is during disconnect sequences when modems are being used.

In the event of parity or framing errors or receive over-run, an appropriate status indicator byte is queued to the display process. At this time, if "Transparent Mode" is set, the next character is sent as is. If "Transparent Mode" is not set, the next character sent is a "!" when a parity error occurs and a "?" for framing error.

When on-line, if data carrier detect drops for a long period, a disconnect sequence is initiated.

When the on-line key is lit, the communications process is either in on-line mode or attempting to get there. In the case where modems are being used, this may mean the process is waiting for data set ready to come on. (Refer to the Extended Communications Supplement for further information regarding optional modem operations.)

The communications interrupt handler loads and stores the ACIA data registers to effect the receiving and sending of data bytes.

3.5.1.5 Display Control Software Process Module Functions

The Display Control Software Process Module accepts data and commands on the interface and interprets/executes them to effect what is presented on the display screen.

The interface is a byte stream of information, implemented as dual in-memory circular queues. Certain commands require many bytes to complete their sequence; other commands are merely single bytes.

The interface described here must be the only means of communicating with the Display Control Process. Likewise, the Display Control Software Process Module looks to the interface as the only means of receiving data and commands or of transmitting data and commands. Hence, this interface is a two-way simultaneous communication mechanism.

The input interface consists of two queues, QSCRL and QCLID. Information from QSCRL has priority; i.e., if a multiple byte command is received on QSCRL, QCLID is not looked at until the command(s) from QSCRL is completed. Likewise, if a multiple byte command is being processed from QCLID and a command is entered on QSCRL, the processing from QCLID is aborted.

Only data from QSCRL is passed to the Input Communications Application Interpreter.

3.5.2 Organization of Application Software

The following rules must be followed when implementing application software.

. The application must start on a 1K boundary (i.e., 8000, 8400, 8800) not currently occupied by the resident software.

- The first twelve bytes of the application software must contain the following data:

1. Byte 0: Must contain the value of the most significant byte of this 1K memory module (i.e., if this module begins at hex memory location 8000, the value in control byte 0 would be "80").

2. Byte 1: A hex value between "1" and "10" representing the program process number. Care should be taken to select a value that has not been previously assigned. The values currently being used by the resident software are as follows:

Process Number	Hex Value	Process Name
1	1	System Initialization and Tone Processes
3	3	Keyboard Control Software Process Module
5	5	Communication Control Software Process Module
7	7	Display Control Software Process Module
14	E	Terminal Control Software Process Module
16	10	Idle Process

The process number reflects the relative priority with which the process is treated by the system monitor. The lower the process number, the higher the priority the program has within the system.

For more detailed information on the monitor and its use of processing priorities, refer to the section concerning the monitor.

3. Bytes 2-3: Must contain the address of the most significant byte of the application program stack area.

4. Bytes 4-8: These bytes contain the values to be used in initializing the application stack area with

   byte 4 = condition codes
   byte 5 = accumulator B
   byte 6 = accumulator A
   byte 7 = high order byte of the X register
   byte 8 = low order byte of the X register

   Normally these bytes are set to zero.

5. Bytes 9-10: The hex address of the start of the initialization procedure for this program.

6. Byte 11: The ones compliment of byte 0. This byte is used by the initialization process as a validity check.

3.5.2.1 Initialization Process - During system initialization, the Monitor Control Software Process Module will give each process, in order of priority, an opportunity to initialize its RAM, PIA, and/or ACIA areas. The entry point for initialization is defined by the value found in bytes 9 and 10 of the process vectors located in the first twelve bytes of the application program (see Figure 3-6). Upon completion of the initialization, the application program should suspend its process by use of the appropriate system suspend command. (Refer to Paragraph 3.4.1.1 for more detailed information.) When the process is reactivated, execution of the program will start with the instruction immediately following the suspend command.

If multiple processes have been assigned the same process number, the last process found will replace the previously created process.

The application initialization process must establish the presence of a user subroutine by storing the address of the routine in one of three RAM locations:

- TKEYI for a keyboard interpreter subroutine
- OCAI for a display control output subroutine
- ICAI for a display control input subroutine

3.5.2.2 Subroutine Interface - The method of interfacing an interpreter subroutine is common for the three areas of possibility (see Figure 3-7). The subroutine is entered via a jump to subroutine (JSR) instruction. Accumulator "A" contains the current character being processed. The user routine then has several options as indicated in Figure 3-8.

If the user queues the data byte(s), the carry bit must be cleared immediately prior to returning from the subroutine. This will cause the calling process to terminate any further processing of the current character and to seek the next character to be processed.

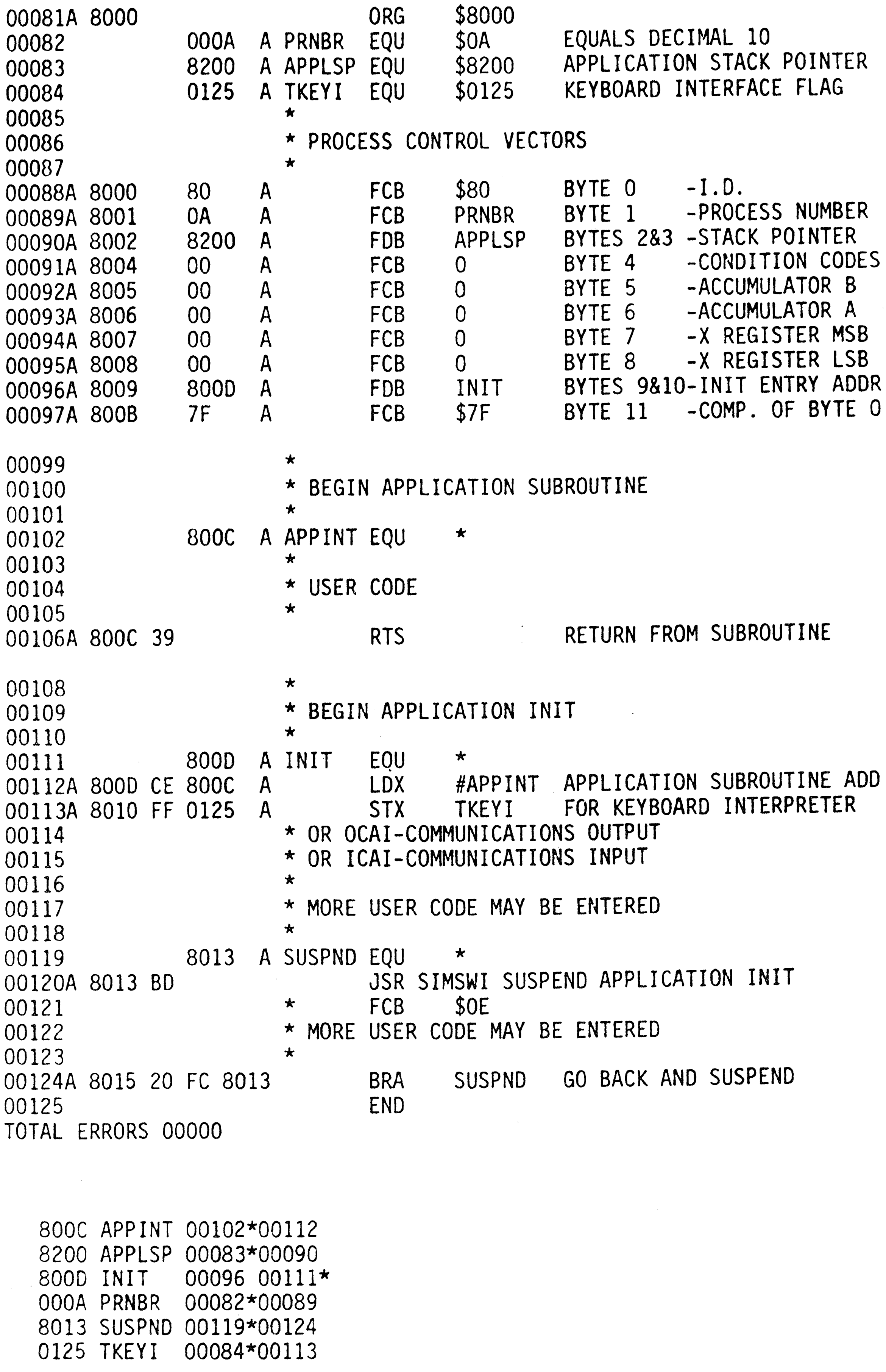
If the user desires the calling process to continue processing the current character found in Accumulator "A", the carry bit must be set immediately prior to returning from the subroutine.

3.5.2.3 Keyboard Interpreter Interface - During initialization, the starting address of the user subroutine must be stored in location TKEYI.

Each key stroke will be presented to the user subroutine in Accumulator "A" as a standard ASCII or command value. The user may then process the character and must exit the subroutine via an RTS instruction.

Care must be taken to ensure that the carry bit is in the proper state prior to exiting the subroutine. The carry bit is to be set if the user wishes the keyboard control process to continue processing the current data byte found in Accumulator "A". The carry bit is to be cleared if the user wishes the Keyboard Control Process to take no further action on the current data byte.

3.5.2.4 Display Control Software Process Communications Input Interface - During initialization, the starting address of the user subroutine must be stored in the memory location ICAI.



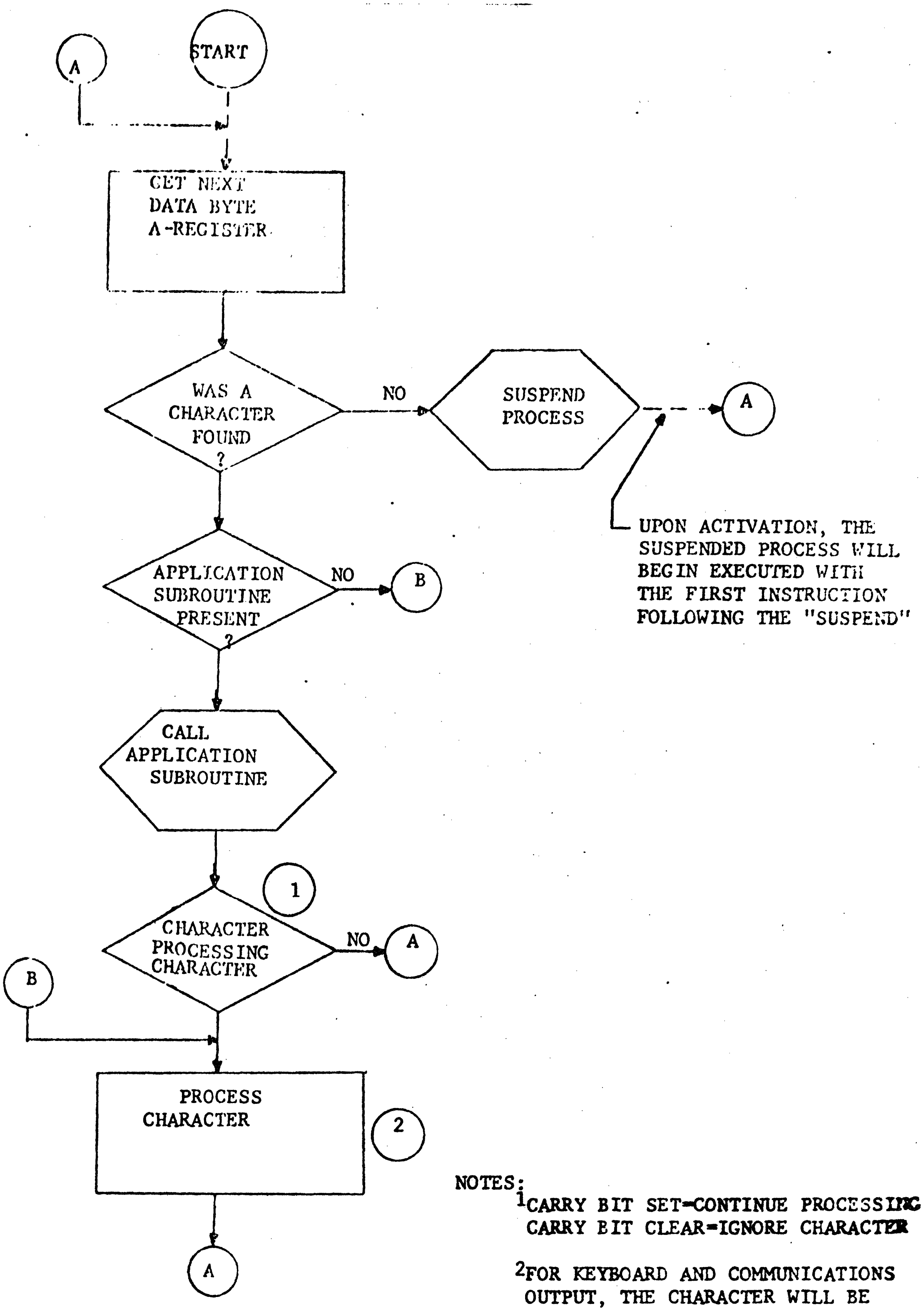
```
00081A 8000                      ORG    $8000
00082          000A  A PRNBR     EQU    $0A       EQUALS DECIMAL 10
00083          8200  A APPLSP    EQU    $8200     APPLICATION STACK POINTER
00084          0125  A TKEYI     EQU    $0125     KEYBOARD INTERFACE FLAG
00085                  *
00086                  * PROCESS CONTROL VECTORS
00087                  *
00088A 8000    80    A           FCB    $80       BYTE 0     -I.D.
00089A 8001    0A    A           FCB    PRNBR     BYTE 1     -PROCESS NUMBER
00090A 8002    8200  A           FDB    APPLSP    BYTES 2&3 -STACK POINTER
00091A 8004    00    A           FCB    0         BYTE 4     -CONDITION CODES
00092A 8005    00    A           FCB    0         BYTE 5     -ACCUMULATOR B
00093A 8006    00    A           FCB    0         BYTE 6     -ACCUMULATOR A
00094A 8007    00    A           FCB    0         BYTE 7     -X REGISTER MSB
00095A 8008    00    A           FCB    0         BYTE 8     -X REGISTER LSB
00096A 8009    800D  A           FDB    INIT      BYTES 9&10-INIT ENTRY ADDR
00097A 800B    7F    A           FCB    $7F       BYTE 11    -COMP. OF BYTE 0

00099                  *
00100                  * BEGIN APPLICATION SUBROUTINE
00101                  *
00102          800C  A APPINT    EQU    *
00103                  *
00104                  * USER CODE
00105                  *
00106A 800C 39                   RTS              RETURN FROM SUBROUTINE

00108                  *
00109                  * BEGIN APPLICATION INIT
00110                  *
00111          800D  A INIT      EQU    *
00112A 800D CE 800C  A           LDX    #APPINT   APPLICATION SUBROUTINE ADD
00113A 8010 FF 0125  A           STX    TKEYI     FOR KEYBOARD INTERPRETER
00114                  * OR OCAI-COMMUNICATIONS OUTPUT
00115                  * OR ICAI-COMMUNICATIONS INPUT
00116                  *
00117                  * MORE USER CODE MAY BE ENTERED
00118                  *
00119          8013  A SUSPND    EQU    *
00120A 8013 BD                   JSR SIMSWI SUSPEND APPLICATION INIT
00121                  *         FCB    $0E
00122                  * MORE USER CODE MAY BE ENTERED
00123                  *
00124A 8015 20 FC 8013           BRA    SUSPND    GO BACK AND SUSPEND
00125                            END
TOTAL ERRORS 00000


   800C APPINT 00102*00112
   8200 APPLSP 00083*00090
   800D INIT   00096 00111*
   000A PRNBR  00082*00089
   8013 SUSPND 00119*00124
   0125 TKEYI  00084*00113
```

FIGURE 3-6. Example of Code to Define an Application Subroutine and its Associated Initialization Process



NOTES:

[1]CARRY BIT SET=CONTINUE PROCESSING
CARRY BIT CLEAR=IGNORE CHARACTER

[2]FOR KEYBOARD AND COMMUNICATIONS
OUTPUT, THE CHARACTER WILL BE
PLACED IN PROPER QUEUE

FOR COMMUNICATION INPUT, THE
CHARACTER WILL BE PROCESSED
BY THE DISPLAY CONTROL PROCESS

FIGURE 3-7. Application Subroutine Interface with Resident Input/Output Modules

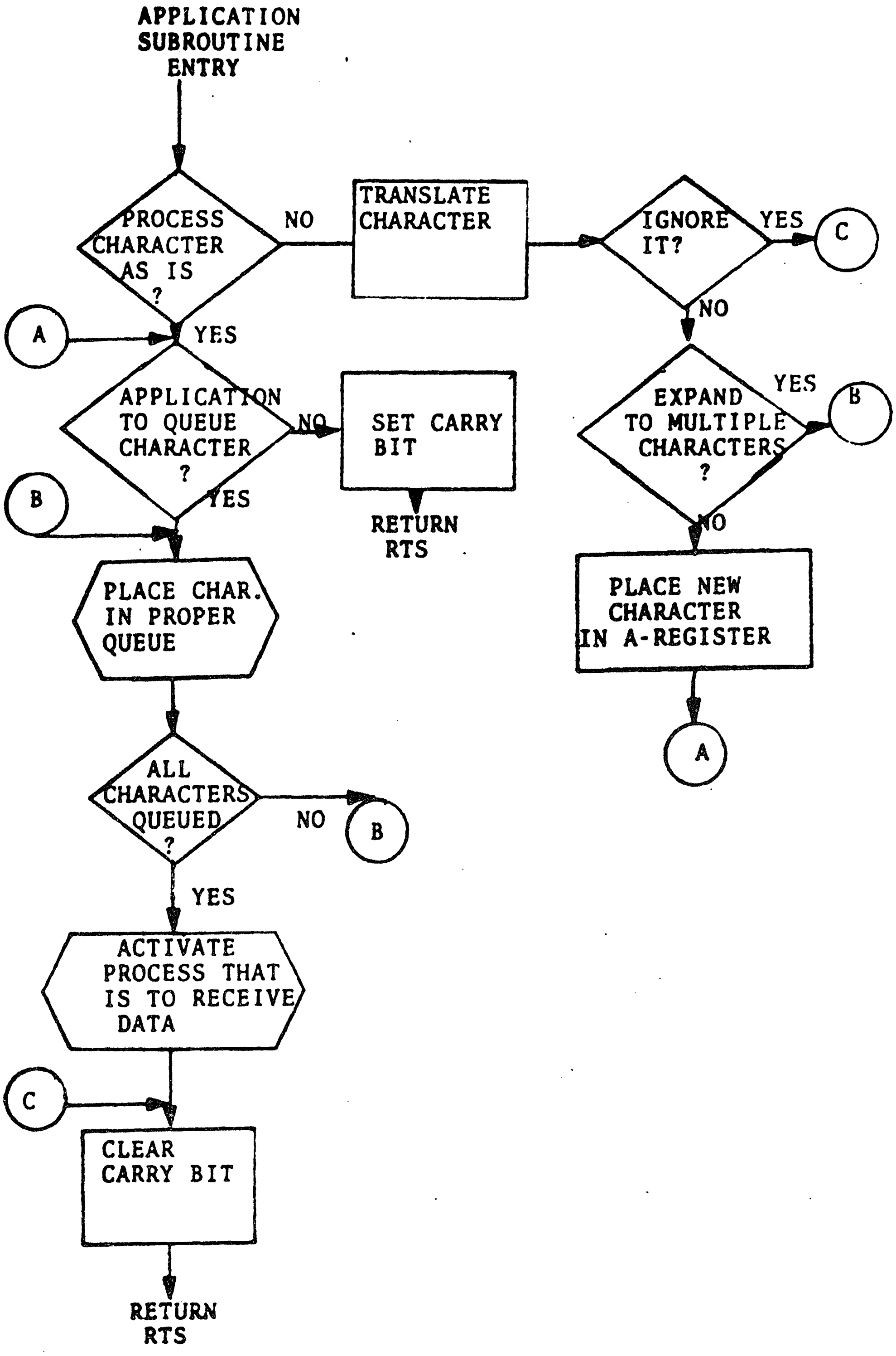

APPLICATION SUBROUTINE ENTRY
PROCESS CHARACTER AS IS ?
NO
TRANSLATE CHARACTER
IGNORE IT?
YES
C
NO
A
YES
APPLICATION TO QUEUE CHARACTER ?
NO
SET CARRY BIT
RETURN RTS
EXPAND TO MULTIPLE CHARACTERS ?
YES
B
YES
B
NO
PLACE CHAR. IN PROPER QUEUE
PLACE NEW CHARACTER IN A-REGISTER
A
ALL CHARACTERS QUEUED ?
NO
B
YES
ACTIVATE PROCESS THAT IS TO RECEIVE DATA
C
CLEAR CARRY BIT
RETURN RTS

FIGURE 3-8. Example of User Application Subroutine Logic

Each character received on the communications line is placed in the Scroll Queue (QSCRL) by the communications process. The Display Control Software Process is then activated and removes bytes from the Scroll Queue. The current character is passed to the user subroutine, if present, in Accumulator "A". The user should restrict the processing done by the subroutine because at the high baud rates (i.e., 9600 baud or higher), the danger exists of creating a receive over-run condition if the characters are not processed quickly enough.

If a character is expanded into a multiple character string that is to be processed by the Display Control Process, the characters must be placed in the QCLID queue. These bytes, however, will not be processed until the Display Control Process has determined the communications input queue (QSCRL) has been emptied.

Care must be taken to ensure that the carry bit is in the proper state prior to exiting the subroutine. The carry bit is to be set if the user wishes the Display Control Process to continue processing the current data byte found in Accumulator "A". The carry bit is to be cleared if the user wishes the Display Control Process to take no further action on the current data byte.

3.5.2.5 Display Control Software Process Communications Output Interface - During initialization, the starting address of the user subroutine must be stored in the memory location OCAI.

Before the Display Control Process queues a data byte to be transmitted, the user subroutine will be called, if present, with the data byte in Accumulator "A". The user may accumulate all data bytes for validity checking and initiate his own code sequence that will place the data bytes in the communications output queue (QCLOD). If no accumulation of data is desired, the user may place the byte in the output queue or may elect to have the Display Control Process queue the data.

If the user detects a validity error in the data, he may send commands and messages to the Display Control Process that can, for example, blink the field in error and display an error message. This data must be queued on the Display Control Process input queue (QCLID).

Care must be taken to return from the subroutine with the carry bit in the proper state. The carry bit is to be set if the user wishes the Display Control Process to queue the current data byte found in Accumulator "A". The carry bit is to be cleared if the user wishes the Display Control Process to take no further action on the current data byte.

3.5.2.6 Spare Switches - The rear panel controls contain three spare switches labeled Switch A, Switch B, and Switch C, which may be read and interpreted by user software. The procedure necessary to read the Peripheral Interface Adapter (PIA) attached to Switches A, B, and C is as follows:

a. Read the PIA (PIA2BD) by performing a load (LDAA or LDAB). The switch settings will be in register A or B.

b. Interogate bits one, two, and three, respectively, for Switches A, B, and C. If the bit value is zero, the switch is in the 'ON" position. If the bit value is one, the switch is in the "OFF" position.

These switches may be used to program the application to inhibit a function, a set of functions, or to cause the application to turn itself off. The switches are generally used during the program's initialization process and then not read again.

However, if the application requires periodic reading of the PIA to see if any switch settings have been changed, the user should initiate a delay of 55 milliseconds after a change has been detected and before a final read of the PIA is made. This will allow sufficient time to "debounce" the switches.

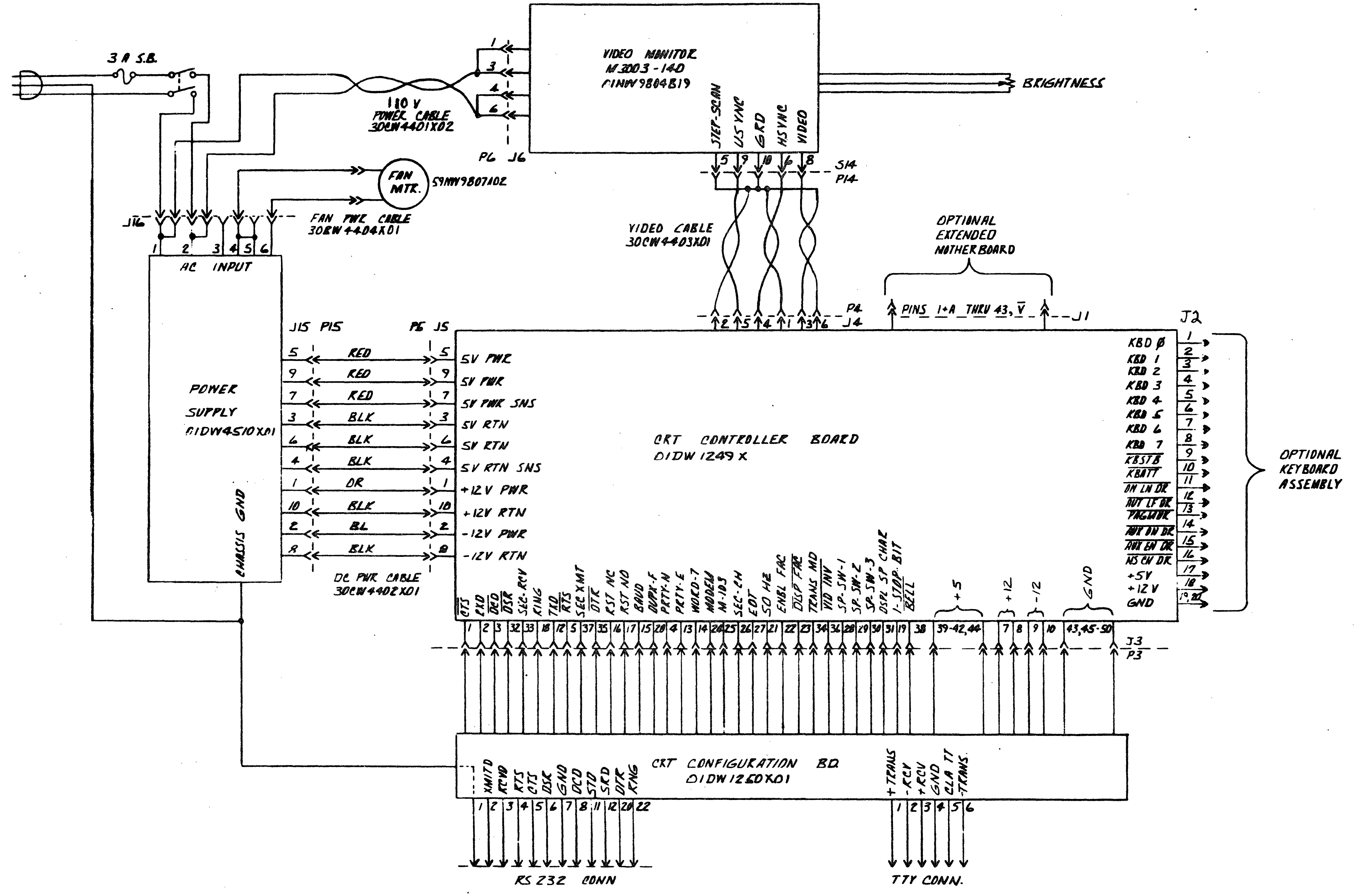
For example, if a program periodically reads the PIA and suspects a switch has changed (i.e., the current value differs from the previous value), a delay would be initiated using the Delay MOC routine. Following the delay, a final read of the PIA would be made. The results could then be tested and saved for future reference.

CHAPTER 4

DIAGRAMS

4.1 SCHEMATIC AND INTERCONNECTION DIAGRAMS

This chapter contains the interconnection diagram for the Basic Display. Also, the schematics for the CRT Controller PC Board, Configuration PC Board, Power Supply Assembly, and the Video Monitor Assembly are provided.



THIS DOCUMENT CONTAINS INFORMATION PROPRIETARY TO MOTOROLA, INC. AND SHALL NOT BE USED FOR ENGINEERING, DESIGN, PROCUREMENT, OR MANUFACTURE IN WHOLE OR IN PART WITHOUT CONSENT OF MOTOROLA, INC.

220V	30CW1558X	02
110V	30CW1558X	01
PROCESS	PART NUMBER	

DIMENSIONS ARE IN INCHES

UNLESS OTHERWISE SPECIFIED TOLERANCES ARE:

2 PLACE DEC ±

3 PLACE DEC ±

HOLES ±

ANGLES ±

SCALE:

DO NOT SCALE DWG.

TITLE

INTERCONNECT, CRT TERMINAL

DR GAL	DATE 8-9-77	ME	DATE	ORIG MODEL	
APPD RWT.	DATE	EE	DATE		

MOTOROLA INC.
DATA SYSTEMS
3102 N. 56TH ST.
PHOENIX, AZ 85018

DWG. NO. 30CW1558X
SHEET 2 OF 2

WHEN REFERRING TO THIS DWG. STATE PT NO. AND APPLICABLE ISSUE SYMBOL

SYM	SIZE AFTER PLATING	QTY

HOLE LIST

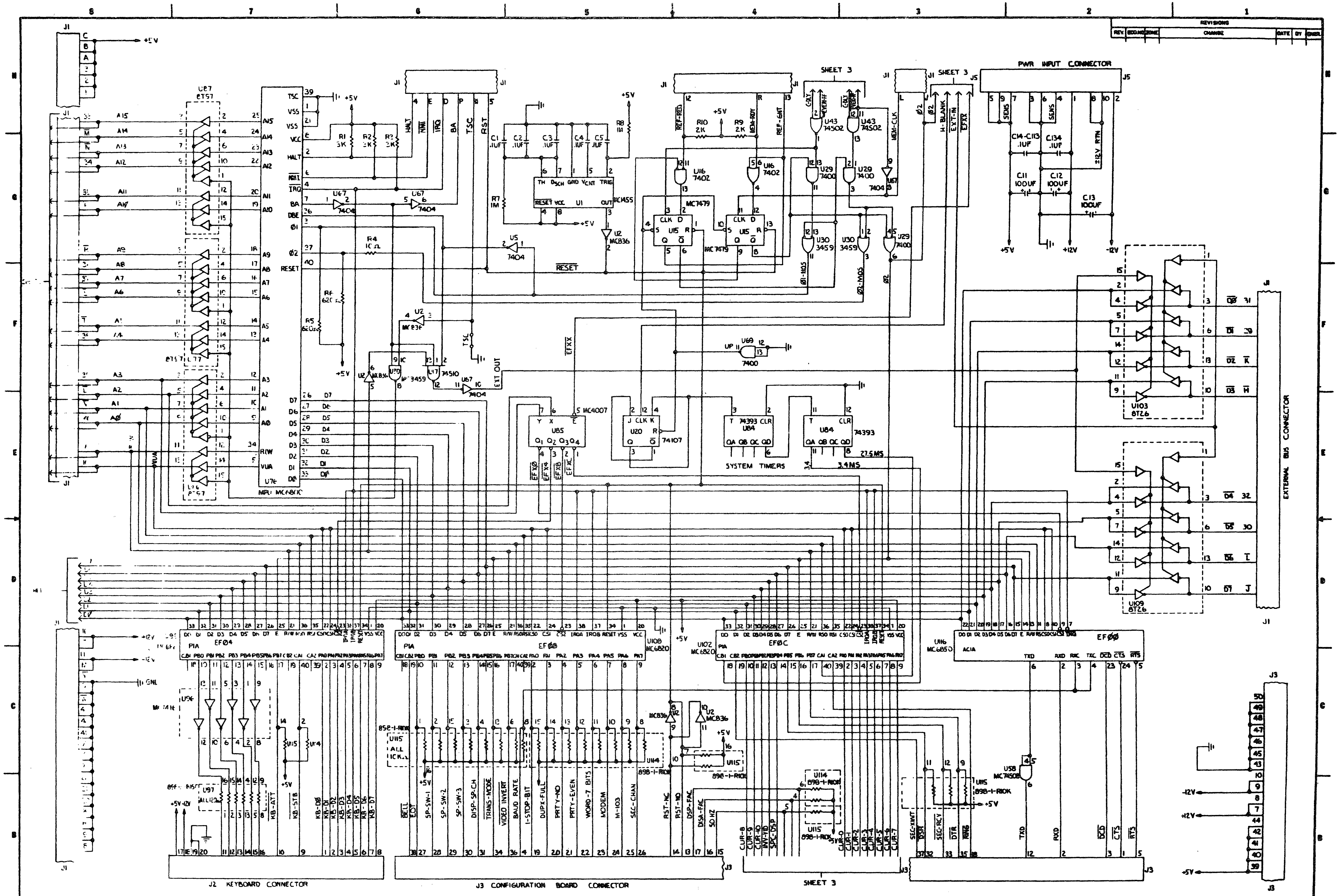
ISSUE	REL/CN NO.	DATE	APPD	DESCRIPTION

REVISION

ISSUE	REL/CN NO.	DATE	APPD	DESCRIPTION
K		8-9-77		RELS'D FOR PRODUCTION

REVISION

DWG NO. 30CW1558X
SHEET 2 OF 2
ISSUE K



NOTES:

FOR PARTS LIST SEE BMC1DW1249X01

NEXT ASSEMBLY	USED ON
APPLICATION	

UNLESS OTHERWISE SPECIFIED, TOLERANCES:
INCHES .XX ± .XXX ±
MILLIMETERS .X ± .XX ±
ANGULAR ±
✓ RMS ALL MACHINED SURFACES.
FEATURE CONTROL SYMBOLS PER ANSI Y14.5.
BREAK ALL SHARP EDGES AND CORNERS, REMOVE BURRS.
UNDERLINED DIMS NOT TO SCALE.
THIRD ANGLE ORTHOGRAPHIC PROJECTION IS USED.

MATERIAL

HEAT TREAT

APPLIED FINISH

DRAWN BY E/M DESIGNS DATE 4-20-77
CHECKED BY DATE
ENGR. APPROVAL DATE

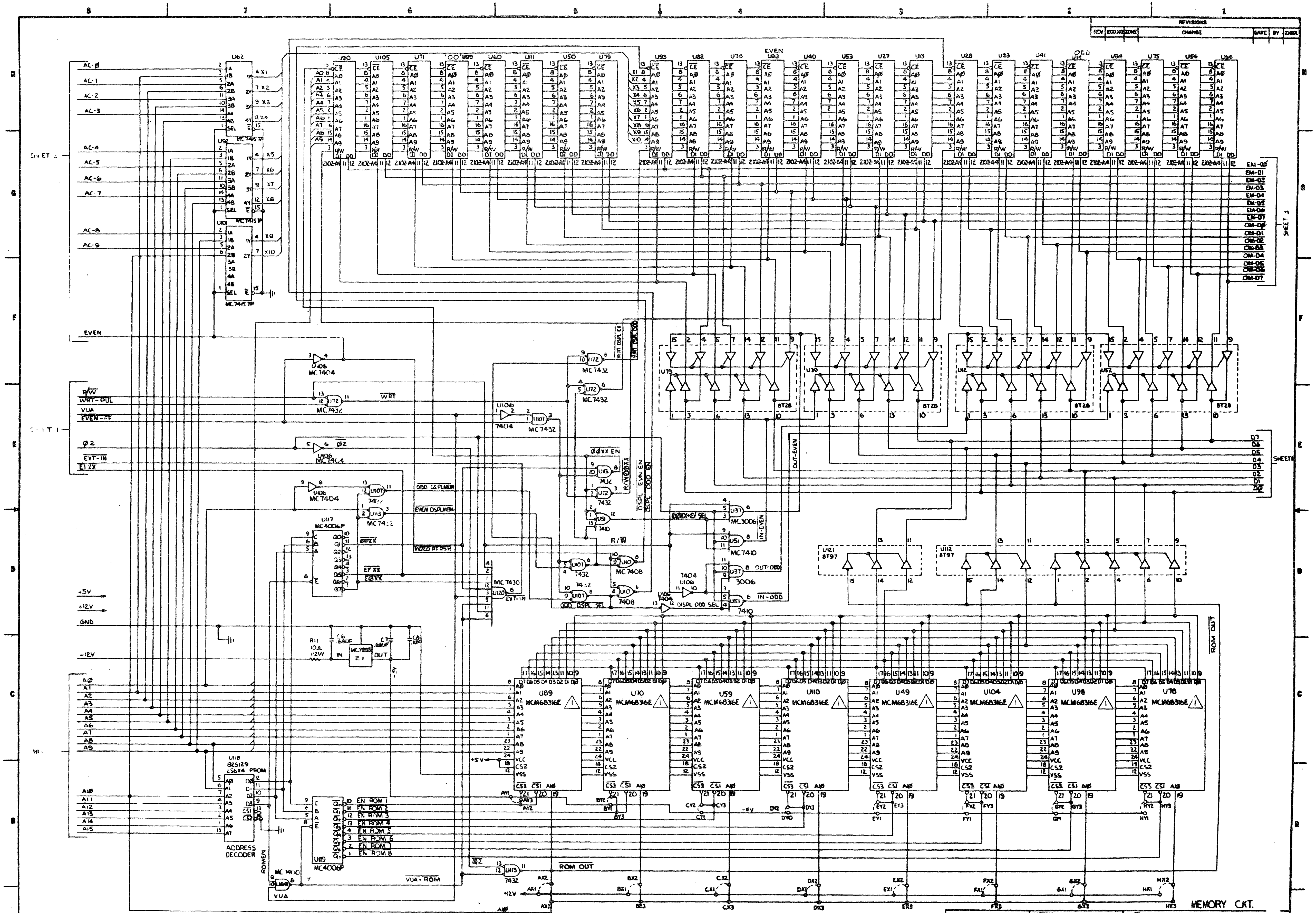
MPU BUS CKT

MOTOROLA INC.
Discrete Semiconductor Division

TITLE: SCHEMATIC-
CRT CONTROLLER BD.

SIZE	CODE IDENT. NO.	DRAWING NO.
E	04713	63EW1249X

SCALE WEIGHT SHEET 1 OF 3



NOTES:

1\ MCM68708 MAY BE USED IF LINES ARE REMOVED FROM X2 TO X3 AND Y2 TO Y3 AND JUMPERS INSTALLED FROM X1 TO X2 AND Y1 TO Y2 IN EACH POSITION USED.

NEXT ASSEMBLY	USED ON
APPLICATION	

UNLESS OTHERWISE SPECIFIED, TOLERANCES:
INCHES .XX± .XXX±
MILLIMETERS .X± .XX±
ANGULAR ±
√ RMS ALL MACHINED SURFACES.
FEATURE CONTROL SYMBOLS PER ANSI Y14.5
BREAK ALL SHARP EDGES AND CORNERS, REMOVE BURRS.
UNDERLINED DIM NOT TO SCALE.
THIRD ANGLE ORTHOGRAPHIC PROJECTION IS USED.

MATERIAL

HEAT TREAT

APPLIED FINISH

DRAWN BY E/M DESIGNS	DATE 6-28-77
CHECKED BY	DATE
ENGR. APPROVAL	DATE

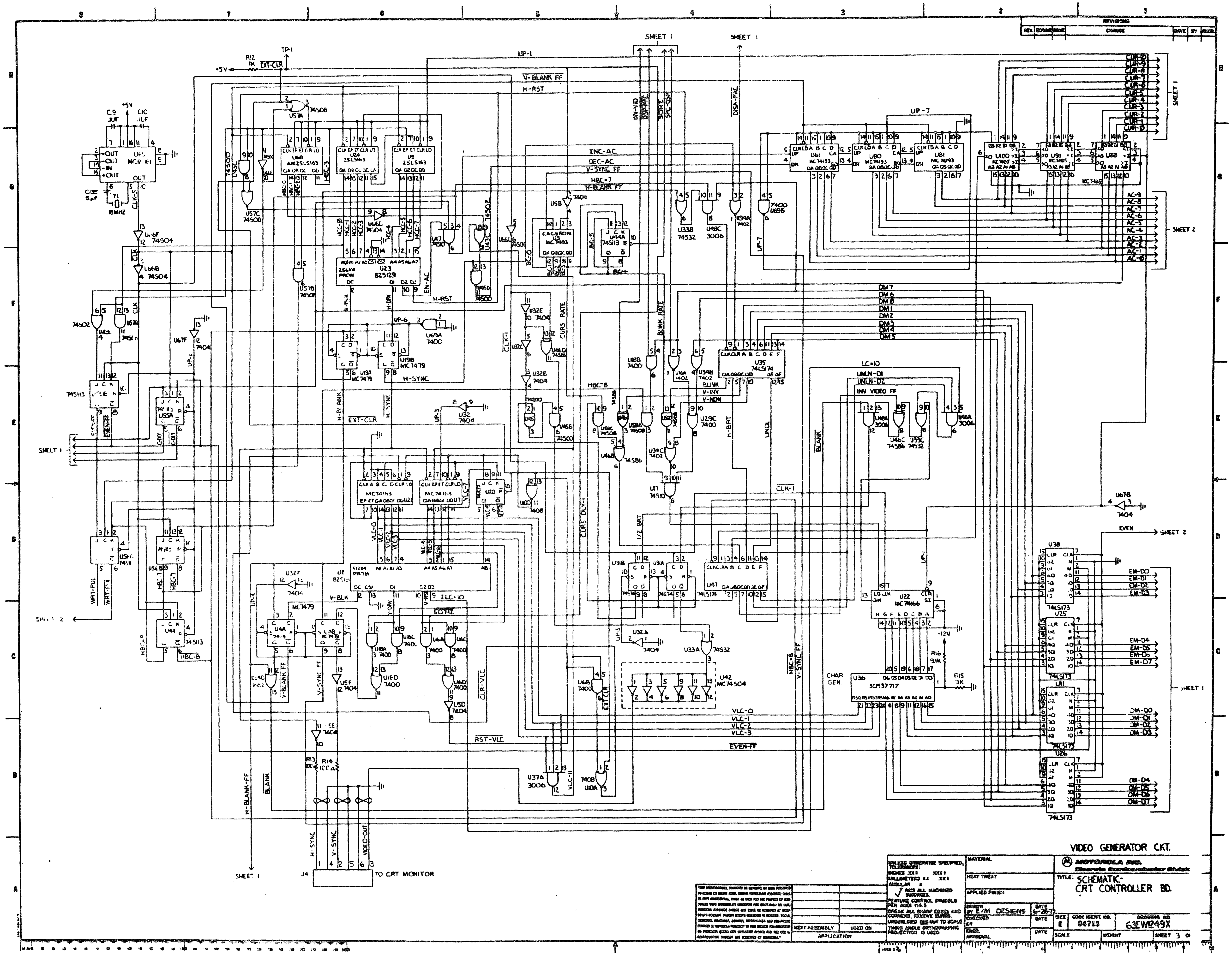
Ⓜ MOTOROLA INC.
Discrete Semiconductor Division

TITLE: SCHEMATIC-
CRT CONTROLLER BD.

SIZE	CODE IDENT. NO.	DRAWING NO.
E	04713	63EW1249X

SCALE	WEIGHT	SHEET 2 OF

4-7/4-8



VIDEO GENERATOR CKT.

"THE SPECIFICATIONS, DRAWINGS OR DESIGNS, OR DATA RESERVED TO HEREIN ARE MOTOROLA'S PROPERTY, ... CONFIDENTIAL ... REPRODUCTION THEREOF ARE RESERVED BY MOTOROLA"

NEXT ASSEMBLY | USED ON
APPLICATION

UNLESS OTHERWISE SPECIFIED, TOLERANCES:
INCHES .XX± .XXX±
MILLIMETERS .X± .XX±
ANGULAR ±
√ R63 ALL MACHINED SURFACES.
FEATURE CONTROL SYMBOLS PER ANSI Y14.5.
BREAK ALL SHARP EDGES AND CORNERS, REMOVE BURRS.
UNDERLINED DIM NOT TO SCALE.
THIRD ANGLE ORTHOGRAPHIC PROJECTION IS USED.

MATERIAL
HEAT TREAT
APPLIED FINISH
DRAWN BY E/M DESIGNS DATE 6-28-7
CHECKED BY DATE
ENGR. APPROVAL DATE

MOTOROLA INC.
Discrete Semiconductor Divisi

TITLE: SCHEMATIC-
CRT CONTROLLER BD.

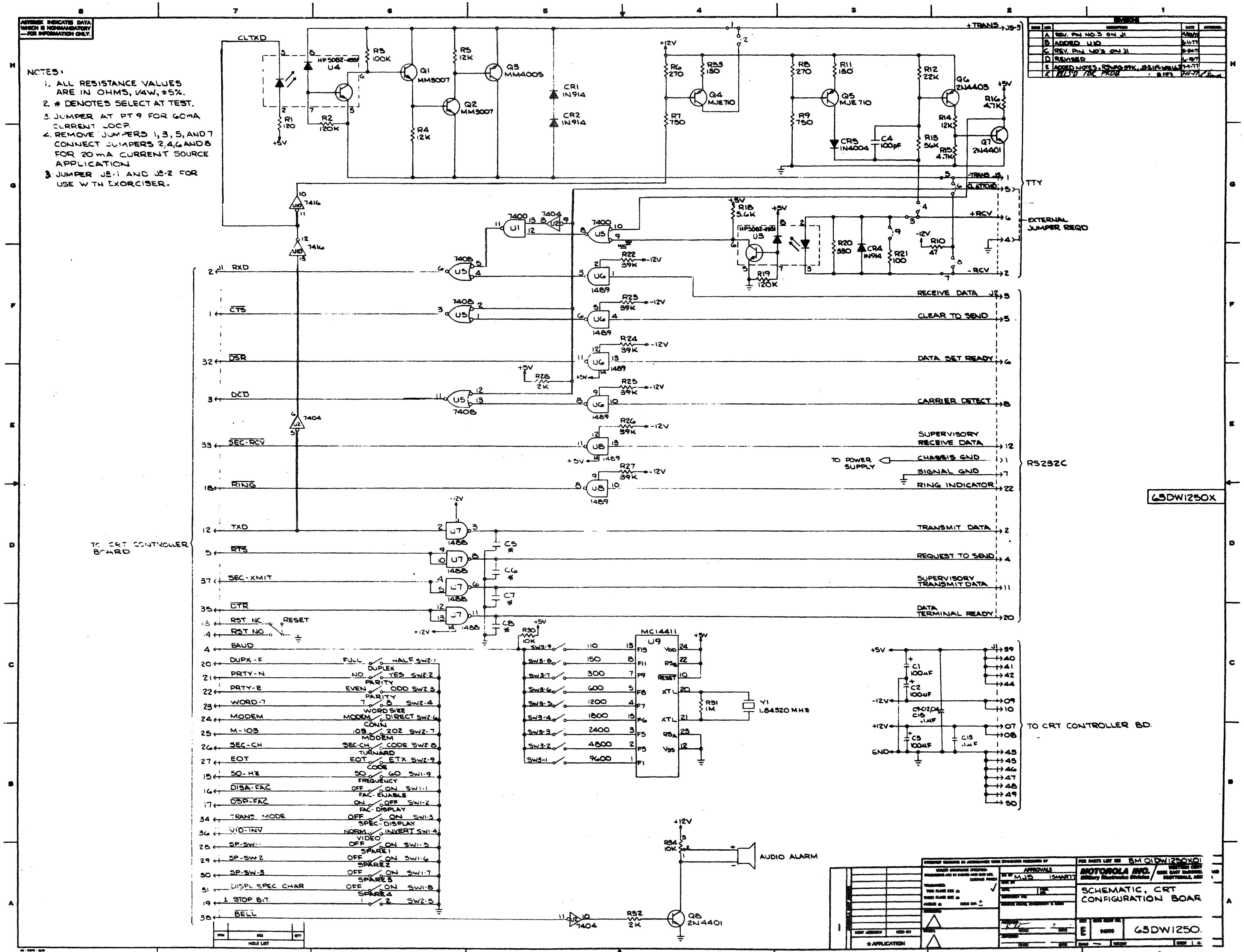
SIZE E | CODE IDENT. NO. 04713 | DRAWING NO. 63EW1249X
SCALE | WEIGHT | SHEET 3 O

ASTERISK INDICATES DATA WHICH IS NONMANDATORY —FOR INFORMATION ONLY.

NOTES:

1. ALL RESISTANCE VALUES ARE IN OHMS, 1/4W, ±5%.
2. * DENOTES SELECT AT TEST.
3. JUMPER AT PT 9 FOR 60mA CURRENT LOOP.
4. REMOVE JUMPERS 1, 3, 5, AND 7 CONNECT JUMPERS 2, 4, 6 AND 8 FOR 20 mA CURRENT SOURCE APPLICATION
5. JUMPER J3-1 AND J3-2 FOR USE WITH EXORCISER.

	A	REV. PIN NO.3 ON J1	4/20/77	
	B	ADDED U10	5-11-77	
	C	REV. PIN NO'S ON J1	5-24-77	
	D	REVISED	6-15/77	
	E	ADDED NOTES, R5 WAS 39K, J3-3 1K, WAS 4.7	7-4-77	
	K	RELS'D FOR PROD, 8153	[illegible]	



FOR PARTS LIST SEE BM 01DW1250X01

MOTOROLA INC. Military Electronics Division / WESTERN CENTER 8201 EAST McDOWELL ROAD SCOTTSDALE, ARIZ.

APPROVALS

MJS 15MAR77

SCHEMATIC, CRT CONFIGURATION BOAR

E 94990 63DW1250.

☆ APPLICATION

NEXT ASSEMBLY USED ON

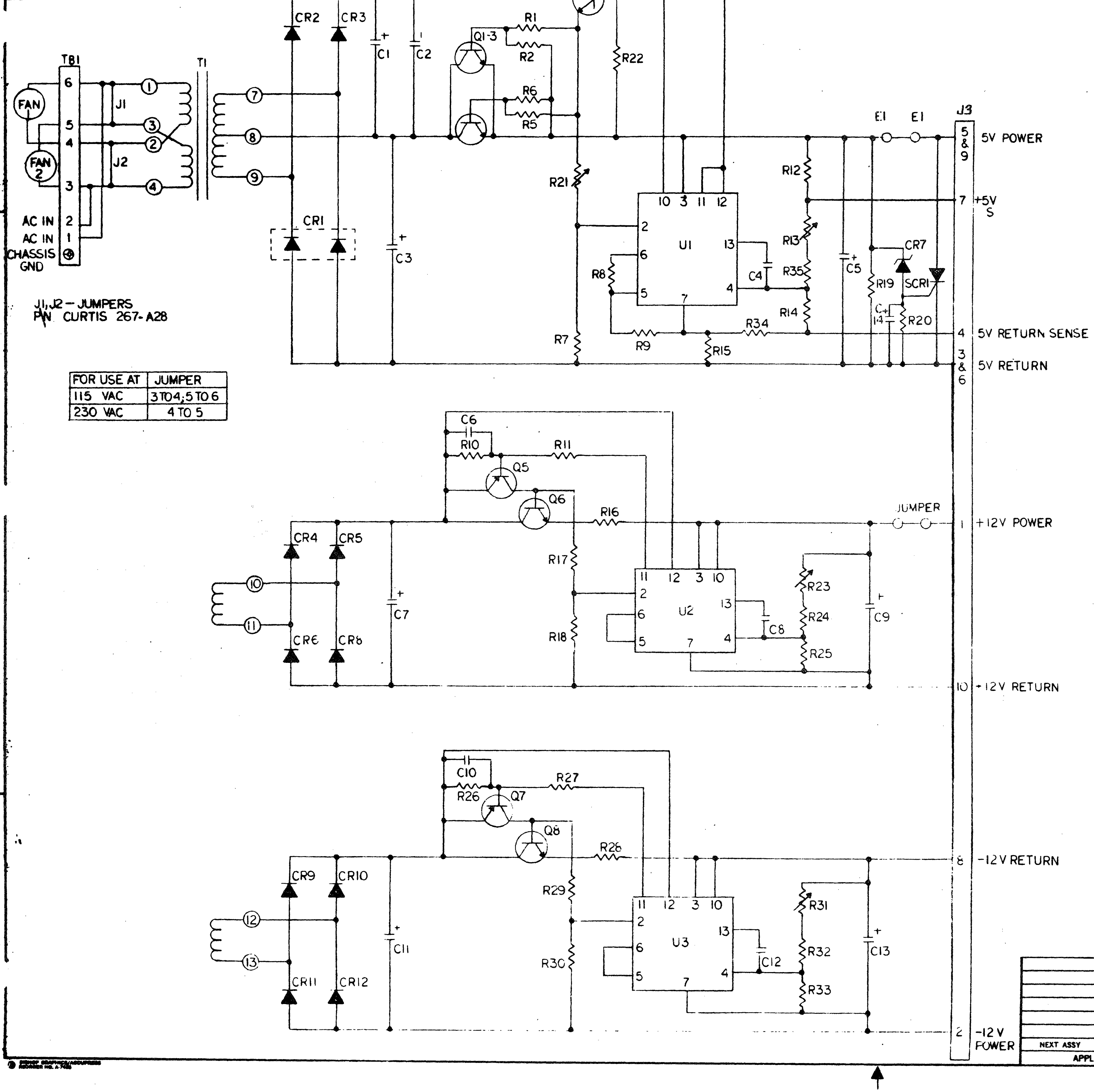
4-11/4 12

REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
	A	ADDED J3, R17, 29 WAS 270 Ω	6-10-77	

PRODUCTION NOTES

* 1) HEATSINK WITH 13920 THERMOTAB HEATSINK

2) U1 U2 U3 MUST BE PRETESTED LO-NOISE I.C.



J1, J2 — JUMPERS
P/N CURTIS 267-A28

FOR USE AT	JUMPER
115 VAC	3 TO 4; 5 TO 6
230 VAC	4 TO 5

R34	BUSS	RESISTOR	
C1,2,5	1000/16	CAPACITOR ELECT	101-10108
C3	64000/15		103-10030
C7	7300/35		102-10098
C9,13	100/35	ELECT	101-10110
C4, 6, 10	.001/100	CAPACITOR MYLAR	104-10093
C11	3300/35	CAPACITOR ELECT	102-10099
CR1	R711A	DIODE BRIDGE	140-10003
CR2, 3	AE1C	1A 200V	111-10251
CR4,5,6,8-12	AE3B	3A 100V	111-10252
CR7	1N753	DIODE ZENER	112-10007
SCR1	S0315L *	SCR	160-10259
C8, 12	.01/1000	CAPACITOR MYLAR	105-10090
Q1, 2, 3	12505-2	TRANSISTOR	171-10262
Q4	12500-3		171-10261
Q5, 7 *	2N6554 *		172-10250
Q6, 8	12505-3	TRANSISTOR	171-10261
C14	1\50	CAPACITOR ELECT	101-10111
R17, 29	300 Ω	RESISTOR ½W 5% C.F.	151-10352
R1, 3, 5, 12, 15, 35	6.8 Ω		151-10313
R2, 4, 6	22 Ω		151-10325
R7, 8, 9, 22	2.2K		151-10373
R18, 30	4.7K		151-10381
R19, 20	47 Ω		151-10333
R10, 11, 26, 27	180 Ω		151-10347
R14	1.6K	½W 2% MF	152-10510
R24, 32	1.2K		152-10507
R25, 33	2.4K	½W 2% MF	152-10514
R16	.12 Ω	2W WW BWH	158-10077
R28	.22 Ω	RESISTOR 2W WW BWH	158-10079
R13, 21, 23 31	1.5K	POT CTS BE54719	155-10085
TB1	*140 SERIES, 6 TER	TERMINAL BLOCK CINCH	901-10778N
U1, 2, 3	μa723	I C VOLTAGE REGULATOR	130-10287
T1	14352	TRANSFORMER	082-14352
CHASSIS	14350	CHASSIS	412-14381
P.C. B	14386	P C BOARD	505-14366
J1 J2	CINCH JONES [illegible]	JUMPERS	901-10781N
J3	1-380991-0	CONNECTOR	
E1, JUMPER	* 16 GA BLK	WIRE 9"	
REF DES	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	ST P/N

PARTS LIST

NEXT ASSY	USED ON
APPLICATION	

TOLERANCE XX = .030
XXX = .010

MATERIAL

FINISH

DO NOT SCALE DRAWING

CONTRACT NO.

APPROVALS	DATE
DRAWN	5/26/77
CHECKED	6/ /77
ENG. APP.	
APPROVED	

POWER-ONE, INC
531 DAWSON DRIVE
CAMARILLO, CALIF. 930
(805) 484-2806

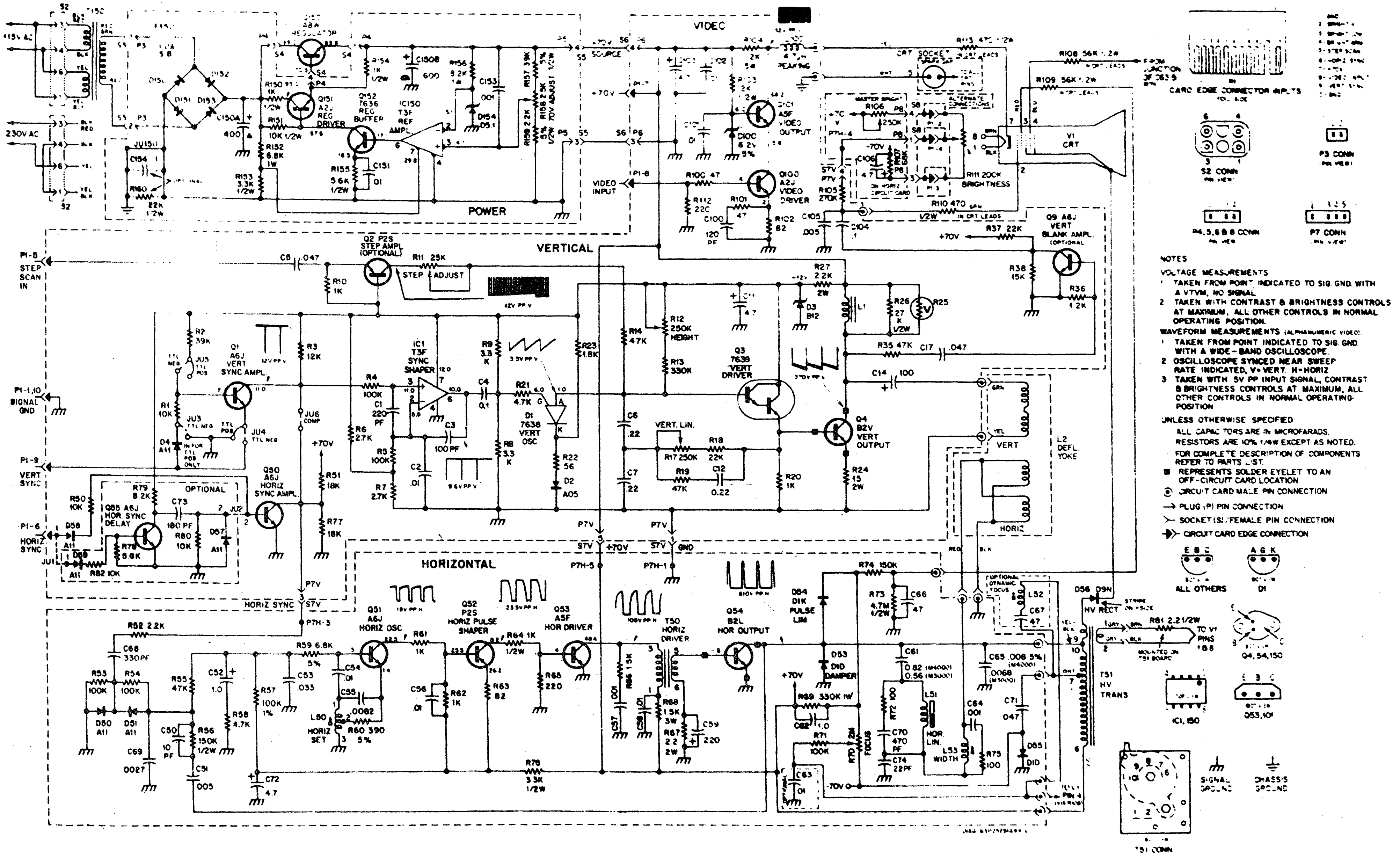
TITLE
SCHEMATIC CP 260

SIZE	CODE IDENT NO.	DRAWING NO.
D	54407	143 E

SCALE

SHEET OF

4-15/4-16



[redacted]
[redacted]

NOTES

VOLTAGE MEASUREMENTS

1 TAKEN FROM POINT INDICATED TO SIG. GND. WITH A VTVM, NO SIGNAL

2 TAKEN WITH CONTRAST & BRIGHTNESS CONTROLS AT MAXIMUM, ALL OTHER CONTROLS IN NORMAL OPERATING POSITION.

WAVEFORM MEASUREMENTS (ALPHANUMERIC VIDEO)

1 TAKEN FROM POINT INDICATED TO SIG. GND. WITH A WIDE-BAND OSCILLOSCOPE.

2 OSCILLOSCOPE SYNCED NEAR SWEEP RATE INDICATED, V= VERT H=HORIZ

3 TAKEN WITH 5V PP INPUT SIGNAL, CONTRAST & BRIGHTNESS CONTROLS AT MAXIMUM, ALL OTHER CONTROLS IN NORMAL OPERATING POSITION

UNLESS OTHERWISE SPECIFIED

ALL CAPACITORS ARE IN MICROFARADS.

RESISTORS ARE 10% 1/4W EXCEPT AS NOTED.

FOR COMPLETE DESCRIPTION OF COMPONENTS REFER TO PARTS LIST.

■ REPRESENTS SOLDER EYELET TO AN OFF-CIRCUIT CARD LOCATION.

⦿ CIRCUIT CARD MALE PIN CONNECTION

→ PLUG (P) PIN CONNECTION

≻ SOCKET (S)/FEMALE PIN CONNECTION

→≻ CIRCUIT CARD EDGE CONNECTION

VIDEO MONITOR SCHEMATIC DIAGRAM

APPENDIX A

GLOSSARY OF TERMS

ACCUMULATOR - A register of the arithmetic/logic unit of a central processor used as intermediate storage during the formation of algebraic sums, or for other intermediate logical and arithmetic operations.

ACIA (Asynchronous Communication Interface Adapter) - An integrated circuit that provides a means of efficiently interfacing the MPU to devices requiring an asynchronous serial data format.

ACTIVATE - To make a process eligible for execution.

ASCII CODE - The acronym for American Standard Code for Information Interchange. This standardized code is used extensively in data transmission. The code includes 128 upper and lower case letters, numerals, and special purpose symbols each encoded by a unique 7-bit binary number.

ASYNCHRONOUS COMMUNICATION - A method of transferring data where the timing of character placement on connecting communication lines is not critical. Each transferred character is preceded by a start bit and followed by a stop bit, permitting the interval between characters to vary.

BAUD RATE - Synonymous with signal events (bits)-per-second and used as a measure of serial data flow between a computer and/or communication devices.

BIT - A binary digit. A bit is the smallest unit of data in a digital computer.

BREAKPOINT - A location in a program at which execution of that program can be halted to permit visual check, printing out, or other performance analysis.

BUFFER - An electronic circuit which forms a temporary store for data or information signals.

BUS - A circuit or group of circuits which provide a communication path between two or more devices, such as between a central processor, memory, and peripherals.

BYTE - A set of contiguous binary bits, usually eight, which are operated on as a unit. A byte can also be a subset of a computer word.

CARRY BIT - A condition (or status) bit that can be set or reset by certain data operations, and its status can be directly tested by a program.

CHARACTER - One of a set of alphabetic, numeric, and/or graphic symbols. Each character is represented in its set by a unique binary code.

CIRCULAR QUEUE - A mechanism that allows variable amounts of data to be stored until a process is ready to receive it.

CODE - A system of characters and rules for representing information in a language capable of being understood and used by a computer. Code can be in the form of alphanumeric characters or binary data that can be directly executed by a computer.

CONTROL CHARACTER - A character whose occurrence in a particular context initiates, modifies, or halts operation.

COUNTER - A device such as a register or a memory location which is used to record the number of events. Counters can typically be incremented, decremented, preset to a value, or cleared.

CYCLE TIME - The time required by a computer to read from or write into the system memory. If system memory is core, the read cycle time includes a write-after read (restore) subcycle. Cycle time is often used as a measure of computer performance, since this is a measure of the time required to fetch an instruction.

EXECUTE - To perform a specified computer instruction. To run a program.

FETCH - The action of obtaining an instruction from a stored program and decoding that instruction. Also refers to that portion of a computer's instruction cycle when that action is performed.

FIRMWARE - A computer program (software) that is implemented in hardware, such as read-only memory.

FLAG - An indicator, usually a single binary bit, whose state is used to inform a later section of a program that a condition, identified with the flag and designated by the state of the flag, had occurred. A flag can be both software- and hardware-implemented.

HALF DUPLEX - Refers to a communication channel which can receive and transmit, but not simultaneously.

INTERRUPT - Suspension of normal program execution to execute a higher priority service routine, as requested by a peripheral device. After completing service routine execution, the interrupted program execution is restored at the point where it was interrupted.

INSTRUCTION - The smallest single operation that the computer can be directed to operate.

INTERRUPT VECTOR - Typically, two memory locations assigned to an interrupting device and containing the starting address and processor status word for its service routine.

JUMP - An instruction which, when executed, can cause the computer to fetch the next instruction to be executed from a location other than the next sequential location. Synonymous with "Branch".

LINE ROLLOVER - Cursor moving to the left on a display screen - proceeds from left-most column of current line to right-most column of previous line, proceeds from left-most column of top line to right-most column of last line. Cursor movement to the right - proceeds from right-most column of current line to left-most column of next line, proceeds from right-most column of last line to left-most column of top line.

MEMORY - A general term which refers to any storage media for binary data. Basic memory functional types include read/write and read-only.

MEMORY MAP - A listing of addresses or symbolic representations of addresses which define the boundaries of the memory address space occupied by a program or a series of programs.

MICROPROCESSOR - A single LSI circuit which performs the functions of a CPU. Some characteristics of a microprocessor include small size, inclusion in a single integrated circuit or a set of integrated circuits, and low cost.

MONITOR - A program, typically part of a larger operating system, which provides a uniform method of program timing, scheduling, and handling of input/output tasks.

OVERFLOW - A condition occurring in a computer when the results of a mathematical operation produces a result which has a magnitude exceeding the capacity of the computer's data word size.

PAGE ROLLOVER - Cursor movement upward on a display screen - proceeds from top line to last line in same column. Cursor movement downward - proceeds from last line to top line in same column.

PARITY CHECK - A method of checking the correctness of binary data after that data has been transferred from or to storage. An additional bit, called the parity bit, is appended to the binary word or character to be transferred. The parity bit is the single-digit sum of all the binary digits in the word or character and its logical state can be assigned to represent either an even or an odd number of 1's making up the binary word. Parity is checked in the same manner in which it is generated.

PC (Program Counter) - A special-purpose CPU register which contains the address of the next instruction to be fetched and executed.

PIA (Peripheral Interface Adapter) - An integrated circuit that provides a flexible method of connecting byte-oriented peripherals to an MPU. The PIA features built in registers which not only make it programmable but also provide temporary data storage to simplify data transfer.

PRIORITY - The sequence in which various entries and tasks are processed or peripheral device are served. Priorities are based on analyses of codes associated with an entry or task, or the positional assignment of a peripheral device within a group of devices.

PROCESS - A body of code, a program, performing a specific function(s).

PROGRAM - A complete sequence of computer instructions necessary to solve a specific problem, perform a specific action, or respond to external stimuli in a prescribed manner. As a verb, it means to develop a program.

RAM (Random-Access Memory) - A computer memory structured so that the time required to access any data item stored in the memory is the same as for any other item.

REGISTER - A temporary storage unit which can be implemented as a hardware device or as a software structure and used to store data for manipulation and/or processing reference. Typically, a register consists of a single computer word or a portion of a word.

SERIAL I/O - A method of data transfer between a computer and a peripheral device in which data is transmitted for input to the computer (or output to the device) bit by bit over a single circuit.

SCRATCH PAD MEMORY - Any memory space used for the temporary storage of data. Typically, scratch pad memories are high-speed integrated circuits which are addressed as internal registers.

SERVICE ROUTINE - A set of instructions to perform a programmed operation, typically in response to an interrupt.

SHIFT REGISTER - A register in which binary data bits are moved as a contiguous group a prescribed number of positions to the right or to the left.

STACK - A dynamic, sequential data list, usually contained in system memory, having special provisions for program.

STATIC MEMORY - A type of semiconductor read/write random access memory which does not require periodic refresh cycles.

SUBROUTINE - A short program segment which performs a specific function and is available for general use by other programs and routines.

USER PROGRAM - A program written by the user to make the system perform any desired task.

VECTOR - See INTERRUPT VECTOR.

WORD - A set of binary bits handled by the computer as the primary unit of information. The length of a computer word is determined by the hardware design. Typically, each system memory location contains one word.

APPENDIX B

ASCII CHARACTER SET

ASCII CHARACTER SET

Name of Character	Comments	7-BIT STD CODE Decimal Value	Octal Value	Hex Value
NUL	Null or tape feed	0	000	00
SOH	Start of heading	1	001	01
STX	Start of text	2	002	02
ETX	End of text	3	003	03
EOT	End of transmission	4	004	04
ENQ	Enquire (who are you, WRU)	5	005	05
ACK	Acknowledge	6	006	06
BEL	Bell	7	007	07
BS	Backspace	8	010	08
HT	Horizontal tab	9	011	09
LF	Line feed	10	012	0A
VT	Vertical tab	11	013	0B
FF	Form feed	12	014	0C
CR	Carriage return	13	015	0D
SO	Shift out (to red ribbon)	14	016	0E
SI	Shift in (to black ribbon)	15	017	0F
DLE	Data Link Escape	16	020	10
DC1	Device control 1	17	021	11
DC2	Device control 2	18	022	12
DC3	Device control 3	19	023	13
DC4	Device control 4	20	024	14
NAK	Negative acknowledge	21	025	15
SYN	Synchronous idle	22	026	16
ETB	End of transmission block	23	027	17
CAN	Cancel	24	030	18
EM	End of medium	25	031	19
SUB	Substitute	26	032	1A
ESC	Escape, prefix	27	033	1B
FS	File separator	28	034	1C
GS	Group separator	29	035	1D
RS	Record separator	30	036	1E
US	Unit separator	31	037	1F
SP	Space or blank	32	040	20

ASCII CHARACTER SET (cont'd)

Name of Character	Comments	7-BIT STD CODE		
		Decimal Value	Octal Value	Hex Value
!	Exclamation point	33	041	21
"	Quotation marks (dieresis)	34	042	22
#	Number sign	35	043	23
$	Dollar sign	36	044	24
%	Percent sign	37	045	25
&	Ampersand	38	046	26
'	Apostrophe (acute accent, closing single quote)	39	047	27
(	Opening parenthesis	40	050	28
)	Closing parenthesis	41	051	29
*	Asterisk	42	052	2A
+	lus	43	053	2B
,	Comma (cedilla)	44	054	2C
-	Hyphen (minus)	45	055	2D
.	Period (decimal point)	46	056	2E
/	Slant	47	057	2F
0	Digit 0	48	060	30
1	Digit 1	49	061	31
2	Digit 2	50	062	32
3	Digit 3	51	063	33
4	Digit 4	52	064	34
5	Digit 5	53	065	35
6	Digit 6	54	066	36
7	Digit 7	55	067	37
8	Digit 8	56	070	38
9	Digit 9	57	071	39
:	Colon	58	072	3A
;	Semicolon	59	073	3B
<	Less than	60	074	3C
=	Equals	61	075	3D
>	Greater than	62	076	3E
?	Question mark	63	077	3F
@	Commercial at	64	100	40

ASCII CHARACTER SET (cont'd)

Name of Character	Comments	7-BIT STD CODE		
		Decimal Value	Octal Value	Hex Value
A	Upper case letter A	65	101	41
B	Upper case letter B	66	102	42
C	Upper case letter C	67	103	43
D	Upper case letter D	68	104	44
E	Upper case letter E	69	105	45
F	Upper case letter F	70	106	46
G	Upper case letter G	71	107	47
H	Upper case letter H	72	110	48
I	Upper case letter I	73	111	49
J	Upper case letter J	74	112	4A
K	Upper case letter K	75	113	4B
L	Upper case letter L	76	114	4C
M	Upper case letter M	77	115	4D
N	Upper case letter N	78	116	4E
O	Upper case letter O	79	117	4F
P	Upper case letter P	80	120	50
Q	Upper case letter Q	81	121	51
R	Upper case letter R	82	122	52
S	Upper case letter S	83	123	53
T	Upper case letter T	84	124	54
U	Upper case letter U	85	125	55
V	Upper case letter V	86	126	56
W	Upper case letter W	87	127	57
X	Upper case letter X	88	130	58
Y	Upper case letter Y	89	131	59
Z	Upper case letter Z	90	132	5A
[	Opening bracket	91	133	5B
\	Reverse slant	92	134	5C
]	Closing bracket	93	135	5D
^	Circumflex	94	136	5E
_	Underline	95	137	5F

ASCII CHARACTER SET

Name of Character	Comments	7-BIT STD CODE		
		Decimal Value	Octal Value	Hex Value
'	Quotation mark	96	140	60
a	Lower case letter a	97	141	61
b	Lower case letter b	98	142	62
c	Lower case letter c	99	143	63
d	Lower case letter d	100	144	64
e	Lower case letter e	101	145	65
f	Lower case letter f	102	146	66
g	Lower case letter g	103	147	67
h	Lower case letter h	104	150	68
i	Lower case letter i	105	151	69
j	Lower case letter j	106	151	6A
k	Lower case letter k	107	153	6B
l	Lower case letter l	108	154	6C
m	Lower case letter m	109	155	6D
n	Lower case letter n	110	156	6E
o	Lower case letter o	111	157	6F
p	Lower case letter p	112	160	70
q	Lower case letter q	113	161	71
r	Lower case letter r	114	162	72
s	Lower case letter s	115	163	73
t	Lower case letter t	116	164	74
u	Lower case letter u	117	165	75
v	Lower case letter v	118	166	76
w	Lower case letter w	119	167	77
x	Lower case letter x	120	170	78
y	Lower case letter y	121	171	79
z	Lower case letter z	122	172	7A
{	Opening brace	123	173	7B
¦		124	174	7C
}	Closing brace	125	175	7D
∿	Equivalent	126	176	7E
DEL	Delete	127	177	7F

MOTOROLA INC.
Integrated Circuits Division

Microsystems • P.O. Box 20912 • Phoenix, Arizona 85036

12437-1 PRINTED IN USA (3/79) MPS 1M