PRELIMINARY

INSTRUCTION MANUAL

MODEL 480

VIDEO COMMUNICATIONS TERMINAL

(TRW)

CONRAC CORPORATION

Conrac Division, 600 North Rimsdale Avenue, Covina, Ca. 91722

This will be a

3/4 right front view (photograph)

of the unit.

Figure 1-1. Model 480 Video Communications Terminal

RC 480 DISPLAY.

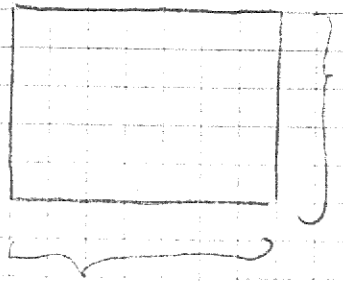
I GENERELT OM SYSTEMET SE AA 241.

II GENERELT OM DISPLAY SE AA 239

III GENERELT OM KEYBOARD SE AA 240.

IV PRINCIP I DISPLAY.

80 CHARACTERS. (80 · 6 = 480)

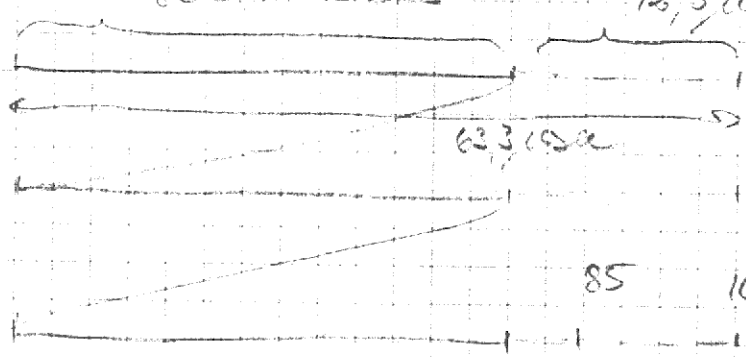


315 LINES SOM I NOR-MALT TV.

HVER RASTERTIME = 63,3 μsec

63,3 μsec · 315 = 50Hz. 50Hz = REFRESHTIME.
50Hz ~ 20 msec.

DA DER SKAL RUGES TID TIL STRÅLENS TIL-BAGELØB INDDELES TIL 100 CHARACTERS



COUNT 80 ⇒ BLANK VIDEO

COUNT 85 ⇒ STOP CLOCKING & MEMORY

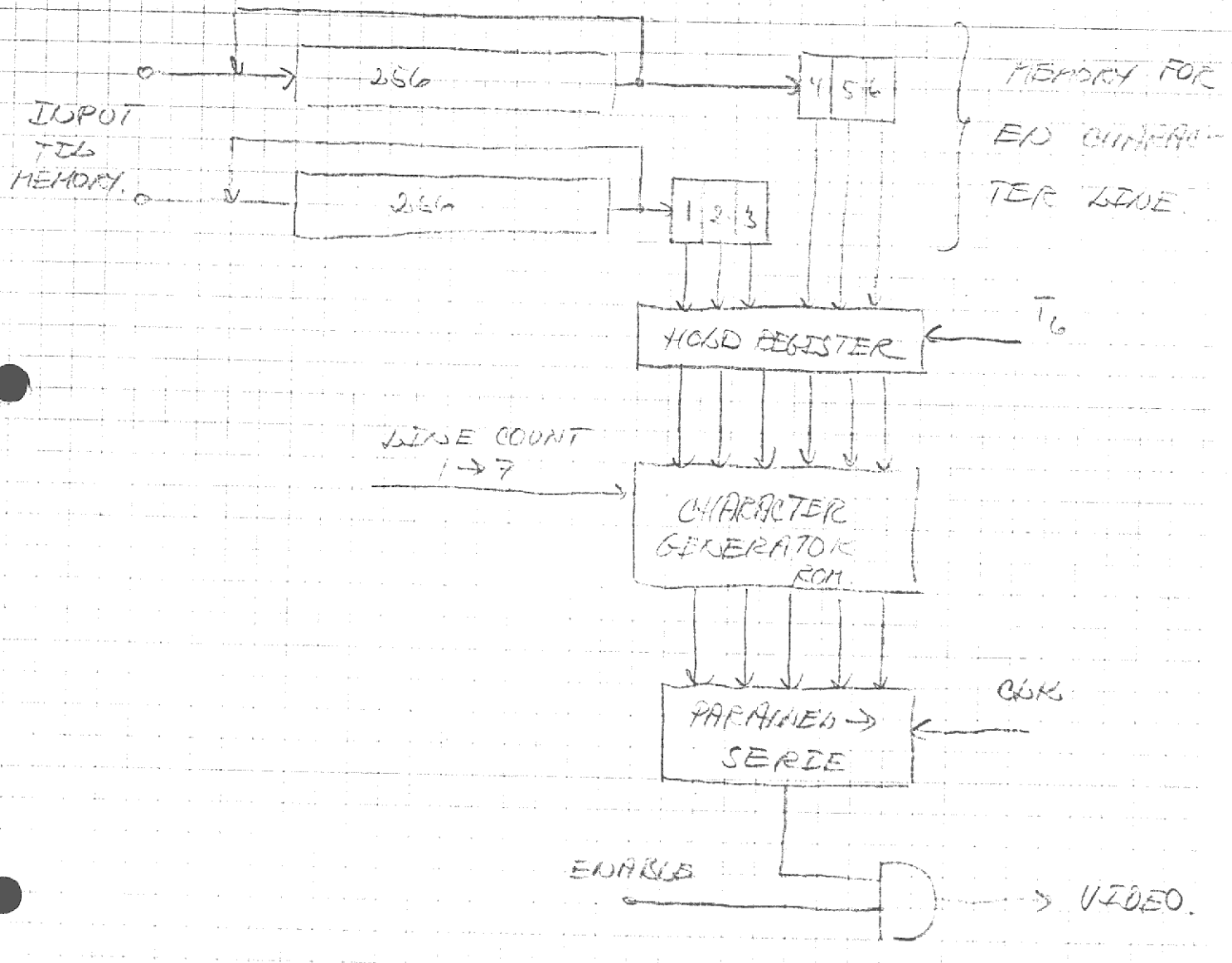
3 CLK / CHARACTER ~ 255 CLK ~ STOPREGISTER AF MEMORY.

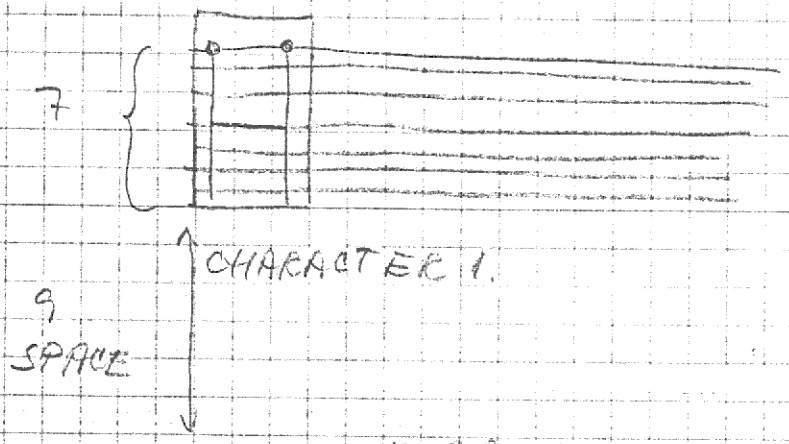
COUNT 80 MEDFØRER OGSÅ START TILBAGEGANG PÅ STRÅLEN.

COUNT 100 ⇒ RESET COUNTER (CHARACTER-COUNTER OG START SCAN MEMORY IGEN)

HVER CHARACTER OPBYGGES AF EN 5·7 MATRIX OG DER BRUGES 1 SØJLE TIL SPACE ⇒ 6·100 = 600 DOTS

FREKVENS = 50 Hz · 315 · 100 · 6 = 9,45 MHz.



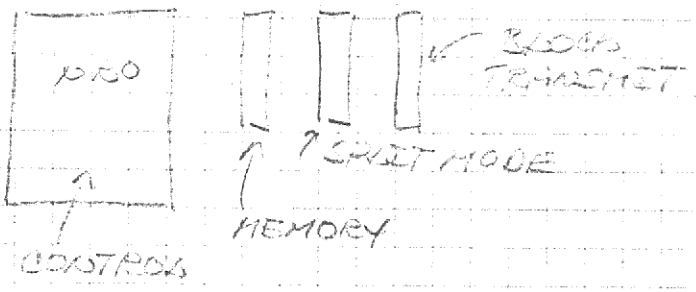


16 RASTERLINJER / CHARACTER LINE.

8 CHARACTERLINJER ⇒ 16·8 = 128 RASTERLINJER

V START PÅ HARDWARE.

PRINCIP FOR SIGNALNAVNE
PRINCIP FOR GATEKOMMENTERING BÅDE PÅ PCB'A OG PÅ SCHEMATICS.



VI TIMING GENERATOR.

VII MEMORY.

VIII CONTROL FUNCTION DECODE.

RC 480 DISPLAY.

I RECEIVE DATA FRÅN DATABASE
(WRITE DATA I MEMORY)

II CHARACTER GENERATOR.

III GRUNDFUNKTIONER MED MODIFIKATIONER.

IV CHARACTER INSERT OCH DELETE.

III (1)

RC 480 DISPLAY

I FORMAT MODE.

II CLEAR DISPLAY, CLEAR END OF DISPLAY, CLEAR ENTRY.

III LOAD CURSOR ADDRESS.

IV VIDEO

V POWER SUPPLY

VI KEYBOARD.

VII PRAKTIK.

TTY TELETYPE → CPU → TELETYPE.

100	MOV 10 (TTI)	060210
101	MOVB 10 (TTI)	060110
102	BKPT 10	063514
103	JMP. -1	000777
104	DEA 2,10	070410
105	MOV 11 (TTO)	060211
106	DOAC 2, 11	073111
107	SKPBZ 11	063511
110	JMP. -1	000777
111	JMP. - 10	000770

FRA REGISTRE → CPU → DISPLAY

1	ICRST		062677
2	READS	0,77	060477
3	DOAS	0,11	060111
4	SKRES	11	063511
5	JMP. -1		000777
6	HALT		063077
7	JMP 1.		000001

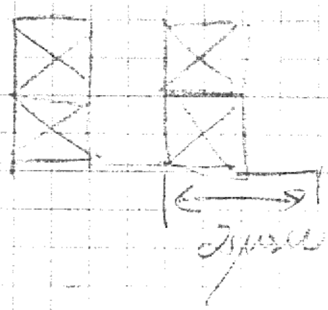
DISPLAY [illegible]

1) generation display methods
2) GRAPHIC DISPLAY
3) LOGIC TYPES of DOT GENERATOR
4) DATA FORMAT
5) MEMORY
6) TIMING SYSTEM
7) CHARACTER GENERATOR

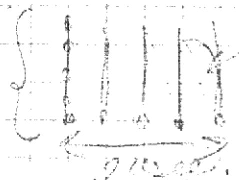
I ① ②

1) Sigma bogplaat

2) dobbelt hung window

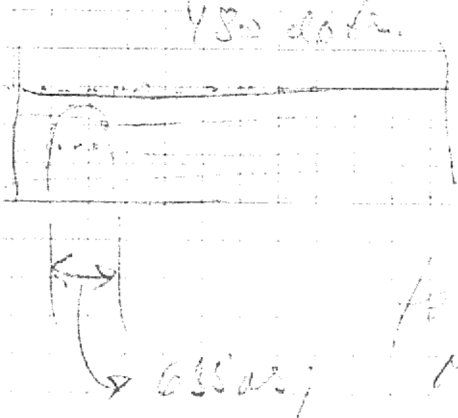


3) 7 dots



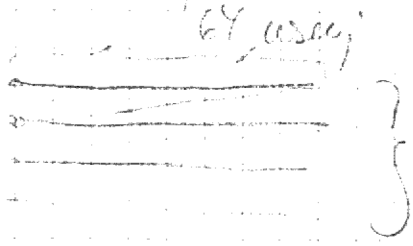
strøm lines 4-5-6 går ned,
krever hurtig
deflektions amp.

4)

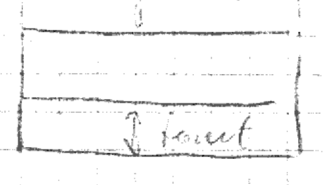


480 dots i hver linie.

first kant mulig efter
MSI og sist



315 lines mulige



kanaler linie
127

100 CHARACTERER
63,5 μsec → 30 μsec
60 Hz

50 CHARACTER → 51 μsec

51 μsec $\frac{51}{60}$ ≈ 635 μs

μsec denne tid bruges således til at komme tilbage.

80 CHARACTER.

7 lines

til 6 DOTS & hver CHARACTER.

5 dots bruges til hvert karakter

vi har altså 5 · 7 matrix.

vi laver skip 9 lines for næste karakter

altså 16 lines for hver karakter

frekvensen for dots

60 Hz · 315 · 100 · 6 = 9,45 MHz,

der bruges 122 lines

16 | 122 | 8

16) 315 | 15 | 155 | 11

20 synlige linier.

da vi kun optyser halvdelen af feltet

⇒ frekvensen = da doblet af 9,45 MHz

6 prod

120 ns til 6 DOTS ⇒

106 ns/DOT

60 Hz ⇒ 3 msec

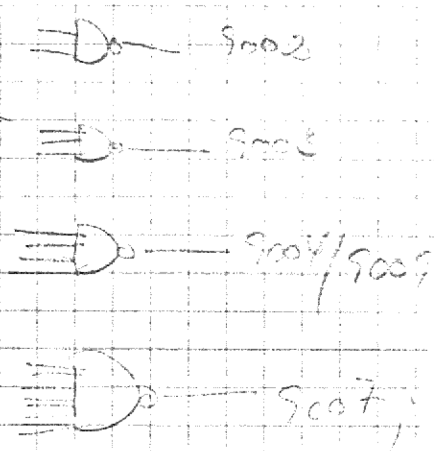
63,3 μsec · 315 linier

⇒ 3 msec,

63,3 · 315

315 · 633,0

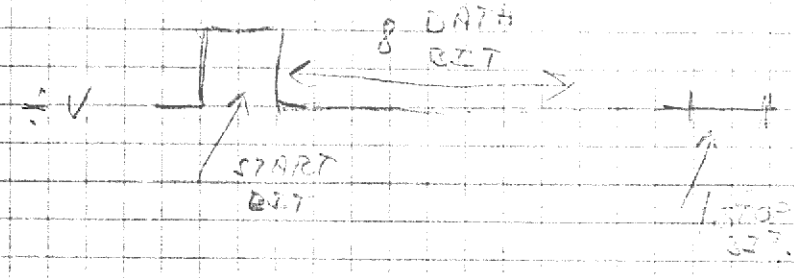
156 ns/DOT, vi oplyser af 18990 · 9 kun halvdelen 19939,5



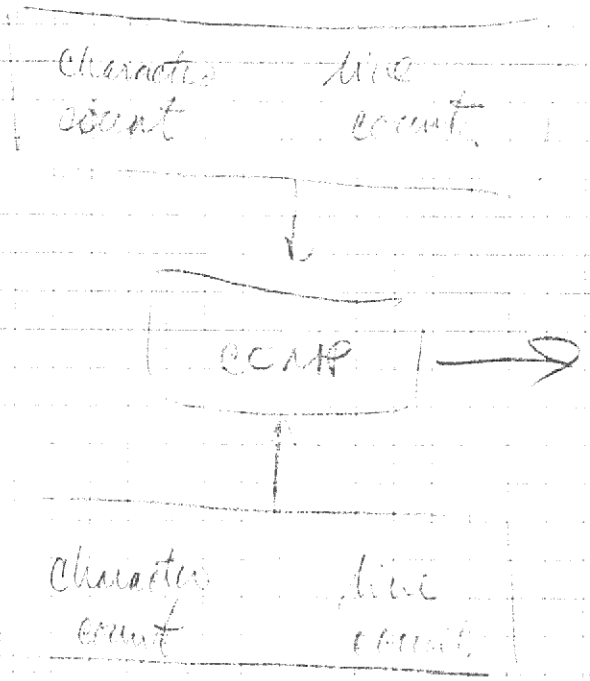
EIA RS-232C
(modem interface ...)

logic zero / ON

logic one / OFF

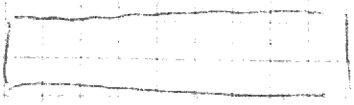

8 DATA BIT
START BIT

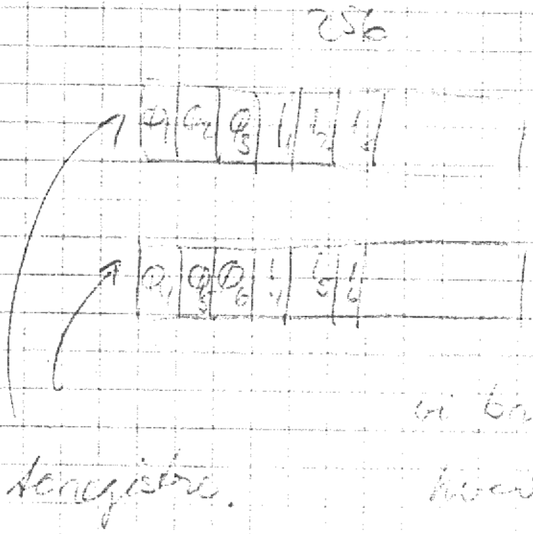
CURSOR REG.


character count
line count
COMP
character count
line count

MEMORY

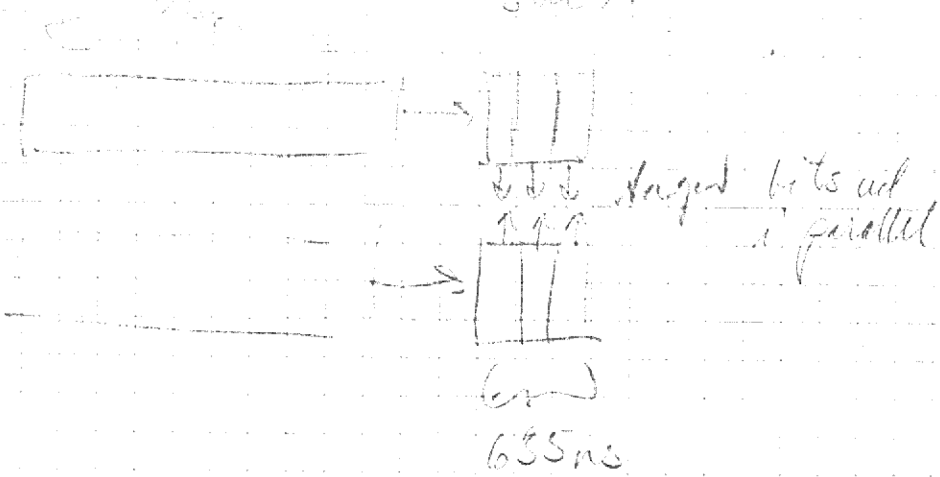
256 bit skifte[illegible]





vi bruger [illegible] 240 positioner i hver.

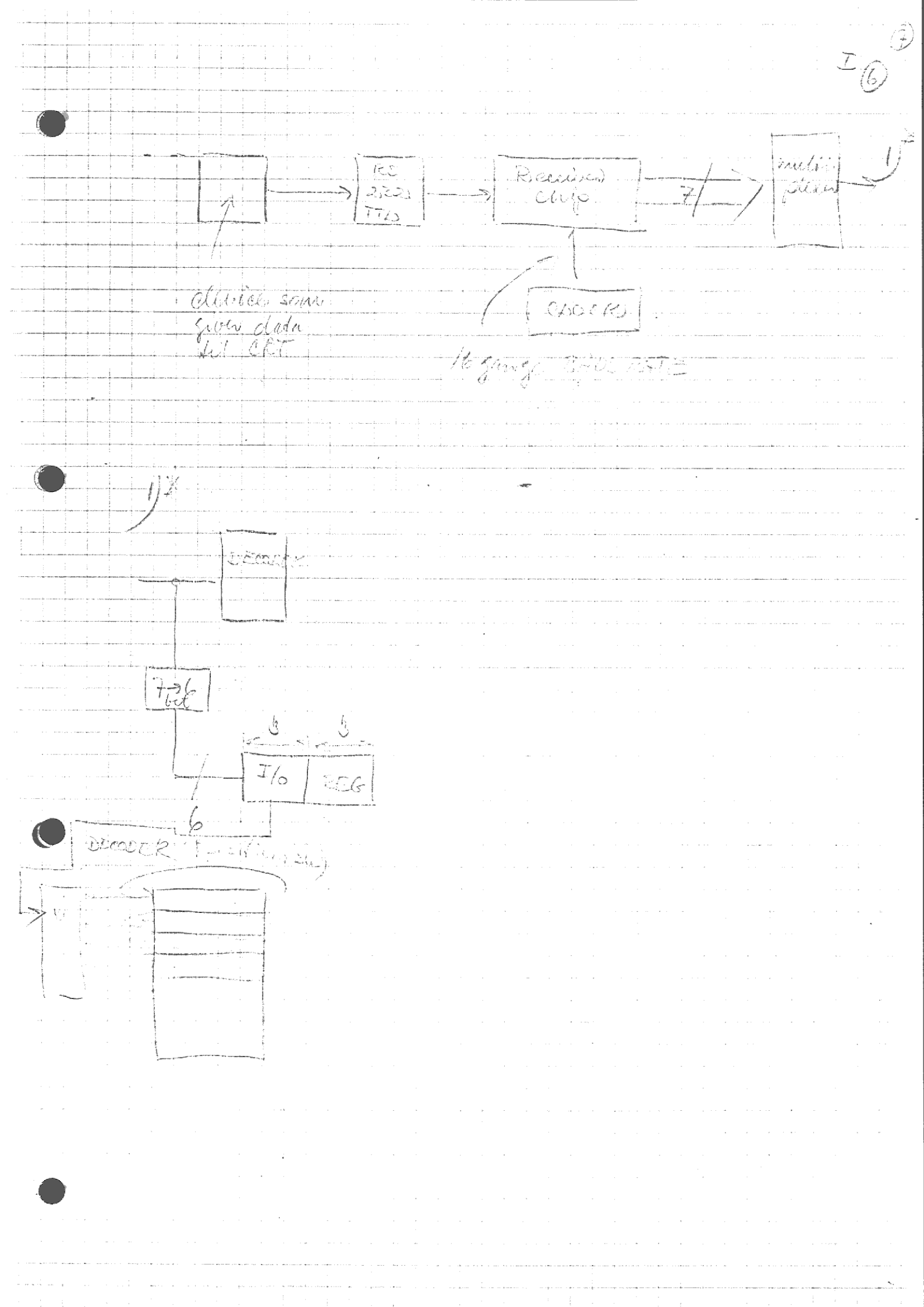
[illegible] dynamiske [illegible] det [illegible] [illegible] [illegible] for at holde informationen.



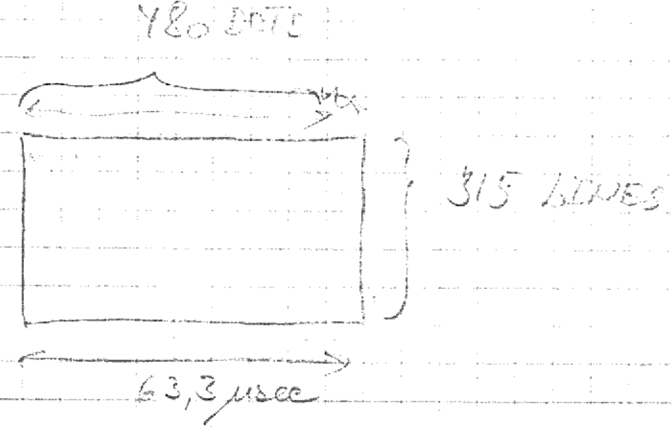
8 CHARACTER / BYTE

6 BIT / CHARACTER

15 MEMORY POSITIONER / BYTE.


I/O
ECG
7 bit
b
b

I (7)

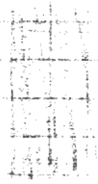


50 Hz ~ 20 msec; REFRESHTIME.

$63{,}3\,\mu sec \cdot 315 \cong 20\,msec$,

480 DOTS ~ 80 CHARACTER.

SKÆRMEN INDDELES SOM TIL 100 KARAKTERER, DE SIDSTE 20 BRUGES TIL TILBAGELØB.



$6{,}33\,\mu sec \cdot 80 \cong 51\,\mu sec$ altså $12{,}3\,\mu sec$ til tilbageløb.

II (1)

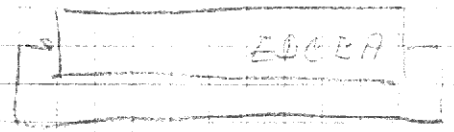
1) PRINCIP. MEMORY

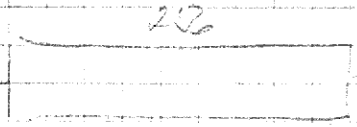
2) CORRELATION

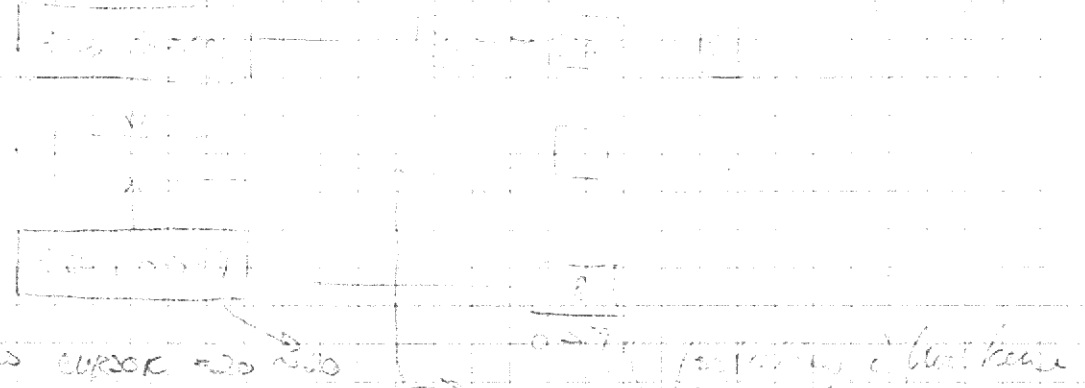
3) MEMORY TIMING, MEMORY EDIT,

4) I/O PROCEDURE I MORE,

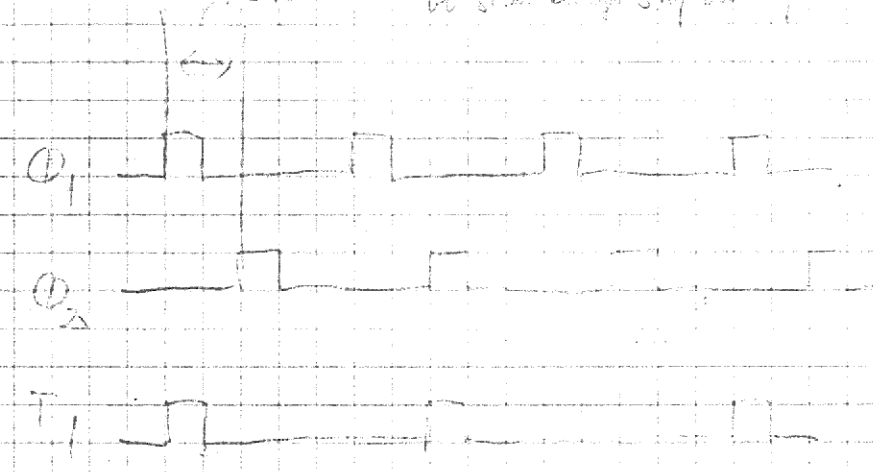
PRIMARY MEMORY

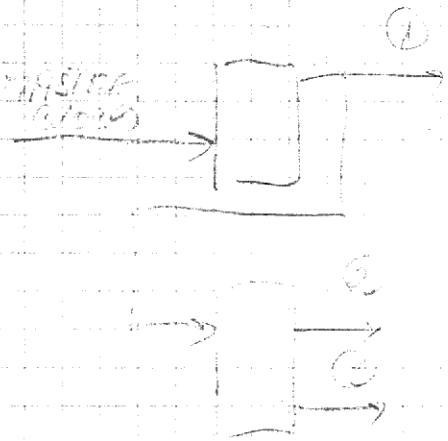


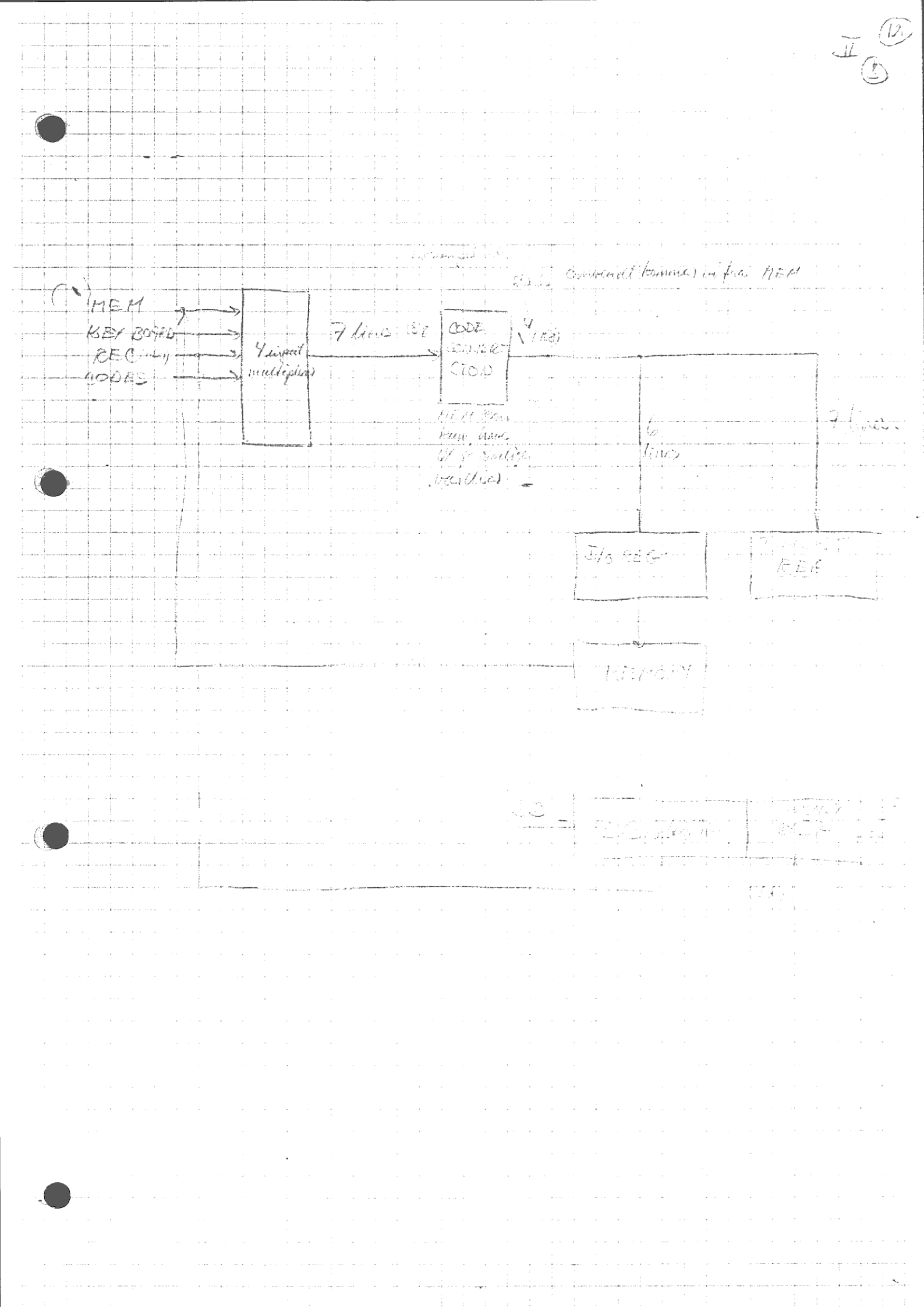




MEMORY TIMING.






MEM
KEY BOARD
REC
MODES
4 input multiplexer
7 lines
CODE CONVERT CODE
6 lines
7 lines
I/O REG
REG
MEMORY

- LCA

LINE 100

MS BYTE

LS BYTE

1) WRITE
2) READ.
3) INSERT og DELETE.
4) POWER SUPPLY
5) VIDEO [illegible]
6) [illegible]
7) MÅLING PÅ MASKINEN.

IV (1)

1) KEYBOARD

2) DECODING

3) TIMING

4) OSC & DIVIDER

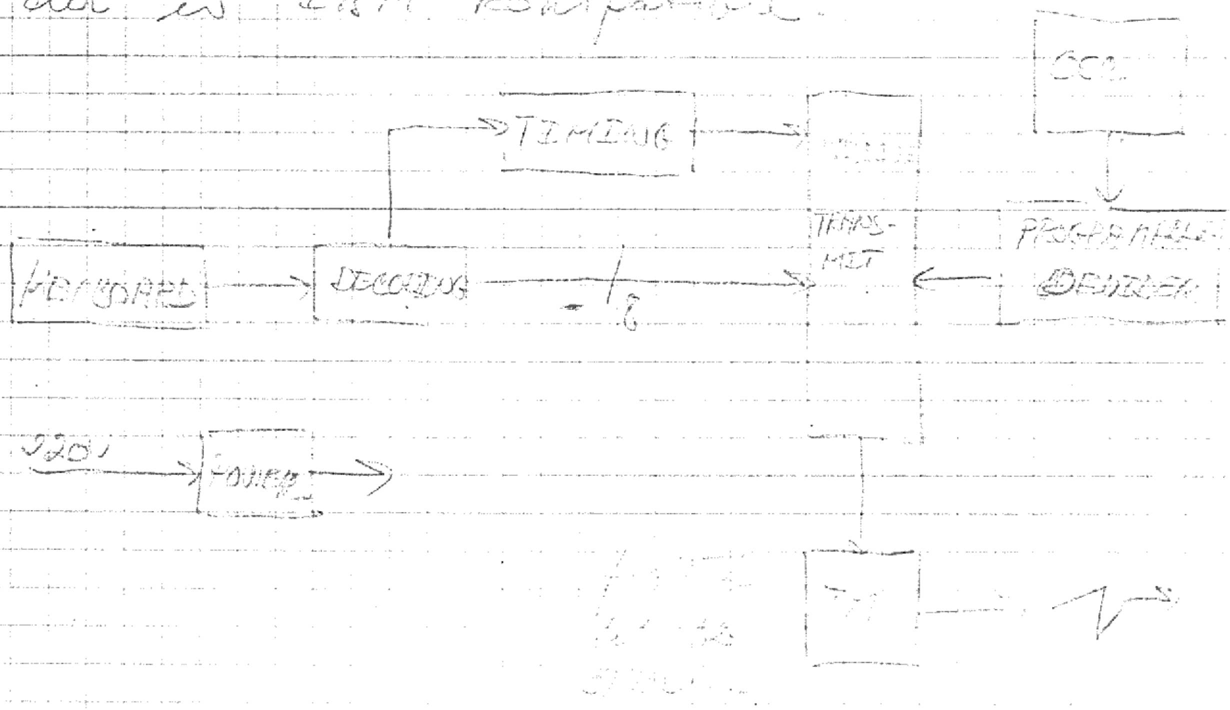
5) RECEIVER / TRANSMITTER

6) TTL → TTY STANDARD

7) POWER SUPPLY

- - KEYBOARD.

Der er lavet modifikationer [illegible] man ønsker [illegible] keyboard der er IBM kompatible.



CHAPTER I

GENERAL DESCRIPTION

1-1. INTRODUCTION

This instruction manual contains a general description, installation instructions, post installation adjustments, a checkout after installation procedure, and operating instructions for the Model 480 Video Communications Terminal (VCT).

1-2. GENERAL DESCRIPTION

The Model 480 Video Communications Terminal (see figure 1-1) is basically a fully buffered terminal that stores and displays up to eight 80 character lines. The unit operates at switchable line rates from 110 to 9600 BAUD in a block transmit mode or a full duplex teletype mode. Additional features include a bell to attract operator attention and addressable cursor position.

The edit functions provided which make for more flexible operation and improve operator convenience include character insert/delete and split format (forms protect).

1-2. GENERAL DESCRIPTION (Cont'd.)

The VCT uses a keyboard essentially identical to the standard teletype keyboard so that no retraining of operators is necessary. For performance and reliability, the VCT uses the latest technologies including an MOS memory. To reduce operator fatigue, the screen is refreshed at a flicker free rate of 60 Hz.

The VCT has three operational modes: 1) on-line (TTY), 2) off-line (local), and 3) block transmit. Each of these modes has availble two types of format operation: 1) general (unprotected) format and 2) split (partially-protected) format. Each mode or type of format is defined by a function code which can be issued by the communication interface or the keyboard control keys.

1-3. PHYSICAL DESCRIPTION

The size and form factor of the VCT is the same as some popular electric typewriters, so that it easily fits on all standard secretarial desks and alcove "L's".

The VCT basically consists of three major units: a 12-inch CRT display; an operator keyboard; and a controller unit. These units are integrated into a single free-standing chassis assembly with overall dimensions as shown in figure 1-2.

1-4. LEADING PARTICULARS

The physical, mechanical and electrical characteristics of the Video Communications Terminal are listed in table 1-1.

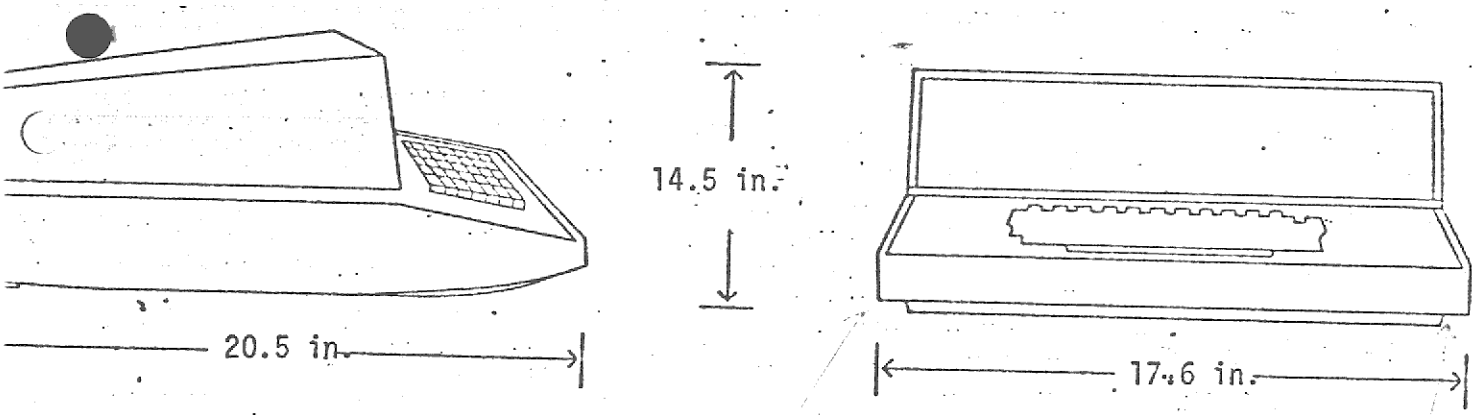


Figure 1-2. Model 480 Video Communications Terminal Overall Dimensions

Table 1-1. Model 480 Video Communications Terminal Quick-Reference Data

Item	Characteristic
Data Interface	Conforms to EIA standard RS232C
Display Characteristics	
Refresh Rate	60 Hertz
Phosphor	EIA-P-4 Aluminized EIA-P-39 Aluminized (Optional)
Deflection Method	Electromagnetic
Focus Method	Electrostatic
Memory	480 or 640 Characters MOS Shift Register
Code Set	64 ASCII
Character Set	64 Displayable Characters
Character Generator	5 x 7 dot matrix
Display Format	The viewable screen area is 9 inches wide by 7.0 inches high with the long axis of the CRT in the horizontal plane.
Character and Display Dimensions	
Lines/Display	8 (max.)
Characters/Line	80 (max.)
Page Width, inches	6.0 inches Nominal
Page Length	8.0.inches Nominal
Character Width	.067 inch Nominal

Table 1-1. Model 480 Video Communications Terminal
Quick-Reference Data (Cont'd.)

Item	Characteristic
Character Origin to Character Origin Spacing	.112 inch Nominal
Character Height	.150 inch Nominal
Line Origin to Line Origin Spacing	.250 inch Nominal
Operating Temperature Range	+10°C to +40°C at 90% Relative Humidity
Operating Barometric Pressure Range	30 to 20.58 inches of Hg.
Storage Temperature Range	-20°C to 75°C
Shock and Vibration	10 to 50 Hz with "G" factor of 1.3 to 5, 30 minutes maximum
Warmup	CRT display stabilizes within 15 minutes after turn-on.
Input Power	120/240Vac ±10%, single phase, 60 ±2.0 Hz, 200 watts maximum
Physical Dimensions	See figure 1-2.
Video Display Assembly	
Input Signals	Three separate wires containing video, horizontal and vertical information at logic levels. Video is up for kinescope illumination, down for cutoff. Horizontal and vertical sync pulses are negative going.
Input Signal Range	For logic level signals, up level must be from +2.4 to 5.5 volts, down level must be from 0.0 to 0.6 volts.

Table 101. Model 480 Video Communications Terminal
Quick-Reference Data (Cont'd.)

Item	Characteristic
Input Signal Timing	For full width of display, video occupies 48.0 to 51.0 microseconds of each horizontal line. Horizontal retrace time is 12.5 microseconds maximum. Vertical retrace time is 1500 microseconds maximum.
Input Connections	All input connections are made through a 15-circuit, printed wiring board edge connector.
Video Gain	30 volts peak-to-peak at cathode for minimum logic level.
Video Response	Transient response to the positive or negative step input of voltage with the intensity control set to produce 30 volts change is 30 nanoseconds (10% to 90%) with less than 5% overshoot measured at the cathode of the kinescope.
Brightness	The peak light output in the center of the video dot is 40 foot Lamberts minimum for a kinescope with no faceplate and a P4 phosphor.
Kinescope	9-inch - 230MB4, 12-inch - ST4500A. Standard tubes are supplied with P4 phosphor and a tension band for implosion protection.

CHAPTER II

INSTALLATION

2-1. INTRODUCTION

This chapter contains unpacking and inspection instructions, an installation procedure including cable connections, post installation adjustments, and a checkout after installation test.

2-2. UNPACKING AND INSPECTION

The VCT is packaged and shipped in a heavy duty corrugated paper carton. An Instant-Pak foam is used for packing to insure against possible damage during shipment. Remove the VCT from the container carefully to avoid breaking or otherwise damaging it. Avoid penetrating the container with sharp tools or hammers.

After the VCT has been unpacked, inspect carefully for obvious physical damge that might impair proper operation. Inventory the contents of the shipping container to assure that all necessary hardware has been supplied. If the VCT is damaged, or parts are missing, report this fact through normal channels.

2-3. INSTALLATION

The VCT is supplied as a complete unit and requires no additional material for installation other than interconnecting cabling.

The VCT is a cabinet-bench mount unit. The cabinet mount merely requires removal of the VCT from the packing carton and positioning it in the desired location on a bench or shelf.

2-3-1. Interconnecting Cabling

Only two cable assembly connections are required to place the VCT into operation. The power cable is hard wired at the VCT end and merely requires plugging into a standard wall outlet.

The data cable assembly is not supplied and must be fabricated by the OEM user. The connector type is a Cinch DB-25S or equivalent with pin assignments as itemized in Table 2-1, and Interface Signal assignments as tabulated in Table 2-2.

2-3-2. Post Installation Adjustments

After mechanical installation of the VCT is complete, and all cabling connections have been made, the adjustments outlined in the following subparagraphs should be checked.

There are only two adjustments required after mechanical installation of the VCT. Place the Power ON/OFF switch to ON, allow a 20 minute warmup and visually determine if either of the following adjustments is necessary:

A. INTENSITY Potentiometer. This is a slide-type potentiometer which is located at the right front side under the keyboard housing. Manually adjust for desired level of intensity.

B. HORIZONTAL POSITIONING Potentiometer (R11). This potentiometer is a slotted-head type, and is located at the left rear side of the unit. To gain access, remove the rear cover plate. Before making this adjustment, at least one full line of characters should be displayed on the screen. R11 is then adjusted as required to produce a centered, undistorted display. Check the first and last characters on a line to ensure that both are properly displayed.

NOTE

The following adjustments apply to the Video Display Assembly. Make no adjustments until the VCT has been operating a minimum of 20 minutes.

C. VIDEO DISPLAY ASSEMBLY, IMAGE ADJUSTMENT. If the image appears skewed relative to the kinescope sides, this may be corrected by slightly rotating the deflection yoke. If the image appears to be off-center relative to the kinescope slides, the centering magnets on the rear of the deflection yoke should be adjusted. The height of the displayed image may be adjusted using R60 on the main board and the width may be adjusted using L1 on the main board.

2-3-3. Post Installation Checkout

After installation, and after all necessary adjustments have been made, perform the following procedural steps in sequence to assure that the VCT is functioning properly.

Before performing the Post Installation Checkout, it is recommended that the reader familiarize himself with the operation of the VCT which is described in Chapter III.

A. KEYBOARD ENTRY AND DISPLAY TEST

In this test, data is entered from the keyboard and displayed on the VCT to verify that the keyboard interface, CRT display, and associated circuits are functioning properly.

1. Place the power ON/OFF switch to ON.

2. Press the CTRL key and hold, then press the "G" key. The tone of the bell will be heard, and no information will be displayed on the VCT screen.

3. From the keyboard, enter and display two lines of characters. Backspace a few characters, then press and hold the CTRL key. Press "I" key and visually observe all characters on the second line have been erased. With CTRL key pressed, press ERS key to clear the screen.

4. From the keyboard, enter and display three lines of characters. Press the HOME key and observe cursor goes to home. From the keyboard, enter and display approximately ten (10) more characters. Press and hold CTRL key, then press "K" key. All characters to the right of the cursor, and all lines below the cursor, will erase. With CTRL key pressed, press ERS key to clear screen.

5. Press and hold the CTRL key, then press "H" key several times in succession. Observe cursor steps down one (1) line each time "H" key is pressed.

6. At the keyboard, enter a message on the screen with a colon at the end. Press and hold the CTRL key, then press the "Y" key. Repeatedly press "BS" key and observe cursor will not backspace past the colon. Enter an additional message from the keyboard and observe characters are displayed to the righ of the colon. No characters can be entered to the left of the colon. With CTRL key pressed, press ERS key to clear the screen.

7. At the keyboard, enter a message on the screen with a colon at the end. Set forms by pressing and holding the CTRL key, then press "Y" key. Enter additional characters on the screen (approximately 10). Press and hold the CTRL key, then press the "X" key. All characters to the right of the colon will be erased. With CTRL key depressed, press thw "L" key and observe all remaining characters on the screen are erased.

8. At the keyboard, verify cursor is in the home position. If not, press HOME key. Press and hold CTRL key, then press "Q" key. Enter address to which cursor is to jump. Press keys 1, 7, and 9. Verify cursor moves to line 1, character location 79. Repeat for the following addresses:

200, 250, 279
310, 360, 379
405, 440, 479
502, 535, 579

At the conclusion of the test, press HOME key, and observe cursor returns to the home position.

9. At the keyboard, enter and display a minimum of three (3) lines of characters. Select a character and line location within the display (i.e., 150), at which you would like to insert a character. Jump cursor to this location as described in step 8. Press the CTRL key and hold, then press the "O" key. Verify all characters from the cursor to the end of the line will move one (1) character to the right. Do not clear the display.

10. Select a character and line location at which you would like to delete a character (i.e., 250). Jump cursor to this location as described in step 8. Press the CTRL key and hold, then press the "P" key. Verify that the character at the cursor is deleted, and all characters to the right of the cursor are shifted one location to the left. Again press the CTRL key and then the "P" key.

B. BACK-TO-BACK CHECKOUT

In this test, the output of the VCT is tied back to the input to provide a complete loop check of the unit. In order to perform this test, a Cinch DB25S or equivalent test connector is required with pins jumpered as follows:

2 to 3
4 to 5 and
6 to 20

After the pins have been jumpered, remove data connector, if any, from J4 at the left rear side of the VCT and install the special test connector in its place.

1. Place Power ON/OFF switch to ON. At the keyboard, press the CTRL key and hold, then press the TTY key (Model 480 is now in TTY mode of operation).

2. From the keyboard, enter all characters available and verify that each and all are displayed.

3. On the keyboard, press CTRL key and hold, then press and release ERS key. Verify VCT screen is cleared and cursor goes to home position.

4. On the keyboard, press CTRL key and hold, then press and release LOCAL key.

5. At the keyboard, enter one (1) complete line of characters, then press "NEW LINE" key (carriage return).

6. Press and hold the shift key, then press and release SOM key (start of message), then enter additional characters for line 2. Press and release "NEW LINE key. Enter a complete group of characters on line three (3), followed by a "NEW LINE".

7. Press CTRL key and hold, then press and release "Q" key. Enter address 578 for cursor location.

8. Press CTRL key and hold, then press and release XMIT key. Verify cursor jumps to location where SOM (start of message) was entered, and travels down the screen at a rate determined by the operating baud rate. Verify no entered characters changed as a result of this operation.

9. On the keyboard, press the CTRL key and hold, then press the TTY key, followed by the "T" key. Verify cursor goes to home, searches for an SOM (start of message), and travels down the screen at a rate determined by the operating baud rate. Verify the cursor halts at address location 579 and is not blinking.

10. Press CTRL key and hold, then press LOCAL key and release. Verify cursor is now blinking.

11. Place Power ON/OFF switch to OFF, remove special test connector, and reinstall data connector at J4.

This will be a rear view (photograph) of the VCT with Data Connector keyed and nomenclature assigned.

Figure 2-1. Video Communications Terminal, Rear Panel View

Table 2-1. Data Connector J4, Pin Assignments

Pin	EIA Designation	Description	Direction
1	AA	Protective Ground	--
2	BA	Transmitted Data	From VCT
3	BB	Received Data	To VCT
4	CA	Request to Send	From VCT
5	CB	Clear to Send	To VCT
6	CC	Data Set Ready	To VCT
7	AB	Signal Ground	--
.			
.			
20	CD	Data Terminal Ready	From VCT

Table 2-2. Data Connector J4, Interface Signals Assignments

EIA Pin Designation	Signal Nomenclature	Description
AA	Protective Ground	Chassis ground
AB	Signal Ground	Ground reference for interface signals
BA	Transmitted Data	This line shall carry data transmission from the VCT. It shall be held in the OFF state when any one or more of the signals CA, CB, CC, and CD are in the OFF state.
BB	Received Data	This line shall carry all data transmission to the VCT.
CA	Request to Send	The VCT shall place this line in the ON state when it wishes to transmit data on BA. The VCT shall wait for signal CB to be ON before transmitting.
CB	Clear to Send	The Data Set (the first device connected to the VCT by this interface) will place this line in the ON state in the event CA is in the ON state.
CC	Data Set Ready	The Data Set will hold this line in the ON state at all times that it is operational.
CD	Data Terminal Ready	The VCT shall hold this line ON at all times it is prepared to input data on BB.

CHAPTER III

OPERATION

3-1. INTRODUCTION

Operation of the Video Communications Terminal is related directly to the use of the system developed by the OEM, and the operator should refer to the operation manual provided by the OEM for special operating instructions.

This chapter contains a general description of the VCT Keyboard Assembly, a description of the operating formats and modes, and a definition of the communication function codes.

3-2. KEYBOARD DESCRIPTION

The keyboard Figure 3-1, provides the interface between the operator and the VCT. As shown, the keyboard is a modified TTY keyboard with special coding as tabulated in Table 3-1. Note that the mode keys TTY, LOCAL, and XMIT are interlocked with the control key (CTRL).

The keyboard combines solid state keys with solid state encoding, thus eliminating all moving parts with the exception of the key plunger.

Figure 3-1. Video Communications Terminal Keyboard Assembly

The keyboard uses only one circuit board for both switch termination and encoding, thus eliminating circuit board interconnections and assuring improved reliability.

3-3. OPERATING FORMATS AND MODES

The VCT has three operational modes: 1) on-line (TTY), 2) off-line (local), and 3) block transmit. Each of these modes has available two types of format operation: 1) general (unprotected) format and 2) split (partially-protected) format. Each mode or type of format is defined by a function code which can be issued by the communication interface or the keyboard control keys.

The VCT always responds to any mode change codes; in the event of communication activity, after the completion of the current character. Keystroke mode changes are not transmitted on the communication interface.

3-3-1. General Format

In this format the display is considered unprotected. The operator, at his discretion, or the communication link may access any area of the screen.

3-3-2. Split Format

This format provides an efficient means for entering and displaying formatted data. When this format is used, the screen can be

divided in fixed (protected) fields for header information and variable (unprotected) fields for data entry. The COLON (:) is used as the delineator which separates the two fields. The colon most recently entered on a line is the delineator. A line containing no colons will not be protected. To cause the terminal to enter the protected (Split Format) mode, a Set Format (SF) command must be issued by either the keyboard or the communications interface.

NOTE

Because the colon locations are stored in holding registers, erasure or overstrike of the colon will neither clear the register nor keep that line unprotected. Only Clear Display (ERS) command will clear the register; another colon will change the register. Colons entered in the variable field after the SF command are not delineators.

The data from the start of the line to the colon is defined as the fixed field. The variable field is from the right of the colon to the end of the line. There may be only one variable field per line and any line which does not have a delineator present is considered a variable field. Keyboard and communication entry is restricted to the variable field unless the protected area is accessed by use of the LF (Control-H) and LCA operations.

The protected data becomes unprotected by the issuance of a Reset Format (RF) Command. The normal operational sequence when using this format

is: the communication interface will erase the screen, then write the format with colons on the screen, and finally will issue the Set Format Command. An ERS (Control-L) command will place the VCT in General Format and will clear both the display and the delineator registers.

3-3-3. TTY Mode

This mode is a conversational echo mode with valid keystrokes being transmitted to the communication line interface ana displayed on the CRT upon receipt of the echo.

This mode is the only mode in which information can be received from the communication interface. Thus, any message sent from the CPU must be preceeded by the TTY function code.

3-3-4. Local Mode

This is an off-line mode and enables the operator to compose and edit text for subsequent block transmit. No keystrokes are transmitted in this mode. The communication interface must begin each transmission to the VCT with the location function code (LOC) if operation requires the VCT to remain off-line.

3-3-5. Block Transmit Mode

This mode is entered from either the local or the teletype mode by communication link Block Transmit (BXT) code entry or by activation of the Block Transmit Key; in thos mode, data is transmitted to the communication interface as described in Chapter IV, paragraph 4-8-1 and with data transmission, the VCT remains in the Block transmit mode until the mode is

changed by code entry or keyboard control. At this time, the entry of a NAK code will force a retransmit of data beginning with the homing of the cursor. The Block Transmit Key and code are inhibited (locked out) from reinitiating a block transmission until the mode is changed.

3-4. FUNCTION CODES

The function codes are commands of which there are two types, those issued by the communication interface and those generated by the operator at the keyboard control keys. A description of each of the function codes is provided in the following sub-paragraphs. The Terminal USASCII Code Set is shown in Table 3-2.

3-4-1. Communication Interface Commands

These are commands that the VCT must respond to when they are presented to the VCT by the communication interface. These commands can be generated by the keyboard but the keycaps do not have legends for operator use.

A. LOAD CURSOR ADDRESS (LCA)

LCA will condition the VCT to move the cursor to a position specified by the succeeding three characters. The VCT will interpret the three characters succeeding the LCA, as follows: First character, vertical position; Second character, most significant horizontal position, and Third character, least significant horizontal position.

Addressing is sequential 0-5 vertically and 0-79 horizontally. For example, to position the cursor to the ninth location of the third line, the commands would be LCA 208. The VCT is not constrained to detect illegal addresses, and an LCA command with an illegal address will cause unpredictable results.

3-4-1. Communication Interface Commands (continued)

B. Set Format (SF)

Causes the VCT to assume the Split Format operation described in paragraph 3-3-2, and position the cursor to home or the first delineator character beyond the home position.

C. Reset Format (RF)

Causes the VCT to assume the General Format operation described in paragraph 3-3-1.

D. Bell

Causes an audible tone to attract the operator's attention.

3-4-2. Communication Interface and Keyboard Commands

These are commands that the VCT must respond to when presented by the communication interface or the keyboard after a character in transit is completed.

A. Local (LOC)

Places the VCT in the local mode. This keystroke is never transmitted.

B. TTY

Places the VCT in the TTY mode. This keystroke is never transmitted.

C. Block Transmit (BXT)

Places the VCT in the block transmit mode. This keystroke is never transmitted.

3-4-2. Communication Interface and Keyboard Commands (continued)

D. Clear Display (ERS)

Erases the entire display and moves the cursor to the left-most position of the first line (Home). This command also causes the VCT to assume the General Format operation and resets the delineator location registers.

E. Clear to End of Display (EOF)

General Format: Erases all data from the cursor position to the end of page. The cursor does not move.

Split Format: Erases all variable data from the cursor position to the end of page. Cursor does not move.

F. Clear Entry (EE)

General Format: Erases entire display and returns the cursor to Home but does not reset delineator location registers.

Split Format: Erases the variable fields and returns cursor to the right of the first delineator; does not reset delineator location registers.

G. Erase Line (EOL)

General Format: Erases entire line on which cursor is positioned and returns cursor to beginning of that line.

Split Format: Erases variable area of line on which cursor is positioned and returns cursor to right of the delineator on that line.

3-4-2. Communication Interface and Keyboard Commands (continued)

H. New Line (NL)

General Format: Places the symbol (\) on the screen and moves the cursor to the beginning of the next line.

Split Format: Places the symbol (\) on the screen and moves cursor to the right of the delineator found in the succeeding line or the first position of a line with no delineator.

With either format, when the cursor is in the bottom line, the next succeeding line is the first line of the display.

I. Line Feed (LF)

Causes the cursor to move down one line or from the last line to the first without altering any data.

J. Backspace (BS)

Causes the cursor to move left one character position and does not alter any character. The cursor will stop movement at the beginning of the line in General Format or the right of the delineator (if present) in Split Format.

K. Message Cancel (MCN)

This is a special code used by the communication interface and no visual display occurs.

L. Start of Message (SOM)

Places the symbol (]) on the screen. This symbol is used in the block transmit mode.

M. End of Message (EOM)

Places the symbol ( ) on the screen. This symbol is used in the block transmit mode.

3-4-2. Communication Interface and Keyboard Commands (continued)

N. Home (HOM)

General Format: Causes the cursor to move to the left-most position of the first line.

Split Format: Causes the cursor to move to the right of the first delineator or to the first position of the first line if no delineator is present.

O. Character Insert (INS)

Starts at cursor location by shifting that character and all subsequent characters on the same line to the right one location. Character at the end of the line is dropped (no wrap around). The cursor does not move.

P. Character Delete (DEL)

Starts at the cursor location, deletes that character, and moves all subsequent characters on the same line to the right of the cursor one location to the left. A blank character will be inserted at the end of the line. The cursor does not move.

Q. Forward Space (FWS)

This command is formed by activity of the shift and space keys, and causes the cursor to move right one character position up to the 80th character without altering any character.

No legend is presented on the space key to show the shifted FWS. The code transmitted in the TTY mode is the standard space code (unshifted). This function is a keyboard-only activity and the communications interface cannot cause this function.

CHAPTER IV

THEORY OF OPERATION

4-1. INTRODUCTION

This chapter provides a general description of the Video Communications Terminal and a flow diagram presentation which references illustrations located in the back of this chapter.

A glossary of signal mnemonics is provided in Appendix "A". The Mnemonics are listed in alphanumeric sequence along with a descriptive title as to function. A complete set of Logic Diagrams may be found in Appendix "B". Signals on the Logic Diagrams are identified using the mnemonic nomenclature assignment.

4-2. BLOCK DIAGRAM DESCRIPTION

Referring to figure 4-1, the Video Communications Terminal may be divided into functional groups as shown. Basically, the Terminal accepts and responds to information in the form of control function signals or data which are received via the keyboard or interface. Information received from the keyboard is in the form of a 7-bit parallel code. Information received at the interface is in the form of a 10-bit serial asynchronous code.

The keyboard provides the interface between the operator and the Terminal and depending on his action, control signal information or data information are applied to the Terminal. Control signal information is

generally those functions as tabulated in Column "0" and Column "1" of Table 3-2 (see Chapter III). For example mode selection, backspace, forward space, line feed, etc. are all control functions. When any of these functions are implemented, the Terminal recognizes the information as a control function and routes the 7-bit code to the Control Function Decoder. The decoded function is then applied to Control Logic which initiates control signals to the other functional groups of the Terminal, resulting in the control function being executed. Control function information is not written into memory.

Data information would consist of those characters tabulated in Columns "2" through "5" of Table 3-2 (see Chapter III). For example, 0, 1, 2, A, B, C, and etc, are all characters or data information. When characters are input at the keyboard, the 7-bit code for each character is compressed into a 6-bit format and stored in memory and also applied to the input/output group for transmission out the interface as serial data. The memory has the capacity to store six (expandable to eight) lines of information with each line containing eighty characters. Once the characters are stored in memory, the Character Generator interprets the 6-bit codes and responds with video data to the CRT. The Character Generator contains a Read-only-Memory (ROM) in which the character information is stored. The 6-bit code is effectively an address which selects a particular character. The output of the Character Generator is applied as video data to the CRT.

The Master Clock and Timing group generates and sends timing signals to the Character Generator which are used to develop the horizontal and vertical sync information applied to the Video Display Assembly at the

required time intervals. This group also generates the basic timing signals which are used throughout the Terminal to sequentially process all received and transmitted information.

Information received from the interface is applied to the Terminal at the input/output group as serial data. This data for each control function or character is a 10-bit code consisting of one start, seven data, one parity, and one stop bit. The input/output group buffers the data, checks for correct parity, and converts the information to a 7-bit parallel code. This code is then routed in parallel to the Control Function Decoder. Assuming the code is a character, it is loaded into memory. Further processing of the 7-bit codes is dependent on whether it is a control function or a data character, and is identical to that described for keyboard operation.

4-3. FLOW DIAGRAM DESCRIPTION

The flow diagram description presents the Terminal theory of operation based on operational modes and control functions implementation. The supporting flow diagrams are divided into functional circuit blocks with the appropriate circuit component designators referenced within the block. The functional blocks are further identified by reference to the applicable logic daigram.

Timing diagrams are provided for reference and to aid in understanding the operation of the Terminal. These are to be used in conjunction with the flow diagrams contained in this chapter, and logic diagrams located in Appendix "B".

4-4. TIMING GENERATION

The timing requirements for the terminal consists of: (1) the Basic Clock (MTTM1N-MTTM8N) which are used to gate certain functions at a predetermined sequence (2) an I/O Register shift clock (MCLPHN) which is used to shift data into or out of the I/O Register during a read or write operation (3) the Memory Phase Clocks which are used to circulate data between Memory and the Memory Multiplexer, and shift memory data to the Memory Bus Multiplexer, and Edit Memory Multiplexer (4) the Edit Memory Clocks which are used to clock memory data into and out of the Edit Memory during an edit function.

Refer to Drawing 162562-001, Sheet 1 and figure 4-2 for the following discussion. All timing signals are derived from an 9.468 MHz crystal-controlled oscillator located on the control logic assembly (Drawing 162566-001, Sheet 2). This signal (XCLOKN) supplies an input on Drawing 162562-001, Sheet 1, at pin 13 of inverter 205.

Generation of the Basic Clocks begins with the buffered XCLOKN, clocking shift register 109 at pin 8, causing data in the register to be shifted to the right. Timing pulses are produced by shifting a single logic zero bit entered at pins 1 and 2 of 109 through the register at a 9.468 MHz rate.

Assume that the data bit is in timing position three, which is producing a low output at pin 5 of 109 and thus producing a low MTTM3N clock pulse. When XCLOKN goes high, the data bit will be shifted to the right, producing a low MTTM4N clock pulse and terminating the MTTM3N pulse. When the next low-to-high transition of XCLOKN occurs, MTTM5N

will go low and MTTM4N will be terminated. At this time, all inputs to the five-input gate, 209 will be high, producing a low condition at nand gate 209-8. The low input is applied to pins 1 and 2 of 109.

At the next transition of XCLOKN, a low MTTM1N will be generated. MTTM2N-MTTM8N will then be produced in succession with each low-to-positive transition of XCLOKN. These master T-times are generally buffered before being used by the system with both polarities being utilized to accomplish all necessary timing functions.

The memory and Edit Memory clocks generation (see figure 4-2) is initiated on receipt of GENDLN, which is applied to the direct set input of flip-flop 407-5. This sets 407-6 high and enables flip-flop 407-9 to be clocked low at the next low-to-high transition of XCLOKN. This flip-flop will then toggle with each low-to-high transition of XCLOKN. Flip-flop 408-10 is clocked high on the low-to-high transition of XCLOKN and remains in this state until count 86 is detected. Flip-flop 409-10 is clocked low at the low-to-high transition of XCLOKN, and will toggle on every other low-to-high transition of XCLOKN. Flip-flop 409-7 is clocked low on receipt of GDENDP (end-of-line plus T2) and remains low until reset by count 85 and T4.

Note that flip-flops 409 supplies enabling levels to the memory clock gates, and flip-flop 409, pins 9/10 supply enabling levels to the Edit Memory clocks. Gate 406-11 then supplies the final enabling levels as shown, and determines the pulse width of both Memory and Edit Memory clocks. The output of this gate is also the I/O Register shift clock MCLPHN.

4-5. LOCAL MODE

The Local Mode (LOC) is an off-line mode and enables the operator to compose and edit text for subsequent block transmit. In this mode, keystrokes are not transmitted to the interface. The Local mode may be entered either by receipt of the LOC code from the interface or by activation of the keyboard LOCAL key.

For the following discussion, refer to figures 4-3 and 4-4. As shown in figure 4-3, signals KMCTLP and KMLOCP are routed from the keyboard, through connector J9, to the Mode Control section. KMCTLP is generated as a result of the operator actuating the Control (CTRL) key. KMLOCP is generated by the operator when the LOCAL key is activated. This results in output HLOCAP being applied to the Keyboard Cycle Logic as Local Mode Enable. Signal KBSHFT is an enabling level for the shifted forward space gating and will generate a non destructive forward space when the space bar is actuated.

Signal KBSTRP is the keyboard strobe pulse and is generated when any key is actuated. The duration of the pulse is determined by the length of time the key is depressed. When a key is activated, signal KBSTRP and parallel data (KDBT1P-KDBT7P) are supplied as inputs from the keyboard assembly via keyboard connector J9. KBSTRP initiates the keyboard cycle by generating KECYLN which is applied to the Data Multiplexer Control. CMSD1P and CMSD2P will switch the Data Multiplexer to accept the keyboard parallel data KDBT1P-KDBT7P. KFCEIN is also generated as an input to Write Control and initiates the write cycle which will write the keyboard character into memory. Signal KFCEIN generates signals HENBIN, HCLKIN, and HWRIRN.

Signal HENBIN is the parallel load enable pulse for the I/O Register, and it occurs from T_1 to T_1 time. Signal HCLKIN is the parallel load clock for the I/O Register, occuring at T_3 time, and will parallel load the 6-bit keyboard character into the I/O Register.

Signal HWRIRN occurs next, at T_5 time, and is applied to the Read/Write/Edit Logic. This signal will be clocked by T_2, and will remain from T_2 until correlation. The output signal UMENAN (T_2-T_2) is produced after correlation is detected, and is routed to the Line Selector as an enable for the Line Selector Decoder. The Line Selector Decoder determines which line the character will be written on.

At the I/O Register, UMENAN enables memory clocks (MCLPHN) to serially shift the 6-bit character (MIDA1P-MIDA6P) into memory after correlation has occurred. Note that signals (ULIN1N-ULIN6N) select which line the character will be written on, in memory. The Memory Multiplexer recirculates the memory data back to memory during the time a write cycle is not being executed, so that no data is lost.

The output of the Memory Multiplexer is parallel data (MCD11N-MCD62N) which is routed to the Memory Bus Multiplexer for routing to the character generator to be displayed. The control of which line the character will be displayed on is determined by signals GRLL1P-GRLL4P; these are supplied as inputs from the Character Generator, Line Counter. The output from the Memory Bus Multiplexer is two lines, each having three serial bits. One line represents the 3-LSB's (MGDAIN) and the remaining line represents the 3-MSB's. In the Character Generator Data Bus, the two lines of serial data are converted to a 6-bit parallel character (MDAT1P-MDAT6P) which is applied to the Character Generation Section.

The Character Generation Section consists of a ROM (401) and two 4-bit universal shift registers (501 and 502). The 6-bit parallel data word is effectively an address for the ROM, and determines which character will be displayed. The output of the ROM is applied as a 5-bit parallel data word to the shift registers. The 5-bit data word (GVIDEO) is then serially shifted to the Video Multiplexer. The output of the Video Multiplexer is a 5-bit serial stream of data which is routed through connector J6 to the CRT display. Additional signals sent to the CRT display include a horizontal sync pulse (GHSYNN) and a vertical sync pulse (GVSYNN). GHSYNN is a negative pulse with a duration of 5.0 $\pm$1 microseconds and a repetition rate of 63.4 microseconds. GVSYNN is a negative pulse with a duration of 1 millisecond and a repetition rate of 16.6 milliseconds. Both signals are developed by the Character Generator.

When the next character is selected at the keyboard, the entire process is repeated. For example, the Keyboard Cycle Logic generates KFCEIN, and another write cycle is initiated.

4.6 TTY MODE

In TTY Mode, data received at the communications interface is written into the Terminal's display memory. Note that TTY Mode is the only mode in which data from the interface is displayed. Keystrokes generated during TTY Mode are not written into the display memory, but are transmitted to the interface. For a keyboard-generated character to be displayed in TTY Mode, it must be echoed back to the Terminal from the CPU. TTY Mode may be entered either from the keyboard "TTY" key or be receipt of a TTY code from the interface.

4-6-1. TTY Receive Mode

The TTY Receive Mode is the only operational mode of the terminal in which information may be received from the interface. When this mode is entered, the terminal is effectively writing data into memory starting at a location determined by the cursor position. Any message received from the interface must therefore be preceeded by the TTY Function Code (TTY).

For the following discussion, refer to figure 4-5 and 4-6. The TTY Receive Mode is entered by receipt of the TTY code from the interface at Data Connector J4. The TTY code is a serial stream of data consisting of 10-bits; 1-start, 7-data, 1-parity, and 1-stop. This data is received at J4 (XCKBBN) and is clocked into the I/O section, receive buffer, by CBCLKP.

In the I/O section, I.C. 909 converts the serial data word with start, data, parity, and stop bits into a 7-bit parallel code and verifies proper code transmission by checking parity and receipt of valid stop bit. The 7-bit parallel code (RDBT17-RDBT7P) is applied to the Data Multiplexer. Additionally, signal RPBFLP is generated, indicating receive buffer full. This signal when applied to Receive cycle Logic, generates RECYLN.

Signal RECYLN produces CMDS1P and CMDS2P as outputs from the Data Multiplexer Control section. These are applied to the Data Multiplexer and switch the Multiplexer to accept receive data as an input source.

The Data Multiplexer essentially consists of four, dual four-input multiplexers which permit selection of four, 7-bit parallel input data sources. The output of the Data Multiplexer is a 7-bit parallel data word (HMBT1P-HMBT7P) which is applied to the Control Function Decode section.

The Control Function Decode section contains a one-of-ten decoder (1102) and a one-of-sixteen decoder (1101). Decoder 1102 accepts bits 5, 6, and 7 of the modified ASCII code, which determine in which column (0-7) the character is located (see Table 3-2). Similarly, decoder 1101 uses bits 1 through 4 to determine the row (0-15) location of the characters. The outputs of the decoders are connected to gating with the output of each gate representing the decode of a particular receive code function. The output signal of the TTY Mode gate (DMTTYN) is applied to the Mode Control section, where it sets the TTY Mode flip-flop, and assures all other mode control flip-flops are in the reset state.

The terminal is now ready to accept the first character of receive data. Processing of the character is the same as with the TTY code up to the point where the data is now at the Data Multiplexer output, and ready to be loaded into the I/O Register. Refer to figure 4-5 and 4-6 for the following discussion.

On receipt of the first character of data, signal RPBFLP is produced in the I/O section and routed to the Receive Cycle Logic. This signal produces RFCE2N as an output which initializes the Write Control logic by generating the following signals: (1) HENBIN, (2) HCLKIN, and (3) HWRIRN.

HENBIN is the parallel load enable pulse for the I/O Register, with a duration of T_1-T_1. HCLKIN is the parallel load pulse for the I/O Register with a duration of T_3. This pulse loads the 6-bit character into the I/O Register. HWRIRN occurring next at T_5, is routed to the Read/Write/Edit Function and initiates the memory write cycle.

HWRIRN sets flip-flop 601, causing 601-9 to go low. This signal is applied to a 4-bit Universal Shift Register (602-5) which is used in a parallel-parallel data transfer mode. This register is clocked at T_2 time, and with correlation (UCCORN), UMENAN is generated. The duration of UMENAN is T_2-T_2. UMENAN enables Line Selector 401, with the output of 401 applied to the Memory Multiplexer to determine which line the character will be written on.

UMENAN is also applied to the I/O Register as an enabling level, permitting the I/O Register shift clocks (MCLPHN) to serially shift the 6-bit character (MIDA1P)-MIDA6P) into the Memory Multiplexer. This character at the input to the Memory Multiplexer is two lines, identified as MIDA1P and MIDA4P with each line having three serial bits. One line represents the 3-LSB's (MIDA1P) and the remaining line represents the 3-MSB's. Note that the Memory Multiplexer recirculates the memory data back to memory during the time a write cycle is not being executed, so that no data is lost.

The output of the Memory Multiplexer is parallel data (MCD11N-MCD62N) which is routed to the Memory Bus Multiplexer for routing to the character generator to be displayed. The control of which line the

character will be displayed on is determined by signals GRLL1P-GRLL4P; these are supplied as inputs from the Character Generator, Line Counter. The output from the Memory Bus Multiplexer is two lines, each having three serial bits. One line represents the 3-LSB's (MGDAIN) and the remaining line represents the 3-MSB's. In the character Generation Data Bus, the two lines of serial data are converted to a 6-bit parallel character (MDAT1P-MDAT6P) which is applied to the Character Generation section.

The Character Generation section consists of a ROM (401) and two 4-bit universal shift registers (501 and 502). The 6-bit parallel data word is effectively an address for the ROM, and determines which character is displayed. The output of the ROM is applied as a 5-bit parallel data word to the shift registers. The 5-bit data word (GVIDEO) is then serially shifted to the Video Multiplexer. The output of the Video Multiplexer is a 5-bit serial stream of data which is routed through connector J6 to the CRT display. Additional signals sent to the CRT display include a horizontal sync pulse (GHSYNN), and a vertical sync pulse (GYSYNN). GHSYNN is a negative pulse with a duration of 5.0 $\pm$ 1 microseconds and a repetition rate of 63.4 microseconds. GVSYNN is a negative pulse with a duration of 1 millisecond and a repetition rate of 16.6 milliseconds. Both signals are developed by the Character Generator.

When the next character is received from the interface, the entire process is repeated. For example, RPBFLP is generated when the receive buffer is full, and this will initiate another write cycle.

4-7. TTY TRANSMIT MODE

The TTY Transmit Mode is the only operational mode of the terminal in which information entered at the keyboard may be transmitted out the interface in real time. The selection of the characters to be transmitted, and the execution of each transmission on a character-by-character basis is solely dependent on the operator. Memory is not accessed, and therefore the contents of memory are not modified in a TTY transmit operation.

For the following discussion, refer to figures 4-7 and 4-8. As shown in figure 4-7, signals KMTTYP and KMCTLP are routed from the keyboard, through Connector J9, to the Mode Control section. KMTTYP is generated by the operator when the TTY key is activated. KMCTLP is generated as a result of the operator activating the control (CTRL) key. These two signals are anded in the Mode Control section. The final output is used to set the TTY mode flip-flop, and also assure all other mode control flip-flops are in the reset state.

The output of the TTY mode flip-flop is signal HTTYAP which is routed to the Transmit Data Cycle Logic as an enabling level for the transmit operation. HTTYAP is alo applied to the Input/Output section where it is sent out the interface as the Request-to-Send signal, ICKCAP. The terminal now waits for Clear-to-Send (XCKCBP) to be received from the interface. When Clear-to-Send is received, XCKCBP provides an additional enabling level to the Transmit Data Cycle Logic, and the keyboard character may now be transmitted.

The terminal is now ready to accept the first character of data from the keyboard, for transmission out the interface.

When the operator activates the desired key, this results in signals KBSTRP (keyboard strobe) and the 7-bit character code (KDBT1P-KDBT7P) being applied through keyboard Connector J9. The 7-bit parallel character code is applied to the input of the Data Multiplexer. The keyboard strobe (KBSTRP) is routed to the keyboard cycle logic.

KBSTRP will set the keyboard cycle flip-flop, initiating the transmission of the first character to the interface.

Signals KECYLP and KECYLN are derived from KBSTRP. These signals are used to switch the Data Multiplexer to accept the keyboard data (KDBT1P-KDBT7P), and route the parallel data (HMBT1P-HMBT7P) to the input of the Transmit Buffer.

Signals KPLD1P and KPLDSP are also derived from KBSTRP. Signal KPLD1P parallel loads the 7-bit character into the transmit buffer at T_5 time. At T_6 time, KPLDSP load the 7-bit parallel character (TDBT1P-TDBT7P) into the Input/Output section, transmit holding register (I.C. 909). This device converts the parallel data into a serial word, and generates a start, parity, and stop bit. The 10-bit data word is then transmitted serially out the interface.

When the next character is to be transmitted, the operator actuates the desired key. Signal KBSTRP occurs along with the 7-bit parallel data word and the entire process is repeated. The terminal remains in the TTY mode until a mode change is initiated from either the keyboard or at the interface.

4-8. BLOCK TRANSMIT MODE

The Block Transmit Mode (BXT) permits a large block of data to be sequentially transmitted from the terminal, to the interface. This mode is entered from either the Local or TTY mode by communication link Block Transmit (BXT) code entry or by activation of the keyboard Block Transmit (XMIT) key. The sequence of operation is shown in flowchart form in figure 4-8.

4-8-1. Block Transmit Sequence

As shown in figure 4-9 activation of the Block Transmit Key (XMIT) or code entry while the terminal is not in BXT, enters an End-of-Message (EOM) symbol ( ) at the cursor location, sends the cursor to the home position and begins a sequential search for the start-of-message (SOM) symbol.

When the first SOM symbol is located, the terminal will turn-on the Request-to-Send signal which is transmitted to the interface. The terminal then waits for a Clear-to-Send signal from the interface. When the Clear-to-Send signal is received, the terminal will transmit an SOM symbol code to the communications interface and wait for the echo SOM code to be returned.

Upon receipt of the echo SOM code, the terminal will commence to transmit the block of data. The transmission of the data will be line-by-line from the first character following the SOM symbol, the cursor

will follow the character being transmitted from the screen, and will not blink. The transmission of each line of data to the interface is ended by the New Line Symbol (NL) or by the end-of-line; in either case, a NL coce is transmitted.

Transmission of the displayed data will always end at the first EOM symbol detected; in the event that no SOM symbol is present on the screen, only the EOM symbol code will be transmitted. Upon completion of transmission, the Terminal will remain in BXT mode until the mode is changed by interface code entry or keyboard control. After transmission of the data has been completed, the keyboard BXT key and interface code entry are inhibited from reinitiating a block transmission until the mode is changed.

A retransmission of the block of data may be accomplished at this time by the entry of a No Acknowledge Code (NAK). After a NAK code has been entered, the Block Transmit sequence is repeated, beginning with the search for the SOM symbol.

4-8-2. Block Transmit Data Transmission Restraints

The terminal will wait indefinitely for the SOM symbol code to be returned from the communications interface. If the operator or the communications interface wishes to abort the transmission at any time, the function code of one of the other modes must be initiated, and any character in transit will be transmitted before the terminal changes modes.

If a new line character is inserted on the line before the only EOM symbol, the EOM symbol will never be found and the message on the screen, including the SOM symbol code, will be transmitted incessantly until operator or communication data link intervention.

BXT flip-flop (HMTYBN) is routed to the Transmit/Receive Clocks section and the Block Transmit Control section. Note that HMTYBN may also be generated at the keyboard via signals KMCTLP and KMMTYP being applied to the mode control section through connector J9.

In Transmit/Receive Clocks, HMTYBN is used as an enabling level for the BXT reset gate. In the Block Transmit Control section, it is used to: (1) generate a signal (HEOMEN) which when routed to the Data Multiplexer results in the EOM character being written into memory (2) generate a signal (BHOMAN) which is routed through the Control Function Decode section with signal HDHOMP resetting the cursor horizontal and vertical position counters (home cursor) (3) initializes block transmit control section for BXT operation (4) generate signals BMDOQP and BTMDEN which are routed to the Control Function Decode section. Signal BMDOQP supplies an enabling level for the SOM decode gate. BTMDEN is an enabling level for the EOM and end-of-line detect gates (5) generate signal BRMNLN when a New Line (NL) code is detected. Signal BRMNLN is gated in the Write Inhibit and New Line Control section with clock T4. The output HDNLPN causes the cursor to advance to the next line and is also applied to Block Transmit Control. Signal BLDCRP is the output applied to the Data Multiplexer Control section which generates the New Line (NL) character during data transmission (6) generate signal BCMRRN which is routed to the Read/Write/Edit Functions section. This signal initiates a memory read cycle, which will result in UREADN being applied to the Memory Bus and I/O Register. UREADN will enable memory clocks (MCLPHN) to do a continuous read of data from memory through the Data

Multiplexer I/O Register, out the Data Multiplexer, to the Control Function Decode section. At this time, no data is being transmitted to the interface and a transmission will not occur until the SOM code is detected.

When the SOM code is detected (see figure 4-11b) in the I/O Register, the read operation is discontinued. Signal HDSOMP, the output of the SOM decode gate, is applied to the Block Transmit Control section. This results in signals BCKCAN and DPLDBN being generated. BCKCAN is routed to the I/O section and out the interface as ICKCAP, the Request-to-Send signal. DPLDBN is an initiate signal for the Transmit Data Cycle logic. The SOM code remains in the I/O Register and the Terminal waits for a Clear-to-Send signal (XCKCBP) from the interface.

On receipt of XCKCBP from the interface, the Transmit Data Cycle logic will implement the transmission of the SOM code. Signal KPLDIP will parallel load the SOM code into the Transmit Buffer. Signal KPLDSP parallel loads the SOM code into the Transmit Holding Register, located in the Asynchronous Receiver/Transmitter, I.C. 909. This device converts the parallel data into a serial word, and generates a start, parity, and stop bit. The 10-bit data word (SOM code) is then serially transmitted out the interface. The terminal will now wait for the SOM code (echo) to be returned from the interface.

The SOM code echo (see figure 4-11c) is received at the I/O section via Data Connector J4. The 10-bit data word is serially shifted into I.C. 909 by CBCLKP. When the receive buffer is full, signal RPBFLP is applied to the Receive Cycle Logic. The output of the Receive Cycle Logic is RECYLN which switches the Data Multiplexer Control section to accept the 7-bit SOM code. The SOM code (HMBT1P-HMBT7P) is routed to the Control Function Decode section where it is decoded as HDSOMP. Signal HDSOMP applied to the Block Transmit Control section will result in BCMRRN being routed to the Read/Write/Edit functions. In this section, BCMRRN generates UREADN which is applied to the Memory Bus and I/O Register. UREADN enables memory clocks (MCLPHN) to read the first character after the SOM from memory, and into the I/O register.

Signals CMDS1P and CMDS2P from Data Multiplexer Control have now been switched to select memory data (the 7-bit character) by removing input signal RECYLN. The memory data (MIDA1P-MIDA6P) is now gated through the Data Multiplexer and to the Transmit Buffer as a 7-bit parallel data word (HMBT1P-HMBT7P).

Signal DPLDBN applied to the Transmit Data Cycle section results in KPLD1P being routed to the Transmit Buffer, loading the 7-bit data word into the buffer. KPLDSP is generated next in the Transmit Data Cycle section and loads the 7-bit data word into the I/O section, Transmit Holding Register, device 909. Again the 7-bit parallel data word is converted to a 10-bit serial data word, including 1-start, 7-data, 1-parity, and 1-stop bit. The 10-bit data word is transmitted out the interface via Data Connector J4.

The next character (see figure 4-11d) and all succeeding characters to be transmitted are controlled by signal TPBFEN, Transmit Holding Register Empty. This signal is generated by the I/O section each time a 7-bit data word has been transmitted out the interface. TDBFEP is applied to the Transmit/Receive Clocks section where it enables clocking for the BXT master reset circuitry. It is also inverted and routed to Block Transmit Control as BPBFEP and used to enable DPLDBN. Signal TDBFEP is also applied directly to Block Transmit Control and used to generate BCMRNN as an input to the Read/Write/Edit Functions. Signal BCMRRN will generate another character read cycle, the character will be routed to the I/O section, and the character will be transmitted as previously described. This sequence will continue for each and every character until an EOM (end-of-message) code is detected.

When the EOM is detected (see figure 4-11d), in the Control Function Decode section, signal HDEOMP is generated at the output of the EOM gate. This signal is applied to the Block Transmit section where it sets the EOM flip-flop. The output of this flip-flop is signal BFEOMP which is routed to the Transmit/Receive Clocks section. BFEOMP removes an inhibit on the BXT master reset circuitry, resulting in the SOM, EOM, and BXT initialized flip-flops being reset. The terminal will remain in BXT mode until a mode change to one of the other modes is implemented from the interface or the keyboard.

4-9. FUNCTION CODES

Function codes are commands which are issued by the communication interface or by the operator actuating a control key at the keyboard. All function code performance is completed without pauses in the incoming data with the exception of the Clear Display (ERS) function code. A description of each function code from the operator's point of view is provided in Chapter III, paragraph 3-4. The terminal USASCII code set is tabulated in Chapter III, Table 3-2.

The following subparagraphs provide a detailed discussion of each function code once it is initiated at the communication interface or the keyboard. This discussion references Timing Diagrams, located at the back of this chapter, and logic diagrams located in the back of this instruction manual. Some function codes have previously been described in the flow diagram description for operational modes contained in this chapter (for example: Modes, Forward Space, Home, etc.). Those codes will not be discussed in this presentation.

4-9-1. Character Insert Function.

The Character Insert Function (INS) provides the capability of inserting a new character, for example during text composition, from either the communication interface or the keyboard. When this function code is implemented, the terminal will insert a space at the cursor location. The terminal then shifts the character at the cursor location,

and all subsequent characters on the same line, to the right one location. The character at the end of the line is dropped (no wrap around), and the cursor is not forward spaced.

On receipt of the character insert code (INS) from either the interface or the keyboard, the INS function is initiated. When the code is received from the interface, a Receive Data Cycle (RPBFLP) is generated. When the code is received from the keyboard, a Keyboard Cycle (KECYLP) is generated. The initiation of RPBFLP or KECYLP results in a Terminal Write Cycle.

RPBFLP or KECYLP switch the Data Multiplexer to select the INS code, and route the 7-bit character to the Control Function Decode gating. The output of the Character Insert decode gate is signal HDINSP which is applied as a clock signal to the Read/Write/Edit Functions control. Refer to figure 4-12, and Logic Diagram 162566-001, Sheet 2 of 6 for the following discussion.

Signal HDINSP is applied as a clock pulse to pin 12 of flip-flop 704. Pin 10 is set high and pin 9 is set low. The pin 9 output (UDWRTN) is an inhibit to the cursor horizontal position counter, preventing the counter from being incremented. Flip-flop 705-9 is set low as a result of the positive-going input clock from flip-flop 704-10. Note that flip-flop 705-9 will be set high at the next T_6 time, generating a negative pulse with duration of approximately 212 nanoseconds. This pulse is routed to gate 108-5 with the output at pin 6 (UCDELP) applied to the I/O Register as a reset pulse.

The output of flip-flop 705-9 is also applied to Gate 603-12. The output 603-11 is a positive clock pulse which sets flip-flop 601-9 low. The 601-9 output is applied to pin 5 of 4-bit register 602. This register is used in the parallel-parallel data transfer configuration. The register is clocked at pin 10 by T_2 time.

Assume at this point correlation has occured; therefore, pin 12 and pin 14 of 602 will be clocked low at the next T_2. The output of inverter 505-10 goes low, enabling flip-flop 704 pin 7 to be clocked high at T_6 time. This results in UMENAN being held low as a constant enabling level to the I/O Register. Continuous serial shift clocks are gated to the I/O Register.

At the same T_2 time, the output of gate 603-3 goes high and at the next T_6 time, signal UWFSPN is generated, resetting flip-flop 601-9 to a high state.

Continuous serial shift clocks are gated to the I/O Register as a result of the enabling level supplied by UMENAN. The contents of the I/O Register, starting with the 6-bits of all zeroes (space) are written into memory. As this occurs, the character previously at the cursor location is being shifted into the I/O Register. Since UMENAN is being held low by 704-7, this character is now written into memory also, but it is written one space to the right of its original position. Similarly, each succeeding character on the line will be rewritten one location to the right.

At character count 80 detected (GCH8XP), flip-flop 601-6 is set high at the next T_2. Gate 604-4 goes low, setting flip-flop 704-6 high, flip-flop 705-6 high, and flip-flop 704-9 high, terminating the character Insert (INS) function. Note that when character count 80 changes, GCH8XP goes low. At the next T_2, flip-flop 601-6 will be reset to a low state.

The cursor has not moved during the preceeding sequence, and a space now appears on the display at the cursor location. A new character may now be written into memory at this location via the communication interface or by the operator from the keyboard.

4-9-2. Character Delete Function

The Character Delete Function (DEL) provides the capability of deleting a character, for example during text composition, from either the communication interface or the keyboard. When this function code is implemented, the terminal will delete the character at the current cursor location. All characters to the right of the cursor are then shifted left one character position, and a space is inserted at the end of the line.

On receipt of the Character Delete Code (DEL) from either the interface or the keyboard, the DEL function is initiated. When this code is received from the interface, a Receive Data Cycle (RPBFLP) is generated. When the code is received from the keyboard, a Keyboard Cycle (KECYLP) is generated.

The 7-bit function code is decoded in the Control Function Decode section. The output of the DEL function gate is signal HDDELP, a positive pulse which occurs at T_4 time.

Refer to figure 4-13, and Logic Diagram 162566-001, Sheet 2 of 6 for the following discussion. HDDELP is applied as a clock pulse to pin 12 of flip-flop 605. The output at pin 10 is set high with the leading edge of the positive-going pulse. No further action occurs until count 80 (GCH8XP) is received from the Character Generator, Character counter, at pin 4 of flip-flop 605.

Count 80 (GCH8XP) is a positive-going pulse which occurs on the trailing-edge of T_6 time. This input sets flip-flop 605-6 high and 605-7 low. The 605-6 output provides an enabling level for the I/O Register reset gate 108-6 (UCDELP). The 605-7 output is a negative going signal occurring at the trailing-edge of T_6 time, and is the Character Delete Command (EDENAN).

EDENAN is applied to Inverter 703-11, through tate 604, setting flip-flop 704-10 high and 704-9 low. The 704-10 output provides a clock pulse to pin 12 of flip-flop 705, setting 705-9 low. The 705-9 output is a negative pulse with a duration of approximately 530 nanoseconds. This pulse is inverted through gate 603 and the positive output at pin 11 clocks flip-flop 601-9 low. This output is applied to 4-bit Universal Shift Register 602-5 and at the next leading edge of T_2, 602-14 will go low. No further action occurs until correlation is detected at 602-7.

The Edit Memory (Logic diagram 162562, Sheet 1, I.C. 207) contains the data of the current line on which the cursor is located, and is always being clocked so that it is two characters ahead of the normal memory (see figure 402 for comparison between Memory clocks and Edit Memory clocks). Until correlation occurs, Edit Memory data is circulated through the I/O Register but is not written back into the normal Memory. Note that as each Edit Memory character is shifted into the I/O Register, at this point the Edit Memory data is one character ahead of the normal memory.

When correlation occurs (see logic diagram 162566-001, Sheet 2), signal UCCORN is applied as a negative input to pin 7 of 602. At the next T_2, the output of 602-12 will go low. Both inputs to Gate 604-9, 10 are now low and the output will go high. This output is applied to pin 11 of gate 604. The output of 604-13 will go low, generating UMENAN at the leading edge of T_2 time. Note that the output of Inverter 505-10 goes low and is applied to pin 3 of flip-flop 704. At the next T_6 flip-flop 704-7 will go high. This output is applied to gate 604-12 and holds UMENAN low. UMENAN is an enabling level for Line Selector 401 (see 162567-001, Sheet 1). The Line Selector determines which line the Edit Memory data will be written on. The inputs to the Line Selector are provided by the cursor vertical position counter.

UMENAN is also an enabling level for the I/O Register input gating which permits I/O Register serial shift clocks (MCLPHN) to shift the Edit Memory data (MIDA1P-MIDA4P) into memory. Since the Edit Memory is clocked so that it is one character ahead of the normal memory in the I/O Register, the character following the deleted character will be written at the cursor location (original character is deleted). All succeeding characters for the remainder of the current line, up to and including the 78th character, are also written into memory.

When character count 79 occurs (see logic diagram 162566-001, Sheet 2) the I/O Register is reset (all zeroes) and this 6-bit pattern is written into memory as a space. Count 70 (GCD70N) and count 9 (GC09N)

are received from the Character Generator, Character Counter. These two inputs along with clock time T_2 (MTM2N) are applied to gate 402, pins 1, 2, and 13 respectively. The high output of the gate is a positive pulse, with a duration of 106 nanoseconds, which is applied to pin 1 of gate 108. The additional input to this gate is high so that the output of pin 3 is a negative pulse. The input at pin 5 of gate 108 is high at this time; therefore, UCDELP is generated as a positive pulse with a duration of 106 nanoseconds. UCDELP resets the I/O Register, and the 6-bits all zero pattern, at the output of the register is serially shifted to, and written into memory. A space now appears on the screen at character location 79.

In the final sequence of events, the edit logic is reset. This is initiated at the next character count (count 80) after the space is written. Count 80 is applied as a set level to flip-flop 601-2. At the next T_2 time, the leading-edge of the clock will set 601-6 high. The output of gate 604-4 is a negative level which sets flip-flop 704-6 high and 704-7 low. Flip-flop 704-6 going high clocks flip-flop 705-6 high and 705-7 low. The 705-6 high output is inverted through 703 and resets flip-flops 605, pins 15 and 1. The low output at 705-7 resets 704-15, thus terminating the Character Delete function.

4-9-3. Set Forms (SF) Function

The Set Forms command causes the Terminal to assume Split Format mode. This function provides an efficient means for entering and displaying formatted data. When this format is used, the screen can be divided in fixed (protected) fields for header information and variable (unprotected) fields for data entry. The COLON (:) is used as the delineator which separates the two fields. Variable (unprotected) data on lines which contain colons are protected by the Set Format (SF) command. The last (most recent) colon entered on a line is the delineator. A line containing no colons will not be protected.

NOTE

Because the colon locations are stored in holding registers, erasure or overstrike of the colon will neither clear the register nor keep that line unprotected. Only Clear Display (ERS) command will clear the register; another colon will change the register. Colons entered in the variable field after the SF command are not delineators.

The data from the start of the line to the colon is defined as the fixed field. The variable field is from the right of the colon to the end of the line. There may be only one variable field per line and any line which does not have a delineator present is considered a variable

field. Keyboard and communication entry is restricted to the variable field unless the protected area is accessed by use of the LF (Control-H) or LCA (Control-Q) operations. Only the variable field data may be transmitted to the interface.

The protected data becomes unprotected by the issuance of a Reset Format Command. This command resets the Terminal to General Format. The normal operational sequence when using this format is: the communication interface erases the screen, then writes the format with colons on the screen, and finally issues the Set Format Command.

Delineators may only be written into the colon memory in General Format mode. On receipt of the colon function code from either the interface or the keyboard, the write colon into colon memory function is initiated. When this code is received from the interface, a Receive Data Cycle (RPBFLP) is generated. When the code is received from the keyboard, a keyboard cycle (KECYLP) is generated. See figure 4-14 and 4-15 for the following duscussion.

The 7-bit function code (KDBT1P-KDBT7P or RDBT1P-RDBT7P) is applied to the control function decode section as parallel data, HMBT1P-HMBT7P. The 7-bits of data are decoded and the output of the colon decode gate is signal HDCOLP, a positive pulse which occurs at T_4 time.

HDCOLP is applied as a clock pulse to the split mode Read/Write control section, (Logic diagram 162565-001, Sheet 1) where it sets flip-flop 408-6 high. When Memory Busy (MBUSYN) is removed, indicating that the colon has been written into the display memory, flip-flop 408-10 will go high. At the next T_4 time, signal BCOLDN is generated. This signal

is applied to gate 409-5 and is then gated with signal ENORMN (which is high at this time). The resulting output is signal EWRITN occurring at T_4 time, which enables the colon address (UCUR1P-UCUR8P) to be written into the colon memory. Signals UBLN1P-UBLN4P define the line on which the colon will be written.

Each time data is written into memory, and a colon is input to define the protected data area, the previously described sequence is repeated. After all lines of data are written and the colons are input, the Set Format function is initiated.

The Set Format (SF) function code is received from the interface and routed to the Control Function Decode section. The output of the SF gate is signal HDSFPP, a positive pulse which occurs at T_4 time. HDSFPP is applied to the Cursor Horizontal Position Control, generating USTARP, (Logic diagram 162566-001, Sheet 3) which resets the Horizontal Position Counter. Signal USHOMN is also generated as a result of HDSFPP and is applied to the cursor vertical position counter, resetting this counter. USTARP is also applied to the Split mode control section (Logic diagram 162565-001, Sheet 1) and on the trailing edge of this pulse, sets flip-flop 407-10 high. This is one of the enabling levels for gate 307-11.

Signal HDSFPP is routed to the Split mode Read/Write control as a dc set pulse for the forms flip-flop, 407-1. Flip-flop 407-6 goes low, generating signal ENORMN and 407-7 goes high, generating ENORMP. Signal ENORMP and the output of flip-flop 407-10 are gated together to form the 636 nanoseconds negative pulse at the output of gate 307-11. This pulse

generates signals EENJ2N and EENJ1N as an output from the jump cursor control section. Signals EENJ2N and EEJN1N are supplied as parallel load enables to the cursor horizontal position counter (Logic diagram 162566-001, Sheet 3). These two signals will load the jump cursor address (EJUM1P-EJUM8P) which is the colon line address plus one into the horizontal position counter. The cursor is then always forced to one character position beyond the colon line location when any manipulation of the cursor occurs which would cause it to enter a protected area. Note that signal EBSBLN at the output of the forms cursor location comparator is routed to the cursor horizontal position counter, gate 801 pin 2. EBSBLN is an inhibit and prevents backspacing of the cursor into the memory protect area when the Split format has been selected.

When it is desired to reset format (RF), returning to the general format where no area of memory data is protected, the RF function code is applied at the input to the interface.

When the RF code is detected, a receive data cycle (RPBFLP) is generated and the 7-bit code is routed to the control function decode section. The output of the RF gate is signal HDRFPP, a positive pulse which occurs at T_4 time.

HDRFPP is applied as a clock pulse to the split mode Read/Write control section where it resets forms flip-flop 407-6, 7. Pin 6 goes high (ENORMN) and pin 7 goes low (ENORMP). ENORMN going high removes the chip select enable from the colon memory. ENORMP going low removes the enable to gate 307, pin 13 in the jump cursor control section which supplies the

jump cursor signals EENJ1N and EENJ2N to the cursor horizontal position counter. Removing signals EENJ1N and EENJ2N, inhibits the cursor horizontal position counter from loading the jump cursor parallel address information (EJUM1P-EJUM8P) into the counter. Note that this does not disable the Load Cursor Address (LCA) function.

4-9-4. Clear Display (ERS) Function

This function provides the capability of erasing the entire display and moving the cursor to the left most position of the first line (HOME). This function also, when initiated, causes the terminal to assume the General format, and resets the delineator (colon) memory.

The ERS command may be initiated from the or the keyboard (see figure 4-16 and 4-17). When initiated from the interface, a Receive Data cycle (RPBFLP) is generated. The ERS command is generated at the keyboard by the operator depressing the control key (CTRL) and the "L" key. When the operator initiates the ERS command, a Keyboard Data Cycle (KECYLP) is generated.

The 7-bit function code (KDBT1P-KDBT7P or RDBT1P-RDBT7P) is applied to the control function decode section as parallel-data HMBT1P-HMBT7P. The 7-bits of data are decoded and the output of the ERS gate is signal HDERSP, a positive pulse which occurs at T_4 time.

HDERSP is applied as a reset pulse to the cursor horizontal posistion control section. This results in output signals USTARP and USHOMN (162566-001, Sheet 3).

USTARP is a reset pulse for the cursor horizontal position counter, resetting it to Home. USHOMN is a reset pulse for the cursor vertical position counter, resetting it to Home.

Signals UCDISP and UCDISN (162566-001, Sheet 2) are also generated at the output of the Read/Write/Edit functions, when HDERSP is received as

as input. UCDISN is routed to the character generator (162566-001, Sheet 1) where it resets video enable flip-flop 308-1, removing signal GENBVP as an input to the character generator video multiplexer. This prevents video from being-multiplexed to the CRT display during the ERS function. UCDISN is also applied to the Split Mode Read/Write control (162565-001, Sheet 1) where it resets flip-flop 407-6, 7, assuring signal ENORMN is high and ENORMP is low. ENORMN is an enabling level for the forms (colon) memory write gate.

UCDISP at gate 801, pin 12 (162566-001, Sheet 2) enables 801-11 to go low with the leading edge of T_5, causing a continuous sequence of memory write cycles. UCDISP is also applied as a constant dc reset to the I/O register. Thus, the I/O register contains all zeroes which are serially shifted into memory by memory phase clocks, MCLPHN. The memory phase clocks (MCLPHN) are enabled by UMENAN at the input to the I/O register. UMENAN occurs for each character, after correlation, from T_2 to T_2 time.

At the end of the first line, at character count 80, signal UCRETP is generated in the cursor horizontal position control section (162566-001, Sheet 3) and is applied as a clock pulse to the cursor vertical position counter, incrementing the counter by a count of line (line two). USTARP is derived from UCRETP and is a reset for the cursor horizontal position counter, resetting the counter to character location zero. UCRETP is also routed to the Split mode Read/Write control section (162565-001, Sheet 1), resulting in EWRITN (T_4 time) at the output, which loads the cursor horizontal

position address (address zero) into the colon memory. This sequence is repeated for each character line of displayed information and is terminated when the last line (ULASTN) is detected (162566-001, Sheet 3).

ULASTN is detected in the cursor vertical position control and counter section and occurs when this counter has been incremented to count six (character line six). ULASTN is applied to the Read/Write/Edit function section (162566-001, Sheet 2) where it resets the ERS command flip-flop. UCDISP goes low and UCDISN goes high. This removes the constant dc reset (UCDISP) on the I/O register, the enable from gate 801-12, and video output inhibit (UCDISN) to the character generator.

4-9-5. Clear to End of Display (EOF) Function

This function may be implemented in the general format or split format. In the general format, the EOF function erases all data from the cursor position to the end-of-page. The cursor does not move. In the split format, the EOF function erases all variable data from the cursor position to the end-of-page. The cursor does not move.

The EOF command (see figures 4-18 and 4-19) may be initiated from the interface or the keyboard. When initiated from the interface, a receive data cycle (RPBFLP) is generated. The EOF command is initiated at the keyboard by the operator depressing the control key (CTRL) and the "K" key. When the operator initiates the EOF command, a keyboard data cycle (KECYLP) is generated.

The 7-bits function code is applied to the control function decode section as parallel-data, HMBT1P-HMBT7P. The 7-bits of data are decoded and the output of the EOF gate is signal HDEOPP, a positive pulse which occurs at T_4 time.

HDEOPP is applied as a clock pulse to the Read/Write/Edit functions section (162566-001, Sheet 2), where it sets flip-flop 702-10 high and 702-9 low. This results in signals UCLFCN and UCLFCP.

If the Terminal is in Split format mode, the colon memory is enabled. The cursor horizontal position address contained in the colon memory will now be the reference character address, starting with the first line succeeding the cursor location, where the clear operation will commence.

UCLFCP is used within the Read/Write/Edit function section to remove the parallel load enable on the cursor vertical position, edit counter (606) (162566-001, Sheet 2). This counter will now be incremented at character count 80, after each character line is cleared. No further action in the Terminal occurs until character count 100 (GENDLP) from the character generator is detected, and received at the input to Read/Write/Edit functions.

Signal GENDLP is a positive pulse which is gated in the Read/Write/Edit functions section with the output of flip-flop 702-10 (UCLFCP). This produces a negative pulse at the cutput of gate 801-8 which is used as a dc set to flip-flop 701-11, generating URRASP. Signal URRASP is applied as a constant dc reset to the I/O register. The I/O register contains all zeroes (MIDA1P-MIDA6P) which are serially shifted into memory by memory phase clock, MCLPHN. Signal UMENAN is an enabling level to the I/O register input gating, allowing the memory phase clocks (MCLPHN) to serially clock the I/O register. UMENAN is initially generated when correlation occurs and would normally be removed within one terminal cycle time, however, the output of flip-flop 704-7 will remain high which holds the output of gate 604-13 low, until character count 80 is detected.

The location on the first character line at which the clear operation will commence is determined by the cursor location. The location at which all succeeding character lines are cleared is dependent on the output from the colon memory since this address (EFOR1P-EFOR8P) is always loaded into the cursor horizontal position multiplexer. An additional determining factor is whether the general or protected format (forms) has been selected.

For example, if the Terminal is set to general format (unprctected), the address at the output of the colon memory at the beginning of the clear sequence for each character line will be address zero, and all characters on that line will be cleared. If forms had been set (Split format), prior to initiation of the EOF command, the address at the output of the colon memory (EFOR1P-EFOR8P) will be one character beyond the colon (:) location for each character line. The clear sequence will commence at the first character succeeding the colon location on each character line and continue through the end of each line.

As each line is cleared (characters through 79) and count 80 is detected, flip-flop 704-7 is reset (signal UCHSEP removed), flip-flop 705-7 is set low generating URESTN. URESTN going low increments the cursor vertical position, edit counter (606). Flip-flop 705-6 going high, sets flip-flop 702-7 low, generating USFORN which switches the inputs to the Horizontal Position Multiplexer from the horizontal position counter to the inputs from the Colon Memory. The address at the output of the colon memory will now be the location at which the clear operation will begin.

URESTN also resets flip-flop 704-9 to the high state and flip-flop 701-10 to the low state. The terminal then again waits for character count 100 to initialize the clear sequence for the next character line, and the process is repeated as previously described.

The EOF function is terminated when the last character line (line 6) has been cleared. This is detected by gate 506-8 located at the

output of the cursor vertical position, edit counter (606). The detected line 6 input to gate 506-8 is gated with clock time T_2, and the output UBLNSN rests the EOF function flip-flop, and also removes USFORN from the cursor horizontal position multiplexer.

4-9-6. Clear Entry (EE) Function

This function may be implemented in the General Format or Split Format. In the General Format, the EE function erases the entire display and returns the cursor to Home; the Colon Memory is not cleared. In the Split Format, the EE function erases the variable field only and returns the cursor to the right of the first delineator (colon); the Colon Memory is not cleared.

The EE command (see figures 4-20 and 4-21) may be initiated from the interface or the keyboard. When initiated from the interface, a receive data cycle (RPPFLP) is generated. The EE command is initiated at the keyboard by the operator depressing the control key (CTRL) and the "X" key. When the operator initiates the EE command, a keyboard data cycle (KECYLP) is generated.

The 7-bits function code is applied to the control function decode section as parallel-data, HMBT1P-HMBT7P. The 7-bits of data are decoded and the output of the EE gate is signal HDEEPP, a positive pulse which occurs at T_4 time.

HDEEPP is applied as a clock pulse to the Read/Write/Edit functions section (162566-001, Sheet 2), where it sets flip-flop 702-10 high and 702-9 low. This results in signals UCLFCP and UCLFCN.

UCLFCP is used within the Read/Write/Edit function section to remove the parallel load enable on the cursor vertical position, edit counter (606). This counter will now be incremented at character count 80, after each character line is cleared.

HDEEPP is also applied to the cursor horizontal position control section (162566-001, Sheet 3), generating USTARP and USHOMN at the output. USHOMN resets the cursor vertical position counter to location zero (Home the cursor). USTARP resets the cursor horizontal position counter to location zero. USTARP also is applied to jump cursor control (162565-001, Sheet 1) as a clock signal to the jump cursor flip-flop 407-2, setting 407-10 high. At this point further action in the Terminal is determined by whether or not forms (Split format) has been set.

Assuming forms had been set, the output of flip-flop 407-10 is gated with USTARP in the jump cursor control section. This results in signals EENJ1N and EENJ2N at the output. These two signals enable the cursor horizontal position counter to parallel load the address (EJUM1P-EJUM7P) at the output of the jump cursor Address Multiplexer into the counter. This address will be one character position beyond where the colon (:) is located, and is the character location at which the erase sequence would begin. Signal ENORMP being high at the input to the colon memory enables the output of the memory and is received as a result of forms being set.

Assuming forms had not been set (all areas unprotected), signals EENJ1N and EENJ2N would not be generated and therefore the jump cursor address (EJUM1P-EJUM7P) would not be parallel loaded into the cursor

horizontal position counter. Also, signal ENORMP would be low and the colon memory output would be inhibited. The address at the output of the cursor horizontal position counter (UCUR1P-UCUR8P) will always be character location zero, the erase sequence will start at location zero, for each character line, and continue through the end of the line (character 79).

The next action that occurs in the Terminal is the detection of character count 100 (GDENLP) (162566-001, Sheet 2). This always occurs in either protected (Split) or unprotected (General) format. Signal GENDLP is a positive pulse which is gated in the Read/Write/Edit functions section with the output of flip-flop 702-10 (UCLFCP). This produces a negative pulse at the output of Gate 801-8 which is used as a dc set to flip-flop 701-11, generating URRASP. Signal URRASP is applied as a constant dc reset to the I/O Register. The I/O Register contains all zeroes (MIDA1P-MIDA4P) which are serially shifted into memory by memory phase clock, MCLPHN. Signal UMENAN is an enabling level to the I/O Register input gating, allowing the memory phase clocks (MDLPHN) to serially clock the I/O Register. UMENAN is initially generated when correlation occurs and would normally be removed within one Terminal cycle time, however, the output of flip-flop 704-7 will remain high which holds the output of gate 604-13 low, until character count 80 is detected.

The location on the first character line at which the clear sequence will commence is determined by the address at the output of the cursor horizontal position counter (UCUR1P-UCUR8P).

For example, as previously described, if the Terminal is set to general format (unprotected), the address at the output of the cursor horizontal position counter at the beginning of the clear sequence for each character line will be address zero, and all characters on that line will be cleared. If forms had been set (Split format), prior to initiation of the EE command, the address at the output of the jump cursor address multiplexer (EJUM1P-EJUM7P) determines the location at which the clear sequence will commence. This address will always be one character beyond the colon (:) location for each character line. The clear sequence will commence at the first character succeeding the colon location on each character line and continue through the end of each line.

After the first line is cleared (characters through 79) and count 80 is detected, flip-flop 704-7 is reset (signal UCHSEP removed), flip-flop 705-7 is set low generating URESTN. URESTN going low increments the cursor vertical position, edit counter (606). Flip-flop 705-6 going high, sets flip-flop 702-7 low, generating USFORN which switches the inputs to the cursor Horizontal Position Multiplexer from the horizontal position counter to the inputs from the Colon Memory. The address at the output of the colon memory (EFOR1P-EFOR8P) will now be the location at which the clear operation will begin for each succeeding character line. If forms had not been set, the outputs address of the Colon Memory is inhibited and EFOR1P-EFOR8P is always address zero. If forms had been set, the output address of the Colon Memory will be one character beyond the colon (:) location.

URESTN also resets flip-flop 704-9 to the high state and flip-flop 701-10 to the low state. The Terminal then again waits for character count 100 to initialize the clear sequence for the next character line, and the process is repeated as previously described.

The EE function is terminated when the last character line (line 6) has been cleared. This is detected by gate 506-8 located at the output of the cursor vertical position, edit counter (606). The detected line 6 input to gate 506-8 is gated with clock time T_2, and the output UBLNSN resets the EE function flip-flop, and also removes USFORN from the cursor horizontal position multiplexer.

4-9-7. Line Erase (EOL) Function

This function may be implemented in the general format or split format. In the general format, the EOL function erases the entire line on which the cursor is positioned and returns the cursor to the beginning of that line. In the split format, the EOL function erases the variable area of the line on which the cursor is positioned and returns the cursor to the right of the delineator (:) on that line.

The EOL command may be initiated from the interface or the keyboard (see figure 4-22 and 4-23). When initiated from the interfaces a receive data cycle (RPBFLP) is generated. The EOL command is initiated at the keyboard by the operator depressing the control key (CTRL) and the "I" key. When the operator initiates the EOL command, a keyboard data cycle (KECYLP) is generated.

The 7-bits function code is applied to the control function decode section as parallel-data, HMBT1P-HMBT7P. The 7-bits of data are decoded and the output of the EOL gate is signal HDEOLP, a positive pulse which occurs at T_4 time.

HDEOLP is applied as a clock pulse to the Read/Write/Edit function section (162566-001, Sheet 2), where it sets flip-flop 701-10 high. This results in signal URRASP. Signal URRASP is applied as a constant dc reset to the I/O Register. The I/O Register will contain all zeroes (MIDA1P-MIDA6P).

In the Read/Write/Edit functions, signal URRASP also sets flip-flop 704-10 which in turn, sets flip-flop 601-9 low, initiating a memory

write cycle. Additionally, HDELOP is applied as a reset pulse to the cursor horizontal position counter (162566-001, Sheet 3), resetting the counter to character location zero. HDELOP is also inverted in the Read/Write/Edit functions at 703-6 (162566-001, Sheet 2) (EDEOLN) as a dc set to flip-flop 407-11 (162565-001, Sheet 1). This output at pin 10 is one of two enabling levels for the split format gate, 307-12. Further action is dependent on whether or not the Terminal is set in General Format or Split Format.

Assuming General Format has been set, the output of the cursor horizontal position counter (UCUR1P-UCUR8P) will be zero, and correlation will occur at this location. When correlation occurs, signal UMENAN (162566-001, Sheet 1) is generated in the Read/Write/Edit functions. UMENAN is an enabling level to the I/O Register input gating and permits serial shift clocks (MCLPHN) to serially shift all zeroes (MIDA1P-MIDA6P) into memory, starting at character location zero through the end-of-line, character location seventy-nine.

Assuming Split Format has been set, signal ENORMP (162565-001, Sheet 1), is generated in the jump cursor control section, and applied as an enabling level to the colon memory. ENORMP enables the address data at the output of the colon memory (EFOR1P-EFOR8P) to be routed to the cursor horizontal position counter. This address will be one character position beyond the colon location (protected area). Signals EENJ1N and EENJ2N are also generated in jump cursor control, and parallel load the address data (EJUM1P-EJUM7P) into the cursor horizontal position counter.

The output of the counter (UCUR1P-UCUR8P) (162566-001, Sheet 3) is the address (one character position beyond the colon) at which the clear sequence will commence and continue through the end-of-line (character position seventy-nine). Further action in the Terminal occurs regardless of format. When character count 80 from the character generation is received at the input to Read/Write/Edit functions (162566-001, Sheet 2), flip-flop 601-6 is set high, the output of gate 604-4 goes low, setting flip-flop 704-6 high and flip-flop 705-7 low. The low output of flip-flop 705-7 is URESTN. Signal URESTN is a dc reset to flip-flop 704-10; it also resets the EOL command flip-flop 701-15, terminating the EOL function.

4-9-8. Forward Space (FWS) Function

This function causes the cursor to move right one character position up to the 80th character (character 79) without altering any characters in memory.

No legend is presented on the space key to show the shifted FWS. The code transmitted in TTY mode will be the space code; this function is a keyboard only activity and the communication interface cannot implement this function. The FWS function (see figure 4-24) is initiated at the keyboard by the operator depressing the SHIFT key and the SPACE bar. When this occurs, a keyboard data cycle (KECYLP) is generated.

The 7-bits function code (KDBT1P-KDBT7P) is applied to the control function decode section as parallel-data, HMBT1P-HMBT7P. The 7-bits of data are decoded and the outputs of the FWS gate are HDMFSN and HDFSXN.

HDMFSN is applied as a negative level to the Read/Write/Edit functions section (162566-001, Sheet 2) during T_1 through T_1 time. The negative level inhibits the write cycle flip-flop at 601-12 and 601-13 and assures it is reset when HWRIRN occurs. This prevents a write cycle from being generated and therefore data in memory is not changed.

HDFSXN occurs at T_3 time and is also supplied to the Read/Write/Edit functions section where it generates UWFSPP at 108-11. This signal is applied to the horizontal position counter (162566-001, Sheet 3) as a count-up-clock. This signal will increment the counter one character position, each time the shift key and space bar are depressed.

4-9-9. Backspace (BS) Function

This function causes the cursor to move lift one character position and does not alter any characters in memory. The cursor will stop movement at the beginning of the line or to the right of the delineator (colon), if Split Format has been set.

The BS command may be initiated from the interface or the keyboard (see figure 4-25). When initiated from the interface, a receive data cycle (RPBFLP) is generated. The BS command is generated at the keyboard by the operator depressing the "BS" key. When the operator initiates the BS command, a keyboard data cycle (KECYLP) is generated.

The 7-bits function code is applied to the control function decode section as parallel-data, HMBT1P-HMBT7P. The 7-bits of data are decoded and the output of the BS gate is signal HDBSPP, a positive pulse which occurs at T_4 time.

HDBSPP (162566-001, Sheet 3) is applied to the cursor horizontal position control where it is gated with signal EBSBLN. Further action in the Terminal is dependent on whether or not Split format has been set.

Assuming General Format, signal HDBSPP is inverted and becomes a negative count down clock pulse, occuring at T_4 time. Each time a BS function code is decoded, this clock will decrement the cursor horizontal position counter by one.

Assuming Split format, the identical action occurs with the exception that the cursor cannot be backspaced into a protected area. The address at the output of the colon memory (EFOR1P-EFOR8P) (162565-001, Sheet 1) is enabled and is the address of the character one position beyond the colon location. This character address is compared in the forms cursor location comparator with the input character address from the cursor horizontal position decode counter (UCUR1P-UCUR8P). When these two addresses compare, and the Terminal is set in Split format, signal EBSBLN is generated. EBSBLN is an inhibit to the cursor horizontal position decode counter and will prevent count down clocks from decrementing the counter.

4-9-10. Line Feed (LF) Function

This function causes the cursor to move down one line or from the last line to the first line, without altering any characters in memory. The LF function may be initiated from the interface or the keyboard (see figure 4-26). When initiated from the interface, a Receive Data Cycle (RPBFLP) is generated. The LF function is generated at the keyboard by the operator depressing the control key (CTRL) and the "H" key. When the operator initiates the LF command, a keyboard Data Cycle (KECYLP) is generated.

The 7-bits function code (KDBT1P-KDBT7P or RDBT1P-RDBT7P) is applied to the control function decode section as parallel-data, HMBT1P-HMBT7P. The 7-bits of data are decoded and the output of the LF decode gate is signal HDLFPP which is a positive pulse occurring at T_4 time. This signal is routed to the cursor vertical position control and counter (line counter) (162566-001, Sheet 3). In the control section it is inverted and then applied to the line counter as a count-up-clock, incrementing the counter one complete character line, each time the function code is initiated. If the cursor is located on the last character line, the LF function will cause the cursor to move to the corresponding character position on the first line.

4-9-11. Load Cursor Address (LCA) Function

This function may be implemented from the interface only. When implemented, LCA conditions the Terminal to move the cursor to a position specified by the next three characters following the LCA command. Without exception, the Terminal interprets the three characters succeeding the LCA, as follows: first character - vertical position (0-5), second character - most significant horizontal position (0-7), and third character - least significant horizontal position (0-9).

The first position in the address is zero (Ø) and addressing is sequential 0-5 vertically and 0-79 horizontally. The Terminal is not constrained to detect an illegal address.

For the following discussion, refer to figures 4-27 and 4-28. When the LCA function is implemented, the 7-bits function code is received from the interface via Data Connector J4 as serial data, XCKBBN. This data is loaded into the input/output section where it is converted to parallel data, and signal RPBFLP is generated (receive buffer full). RPBFLP is applied to the Receive Cycle Logic, resulting in signals RECYLN and RECYLP at the output. RECYLN switches the Data Multiplexer to accept the 7-bits function code RDBT1P-RDBT7P and route this code as parallel data (HMBT1P-HMBT7P) to the Control Function Decode section. Signal RECYLP is an enabling level which permits the parallel data to be decoded in the Control Function Decode section, and the output is signal HDLCAP (LCA function decode) which occurs at T_1 through T_1 time.

HDLCAP is applied to jump cursor control (162565-001, Sheet 1) as an enabling level to the J-K inputs of a 4-bit shift register (405), which is clocked by the positive going trailing edge of HCLKIN, occurring each receive data cycle at T_3 time. The register is initially cleared by master reset and therefore the four outputs stages are all at a low (logic "0") level. This register is used in the serial-to-parallel data transfer configuration, and is clocked at each receive data cycle. To detect the (1) LCA code (2) cursor vertical position data, and (3) cursor horizontal position data, requires four receive data cycles.

During the first cycle when the LCA code is received, HDLCAP at the input to the register is high and signal HCLKIN clocks a high input logic ("1") into the first parallel output stage of the register. No further action occurs until the next receive data cycle.

In the second receive data cycle, the cursor vertical position (0-5) character information is received. Signal HDLCAP is now low and will stay low for the remainder of the LCA sequence. Clock signal HCLKIN therefore clocks the logic "1" presently in the first parallel output stage of the 4-bit register, into the second position; at the same time a logic "0" is clocked into the first stage. The logic "1" output is used to generate signals CLCABN and CLCLCN at the output of jump cursor control.

CLCABN is a negative level and will stay at this level for the remainder of the LCA sequence. CLCABN is routed to write inhibit (162566-001, Sheet 5) where it generates signal HLRMBN. This signal is applied to

Write Control as an inhibit which prevents the LCA code and succeeding character position information from being written into memory. HLRMBN is also a negative level with the same duration time as CLCABN. Signal CLCABN is also applied as an enabling level to the jump cursor address multiplexer (162565-001, Sheet 1). This will allow the character horizontal position information (MIDA1P-MIDA4P), when received, to be selected by the multiplexer.

Signal CLCLCN is generated as a result of the logic "1" output, being gated with T_5 time. It is therefore a negative level, occurring at T_5 time. CLCLCN is applied to the cursor vertical position control and counter section (162566-001, Sheet 3) as a parallel load enable, and on the trailing edge of this pulse, the cursor vertical position data (MIDA1P-MIDA3P) is loaded into the counter. The cursor has now been positioned to the appropriate decoded line. No further action occurs in the Terminal until the next receive data cycle.

When the third receive data cycle occurs, the cursor most significant horizontal position character (0-7) information is received. In the jump cursor control section, HCLKIN again clocks the 4-bit shift register, shifting the logic "1" from the second output stage of the register into the third output stage of the register. The preceeding two outputs of the register will now be at a logic "0" level. The logic "1" output from the third stage of the register is gated with T_5 time which produced signal EENJ2N at the output of Jump Cursor Control.

EENJ2N is applied to the cursor horizontal position counter (162566-001, Sheet 3) as a parallel load enable. EENJ2N loads the cursor most significant character horizontal position data (EJUM5P-EJUM7P) into the counter. No further action in the Terminal occurs until the fourth receive data cycle. The cursor vertical position counter presently contains the line location (0-5) information at which the cursor will be positioned, and the cursor horizontal position counter contains the MSD (0-7) character information at which the cursor will be positioned.

When the fourth receive data cycle occurs, the cursor least significant horizontal position character (0-9) information is received. In the jump cursor control section, HCLKIN again clocks the 4-bit shift register, shifting the logic "1" from the third output stage of the register into the fourth and final output stage. The logic "1" output from the fourth stage of the register is gated with T_5 time which produces signal EENJ1N at the output of jump cursor control. EENJ1N is applied to the cursor horizontal position counter as a parallel load enable. EENJ1N loads the cursor least significant character horizontal position data (EJUM1P-EJUM4P) into the counter. The cursor has now been positioned first to the decoded line location and secondly to the requested horizontal position on line.

Additionally, in the jump cursor control section, the logic "1" output from the fourth stage is gated with T_6 time. This produces a negative pulse with a duration time of T_6 clock which resets the 4-bit register on the lead edge of the pulse terminating the LCA sequence.

4-9-12. Bell Function Code

This function code may be implemented from either the keyboard or the interface. When implemented, it causes an audible tone for the purpose of attracting the operator's attention. Refer to control logic diagram 162566-001, sheet 5 of 6 for the following discussion.

The 7-bits bell function code (HMBT1P-HMBT7P) is received at the input to Column Decoder 1102 and Row Decoder 1101 as the result of a receive data cycle. Also received as a result of the receive data cycle is signal RECYLP which supplied an enabling level to the Row Decoder. The decoded outputs of the decoders for a bell function code are D00XXN for column, and DX07N for row. The two outputs are applied to the bell decode gate 1103, pins 3 and 4. These are gated with signal DEIPBN which occurs at T_4 time, and is also generated during a receive data cycle. The output of this gate is signal HDBELP (bell decode), a positive pulse occuring at T_3 time. This pulse drives the output of one-shot 1201-7 low, and is one of the enabling levels to gate 1205. An additional enabling level is signal GRLL1P which is received from the Character Generator, occuring at a rate of approximately 1.0 KC.

The output of the one-shot remains low for approximately 500 milliseconds, so that the input to the base of Q1 via gate 1205 is controlled by GRLL1P and switching occurs at the 1.0 KC rate. When GRLL1P is low, Q1 is conducting and current flows through the speaker coil, producing the audible signal (BABELP). The audible signal is removed when the output of one-shot 1201-7 goes high. This time is determined by the RC circuit consisting of C16 and R9, and is approximately 500 milliseconds.

4-10. CHARACTER GENERATOR

The character generator reads six bit characters from successive locations in the memory, and decodes them as required to produce a display of 80 characters per line by 6 lines on the video display unit, or CRT. Characters are displayed on the CRT in a 5 by 7 dot matrix format, with vertical character spacing the equivalent of nine dots and horizontal character spacing the equivalent of one dot. Each character thus occupies seven dots horizontally, and each line of characters on the screen requires sixteen raster lines.

Character generation begins when the first character of a row is read from memory. The six bit character is presented to a memory bus multiplexer in Memory along with control signals from the Character Generator that select which line of the 5 x 7 matrix is being addressed. The output of the memory bus multiplexer is the selected six bit character which is applied to the ROM in the Character Generator. The ROM output is a five-bit word representing the five dots making up one raster line of the 5 x 7 matrix. These five bits are presented to a parallel-to-serial converter, where they are shifted out serially and are then gated in a video multiplexer prior to being routed to the video display unit.

In a similar manner, the succeeding characters of the row are read and processed. When the 80th character of the row has been read, a horizontal sync pulse is generated, and the character generator address is reset to the first character of the row. The process of reading 80 characters is repeated except that the control signals within the ROM have been changed to permit the

second line of the 5 x 7 matrix to be read out. The same sequence occurs for each of the seven raster lines making up a row of characters. Each row of characters is thus read from memory seven different times in the process of displaying the row on the screen. Following these seven lines, the display is blanked for eight raster lines to provide a space before beginning the next row of characters. After the sixteenth raster line, the character generator address is set to the first character of the next row and the process repeats. At the end of the ~~sixth~~ row of characters, a vertical sync pulse is produced and the entire cycle is repeated.

For the following discussion, refer to figures 4-29, 4-30, and Logic Diagram 162566-001, sheet 1 of 6. The Character Generator receives memory data (MGPA1N-MGDA2N) in a serial format which is clocked by memory clock MCLPHN. Memory data MGDA1N is the three LSB's of the six bit character and MGDA2N is the three MSB's of the six bit character. This data is serially shifted in two-bit increments into the character generator. For example during the first MCLPHN clock, LSB bit one and MSB bit one are simultaneously shifted. Therefore, during one terminal cycle (T_1-T_6), only three MCLPHN clocks are required to load one six-bit Character into the Character Generator serial-parallel register. When the 6-bit character is serially shifted into the serial-parallel register, a parallel load clock is generated which strobes the character into the ROM. The parallel load clock is the output of gate 108-8 and is generated by combining T_2 time (MTTM2P) and memory clock MCLPHN. This results in a positive-going clock with a duration of 53 nanoseconds which is developed during clock T_2 time. The positive transition occurs at the mid-point of the T_2 clock.

The parallel-serial ROM output shift register (501, 502)generates a negative-going parallel load clock with a duration of 106 nanoseconds at the output of gate 503-8 which loads the 5-bit character into the register, once each machine cycle. For example, assume that the parallel load clock has been generated, loading the first 5-bit character. Note that the 5-bit character is loaded in the first five parallel positions. In the sixth parallel position, a logic zero is always loaded and in the seventh position, a logic one is loaded. The logic one in the seventh position will be serially shifted through the register immediately following the data. After five positive transitions of the master clock (XCLOKN), the 5-bit character has been shifted out of the register, through the Video Multiplexer and to the CRT. The logic one immediately following the 5-bit character causes all inputs to Gate 503 to go high and the output will go low. This enables the second 5-bit character to be loaded. The logic zero loaded into the sixth position of the register prevents a parallel load pulse from being generated in the event the 5-bit character being loaded is an all logic ones pattern. This process is repeated for each character until the character counter has been incremented to a character count of 80. At this time, character 80 has been loaded into the ROM.

The character counter and decoder is clocked by the trailing edge of T_6 time and, therefore, is incremented each time a character is processed. At character count 80, signal GCD80P resets the Blanking Control, incrementing the ROM to the next row address for character data via signal GDISPP going low. Signal GBENVP is also removed as an enabling level to the video multiplexer, inhibiting data from being transferred to the CRT. Additionally, the horizontal sync pulse GHSYNN is generated and routed to the Video Display Unit. This is a negative pulse with a duration of approximately 5 microseconds and a repetition rate of 63.5 microseconds.

At character count 100, the Raster Lines Counter (103) is incremented and the character counter will repeat its cycle, starting at count "0". Signal GDISPP will go high and the video multiplexer is again enabled by signal GBENVP being set high. This process is repeated for the seven raster lines which are required to display one complete character line on the CRT.

When the eighth raster line is detected by the Raster Lines Counter, signal GROW8P (raster line 8) will go high maintaining a constant reset on the ROM outputs. All logic ones will now be loaded into the ROM parallel-serial shift register. This data is inverted at the output of the register to all logic zeroes and therefore, the CRT display will be blanked from raster line 8 (GROW8P) through raster line 15. The exception to this is during raster lines 10 and 11. Assuming correlation occurs, logic zeroes are serially shifted through the parallel-serial shift register. The data is inverted at the output to all logic ones and and video is constantly displayed on the CRT, forming the cursor. At raster lines 12 through 15, the CRT would again be blanked.

Each time the Raster Lines Counter (103) reaches a count of 16, it increments the Display Lines Counter (104, 203). The Display Lines Counter is therefore counting displayed character lines. The output of the Display Lines Counter is routed to Raster Line - 128 Detect logic (105-3, 203-6) and the Vertical Sync and Reset Control logic (raster line 262).

At raster line 128, the Blanking Control is reset and the video enable signal (GBENVP) to the Video Multiplexer is removed. No data is transformed to the Video Display Unit and the CRT is blanked. At raster line 176, the Vertical Sync and Reset Control generates a negative-going Vertical Sync pulse (GVSYNN) which has a duration of approximately 1 millisecond and a repetition rate of 16.6 milliseconds. This pulse is routed to the video display and is not multiplexed with the normal video. Additionally, a reset pulse is at raster line 262 which resets the Raster Lines (103) and Display Lines (104, 203) counters to zero and the display of a full screen of characters may now be repeated beginning with the first raster line of data. The raster line 262 detected output is also used as a clock pulse to the cursor blink control. The raster line 262 count occurs at a rate of 60 cps and is applied to a divider chain (306, 305) which has a final output (GBLINK) occurring at 2 cps. GBLINK is routed to the Cursor Vertical Position Comparator where it is used to blink the cursor.

Additional circuitry in the Character Generator is provided to detect an end-of-message (EOM) code. This code may be input from the interface or from the keyboard as a shifted 1 (one). Special detect gating (402, 506) is used to recognize this code at the output of Serial-Parallel Shift Register (405, 403, 404) and generate the 6-bit serial data character GEOMDP. The 6-bit EOM character is routed through the video multiplexer gating to be displayed on the CRT. The EOM symbol is displayed as a five by eight dot matrix with all dots illuminated.

4-11. VIDEO DISPLAY ASSEMBLY

The video display assembly (figure 4-31) is specifically designed to display alpha-numeric information. The unit consists of two circuit boards, one containing the horizontal and vertical deflection systems and the video amplifier, the other containing the horizontal sync phase detector, plus a chassis containing the kinescope, yoke, high voltage power supply and vertical choke. The fully assembled chassis is mounted as an integral part of the Video Communications Terminal. The input power is regulated +10 vdc; and the input horizontal, vertical and video information are three separate signals at logic levels (+5 vdc and ground).

The following information presents a detailed circuit description of the Video Display Assembly. For this discussion, refer to Schematic Diagram 452594 located in Appendix B, at the back of this manual.

4-11-1. Video Amplifier.

The video amplifier consists primarily of a single stage (Q2) with very high gain. Up levels on the input drive Q2 toward saturation while down levels cause Q2 to cut off. Fast transient response for positive going signals at the collector of Q2 is obtained by charging the output capacitance of Q2, the kine inner electrode capacitance and stray capacitance toward +90 volts. However, the signal is cut off at about +55 volts with CR1, R15 and C8, resulting in a rise time of about one time constant.

G2 is controlled by the brightness control (R22) which is part of a voltage divider between GRD and +500 vdc. The DC range of G2 is approximately +170 to +500 volts.

Q1 is a current sink which is controlled by the intensity control. A current node is formed at the base of Q2 so that the current through R8 resulting from the input signal must exceed the current through the sink in order to turn on Q2. The difference between these two currents determines the down (white) level at the collector of Q2. In order to avoid forward biasing, the base-collector junction of Q1, the current sink is operated from a -10 volt supply formed by R16 and CR2.

4-11-2. Vertical Section

Unijunction transistor Q9 and emitter follower Q10 are employed to generate the vertical deflection ramp. Feedback through C29 assures that a linear charging current flows through R55 to C27 and C28. A second feedback path through R58 and R57 (linearity adjustment) adds an exponential correction signal to compensate for the effect of L101. The ramp signal is AC coupled to the vertical size control, R60, and output amplifier, Q11 and Q12 so that no special selection of unijunctions or components will be needed for repairs. Q8 operates in the grounded base mode amplifying the incoming vertical sync signals to a level sufficient to trigger the unijunction oscillator.

4-11-3. Horizontal Deflection and High Voltage

In addition to the primary function of horizontal deflection, this system furnishes the high voltage (12 KV) for the kinescope anode, 500 volts DC for focus and G2 electrodes, and +90 and -90 volts for the video amplifier. It consists of an astable multivibrator (Q3 and Q4) amplifiers Q6 and Q7 and the horizontal output switch Q5.

The horizontal oscillator is triggered by each incoming horizontal sync pulse through C10 and CR3. The horizontal frequency control, R26, is adjusted to produce a free-running frequency slightly lower than the frequency of horizontal sync pulses. Q6, Q7, and T1 provide the necessary current gain to drive the output stage, Q5. Q5 and Q7 operate asynchronously so that when Q7 is on, energy is stored in T1 and when Q7 is off, this energy is used to drive Q5.

When Q5 is turned on, a step voltage is applied to the yoke through C21, a portion of L1, and L2. C21, the flat face or "S-shaping" capacitor introduces a parabolic correction to compensate for kinescope curvature. L1, the width coil, changes the circuit inductance thereby controlling the amount of current entering the yoke. A second portion of L1, in parallel with the yoke, is increased as the series portion of L1 is decreased. This action causes the total inductance of the yoke circuit to remain essentially constant so that the high voltage is unaffected by the width coil adjustment L2, the linearity coil, is a saturable reactor whose inductance decreases as the current through it increases. The effect compensates for the resistive losses in the yoke circuit.

At the end of the sweep, Q5 is turned off and the voltage at its emitter takes the form of a negative going half sine wave with a frequency determined by the variances inductances and capacitances in the horizontal output circuit. As this waveform crosses the supply voltage in the positive going direction, the damper diode, CR9 starts to conduct. By this time,

the current in the yoke has been completely reversed and the sweep begins again with the damper diode conducting for approximately the first half of sweep. During this interval, Q5 is turned on again to generate the second half of the sweep.

The network consisting of CR11, C24, R42, and R43 reduces yoke ringing and improves sweep linearity at the start of sweep. CR11 is forward biased during sweep so that C24 is charged to the supply voltage. During horizontal retrace, CR11 is reverse biased and C24 discharges slightly through R42 and R43. At the end of retrace, the cathode of CR11 is at a slightly lower voltage than the cathode of CR9. CR11, therefore, starts to conduct just before the start of sweep allowing C24 to absorb any transient signals which may be present.

R35 is used to reduce picture size variations. In any conventional transistorized television flyback system, an increase in beam current causes the high voltage to drop and the picture size to increase bloom), and the load current from the power supply also increases. In this unit, the load current flows through R35, and the increase results in a reduction of the voltage switched across the yoke, which tends to reduce the scan current and partially compensates for the width change. In addition, the vertical deflection circuit is supplied from this same point, which tends to maintain constant output ratio. The result is about a 2:1 improvement in size change over conventional circuits.

4-11-4. Horizontal Phase Detector.

During horizontal sync, the phase splitter, Q7, turns on the AFC phase detector, CR2 and CR3. At this time, a portion of the reference sawtooth formed by integrating the horizontal flyback pulse is transmitted to the AFC filter consisting of R34, C15, and C16. A DC component generated by the horizontal hold control, R30, is also transmitted. The filtered AFC signal is then applied to the horizontal oscillator through R28. This phase locked horizontal system is used to reduce the effects of noise on the input signal.