

**Technical  
Reference**

**DASHER™  
DISPLAY  
TERMINALS  
6052, 6053**

014-000077-01

## NOTICE

Data General Corporation (DGC) has prepared this manual for use by DGC personnel and customers as a guide to the proper installation, operation, and maintenance of DGC equipment and software. The drawings and specifications contained herein are the property of DGC and shall neither be reproduced in whole or in part without DGC prior written approval nor be implied to grant any license to make, use, or sell equipment manufactured in accordance herewith.

DGC reserves the right to make changes without notice in the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical or arithmetic errors, company policy and pricing information. The information contained herein on DGC software is summary in nature. More detailed information on DGC software is available in current released publications.

NOVA, SUPERNOVA and ECLIPSE are registered trademarks of Data General Corporation, Westboro, Massachusetts. DASHER is a trademark of Data General Corporation, Westboro, Massachusetts.

Ordering No. 014-000077  
©Data General Corporation, 1977  
All Rights Reserved  
Printed in the United States of America  
Rev. 01, September 1977

# TABLE OF CONTENTS

## CHAPTER I

### INTRODUCTION TO DASHER DISPLAY TERMINALS MODELS 6052, 6053

I-1	INTRODUCTION
I-2	6054/6055 Printer Options
I-3	ARCHITECTURE
I-3	OVERVIEW OF THEORY OF OPERATION
I-4	The Keyboard
I-4	Asynchronous Interface
I-4	Transmit Keyboard Data
I-4	Transmit Cursor Address
I-4	Receive Computer Data
I-5	Control Logic
I-6	Memory
I-8	Video Timing and Character Generator
I-8	Video Timing Generator
I-8	Video Character Generator
I-9	Video Monitor
I-9	6054/6055 Printer Option

## CHAPTER II

### PROGRAMMER'S REFERENCE

II-1	INTRODUCTION
II-1	Keyboard
II-1	User Function Codes
II-2	Display
II-2	Character Blinking
II-2	Character Dimming/Underscoring
II-2	Screen Rolling
II-2	Cursor Addressing
II-2	Cursor Sensing

## TABLE OF CONTENTS (Continued)

II-3	DISPLAY COMMANDS
II-6	INSTRUCTIONS
II-6	Read Character Buffer
II-6	Load Character Buffer
II-7	PROGRAMMING
II-7	Input
II-7	Output
II-7	TIMING CONSIDERATIONS
II-7	Input
II-7	Output
II-8	PROGRAMMING EXAMPLES
II-9	Cursor Address Codes
II-10	User Function Codes
II-10	Print and Print Form Codes
II-11	Program Interrupt

## CHAPTER III

## OPERATOR CONTROLS AND PROCEDURES

III-1	INTRODUCTION
III-1	DISPLAY
III-2	KEYBOARD
III-2	Main Keypad
III-3	Numeric Keypad
III-3	Screen Management Keypad
III-4	User Function Keypad
III-4	DISPLAY EXTERNAL CONTROLS
III-4	Front Panel
III-4	Power ON/OFF and Screen Brightness
III-4	Rear Panel
III-4	Data Rate
III-4	Parity
III-4	Mode
III-4	Screen
III-5	OPERATING NOTE
III-5	CLEANING
III-5	PREVENTIVE MAINTENANCE

## TABLE OF CONTENTS (Continued)

### APPENDICES

A-1

#### APPENDIX A

A-1

#### ASCII Character Codes with Terminal Control Functions

A-2

##### ASCII Character Codes Plotted by the 6052 Terminal

A-3

##### ASCII Character Codes Plotted by the 6053 Terminal

A-5

#### Display Commands

A-7

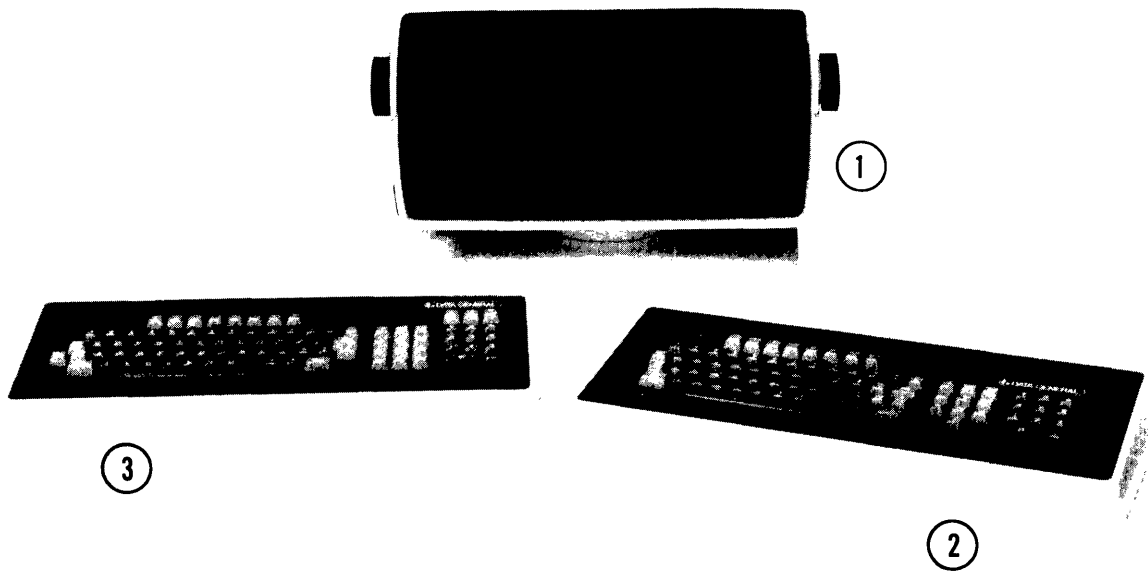
#### Protocol Characters

B-1

#### APPENDIX B

B-1

#### Code Sequences Generated by the User Function Keypad



DG-03834

COMPONENT
1. CRT DISPLAY, MODELS 6052, 6053
2. KEYBOARD, MODEL 6052
3. KEYBOARD,MODEL 6053

# **CHAPTER I**

## **INTRODUCTION TO DASHER DISPLAY TERMINALS MODELS 6052, 6053**

### **INTRODUCTION**

Data General's DASHER 6052/6053 display terminals are soft-copy I/O devices, designed for use in a wide range of interactive terminal applications. Model 6052 is the basic, upper case ASCII terminal; model 6053 is enhanced, upper/lower case ASCII terminal. Both models operate in full-duplex mode only.

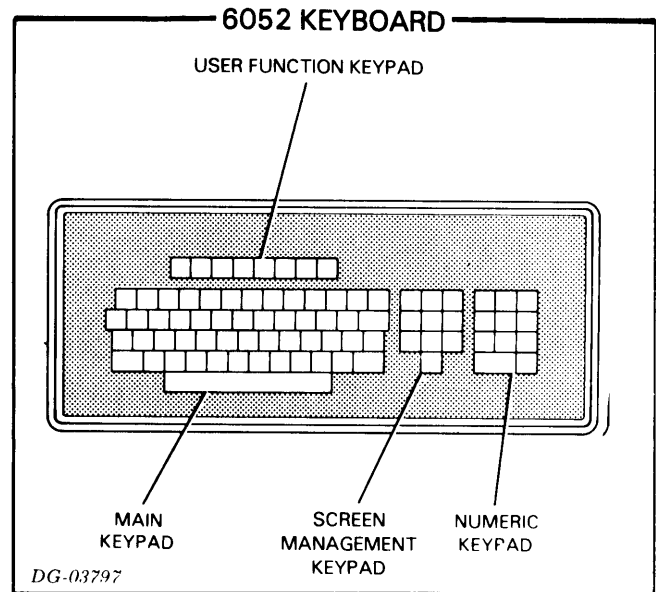
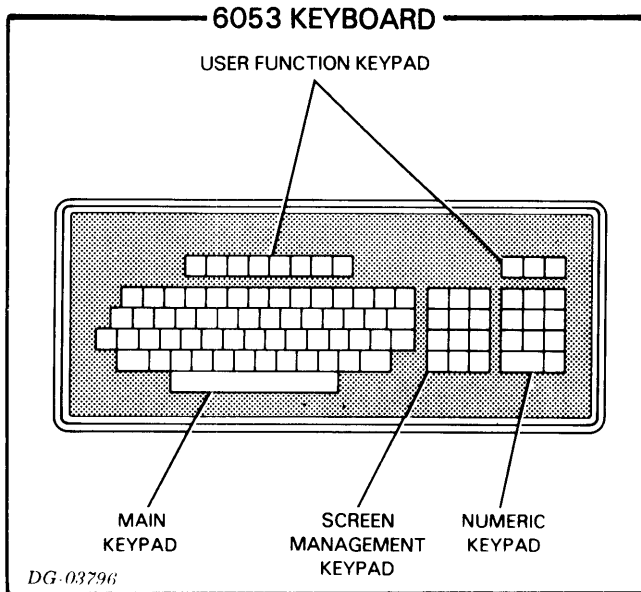
The display terminals consist of two units: an alphanumeric CRT display that is mounted on a fully articulated base and a separate, solid state, capacitive switching keyboard. The units are connected by a 4 foot, plug-in, external cable attached to the keyboard. Online, the display and keyboard function independently while sharing a common asynchronous interface to communicate with the computer. The keyboard is a transmit-only device. The display is a receive-only device with the additional capability of transmitting cursor position information when requested.

Communication with the computer is via either a 20mA current loop or an EIA RS-232C communications line. Both the rate at which data is transmitted and the type of parity check are externally switch-selectable on the display's accessory panel. Data transmission rates range from 110 baud to 19.2K baud. Parity may be either even, odd or marking.

The CRT display has a 1920-character, dynamic RAM memory and a non-interlaced video monitor with a 12 inch diagonal screen. The screen has an active area of 8.5 inches by 5.5 inches, formatted as 24 lines, 80 characters per line. A non-destructive cursor is displayed on the screen as a blinking underscore. Printing characters received by the display are stored in memory and displayed on the screen as either 5x7 upper case dot matrices or 5x8 lower case dot matrices.

Using display command codes, characters can be blinked on a character by character basis, the cursor can be directly addressed or sensed, and the screen can be rolled up one line at a time. On the model 6053, characters can also be displayed at reduced intensity (dimmed) and/or underscored on a character by character basis.

The keyboard of the 6052 display terminal has a teletypewriter-style main keypad while the 6053 keyboard has a typewriter-style main keypad. Both keyboards have three supplementary keypads: numeric, screen management and user function. See the keyboard illustrations on the following page. The user function keys expand the capabilities of these terminals. Model 6052 generates 32 and model 6053 generates 44 user-defined codes with which the programmer can implement his application.



### 6054/6055 Printer Options

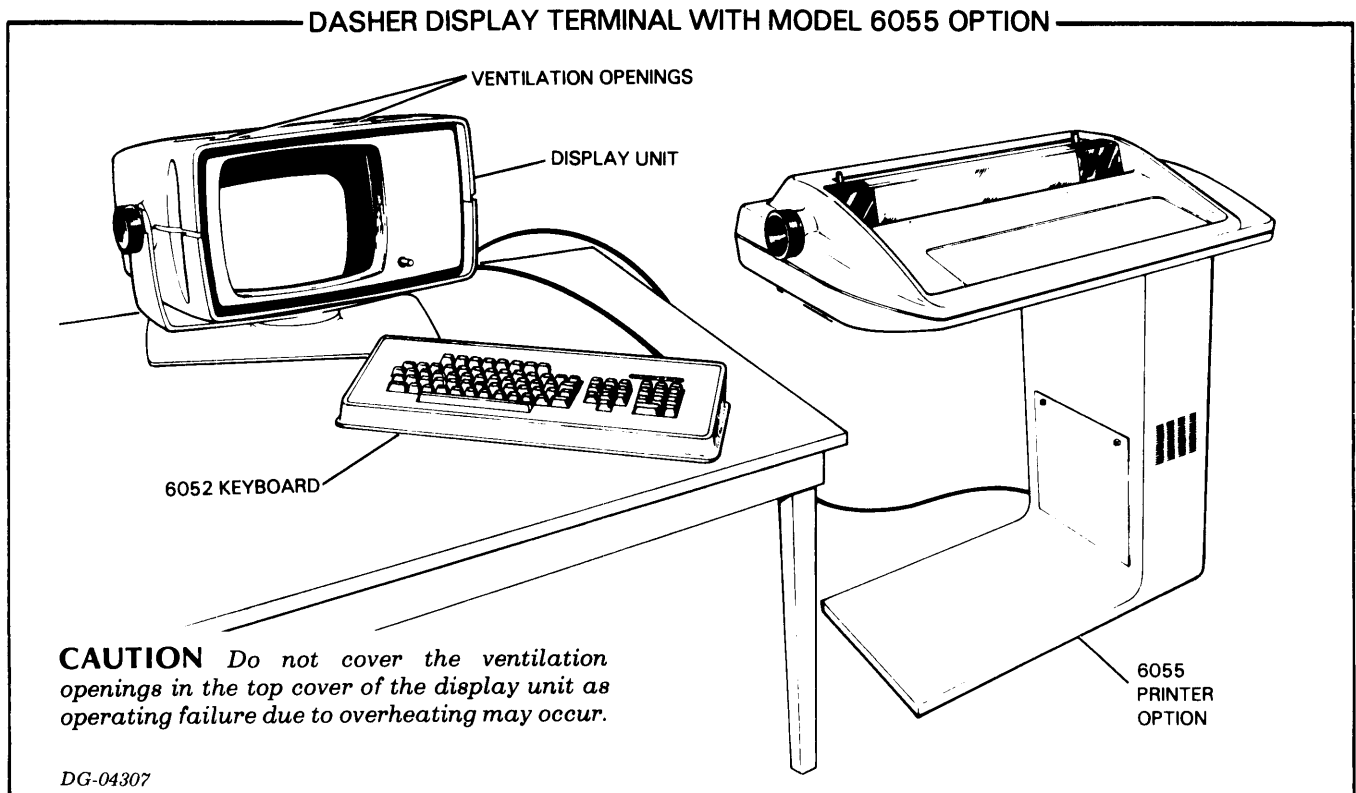
Two video enhancement packages are offered with the DASHER 6052/6053 display terminals which provide facilities for hardcopy output of information displayed on the CRT screen.

The first, model 6054, consists of a printed circuit board, housed in the display chassis, which communicates with any ASCII-based printer with an EIA RS-232C interface. Data transmission rates range from 110 to 1200 baud. An internal cable and an

external EIA RS-232C connector are also included with the model 6054 printer option.

The second, model 6055, consists of the printed circuit board, an internal cable, an external cable and a model 6041 DASHER Receive-Only (RO) 60 cps printer.

The screen formatting capability of the model 6053, implemented through the dimming feature, combined with either of the above printer options allows the use of preprinted forms.





## ARCHITECTURE

The 6052/6053 display terminals incorporate a modular design that facilitates maintenance and provides high reliability. This modularity allows the 6052 terminal to be easily upgraded to a model 6053.

Both terminals consist of the following main assemblies:

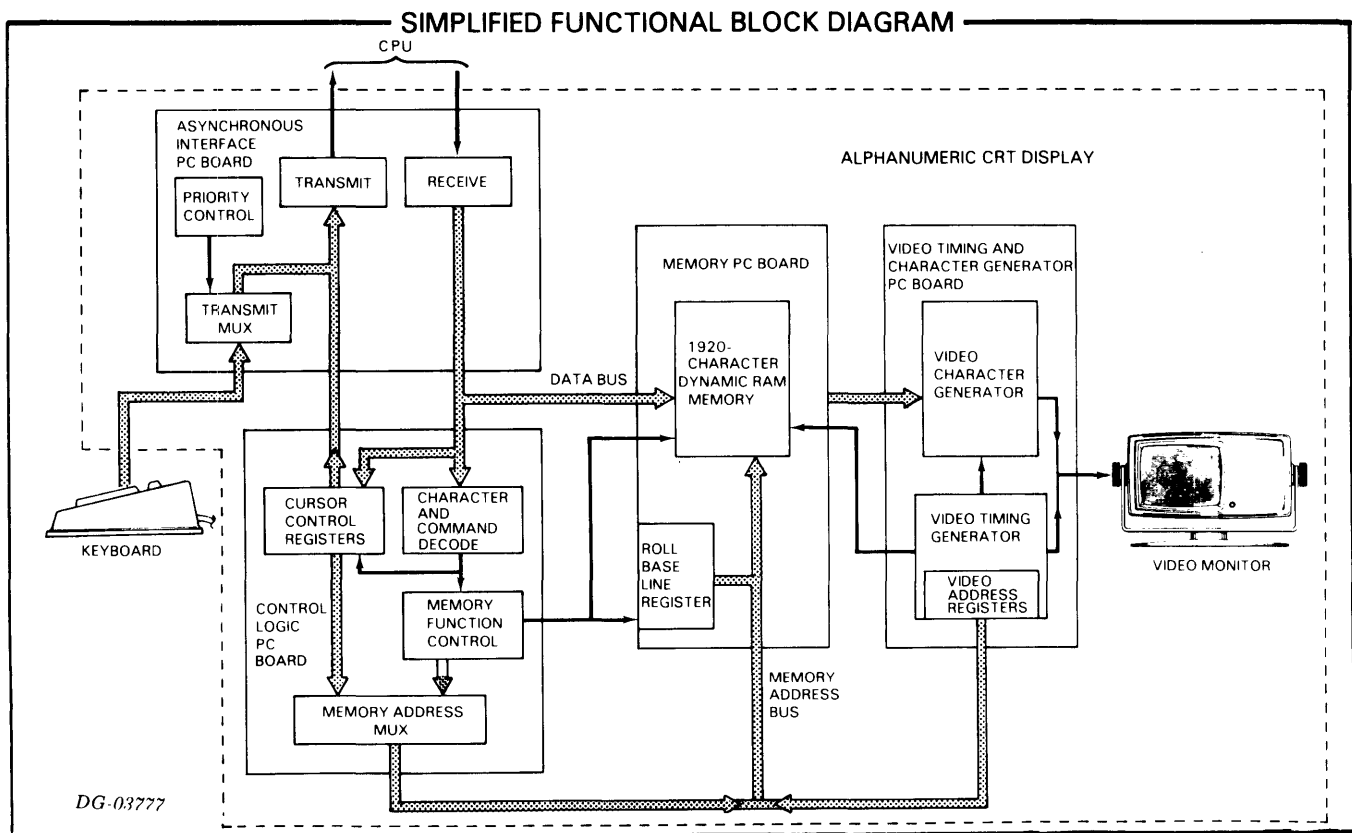
- A keyboard and an external cable.
- An alphanumeric CRT display consisting of the following subassemblies:
  - An accessory panel containing the following: a receptacle for the keyboard external cable; a data rate rotary switch; a parity slide switch; an online/local mode slide switch; interface connections for a 20mA current loop, an EIA RS-232C communications line, and a printer; and a composite video output jack.
  - An asynchronous interface, which forms the communications link between the keyboard, the computer and the display.
  - Control logic, which decodes the characters received via the asynchronous interface, controls the information written into

memory including character blinking, dimming and underscoring, and maintains cursor and memory addresses.

- A video timing and character generator, which provides the system timing and generates the signals necessary to display the printing characters on the screen of the video monitor.
- A 1920-character dynamic RAM memory which stores the information displayed on the screen.
- A video monitor that displays the information stored in memory.
- A power supply, which provides power for the unit.

## OVERVIEW OF THEORY OF OPERATION

The 6052/6053 DASHER display terminals transmit and respond to a subset of the 128-ASCII character set listed in Appendix A. As described previously, both terminals consist of two I/O devices, a keyboard and a display, which function independently when operating online with the computer. A simplified block diagram of the operation of these terminals is shown below.



## The Keyboard

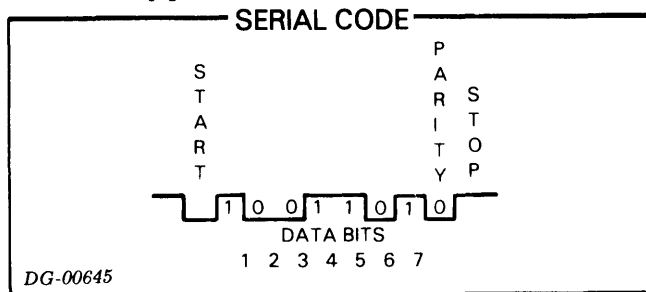
When the operator strikes a key on either the main, screen management or numeric keypad, the appropriate ASCII character code is transmitted to the computer. When the operator strikes a user function key, a sequence of two codes is sent to the computer. The first code is the ASCII control character, Record Separator (036<sub>8</sub>). This code, referred to as the user function header code, notifies the executing program that the next code received should be interpreted as a user-defined code. The second code is one of the ASCII characters listed in Appendix B.

The hardware of the keyboard and the asynchronous interface identifies the type of code generated by the keyboard, standard data or user function, and transmits the appropriate code, or sequence of two codes, in the following manner. The keyboard converts each keystroke into an 8-bit code consisting of a 7-bit ASCII character code and a user function bit. A user function key, when struck, generates a 1 in this bit; all other keys generate a 0. These 8-bit codes are sent to the asynchronous interface via a 4 foot external cable connected to the display's accessory panel.

## Asynchronous Interface

All information transfers between the keyboard, the computer and the CRT display are via the asynchronous interface. The rate at which data is transmitted is determined by the setting of the data rate rotary switch on the accessory panel.

The asynchronous interface transmits and responds to 10-bit serial codes (or 11-bit codes when operating at 110 baud) containing 7 data bits and 1 parity bit. The parity computation for data received from the computer must be the same as that selected for the display terminal by the setting of the parity switch on the accessory panel.



## Transmit Keyboard Data

As each keyboard code is received by the asynchronous interface, the user function bit is tested and then stripped from the code. If it is not a user function code, the multiplexor (refer to the asynchronous interface block diagram) passes the remaining 7-bit ASCII character to a transmitter. If it is, the multiplexor sends the user function header code, 036<sub>8</sub>, to the transmitter, followed by the 7-bit ASCII character, thus providing the two-code, user-function sequence. The transmitter responds to each code received by performing the selected parity computation, adding the parity bit and the start and stop bits, and transmitting the resulting serial code to the computer via either the 20mA current loop or EIA RS-232C communications line.

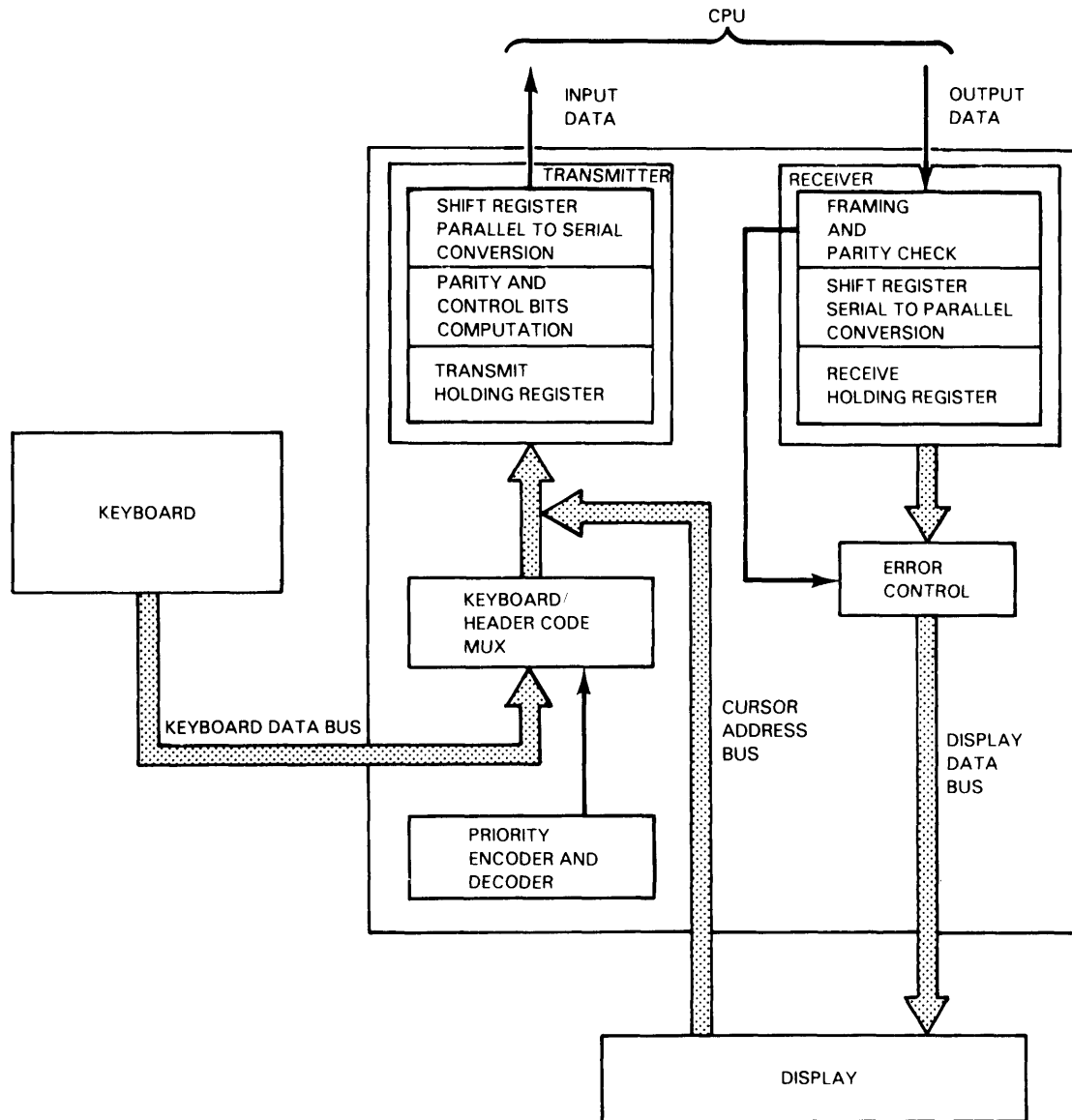
## Transmit Cursor Address

The current position of the cursor is requested by sending a Read Cursor Address code to the display (see Display Commands in Chapter II). When this occurs, the multiplexor sends a cursor address header code (Unit Separator, 037<sub>8</sub>) to the transmitter. The multiplexor is then temporarily disabled by the priority encoder/decoder while two codes, consisting of the cursor's 7-bit column and 5-bit row address, respectively, are sent from the display control logic to the transmitter. Thus, a three-code sequence is sent to the computer. The cursor header code notifies the executing program that the next two codes should be interpreted as the cursor's horizontal and vertical coordinates, respectively.

## Receive Computer Data

As each code is received from the computer, the receiver performs the following functions: it checks for parity and framing errors; strips the parity, start and stop bits from the code; performs a serial to parallel conversion; and passes the parallel code, together with the appropriate error signals, to the error detection logic. If the code is received error-free, it passes to the display's data bus. If the code contains a parity or framing error, it is converted to a code containing all ones and then passes to the data bus. This code is displayed on the model 6052 terminal screen as a question mark and on the model 6053 terminal screen as a 5x7 block with all dots intensified. A Delete character, 177<sub>8</sub>, decoded by the error detection logic is ignored.

## ASYNCHRONOUS INTERFACE



DG-03778

### Control Logic

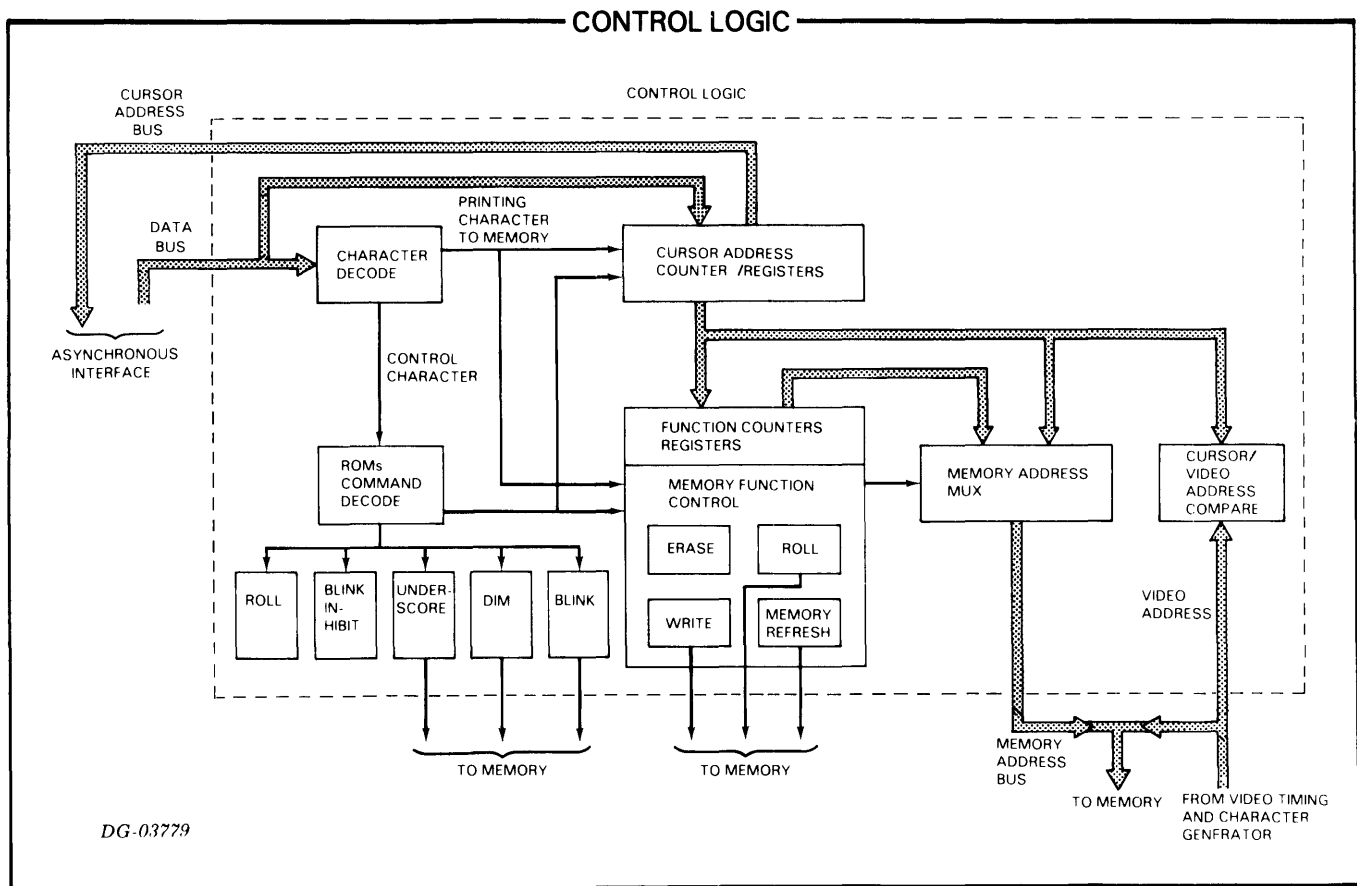
The control logic decodes each code as it appears on the data bus to ascertain if the code is a printing character to be displayed on the screen or if it is a control character. When an ASCII printing character is decoded, the memory function control logic (refer to the control logic block diagram) generates the proper signals to enable the code on the data bus to be written into memory. It also enables the memory address multiplexor to pass the cursor address to the memory address bus. When a control character is

decoded, it is sent to three ROMs for decoding. If it is a display command, the ROMs generate the appropriate signal(s) to carry out the command; if it is not, the code is ignored.

The control logic maintains cursor and memory addresses in counter/registers. These registers provide the memory addresses used during memory write operations. During read operations, the cursor address is compared to the video address appearing on the memory address bus.

The contents of the cursor address registers can be altered or read when the appropriate commands are sent to the display. The contents can be altered in two ways: either by commands which move the cursor relative to its current position (e.g., Cursor Up, Cursor Left) or by a Write Cursor Address command. When the latter is decoded, the next two codes appearing on the data bus are directed to the cursor address registers. The first code is parallel loaded into the cursor's 7-bit column register and the low order five bits of the second code are loaded into its 5-bit row register. The contents are read when a Read Cursor Address command code is sent to the display. When this code is received, two codes containing the cursor's 7-bit column and 5-bit row address, respectively, are sent to the asynchronous interface.

Five flags, Blink, Blink Inhibit, Dim, Underscore, and Roll Screen Enable, reside on the control logic board. These flags are set by display commands. Signals which reflect the state, 0 or 1, of the Blink flag and the Dim and Underscore flags (6053 terminal only) are sent to memory. The state of the Blink Inhibit flag is sent to the video timing and character generator and the state of the Roll flag is sent to the memory function control. When the display is roll enabled by the setting of this flag, the memory function control generates a signal which updates the memory row addresses on the memory board. This causes the information on the screen to roll up one line each time the cursor overflows the bottom line or a New Line code is decoded when the cursor is on the bottom line.



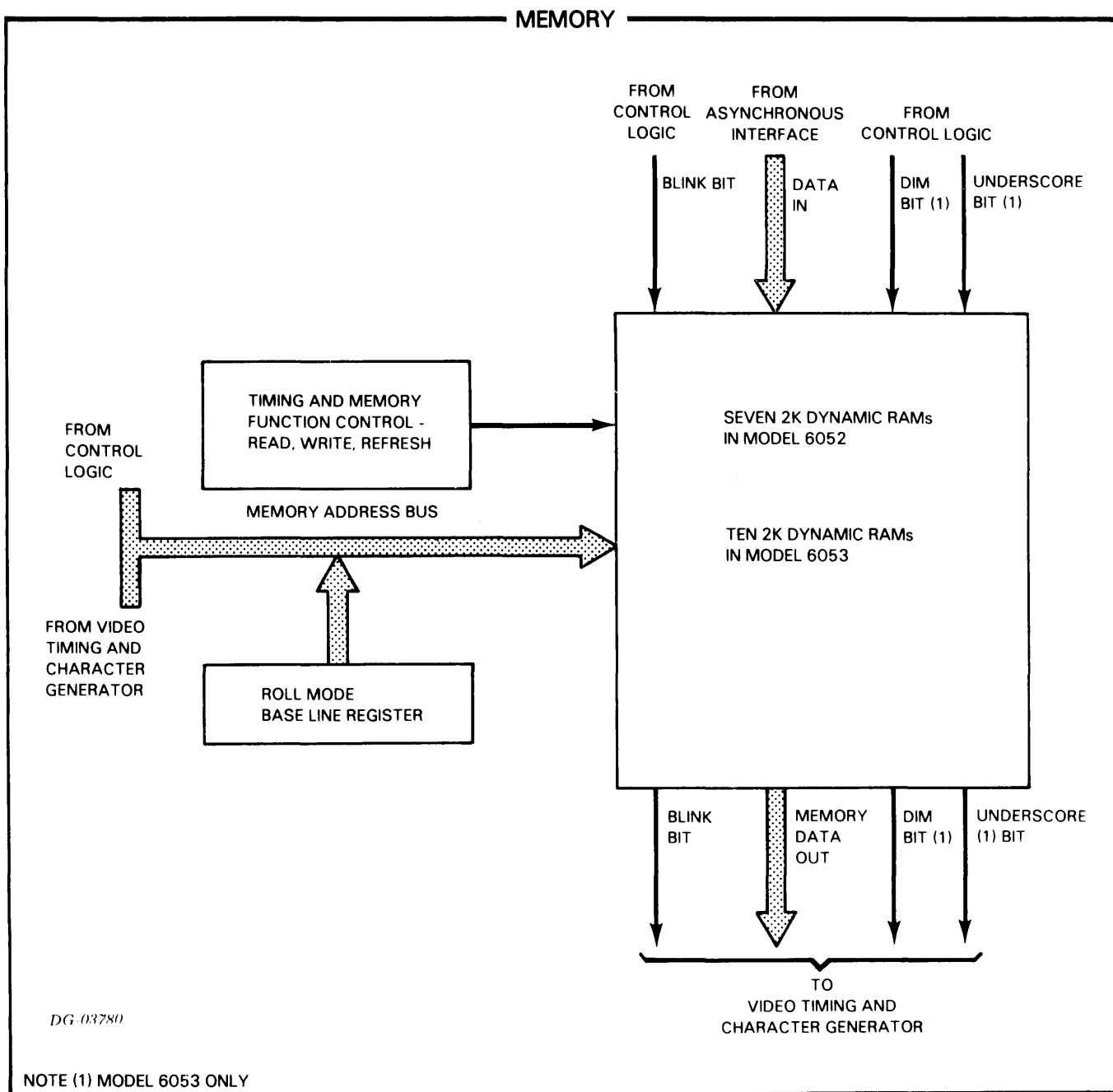
## Memory

The memory provides storage for 1920 characters. These characters are stored in seven 2K dynamic RAMs in the 6052 terminal and in ten 2K dynamic RAMs in the 6053 terminal. Each memory location in the 6052 terminal provides storage for a 6-bit character code (1 bit is dropped from the code when it is stored in memory) plus the blink bit. The 6053 terminal provides storage for a 7-bit character code plus the blink, dim, and underscore bits.

Memory writing is initiated by signals received from the control logic. When a character is written into the 6052 terminal memory, the signal reflecting the state of the Blink flag is sampled and written into the blink bit of that memory location. When a character is written into the 6053 terminal memory, the signals reflecting the state of the Blink, Dim and Underscore flags are sampled and written into the blink, dim and underscore bits of that memory location. In this way, the blink, dim and underscore attributes (on or off) become an integral part of each character stored in memory.

Memory reading is initiated by signals received from the video timing and character generator, which also provides the memory addresses. Eighty characters (one line) are read at a time and are passed to the video timing and character generator via the memory data bus.

The roll mode base line register (refer to the memory block diagram) updates the memory row addresses at the appropriate time when the Roll Screen Enable flag is set.



## Video Timing and Character Generator

This printed circuit board consists of a video timing generator and a video character generator.

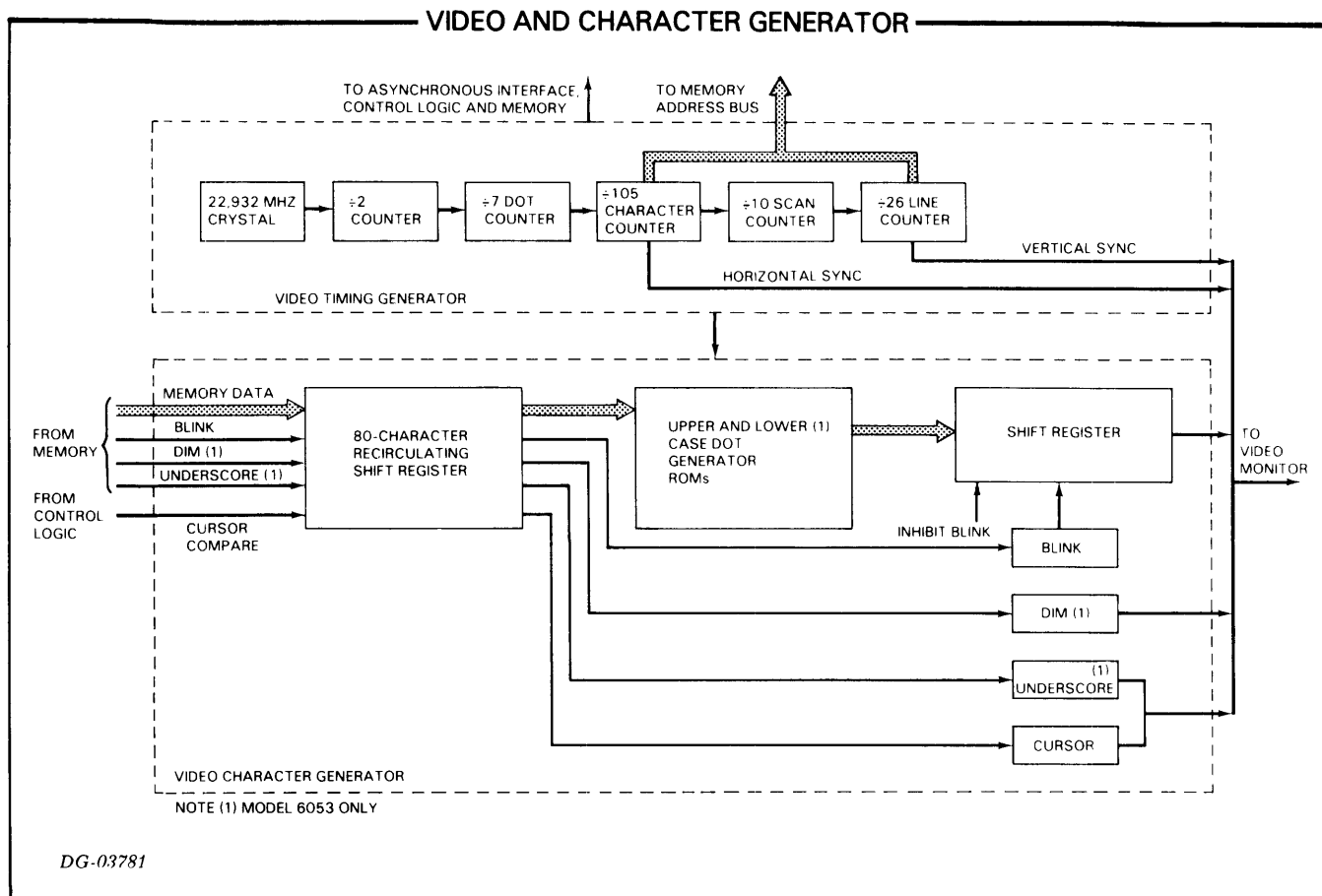
### Video Timing Generator

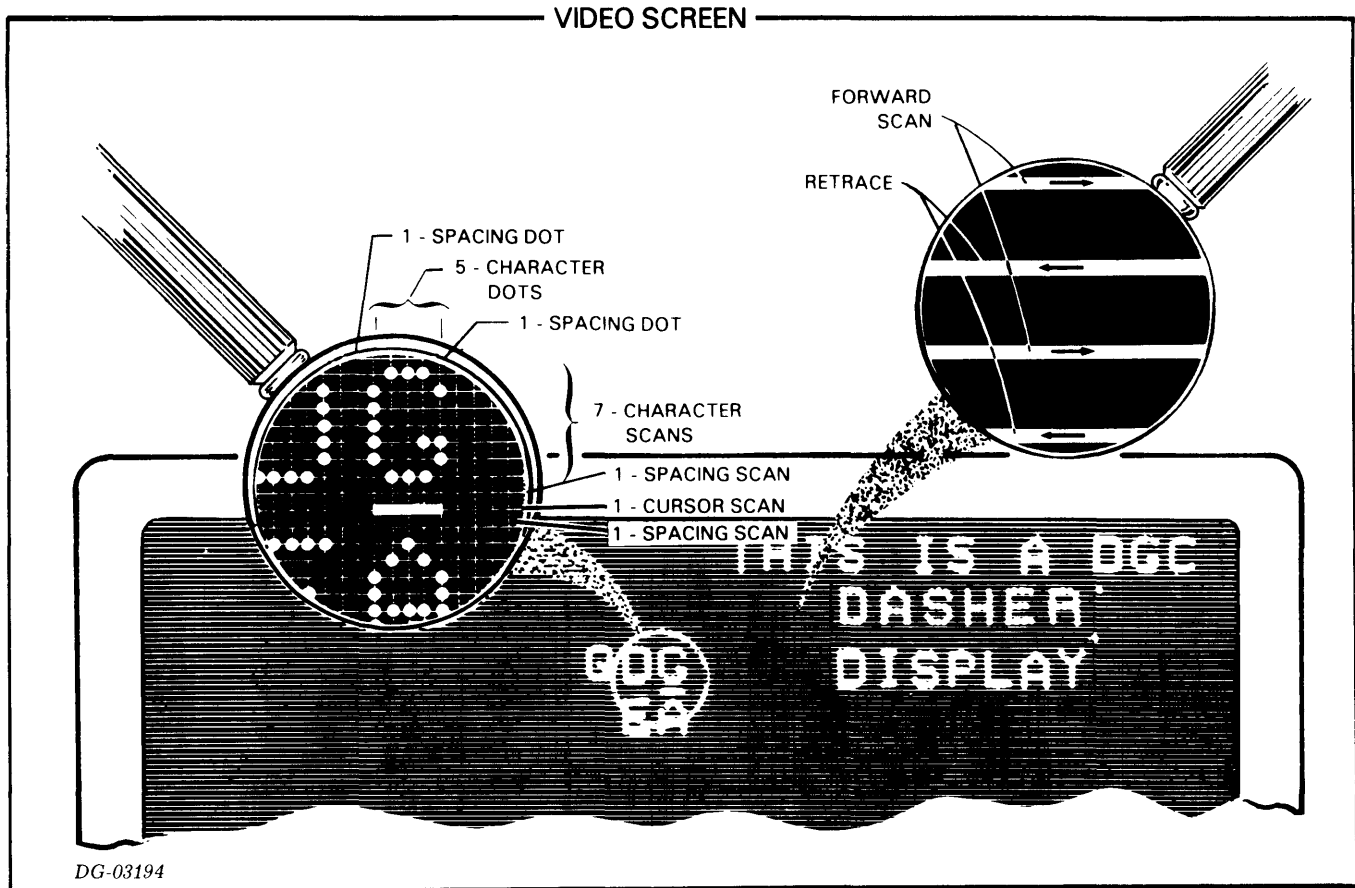
A 22.932MHz crystal oscillator and a group of counters produce the timing signals which control the operation of the display unit. In addition, the video timing logic generates the horizontal/vertical sync and dimming signals, the cursor blink rate and the character blink rate sent to the video monitor and the video addresses sent to memory during read operations.

### Video Character Generator

As memory is read, the information on the memory data bus is loaded into an 80-character recirculating shift register in the video character generator. During this time the memory addresses are compared with

the cursor address to determine if the cursor will appear on the next line of characters to be displayed. Information in the recirculating shift register is read one scan at a time (see the description of the video monitor) and passed either to an upper case dot generator ROM in the 6052 terminal or to the appropriate upper and lower case dot generator ROMs in the 6053 terminal. In the 6052 terminal, the character codes are manipulated in such a way that lower case characters received by this terminal are converted to their upper case equivalents. The output of the dot generator ROM(s) is sent to a shift register. The serial output of this register provides the proper signal to display upper case character codes as 5x7 dot matrices and lower case as 5x8 dot matrices on the CRT screen. The signal reflecting the state of the Blink Inhibit flag is sent to the shift register. If blinking is enabled, the characters whose blink bit contains a one are blinked on the screen; if blinking is inhibited, none of the characters displayed on the screen are blinked. In the 6053 terminal, characters whose dim and/or underscore bits are set to 1 are dimmed and/or underscored, respectively.





DG-03194

## Video Monitor

The video monitor is the display's output, providing a visual image of each character stored in memory. The monitor's 12 inch diagonal screen has an active area of 8.5 inches by 5.5 inches with a margin at the top, bottom and both sides. It displays alphanumeric characters as dot matrices by using a pattern of lines traced with an electron beam.

The monitor is a magnetically deflected device which uses a non-interlaced raster pattern of scanning. A raster is a system of parallel horizontal lines to display characters on the CRT screen. Discrete positions within the raster are defined in the video timing logic by a number of parameters: line, scan, character and dot positions.

The monitor receives horizontal and vertical synchronizing signals from the video timing logic. These signals synchronize and trigger electromagnetic deflection circuits which move the CRT's electron beam horizontally and vertically around the screen. The horizontal deflection circuit is triggered by a 15.6kHz signal while the vertical deflection circuit is triggered by a 50Hz or 60Hz signal. This combination of scanning frequencies produces 260 horizontal scans for each vertical scan.

The electron beam scans the screen to display 24 lines, 80 characters wide. Each line consists of 10 horizontal

scans. As the monitor's electron beam makes each horizontal scan, it traces one slice of a line. These scans are divided into 80 character positions across the screen. Each character position is divided into 7 dot positions in which the electron beam is either intensified or held at a low intensity so that a visible trace does not appear on the screen. This scanning provides a character field made up of 7x10 distinct dots.

## 6054/6055 Printer Option

The 6054/6055 printer I/O controller transmits information stored in the display's memory to the printer in 10-bit serial form, or 11-bit form when operating at 110 baud. The rate at which information is transmitted is switch-selectable on the printed circuit board.

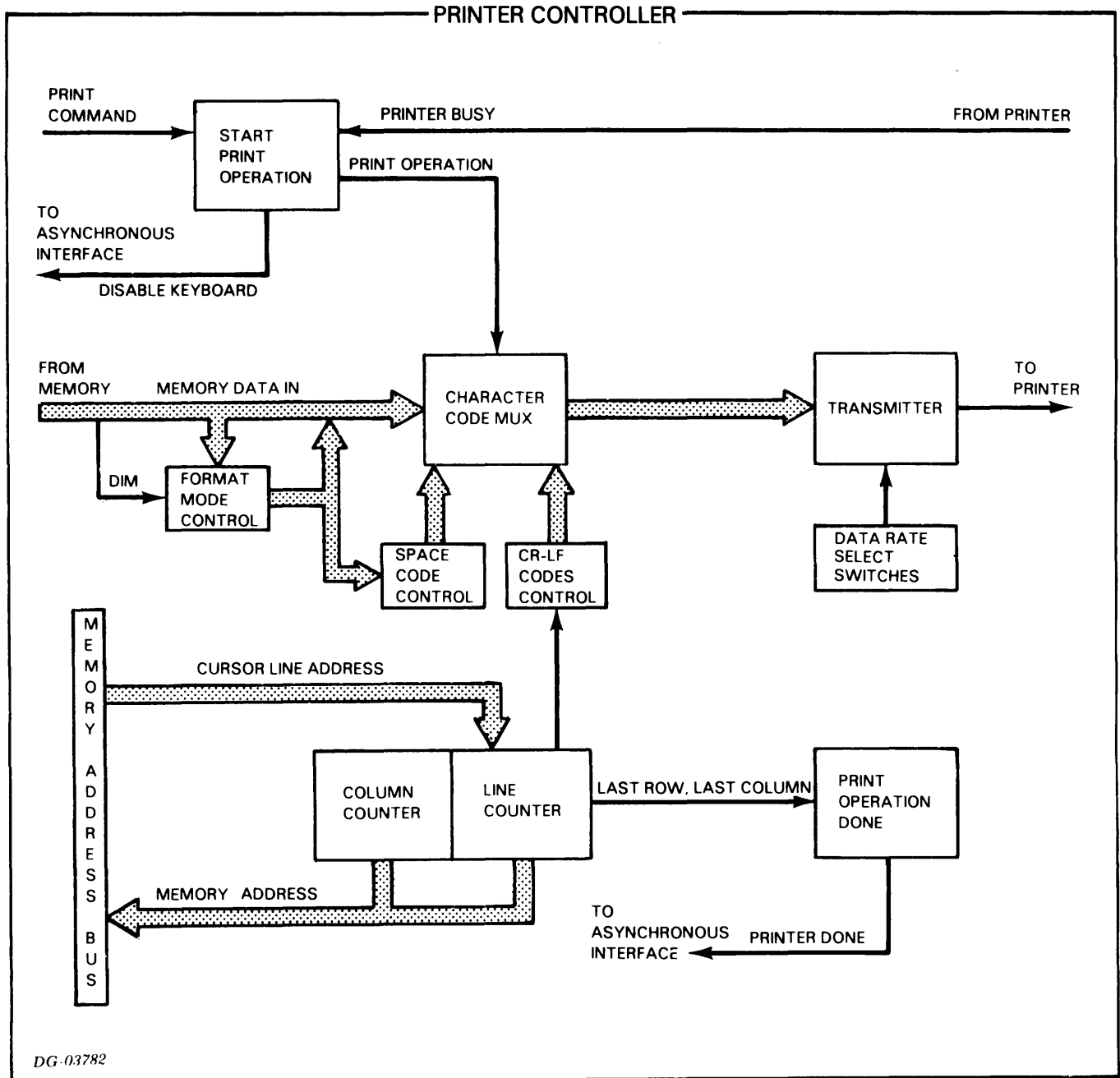
The printer controller transmits information beginning with the leftmost character of the line on which the cursor resides to the end of the screen. Column and line counters generate the addresses sent to memory during a print operation.

When a print command is received, a printer busy signal is sent to the asynchronous interface which disables the keyboard. The cursor line address is loaded into the line counter, the column counter is set to 0, and a Carriage Return and a Line Feed code are sent to the printer.

Memory reading begins with the first scan of the cursor line. As information is read from memory, it is transmitted to the printer and the address counters are incremented appropriately. At the end of each line, a Carriage Return and a Line Feed code are transmitted. Trailing spaces at the end of lines are not transmitted. When all information has been sent, the controller sends a printer done signal to the asynchronous interface. This signal frees the

keyboard and generates a printer done code (006<sub>8</sub>) which is transmitted to the computer.

When a print format command is sent to the 6053 terminal, the controller assumes format mode. In this mode, a character containing the dim attribute (dim bit is set to 1) is interpreted by the controller as a space and only a character whose dim bit is set to 0 is transmitted to the printer.



For more detailed information concerning the theory of operation, refer to the *Technical Manual, 6052-6053 DASHER Display Terminals*, DGC No. 015-000065.



# THE 6052, 6053 DISPLAY TERMINAL SPECIFICATIONS

## PROPERTIES

Mounting .....	Tabletop
	Display dimensions: 12 5" X 14 5" X 20 5"
	Keyboard dimensions: 2 75" X 9" X 20"
Weight .....	Display: 45 pounds Keyboard: 7 pounds
Operating Temperature .....	0 degC to 45 degC
Relative Humidity Range .....	10% to 95%, non-condensing, operating
Operating Altitude .....	10,000 feet maximum
Power Requirements .....	Single Phase, 120Volts, 60Hz ( $\pm 2$ Hz) 100Volts, 50Hz ( $\pm 2$ Hz) 240Volts, 50Hz ( $\pm 2$ Hz) 220Volts, 50Hz ( $\pm 2$ Hz)

## SYSTEM CAPACITY

Number of displays/system .....	Depends on controller
Display Information Capacity .....	1920 characters, formatted as 24 lines, 80 characters/line in a viewing area of 8 5 inches by 5 5 inches on a CRT screen

## CHARACTERISTICS

Data Transmission Mode .....	Full-duplex
Data Transmission Type .....	Asynchronous
Code Format .....	ASCII, 10 or 11-bit, serial
Parity .....	Even, odd or mark, switch selectable
Data Rate .....	110 to 19 2K baud, switch selectable
Interface .....	20mA EIA RS-232C (depending on cable)
Keyboard .....	Model 6052                      Model 6053
Main keypad style .....	Teletypewriter style with      Typewriter style, with 4 4-foot external, plug-in      foot external, plug-in cable to display              cable to display
Type .....	Solid state capacitive switching
Numeric keypad keys .....	11                      11
Screen management keypad keys .....	10                      12
User function keypad keys .....	08                      11
Character Set .....	Model 6052                      Model 6053
	ASCII 64 upper case              ASCII 96 upper and lower alphanumeric              case alphanumeric
International Fonts .....	U K, German, Swedish, French
Display	
Program addressable cursor .....	Read current cursor address or send new cursor position
Blink .....	Character by character
Underscore .....	Model 6053 only, character by character
Dim .....	Model 6053 only, character by character
Roll Screen Enable .....	Program selectable
Video Monitor	
Type .....	Cathode Ray Tube
Screen size .....	8 5 inches by 5 5 inches viewing area, 12 inches diagonal
Character style .....	Model 6052 5 X 7 dot matrix Model 6053 5 X 8 dot matrix
Display .....	Non-interlaced raster scan
Phosphor .....	P4 White
Deflection .....	Magnetic
Focus .....	Dynamic and static

This page intentionally left blank.

# CHAPTER II

## PROGRAMMER'S

## REFERENCE

### INTRODUCTION

The 6052/6053 display terminals consist of two independent I/O devices, a keyboard console as the input device and an alphanumeric CRT display as the output device. The keyboard transmits and the display responds to a subset of the 128-ASCII character set listed in Appendix A.

Because the keyboard and the display function as independent devices, the program must echo the characters received from the keyboard if these characters are to be displayed as they are typed. The table at right summarizes the programming instructions and conventions used when the terminals operate with one of Data General's basic I/O asynchronous controllers.

### Keyboard

When a code-generating key is struck on the main, numeric or screen management keypads, an 8-bit code consisting of 7 data bits and 1 parity bit is transmitted to the computer. Appendix A lists the ASCII character codes generated by these keypads.

### User Function Codes

When a user function key is struck a sequence of two codes is transmitted to the computer. The first code of the sequence is the ASCII control character Record Separator (036<sub>8</sub>). The second code is the appropriate ASCII character generated by the user function key. These codes are listed in Appendix B.

### PROGRAMMING SUMMARY

Mnemonic (First Controller)	
Input	TTI
Output	TTO
Device Code (First Controller)	
Input	10 <sub>8</sub>
Output	11 <sub>8</sub>
Mnemonic (Second Controller)	
Input	TTI1
Output	TTO1
Device Code (Second Controller)	
Input	50 <sub>8</sub>
Output	51 <sub>8</sub>
Priority Mask Bit	
Input	14
Output	15
Characters per Line	80
Lines per Display	24
Total Storage Capacity (Characters)	1920
Data Transfer Rate Max (Baud)	19,200

### ACCUMULATOR FORMATS

Read Character Buffer (DIA)

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

Load Character Buffer (DOA)

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

### S, C AND P FUNCTIONS

- f = S Set Busy to 1, Done to 0 and either load the Input Buffer or write a character into the display's memory.
- f = C Set both Busy and Done to 0 and terminate all data transfers. If issued before transmission is complete, partial character codes are received.
- f = P No effect.

## Display

The display responds to ASCII character codes received from the computer which contain 7 data bits and a parity bit. The parity computation for received data must be the same as that selected for the terminal by the setting of the parity switch. When either a parity or framing error occurs in a code received by the display, a question mark is displayed on the screen of the 6052 terminal and a 5x7 block is displayed on the screen of the 6053 terminal.

Printing characters received by the display are plotted on its screen (lower case characters received by the 6052 terminal display are plotted as their upper case equivalents). ASCII control characters representing display commands move the cursor around the screen, control blinking, dimming and underscoring of printing characters, erase information from the screen, and enable rolling of the screen. One command causes the display to transmit the current horizontal and vertical coordinates of the cursor to the computer. (Refer to the display commands described later in the text.) ASCII control characters which are not display commands are ignored by the display.

### Character Blinking

The display unit of both terminals contains two flags, Blink and Blink Inhibit, which control the blinking of characters on the screen. These flags are set by the appropriate display commands. The current state of the Blink flag, 0 or 1, is stored in memory with each printing character as it is received. Thus, each memory location contains the character bits plus the blink bit. The display is blink enabled or disabled by the state of the Blink Inhibit flag. When it is blink enabled, each printing character whose blink bit is set to 1 is blinked on the screen; when it is blink disabled, none of the characters blink.

In addition to the appropriate blink display commands, each time an ERASE PAGE command is generated the blink flag is set to 0 and the display is blink enabled.

## Character Dimming/Underscoring

The 6053 terminal display contains two flags, Dim and Underscore, which control the dimming and underscoring of printing characters. These flags are set by the appropriate display commands. The current state, 0 or 1, of both flags is stored in memory with each printing character as it is received. Thus, each memory location in the 6053 display contains the character bits, the blink bit, the dim bit and the underscore bit. Each printing character whose dim bit is set to 1 is displayed at reduced intensity; each printing character whose underscore bit is set to 1 is underscored on the screen.

In addition to the appropriate dim and underscore commands, each time an ERASE PAGE command is generated, both flags are set to 0.

### Screen Rolling

The display unit of both terminals contains a Roll Enable flag, which controls the rolling of the screen. This flag is set by the appropriate display commands. When the display is roll enabled, the screen rolls up one line each time the cursor overflows the bottom line or a New Line code is received when the cursor is on the bottom line. When the display is not roll enabled, the cursor moves to the first character position on the top line each time the cursor overflows the bottom line or a New Line code is received when the cursor is on the bottom line.

The ERASE PAGE command does not affect the Roll Enable flag.

### Cursor Addressing

The non-destructive cursor defines the location in memory where a character is stored and subsequently the position on the screen where it is displayed. The cursor is moved around the screen in two ways. It is moved relative to its current position (e.g., up, down, left or right). And, it is moved by sending a WRITE CURSOR ADDRESS command to the display followed by two binary codes containing the new cursor 7-bit column and 5-bit row address, respectively.

### Cursor Sensing

The current position of the cursor is requested by sending a READ CURSOR ADDRESS command to the display. When this occurs, the terminal transmits a three-code sequence to the computer. The first code of the sequence is the ASCII control character Unit Separator (037<sub>8</sub>), the second is the cursor's 7-bit column address, and the third is the cursor's 5-bit row address. The program retrieves this information by issuing data in instructions to the keyboard.

## DISPLAY COMMANDS

The display commands are described in the following format:

**COMMAND NAME** (octal command code)

*Function:* Functional description

*6052 Keyboard:*  
Ways to generate on  
main keypad

*6053 Keyboard:*  
Ways to generate on  
main keypad

### ENABLE BLINK (003<sub>8</sub>)

*Function:* Each character whose blink bit contains a 1 is blinked on the CRT screen.

*6052 Keyboard:*  
CTRL-C, CTRL-SHIFT-C

*6053 Keyboard:*  
CTRL-C, CTRL-SHIFT-C

### DISABLE BLINK (004<sub>8</sub>)

*Function:* None of the characters displayed on the screen are blinked.

*6052 Keyboard:*  
CTRL-D, CTRL-SHIFT-D

*6053 Keyboard:*  
CTRL-D, CTRL-SHIFT-D

### START BLINK (016<sub>8</sub>)

*Function:* The blink bit of each succeeding printing character received is set to 1. When the display is blink enabled, each character whose blink bit contains a 1 is blinked on the screen.

*6052 Keyboard:*  
CTRL-N, CTRL-SHIFT-N

*6053 Keyboard:*  
CTRL-N, CTRL-SHIFT-N

### END BLINK (017<sub>8</sub>)

*Function:* The blink bit of each succeeding character received is set to 0.

*6052 Keyboard:*  
CTRL-O, CTRL-SHIFT-O

*6053 Keyboard:*  
CTRL-O, CTRL-SHIFT-O

### START DIM (034<sub>8</sub>)

This command applies to the 6053 terminal only. While the 6052 terminal's keyboard can transmit this command, its display cannot respond to it.

*Function:* The dim bit of each succeeding printing character received is set to 1. Each character whose dim bit contains a 1 is displayed at reduced intensity (dimmed) on the screen.

*6052 Keyboard:*  
CTRL-SHIFT-L

*6053 Keyboard:*  
CTRL-\ , CTRL-SHIFT-\

### END DIM (035<sub>8</sub>)

*Function:* The dim bit of each succeeding printing character received is set to 0.

*6052 Keyboard:*  
CTRL-SHIFT-M

*6053 Keyboard:*  
CTRL-SHIFT-|

### START UNDERSCORE (024<sub>8</sub>)

This command applies to the 6053 terminal only. While the 6052 terminal's keyboard can transmit this command, its display cannot respond to it.

*Function:* The underscore bit of each succeeding printing character received is set to 1. Each character whose underscore bit is set to 1 is underscored on the screen.

*6052 Keyboard:*  
CTRL-T, CTRL-SHIFT-T

*6053 Keyboard:*  
CTRL-T, CTRL-SHIFT-T

### END UNDERSCORE (025<sub>8</sub>)

*Function:* The underscore bit of each succeeding printing character received is set to 0.

*6052 Keyboard:*  
CTRL-U, CTRL-SHIFT-U

*6053 Keyboard:*  
CTRL-U, CTRL-SHIFT-U

### BELL (007<sub>8</sub>)

*Function:* A short audible tone is produced on the speaker located in the display unit.

*6052 Keyboard:*  
CTRL-G, CTRL-SHIFT-G

*6053 Keyboard:*  
CTRL-G, CTRL-SHIFT-G

## HOME (010<sub>8</sub>)

*Function:* The cursor moves to the first (leftmost) character position on the top line of the screen, which is the cursor home position.

*6052 Keyboard:*  
CTRL-H,CTRL-SHIFT-H  
*Screen Management Keypad:*  
HOME

*6053 Keyboard:*  
CTRL-H,CTRL-SHIFT-H  
*Screen Management Keypad:*  
HOME

## NEW LINE (012<sub>8</sub>)

*Function:* The cursor moves to the first character position on the next line of the screen. If the cursor is on the bottom line, it moves to the home position, unless the terminal is roll enabled (refer to the ROLL ENABLE command described below.)

*6052 Keyboard:*  
NEW LINE,SHIFT-NEW LINE,  
CTRL-NEW LINE,  
CTRL-SHIFT-NEW LINE,  
CTRL-J,CTRL-SHIFT-J

*6053 Keyboard:*  
NEW LINE,SHIFT-NEW LINE,  
CTRL-NEW LINE,  
CTRL-SHIFT-NEW LINE,  
CTRL-J,CTRL-SHIFT-J

## ERASE TO END OF LINE (013<sub>8</sub>)

*Function:* The screen clears from the position of the cursor to the end of the line on which the cursor resides.

*6052 Keyboard:*  
CTRL-K  
*Screen Management Keypad:*  
ERASE EOL

*6053 Keyboard:*  
CTRL-K,CTRL-SHIFT-K  
*Screen Management Keypad:*  
ERASE EOL

## ERASE PAGE (014<sub>8</sub>)

*Function:* The screen clears and the cursor moves to the home position. Additionally, this command performs an End Blink, Enable Blink, End Dim (model 6053) and End Underscore (model 6053) operation.

*6052 Keyboard:*  
CTRL-L  
*Screen Management Keypad:*  
ERASE PAGE

*6053 Keyboard:*  
CTRL-L,CTRL-SHIFT-L  
*Screen Management Keypad:*  
ERASE PAGE

## CARRIAGE RETURN (015<sub>8</sub>)

*Function:* The cursor moves to the first character position on the line on which the cursor resides.

*6052 Keyboard:*  
CR,CTRL-CR,CTRL-M  
SHIFT-CR,CTRL-SHIFT-CR

*6053 Keyboard:*  
CR,CTRL-CR,CTRL-M  
SHIFT-CR,CTRL-SHIFT-CR  
CTRL-SHIFT-M

## WRITE CURSOR ADDRESS (020<sub>8</sub>)

*Function:* The display is forced to use the next two codes received as the cursor's new 7-bit column address and 5-bit line address, respectively. Both the columns and lines are numbered beginning with column 0, line 0. The columns are numbered from left to right across the screen; the lines are numbered from top to bottom. After the second character is received, the cursor moves to the location on the screen specified by the new coordinates.

*6052 Keyboard:*  
CTRL-P

*6053 Keyboard:*  
CTRL-P,CTRL-SHIFT-P

## READ CURSOR ADDRESS (005<sub>8</sub>)

*Function:* A sequence of three codes is sent from the display to the computer. The first code is the ASCII control character, Unit Separator (US, 037<sub>8</sub>); the second is the cursor's current 7-bit column address; and the third is the cursor's current 5-bit line address.

*6052 Keyboard:*  
CTRL-E,CTRL-SHIFT-E

*6053 Keyboard:*  
CTRL-E,CTRL-SHIFT-E

## ROLL ENABLE (022<sub>8</sub>)

*Function:* The display is roll enabled; that is, the screen rolls up one line each time either the cursor overflows the bottom line or a New Line character is decoded when the cursor is on the bottom line. When this occurs, the cursor moves to the first character position on the bottom line, the bottom line becomes blank, and the information previously displayed on the top line is lost.

*6052 Keyboard:*  
CTRL-R,CTRL-SHIFT-R

*6053 Terminal:*  
CTRL-R,CTRL-SHIFT-R

## ROLL DISABLE (023<sub>8</sub>)

*Function:* The roll enable mode is terminated. When the display is not roll enabled, the cursor moves to the home position each time either the cursor overflows the bottom line or a New Line code is decoded when the cursor is on the bottom line. While the cursor position changes, the information displayed on the screen remains unchanged.

*6052 Keyboard:*  
CTRL-S, CTRL-SHIFT-S

*6053 Keyboard:*  
CTRL-S, CTRL-SHIFT-S

## CURSOR UP (027<sub>8</sub>)

*Function:* The cursor moves up one line while remaining in the same column position. If the cursor is on the top line, it moves to the bottom line of the screen.

*6052 Keyboard:*  
CTRL-W, CTRL-SHIFT-W  
*Screen Management Keypad:*  
↑

*6053 Keyboard:*  
CTRL-W, CTRL-SHIFT-W  
*Screen Management Keypad:*  
↑

## CURSOR RIGHT (030<sub>8</sub>)

*Function:* The cursor moves one character (column) position to the right. If the cursor is at the end of the line, a New Line operation is performed.

*6052 Keyboard:*  
CTRL-X, CTRL-SHIFT-X  
*Screen Management Keypad:*  
→

*6053 Keyboard:*  
CTRL-X, CTRL-SHIFT-X  
*Screen Management Keypad:*  
→

## CURSOR LEFT (031<sub>8</sub>)

*Function:* The cursor moves one character position to the left. If the cursor is in the leftmost position on the line, it moves to the rightmost position and a Cursor Up operation is performed.

*6052 Keyboard:*  
CTRL-Y, CTRL-SHIFT-Y  
*Screen Management Keypad:*  
←

*6053 Keyboard:*  
CTRL-Y, CTRL-SHIFT-Y  
*Screen Management Keypad:*  
←

## CURSOR DOWN (032<sub>8</sub>)

*Function:* The cursor moves down one line while remaining in the same column position. If the cursor is on the bottom line, it moves to the top line.

*6052 Keyboard:*  
CTRL-Z, CTRL-SHIFT-Z  
*Screen Management Keypad:*  
↓

*6053 Keyboard:*  
CTRL-Z, CTRL-SHIFT-Z  
*Screen Management Keypad:*  
↓

## PRINT (021<sub>8</sub>)

This command applies to terminals equipped with either a model 6054 or 6055 printer option.

*Function:* Information displayed on the CRT Screen, beginning with the leftmost character on the cursor line and ending with the rightmost character on the bottom line, is printed. During the print operation, the keyboard is disabled.

*6052 Keyboard*  
CTRL-Q, CTRL-SHIFT-Q  
*Screen Management Keypad*  
PRINT

*6053 Terminal:*  
CTRL-Q, CTRL-SHIFT-Q  
*Screen Management Keypad*  
PRINT

## PRINT FORM (001<sub>8</sub>)

This command applies to the 6053 terminal equipped with either a model 6054 or 6055 printer option. It can be used for data entry with pre-printed forms.

*Function:* This command is the same as the PRINT command described above, except that dimmed characters are interpreted as spaces. Only characters displayed at full intensity are transmitted to the printer.

*6052 Keyboard*  
CTRL-A, CTRL-SHIFT-A  
*Screen Management Keypad*  
SHIFT-PRINT

*6053 Keyboard*  
CTRL-A, CTRL-SHIFT-A  
*Screen Management Keypad*  
SHIFT-PRINT

## INSTRUCTIONS

The following instructions and timing information apply when the display terminals are used in conjunction with Data General's 4010 controller.

The 4010 controller contains an 8-bit input buffer and an independent 8-bit output buffer. Since the keyboard and display function as independent devices, one set of Busy and Done flags is available for input operations and another set for output operations.

The controller's Busy and Done flags are controlled using two of the device flag commands as follows:

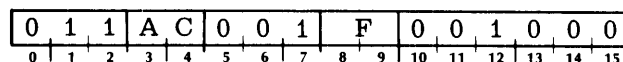
**f=S** Sets Busy to 1, Done to 0 and either reads a character into the input buffer or writes the character in the output buffer into the display's memory.

**f=C** Sets Busy and Done to 0 and terminates all data transfer operations. A Clear command issued during a transfer will result in the partial reception of the code being transferred.

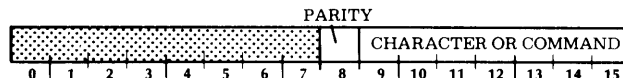
**f=P** No effect.

## Read Character Buffer

**DIA [f] ac, TTI**



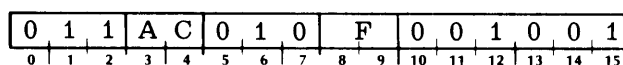
The contents of the input buffer are placed in bits 8-15 of the specified AC. Bits 0-7 of the specified AC are set to 0. After the data transfer, the controller's input Busy and Done flags are set according to the function specified by F. The format of the specified AC is as follows:



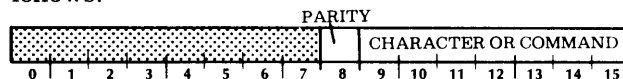
Bits	Name	Contents
0-7	----	Reserved for future use.
8	Parity	Parity bit selected at the terminal: even, odd or mark.
9-15	Character	The 7-bit character code read from the input buffer.

## Load Character Buffer

**DOA [f] ac, TTO**



Bits 8-15 of the specified AC are loaded into the output buffer. After the data transfer, the controller's output Busy and Done flags are set according to the function specified by F. The format of the specified AC is as follows:



Bits	Name	Contents
0-7	----	Reserved for future use.
8	Parity	Even, odd or mark parity. Tested by the display.
9-15	Character	The 7-bit character code transmitted to the output buffer.



## PROGRAMMING

Since the terminal is actually two separate devices, input and output are discussed separately.

### Input

Striking any code-generating key or key combination, code-generating key plus CTRL and/or SHIFT, transmits a serial code containing 7 data bits and a parity bit to the controller. The controller assembles the data and parity bits into an 8-bit parallel code in its input buffer. When the code is fully assembled, the controller's input Busy flag is set to 0, its input Done flag is set to 1, and a program interrupt request is initiated. The program reads the character in the input buffer by issuing a READ CHARACTER BUFFER instruction (DIA) to the keyboard (TTI), after which it sets the Done flag to 0 by issuing a Start or Clear flag command. This allows the next character assembled in the controller's input buffer to generate a program interrupt request.

### Output

The program sends a character to the controller's output buffer by issuing a LOAD CHARACTER BUFFER instruction (DOA) to the display (TTO). A Start flag command initiates a parallel to serial conversion and transmits the code to the display. While the character is being transmitted, the output Busy flag is set to 1 and the output Done flag is set to 0. When the transmission is completed, the output Busy flag is set to 0, the output Done flag is set to 1, and a program interrupt request is initiated.

Each time the program sends a character to the display, the output buffer must be reloaded with a LOAD CHARACTER BUFFER instruction. A sequence of LOAD CHARACTER BUFFER instructions together with Start flag commands transmit a multi-character message. The program must allow each character to be transmitted before sending another character to the controller's output buffer.

## TIMING CONSIDERATIONS

### Input

After the controller's input Done flag is set to 1, the character in the input buffer is available to the program for a time interval determined by the baud rate before another key strike can destroy the character. To avoid possible data loss, the program must respond to the program interrupt request and read the character within the time interval indicated in the following table.

Baud Rate	Maximum Allowable Programmed I/O Latency (ms)
110	21.59
150	15.84
300	7.92
600	3.96
1,200	1.98
1,800	1.48
2,400	0.98
4,800	0.49
9,600	0.24
19,200	0.12

### Output

After the controller's output Done flag is set to 1, the program should provide another character within a time period determined by the baud rate to keep the transmission line operating at its maximum rate.

Baud Rate	Time Limit (ms)
110	9.15
150	6.64
300	3.32
600	1.66
1,200	0.83
1,800	0.62
2,400	0.42
4,800	0.21
9,600	0.11
19,200	0.06

## PROGRAMMING EXAMPLES

The following examples show how a character passes from the keyboard to the computer and from the computer to the display. The first example reads a character from the keyboard, and the second example sends a character to the display.

### Example 1

#### Read a character from the keyboard

```
SKPDN    TTI        ;CHARACTER BUFFER
          ; LOADED?
JMP       :-1        ;NO, TRY AGAIN
DIAC      0,TTI      ;READ CHARACTER AND
          ; CLEAR DONE
```

### Example 2

#### Send a character to the display

```
SKPBZ     TTO        ;DISPLAY FREE?
JMP       :-1        ;NO, TRY AGAIN
DOAS      0,TTI      ;SEND CHARACTER TO
          ; DISPLAY
```

The subroutine shown in example 3 is called by a JUMP TO SUBROUTINE instruction (JSR to TTYRD). It illustrates both reading and echoing characters, using AC0 to store the character.

### Example 3

#### Subroutine to read and echo characters (interrupts disabled)

```
TTYRD:    SKPDN      TTI        ;CHARACTER BUFFER
          ; LOADED?
          JMP        :-1        ;NO, TRY AGAIN
          DIAC       0,TTI      ;READ CHARACTER AND
          ; CLEAR DONE
          SKPBZ      TTO        ;DISPLAY FREE?
          JMP        :-1        ;NO, TRY AGAIN
          DOAS       0,TTO      ;SEND CHARACTER
          ; TO DISPLAY
```

## Cursor Address Codes

The cursor address is sent to the computer in response to a read cursor address code (005<sub>8</sub>) received by the display. The address is contained in two codes of a three-code sequence. The first code of the sequence is always the cursor address header code (037<sub>8</sub>), the second is the 7-bit column address, and the third is the 5-bit row address.

Example 4 illustrates how cursor address codes can be retrieved from the data stream of an echo program. It uses the header code as a flag. When the flag is set, a counter is initialized and the next two characters read are stored in locations CCOL and CROW, respectively. CCOL and CROW are located in page zero of memory (first 377<sub>8</sub> locations), allowing global access to this information. After the second character is stored, the program jumps to a user-provided subroutine, called CADD in the example.

When a cursor address subroutine is not provided, the programmer should ensure that cursor address codes are not echoed to the display, since these codes will be interpreted as ASCII characters.

## Example 4

Read/echo characters and go to subroutine  
on receipt of cursor addresses  
(interrupts disabled)

```

LOC 300
CADD      ;ADDRESS OF USER CURSOR ADDRESS
          ; SUBROUTINE
CCOL:     0      ;CURSOR COLUMN ADDRESS
          ; IS STORED HERE
CROW:     0      ;CURSOR ROW ADDRESS
          ; IS STORED HERE

LOC 400
TTYRD:    LDA     1,CAHDR ;GET CURSOR ADDRESS
          ; HEADER CODE
START:    LDA     2,FLAG  ;INITIALIZE FLAG
          SKPDN    TTI     ;CHARACTER BUFFER
          ; LOADED?
          JMP      -1     ;NO, TRY AGAIN
          DIAC     0,TTI  ;READ CHARACTER AND
          ; CLEAR DONE
          SUB#     0,1,SZR ;HEADER CODE?
          JMP      +5     ;NO
          MOV      0,2    ;YES, SET FLAG
          LDA      3,CCNT ;START COUNTER
          STA      3,ACNT ;
          JMP      START ;READ NEXT CHARACTER
          MOV#     2,2,SNR ;FLAG SET?
          JMP      +8     ;NO
          DSZ      ACNT   ;YES, ROW ADDRESS?
          JMP      +4     ;NO, COLUMN ADDRESS
          STA      0,CROW ;STORE ROW ADDRESS
          LDA      2,FLAG ;CLEAR FLAG
          JMP      @CADD ;GO TO USER SUBROUTINE
          STA      0,CCOL ;STORE COLUMN ADDRESS
          JMP      START ;READ NEXT CHARACTER
          SKPBZ    TTO    ;DISPLAY FREE?
          JMP      -1     ;NO, TRY AGAIN
          DOAS     0,TTO  ;SEND CHARACTER TO
          ; DISPLAY

CAHDR:    037
FLAG:     0
CCNT:     2
ACNT:     0

```

## User Function Codes

User function codes are received by the computer in a two-code sequence. The first code is always the user function header (036<sub>8</sub>); the second is an ASCII character whose function must be defined by the user program.

When a user function subroutine is not provided, the program should ensure that the ASCII characters representing function codes are not echoed to the display, since some of these characters will introduce data errors.

### Print and Print Form Codes

When the display is equipped with a printer option, the PRINT (021<sub>8</sub>) and PRINT FORM (001<sub>8</sub>) codes must be echoed to the display. These codes are transmitted as user function sequences when they are generated by the PRINT key. Example 5 illustrates how user function codes, other than the PRINT and PRINT FORM codes, can be retrieved from the data stream of an echo program. It uses the header code as a flag. When the flag is set, the next character received is tested. If it is a PRINT or PRINT FORM code, the character is echoed. If it is not, the program ignores the character and waits for the character buffer to be loaded with the next character. When a print operation is started, keyboard input is disabled; when it is finished, the terminal sends an 006<sub>8</sub> code to the computer and the keyboard is reenabled.

## Example 5

```

      .LOC 400
START: LDA      1,UFHDR  ;GET USER FUNCTION
                        ; HEADER CODE
      SKPDN     TTI      ;CHARACTER BUFFER
                        ; LOADED?
      JMP      -1        ;NO, TRY AGAIN
      DIAC      0,TTI    ;READ CHARACTER AND
                        ; CLEAR DONE
      SUB#      0,1,SZR  ;USER FUNCTION HEADER?
      JMP      +3        ;NO
      STA      0,FFLAG  ;YES, SET USER
                        ; FUNCTION FLAG
      JMP      START    ;WAIT FOR NEXT
                        ; CHARACTER
      LDA      1,FFLAG  ;GET FLAG
      MOV      1,1,SNR  ;FLAG SET?
      JMP      ECHO     ;NO, ECHO CHARACTER
      SUBC     1,1      ;YES, CLEAR FLAG
      STA      1,FFLAG  ;
      LDA      1,PR     ;GET PRINT CODE
      SUB#     0,1,SNR  ;CHAR PRINT?
      JMP      ECHO     ;YES, ECHO IT
      LDA      1,PF     ;NO, GET PRINT-FORM
                        ; CODE
      SUB#     0,1,SZR  ;IS CHAR PRINT-FORM?
      JMP      START    ;NO, DO NOT ECHO
                        ; USER FUNCTION CODE,
                        ; IF NOT USED
ECHO:  SKPBZ    TTO     ;TTO BUSY?
      JMP      -1        ;YES, WAIT FOR
                        ; NOT BUSY
      DOAS     0,TTO    ;ECHO CHARACTER
      JMP      START    ;SET UP FOR
                        ; NEXT CHARACTER
UFHDR: 036          ;USER FUNCTION HEADER
                        ; CODE
FFLAG: 0           ;FUNCTION FLAG
PR:    021         ;PRINT CODE
PF:    001         ;PRINT-FORM CODE

```

## Program Interrupt

The display terminals can be programmed using the program interrupt facility. This technique is useful in cases where a number of calculations can be performed in the time between receipt of characters in the input buffer.

The routine shown in example 6 reads a line, stores the characters in memory beginning at location 1000, and echoes them to the display using the interrupt priority structure. The user function header is used as a flag for user function codes. Both the header and the function code are stored in memory but are not echoed to the display. Cursor address sequences are handled in a manner similar to example 4 above. The program is terminated by a New Line code, a line overflow, or a cursor address sequence.

### Example 6

#### Read a line using interrupts

.LOC 0				STA	1,SAVE1	;
0		;PC IS STORED HERE WHEN AN		DIAC	0,TTI	;READ CHARACTER AND
		; INTERRUPT OCCURS				; CLEAR DONE
IHAND		;ADDRESS OF INTERRUPT HANDLER		LDA	1,CAHDR	;GET CURSOR ADDRESS
.LOC 300						; HEADER
CADD		;ADDRESS OF USER CURSOR		SUB#	0,1,SZR	;CURSOR ADDRESS HEADER?
		; ADDRESS SUBROUTINE		JMP	+5	;NO
CCOL:	0	;CURSOR COLUMN ADDRESS		STA	0,CFLAG	;SET CURSOR ADDRESS
		; IS STORED HERE				; FLAG
CROW:	0	;CURSOR ROW ADDRESS		LDA	1,CCNT	;START COUNTER
		; IS STORED HERE		STA	1,ACNT	
				JMP	OUT	;WAIT FOR NEXT CHARACTER
.LOC 400				LDA	1,CFLAG	;FLAG SET?
START:	LDA	1,BUFR	;SET UP BUFFER POINTER IN	MOV	1,1,SNR	
	STA	1,23	; AUTO-INCREMENT	JMP	+9	;NO
			; LOCATION 23	DSZ	ACNT	;YES. ROW ADDRESS?
	LDA	1,MAXLL	;GET MAXIMUM LINE LENGTH	JMP	+5	;NO. COLUMN ADDRESS
	STA	1,CNTR	;INITIALIZE LINE	STA	0,CROW	;STORE ROW ADDRESS
			; OVERFLOW COUNTER	LDA	1,FLAG	;CLEAR FLAG
SUBZL	1,1	;SET AC1 = 1		STA	1,CFLAG	
DOBS	1,CPU	;MASK OUT TTO AND		JMP	@CADD	;GO TO USER SUBROUTINE
		; TURN ON INTERRUPTS		STA	0,CCOL	;STORE COLUMN ADDRESS
-				JMP	OUT	;WAIT FOR NEXT CHARACTER
		;PROGRAM CAN DO USEFUL		STA	0,@23	;STORE CHARACTER IN BUFFER
		; THINGS WHILE LINE IS		LDA	1,UHDR	;GET USER FUNCTION HEADER
		; BEING READ		SUB#	0,1,SZR	;USER FUNCTION HEADER
				JMP	+3	;NO
HANG:	LDA	0,CNTR	;WHEN NEED FULL LINE TO	STA	0,FFLAG	;YES. SET USER FUNCTION FLAG
			CONTINUE,	JMP	OUT	;WAIT FOR NEXT CHARACTER
	MOV#	0,0,SZR	; HANG UP	LDA	1,FFLAG	;FLAG SET?
	JMP	+2	; HERE UNTIL READING	MOV	1,1,SNR	
			; IS ALL DONE	JMP	+4	;NO
				LDA	1,FLAG	;YES. CLEAR FLAG
				STA	1,FFLAG	
				JMP	OUT	;DO NOT ECHO USER
						; FUNCTION CODE
				SKPBZ	TTO	;MAKE SURE TTO NOT BUSY
				JMP	+1	
				DOAS	0,TTO	;ECHO CHARACTER
				LDA	1,NL	;IS IT A NEW LINE?
				SUB#	0,1,SZR	
				JMP	+4	;NO
				SUBC	0,0	;YES. CLEAR ACO WITHOUT
						; CHANGING CARRY
				STA	0,CNTR	;ZERO OUT CNTR TO
						; INDICATE LINE DONE
				JMP	+3	
				DSZ	CNTR	;IF NOT A NEW LINE,
						; DECREMENT CNTR
				JMP	OUT	;LINE NOT YET DONE,
						; GO DISMISS
				LDA	0,TTMSK	;LINE IS DONE
				MSKO	0	;MASK OUT TTI (AND TTO)
						; TO INHIBIT INPUT
				OUT:	LDA	0,SAVO
					LDA	1,SAV1
						;RESTORE ACCUMULATORS
				INTEN		;TURN INTERRUPTS BACK ON
				JMP	@0	;RETURN TO INTERRUPTED
						; PROGRAM
BUFR:	777		;BUFFER BEGINS AT	SAVO:	0	
			; LOCATION 1000	SAV1:	0	
MAXLL:	120		;MAXIMUM OF 80 CHARACTERS	CAHDR:	037	
			; PER LINE	UFHDR:	036	
CNTR:	0		;LINE OVERFLOW COUNTER	NL:	012	
				FLAG:	0	
IHAND:	SKPDN	TTI	;MAKE SURE TTI CAUSED	CFLAG:	0	
			; THE INTERRUPT	FFLAG:	0	
	HALT		;ERROR, SOME OTHER	CCNT:	2	
			; PERIPHERAL INTERRUPTED	ACNT:	0	
	STA	0,SAVO	;SAVE ACCUMULATORS THAT	TTMSK:	3	
			; WILL BE USED			

This page intentionally left blank.

# CHAPTER III

## OPERATOR CONTROLS AND PROCEDURES

### INTRODUCTION

The 6052/6053 display terminals serve as a communications link between the operator and the computer. The keyboard is the input device, transmitting the information entered by the operator to the computer. The display is the output device, receiving information from the computer and displaying the alphanumerics on the CRT screen, which is formatted as 24 lines, 80 characters per line.

To display information entered in the keyboard, an echo program in the computer must transmit the keyboard characters as they are received to the display. Since the terminal operates in full-duplex mode at data transmission rates up to 19.2K baud, information can be sent to and from the computer simultaneously at high speeds. When operating with suitable software, the time delay between keyboard entry and display is almost imperceptible to the operator and the terminal appears to operate in a manner similar to an office typewriter.

### DISPLAY

ASCII character codes received by the display are decoded and classified into three categories:

- alphanumeric, printing characters which can be displayed on the CRT screen;
- control characters which represent commands to the display; and
- control characters other than display commands.

When the display receives an alphanumeric, printing character, that character is stored in memory and is displayed on the CRT screen. The location in memory where the character is stored, and subsequently the location on the screen where the character is displayed, is defined by the position of a pointer, called the cursor. The position of the cursor on the screen is represented by a blinking underscore. After a character is entered in memory and displayed, the cursor moves one character position to the right. If the cursor is at the end of a line, it moves to the beginning of the next line. If the cursor is at the end of the bottom line on the screen, it moves to the beginning of the top line, unless the display is roll enabled. When the display is roll enabled, the screen rolls up one line each time the cursor overflows the bottom line or a New Line code is decoded when the cursor is on the bottom line. In this case, the cursor moves to the first character position on the bottom line, the bottom line becomes blank, and the information previously displayed on the top line is lost.

When the display receives a control character which represents a command to the display, the command is carried out. These commands manipulate the display's memory by moving the cursor, adding blink, dim (model 6053) and/or underscore (model 6053) attributes to designated characters, and clearing areas of the screen. A READ CURSOR ADDRESS command causes the current cursor column and row address to be transmitted to the computer.

When the display receives a control character which is not a display command, that character is ignored.

Appendix A lists the characters displayed by both the 6052 and 6053 terminals, the display commands and control characters which are not display commands.

## KEYBOARD

The keyboard consoles of both terminals contain four keypads: main, numeric, screen management and user function.

### Main Keypad

The 6052 terminal has a teletypewriter-style, 52-key, main keypad. This keypad transmits the 64 upper case graphic members of the ASCII character set. The 6053 terminal has a typewriter-style, 56-key, main keypad. This keypad transmits the 96 upper/ lower case graphic members of the ASCII character set. Both main keypads transmit 30 ASCII control characters.

With the exception of the CTRL, SHIFT, REPT (model 6052 only) and BREAK (model 6052 only) keys, each key on the main keypad generates a single

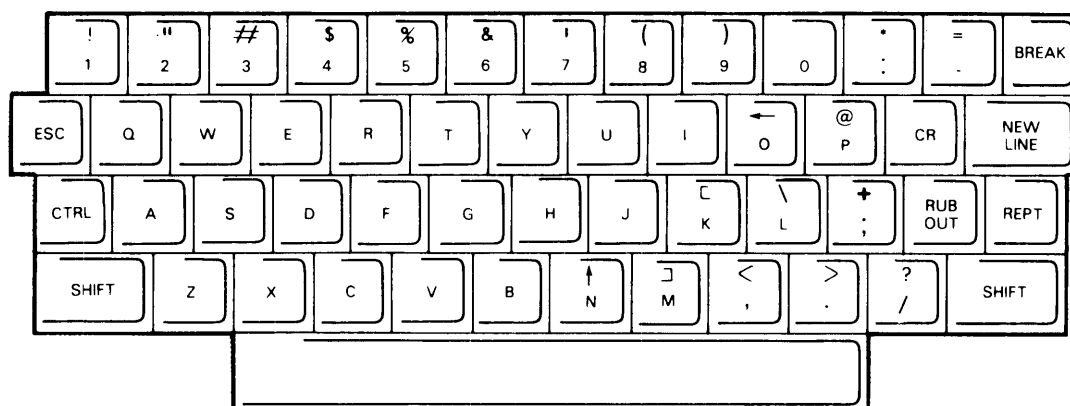
code when struck. On the 6053 keyboard, the REPT and BREAK keys reside on the screen management keypad; however, the function of these keys is identical on both models.

The CTRL and SHIFT keys do not generate codes when pressed alone; rather, they are used separately and in conjunction to alter the code produced by most code-generating keys.

The REPT key does not generate a code when pressed alone; however, when this key is pressed in conjunction with a code-generating key, the resulting code is repeated until one of the keys is released.

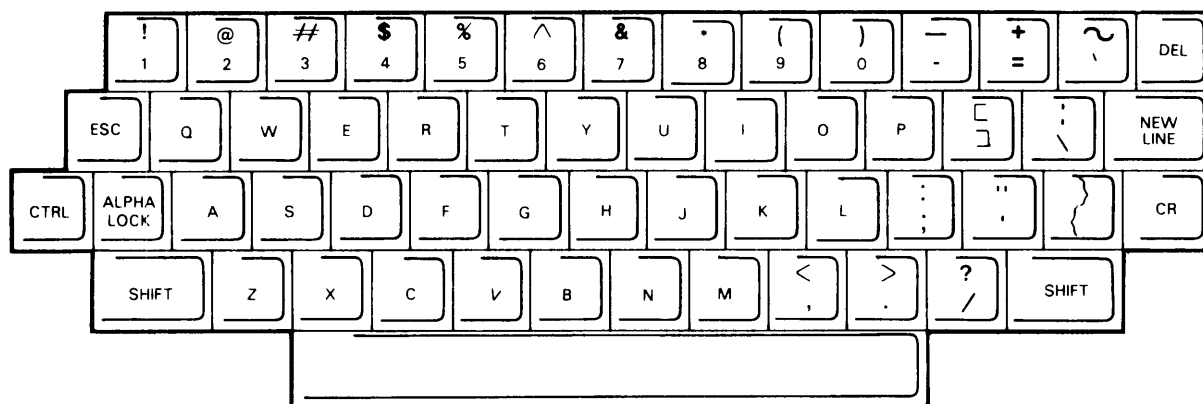
The BREAK key, when pressed, places the transmission line in a spacing state. While this key is pressed other key-entered codes are not transmitted. When the display terminal is equipped with a printer option, pressing this key aborts a print operation.

6052 TERMINAL MAIN KEYBOARD



DG-03189

6053 TERMINAL MAIN KEYBOARD

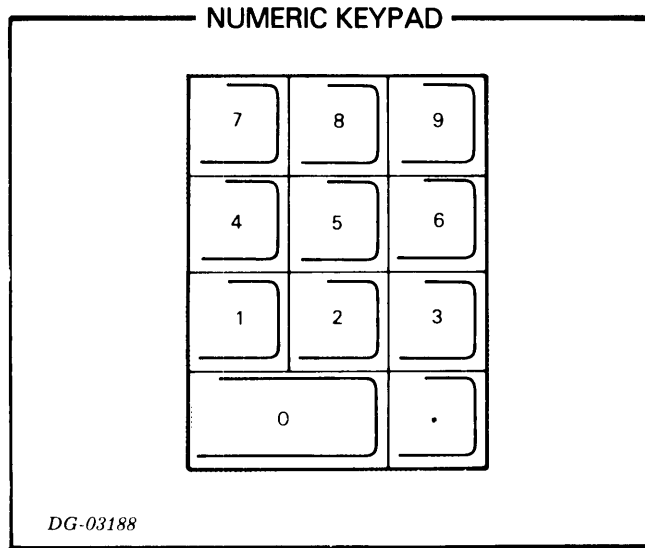


DG-03191



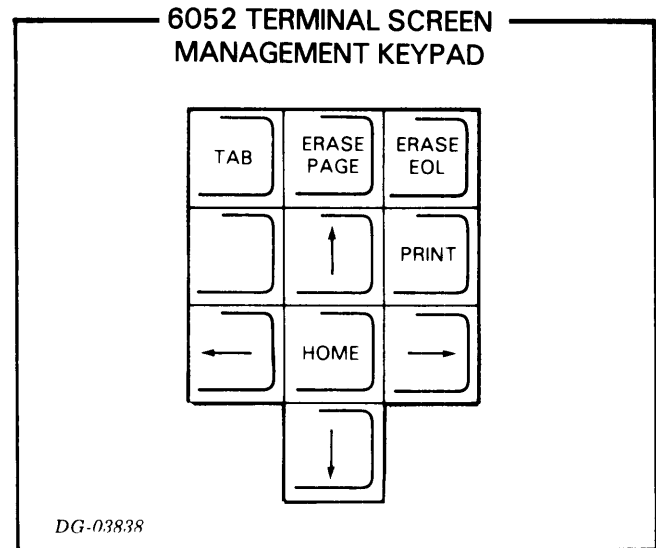
## Numeric Keypad

The numeric keypads of both terminals are the same. Each contains 11 keys which generate the ASCII codes for the numerals 0 through 9 plus the period character. This keypad is arranged like a ten-key adding machine for ease in entering numeric data.



The PRINT key is also a user function key when the terminal is not equipped with a printer option. When it is, pressing this key places the terminal in print mode and the keyboard is disabled until the print operation is completed.

The use of the TAB key must be defined by the program; otherwise, the code generated by this key is ignored when it is received by the display.

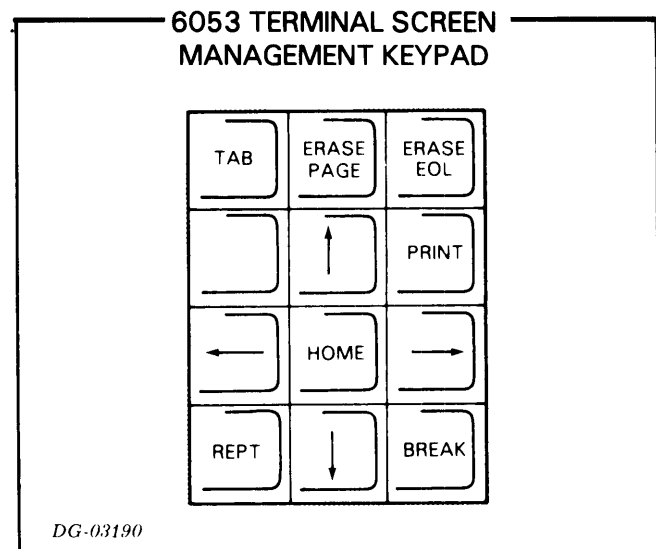


## Screen Management Keypad

The screen management keypads of both terminals are the same, except that the REPT and BREAK keys are added on the 6053 terminal.

The four keys marked with arrows move the cursor in the direction indicated. The HOME key moves the cursor to the leftmost character position on the top line of the screen. The ERASE PAGE key clears the screen and moves the cursor to the home position. The ERASE EOL key clears the characters on the screen from the position of the cursor to the end of the line on which the cursor resides.

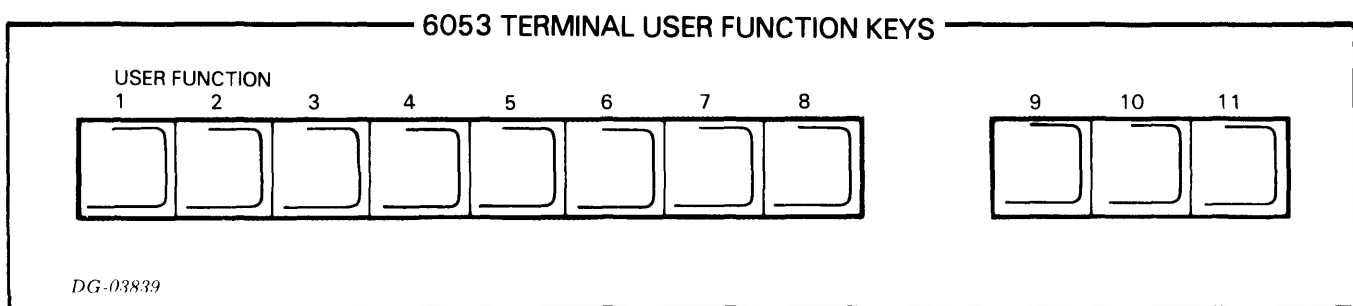
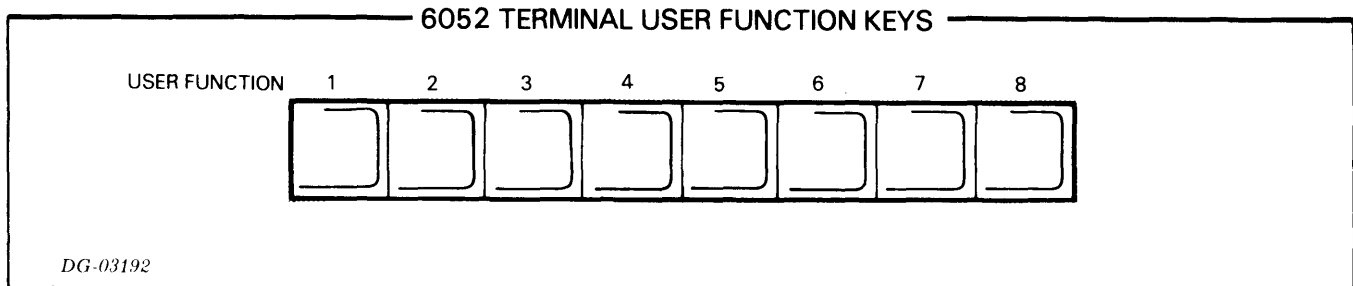
The unlabeled key is an additional user function key. See the User Function Keypad.



## User Function Keypad

The user function keypads of the 6052 and 6053 terminals consist of 8 and 11 keys, respectively. Each key, when pressed, generates a two-code sequence. The first code of the sequence is the ASCII control

character RS (Record Separator, 036<sub>8</sub>). When this code is received by the computer, it notifies the executing program that the next character received is a user-defined control character. The codes generated by these keys are listed in Appendix B.



## DISPLAY EXTERNAL CONTROLS

### Front Panel

#### Power ON/OFF and Screen Brightness

The power switch and CRT screen brightness control are combined in a single control knob. Pulling this knob out turns the display terminal power on; pushing it in turns the power off. Rotating the knob clockwise makes the screen image brighter; rotating it counter-clockwise makes it darker.

### Rear Panel

#### Data Rate

The data transmission rate is selected by moving the DATA RATE rotary switch to one of its ten positions. The baud rate selected by each position is marked on the panel.

#### Parity

The parity computation performed by the display terminal is selected by moving the PARITY slide switch to one of its three positions: EVEN, ODD or in the center, unmarked position for mark parity.

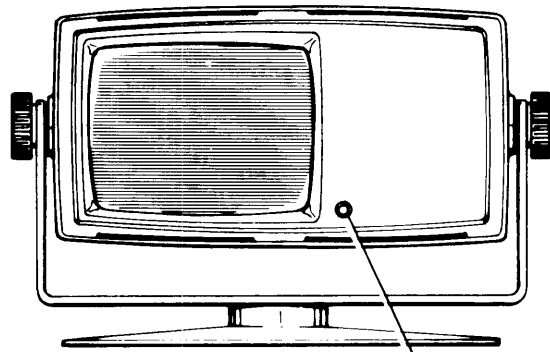
#### Mode

The operating mode of the terminal is selected by moving the MODE slide switch to one of its two positions: LINE or LOCAL. In LINE mode, information entered in the keyboard is transmitted to the computer and information received from the computer is sent to the display. In LOCAL mode, information entered in the keyboard is sent directly to the display and the display does not receive information from the computer.

#### Screen

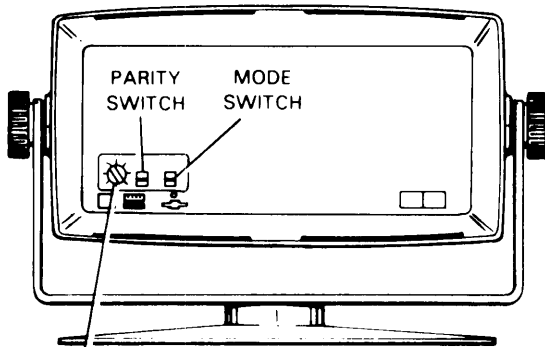
The operator can position the display screen by moving it in both a horizontal and vertical direction. This feature can reduce operator fatigue.

## DISPLAY EXTERNAL CONTROLS



ON/OFF  
AND BRIGHTNESS

FRONT VIEW



DATA RATE  
ROTARY SWITCH

REAR VIEW

DG-03194

## OPERATING NOTE

**CAUTION** - *Do not cover the ventilation openings in the top cover of the display unit as operating failure due to overheating may occur.*

## CLEANING

Wipe the external surfaces of the keyboard console and the display unit with a soft, dry, lint-free cloth. If necessary, clean the covers with a soft cloth dampened with a mild detergent solution, rinse with a soft cloth dampened with water, and buff dry.

Clean the screen by wiping the surface with a soft, lint-free cloth dampened with either water or a window-cleaning solution, such as Windex.

## PREVENTIVE MAINTENANCE

The display diagnostic tape and listing, Nos. 095-000477 and 096-000477, respectively, are included in the documentation package. This test can be run on an "as needed" basis.

This page intentionally left blank.

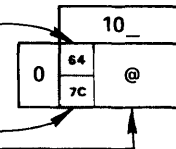
# APPENDIX A

## ASCII CHARACTER CODES

### WITH TERMINAL CONTROL FUNCTIONS

LEGEND:

CHARACTER CODE IN DECIMAL  
EBCDIC EQUIVALENT HEXADECIMAL CODE



↑ means CONTROL

CHARACTER

OCTAL	00_	CONTROL FUNCTION	01_	CONTROL FUNCTION	02_	CONTROL FUNCTION	03_	CONTROL FUNCTION	04_	05_
0	0 00	NUL NULL	8 16	BS (BACK-SPACE) HOME	16 10	DLE ↑P WRITE CURSOR HEADER	24 18	CAN ↑X CURSOR RIGHT	32 40	SPACE 4D (
1	1 01	SOH ↑A PRINT FORM	9 05	HT (TAB) HORIZONTAL TAB	17 11	DC1 ↑Q PRINT PAGE	25 19	EM ↑Y CURSOR LEFT	33 5A	! 5D )
2	2 02	STX ↑B START OF TEXT	10 15	NL (NEW LINE) NEW LINE	18 12	DC2 ↑R ROLL ON	26 3F	SUB ↑Z CURSOR DOWN	34 7F	" (QUOTE) 7B 7C 7D 7E 7F 80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F 90 91 92 93 94 95 96 97 98 99 A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
3	3 03	ETX ↑C ENABLE BLINK	11 0B	VT (VERT. TAB) ERASE LINE	19 13	DC3 ↑S ROLL OFF	27 27	ESC (ESCAPE) ESCAPE	35 7B	# 4E +
4	4 37	EOT ↑D INHIBIT BLINK	12 06	FF (FORM FEED) ERASE PAGE	20 3C	DC4 ↑T UNDERSCORE ON	28 1C	FS ↑\n DIM ON	36 5B	\$ 6B (COMMA)
5	5 2D	ENQ ↑E CURSOR ADDRESS READ	13 0D	RT (RETURN) RETURN	21 3D	NAK ↑U UNDERSCORE OFF	29 1D	CS ↑] DIM OFF	37 6C	% 60 -
6	6 2E	ACK ↑F PRINT DONE	14 0E	SO ↑N BLINK ON	22 32	SYN ↑V SYNCHRONOUS IDLE	30 1E	RS ↑↑ FUNCTION KEY HEADER	38 50	& 4B (PERIOD)
7	7 2F	BEL ↑G BELL	15 0F	SI ↑O BLINK OFF	23 26	ETB ↑W CURSOR UP	31 1F	US ↑← READ CURSOR HEADER	39 7D	' (APOS) 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F 80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F 90 91 92 93 94 95 96 97 98 99 A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF

OCTAL	06_	07_	10_	11_	12_	13_	14_	15_	16_	17_
0	48 F0	0 F8	56 7C	8 7C	64 C8	@ C8	72 D7	H D7	80 E7	P E7
1	49 F1	1 F9	57 C1	9 C1	65 C9	A C9	73 D8	I D8	81 E8	Q E8
2	50 F2	2 FA	58 C2	: C2	66 D1	B D1	74 D9	J D9	82 E9	R E9
3	51 F3	3 FB	59 C3	; C3	67 D2	C D2	75 DA	K DA	83 EA	S EA
4	52 F4	4 FC	60 C4	< C4	68 D3	D D3	76 DB	L DB	84 EB	T EB
5	53 F5	5 FD	61 C5	= C5	69 D4	E D4	77 DC	M DC	85 EB	U EB
6	54 F6	6 FE	62 C6	> C6	70 D5	F D5	78 DD	N DD	86 EC	V EC
7	55 F7	7 FF	63 C7	? C7	71 D6	G D6	79 DE	O DE	87 ED	W ED

CHARACTER CODE IN OCTAL AT TOP AND LEFT OF CHARTS.

## ASCII Character Codes Plotted by the 6052 Terminal

NAME	SYMBOL DISPLAY	7 BIT OCTAL CODE	KEYPAD	
			MAIN KEYPAD	NUMERIC KEYPAD
Space	Non-printing	040	Space Bar	-
Exclamation Mark	!	041	SH 1	-
Quotation Marks	"	042	SH 2	-
Number Sign	#	043	SH 3	-
Dollar Sign	\$	044	SH 4	-
Percent	%	045	SH 5	-
Ampersand	&	046	SH 6	-
Apostrophe	'	047	SH 7	-
Opening Parenthesis	(	050	SH 8	-
Closing Parenthesis	)	051	SH 9	-
Asterisk	*	052	SH :	-
Plus	+	053	SH ;	-
Comma	,	054	,	-
Hyphen (Minus)	-	055	-	-
Period (Decimal Point)	.	056	.	-
Slant	/	057	/	-
Zero	0	060	0	0
One	1	061	1	1
Two	2	062	2	2
Three	3	063	3	3
Four	4	064	4	4
Five	5	065	5	5
Six	6	066	6	6
Seven	7	067	7	7
Eight	8	070	8	8
Nine	9	071	9	9
Colon	:	072	:	-
Semicolon	;	073	;	-
Less Than	<	074	SH ,	-
Equals	=	075	SH -	-
Greater Than	>	076	SH .	-
Question Mark	?	077	SH /	-
Commercial At	@	100	SH P	-
A	A	101	A, SH A	-
B	B	102	B, SH B	-
C	C	103	C, SH C	-
D	D	104	D, SH D	-
E	E	105	E, SH E	-
F	F	106	F, SH F	-
G	G	107	G, SH G	-
H	H	110	H, SH H	-
I	I	111	I, SH I	-
J	J	112	J, SH J	-
K	K	113	K	-
L	L	114	L	-
M	M	115	M	-
N	N	116	N	-
O	O	117	O	-
P	P	120	P	-
Q	Q	121	Q, SH Q	-
R	R	122	R, SH R	-
S	S	123	S, SH S	-
T	T	124	T, SH T	-
U	U	125	U, SH U	-
V	V	126	V, SH V	-
W	W	127	W, SH W	-
X	X	130	X, SH X	-
Y	Y	131	Y, SH Y	-
Z	Z	132	Z, SH Z	-
Opening Bracket	[	133	SH K	-
Reverse Slant	\	134	SH L	-
Closing Bracket	]	135	SH M	-
Circumflex	^	136	SH N	-
Underline	_	137	SH O	-

## ASCII Character Codes Plotted by the 6053 Terminal

NAME	SYMBOL DISPLAY	7 BIT OCTAL CODE	KEYPAD	
			MAIN KEYPAD	NUMERIC KEYPAD
Space	Non-printing	040	Space Bar	-
Exclamation Mark	!	041	SH 1	-
Quotation Marks	"	042	SH '	-
Number Sign	#	043	SH 3	-
Dollar Sign	\$	044	SH 4	-
Percent	%	045	SH 5	-
Ampersand	&	046	SH 7	-
Apostrophe	'	047	'	-
Opening Parenthesis	(	050	SH 9	-
Closing Parenthesis	)	051	SH 0	-
Asterisk	*	052	SH 8	-
Plus	+	053	SH =	-
Comma	,	054	,	-
Hyphen (Minus)	-	055	-	-
Period (Decimal Point)	.	056	.	.
Slant	/	057	/	-
Zero	0	060	0	0
One	1	061	1	1
Two	2	062	2	2
Three	3	063	3	3
Four	4	064	4	4
Five	5	065	5	5
Six	6	066	6	6
Seven	7	067	7	7
Eight	8	070	8	8
Nine	9	071	9	9
Colon	:	072	SH :	-
Semicolon	;	073	;	-
Less Than	<	074	SH ,	-
Equals	=	075	=	-
Greater Than	>	076	SH .	-
Question Mark	?	077	SH /	-
Commercial At	@	100	SH 2	-
A	A	101	SH A	-
B	B	102	SH B	-
C	C	103	SH C	-
D	D	104	SH D	-
E	E	105	SH E	-
F	F	106	SH F	-
G	G	107	SH G	-
H	H	110	SH H	-
I	I	111	SH I	-
J	J	112	SH J	-
K	K	113	SH K	-
L	L	114	SH L	-
M	M	115	SH M	-
N	N	116	SH N	-
O	O	117	SH O	-
P	P	120	SH P	-
Q	Q	121	SH Q	-
R	R	122	SH R	-
S	S	123	SH S	-
T	T	124	SH T	-
U	U	125	SH U	-
V	V	126	SH V	-
W	W	127	SH W	-
X	X	130	SH X	-
Y	Y	131	SH Y	-
Z	Z	132	SH Z	-

### ASCII Character Codes Plotted by the 6053 Terminal (Continued)

NAME	SYMBOL DISPLAY	7 BIT OCTAL CODE	KEYPAD	
			MAIN KEYPAD	NUMERIC KEYPAD
Opening Bracket	[	133	[	-
Reverse Slant	\	134	\	-
Closing Bracket	]	135	]	-
Circumflex	^	136	SH 6	-
Underline	_	137	SH -	-
Grave Accent	`	140	`	-
a	a	141	A	-
b	b	142	B	-
c	c	143	C	-
d	d	144	D	-
e	e	145	E	-
f	f	146	F	-
g	g	147	G	-
h	h	150	H	-
i	i	151	I	-
j	j	152	J	-
k	k	153	K	-
l	l	154	L	-
m	m	155	M	-
n	n	156	N	-
o	o	157	O	-
p	p	160	P	-
q	q	161	Q	-
r	r	162	R	-
s	s	163	S	-
t	t	164	T	-
u	u	165	U	-
v	v	166	V	-
w	w	167	W	-
x	x	170	X	-
y	y	171	Y	-
z	z	172	Z	-
Open Brace	{	173	{	-
Vertical Line		174	SH \	-
Closing Brace	}	175	SH {	-
Tilde	~	176	SH `	-

DG-03824



## DISPLAY COMMANDS

NAME	OCTAL CODE	MAIN KEYPAD		SCREEN MANAGEMENT KEYPAD	FUNCTION
		MODEL 6052	MODEL 6053		
ENABLE BLINK	003	CTRL-C or CTRL-SHIFT-C	CTRL-C or CTRL-SHIFT-C	—	Blink characters whose blink bit is set to 1.
DISABLE BLINK	004	CTRL-D or CTRL-SHIFT-D	CTRL-D or CTRL-SHIFT-D	—	Disable character blinking
READ CURSOR ADDRESS	005	CTRL-E or CTRL-SHIFT-E	CTRL-E or CTRL-SHIFT-E	—	Cursor address header code, 037 <sub>8</sub> , followed by 2 codes containing 7-bit column and 5-bit row address, respectively, are sent to computer.
BELL	007	CTRL-G or CTRL-SHIFT-G	CTRL-G or CTRL-SHIFT-G	—	Short tone produced by speaker in display.
HOME	010	CTRL-H or CTRL-SHIFT-H	CTRL-H or CTRL-SHIFT-H	HOME	Cursor moves to first (leftmost) character position on top line of screen.
NEW LINE	012	NEW LINE, CTRL-NEW LINE, CTRL-SHIFT-NEW LINE, CTRL-J, or CTRL-SHIFT-J	NEW LINE, CTRL-NEW LINE, CTRL-SHIFT-NEW LINE, CTRL-J, or CTRL-SHIFT-J	—	Cursor moves to first character position on next line.
ERASE TO END OF LINE	013	CTRL K	CTRL-K or CTRL-SHIFT-K	EOL	Erase characters from and including the position of the cursor to the end of the line.
ERASE PAGE	014	CTRL-L	CTRL-L CTRL-SHIFT-L	EOP	Erase all characters on screen.
CARRIAGE RETURN	015	CR, CTRL-M, CTRL-CR, SHIFT-CR, or CTRL-SHIFT-CR	CR, CTRL-CR, SHIFT-CR, CTRL-SHIFT-CR, CTRL-M, or CTRL-SHIFT-M	—	Cursor moves to first character position on the line it is on.
START BLINK	016	CTRL-N or CTRL-SHIFT-N	CTRL-N or CTRL-SHIFT-N	—	Set blink bit to 1 in succeeding printing characters received.
END BLINK	017	CTRL-O or CTRL-SHIFT-O	CTRL-O or CTRL-SHIFT-O	—	Set Blink bit to 0 in succeeding printing characters received.
WRITE CURSOR ADDRESS	020	CTRL-P	CTRL-P or CTRL-SHIFT-P	—	Next 2 characters received are used as cursor's 7-bit column and 5-bit row address, respectively, and the cursor is moved accordingly.
ROLL ENABLE	022	CTRL-R or CTRL-SHIFT-R	CTRL-R or CTRL-SHIFT-R	—	Screen rolls up one line each time the cursor overflows the bottom line or a New Line code is received when the cursor is on the bottom line. Cursor moves to first character position on bottom line.
ROLL DISABLE	023	CTRL-S or CTRL-SHIFT-S	CTRL-S or CTRL-SHIFT-S	—	When New Line code is received and the cursor is on the bottom line, the cursor moves to Home position.
START UNDERSCORE	024	CTRL-T or CTRL-SHIFT-T	CTRL-T or CTRL-SHIFT-T	—	Set Underscore bit to 1. Each succeeding printing character received is underlined on screen.(1)
END UNDERSCORE	025	CTRL-U or CTRL-SHIFT-U	CTRL-U or CTRL-SHIFT-U	—	Set Underscore bit to 0. Each succeeding printing character received is not underlined.(1)

## DISPLAY COMMANDS, CONT.

NAME	OCTAL CODE	MAIN KEYPAD		SCREEN MANAGEMENT KEYPAD	FUNCTION
		MODEL 6052	MODEL 6053		
CURSOR UP	027	CTRL-W or CTRL-SHIFT-W	CTRL-W or CTRL-SHIFT-W	↑	Cursor moves up one line while remaining in same column (character) position.
CURSOR RIGHT	030	CTRL-X or CTRL-SHIFT-X	CTRL-X or CTRL-SHIFT-X	→	Cursor moves one character position to the right.
CURSOR LEFT	031	CTRL-Y or CTRL-SHIFT-Y	CTRL-Y or CTRL-SHIFT-Y	←	Cursor moves one character position to the left.
CURSOR DOWN	032	CTRL-Z or CTRL-SHIFT-Z	CTRL-Z or CTRL-SHIFT-Z	↓	Cursor moves down one line while remaining in same character position
START DIM	034	CTRL-SHIFT-L	CTRL- \ or CTRL-SHIFT- \	—	Set Dim bit to 1. Each succeeding printing character received is dimmed on the screen. (1)
END DIM	035	CTRL-SHIFT-M	CTRL-SHIFT-]		Set Dim bit to 0. Each succeeding printing character received is displayed at full intensity on the screen. (1)

(1) Model 6053 only Model 6052 display will not respond to these commands.

DG-03826

## PROTOCOL CHARACTERS

NAME	STANDARD MNEMONICS	OCTAL CODE	MAIN KEYBOARD		SCREEN MGNT KEYBOARD
			MODEL 6052	MODEL 6053	
NULL	NUL	000	CTRL-SHIFT-P	CTRL-SHIFT-2	—
START OF HEADING	SOH	001	CTRL-A CTRL-SHIFT-A	CTRL-A CTRL-SHIFT-A	—
START OF TEXT	STX	002	CTRL-B CTRL-SHIFT-B	CTRL-B CTRL-SHIFT-B	—
ACKNOWLEDGE	ACK	006	CTRL-F CTRL-SHIFT-F	CTRL-F CTRL-SHIFT-F	—
HORIZONTAL TAB	HT	011	CTRL-I CTRL-SHIFT-I	CTRL-I CTRL-SHIFT-I	TAB
DEVICE CODE 1	DC1	021	CTRL-Q CTRL-SHIFT-Q	CTRL-Q CTRL-SHIFT-Q	—
SYNCHRONOUS IDLE	SYN	026	CTRL-V CTRL-SHIFT-V	CTRL-V CTRL-SHIFT-V	—
ESCAPE	ESC	033	ESC CTRL-SHIFT-K	ESC CTRL-L	—

NOTE: The protocol characters, Record Separator (036) and Unit Separator (037) cannot be generated by the keyboard. These codes are generated by the display hardware and are used as user function and cursor address header codes, respectively.

DG-03827

This page intentionally left blank.

# APPENDIX B

## CODE SEQUENCES

### GENERATED BY

## THE USER FUNCTION KEYPAD

SEQUENCE OF CODES	KEYS
036, 041	CTRL-SHIFT-[user function 1]
036, 042	CTRL-SHIFT-[user function 2]
036, 043	CTRL-SHIFT-[user function 3]
036, 044	CTRL-SHIFT-[user function 4]
036, 045	CTRL-SHIFT-[user function 5]
036, 046	CTRL-SHIFT-[user function 6]
036, 047	CTRL-SHIFT-[user function 7]
036, 050	CTRL-SHIFT-[user function 8]
036, 051	CTRL-SHIFT-[user function 9]*
036, 052	CTRL-SHIFT-[user function 10]*
036, 053	CTRL-SHIFT-[user function 11]*
036, 061	CTRL-[user function 1]
036, 062	CTRL-[user function 2]
036, 063	CTRL-[user function 3]
036, 064	CTRL-[user function 4]
036, 065	CTRL-[user function 5]
036, 066	CTRL-[user function 6]
036, 067	CTRL-[user function 7]
036, 070	CTRL-[user function 8]
036, 071	CTRL-[user function 9]*
036, 072	CTRL-[user function 10]*
036, 073	CTRL-[user function 11]*
036, 141	SHIFT-[user function 1]
036, 142	SHIFT-[user function 2]
036, 143	SHIFT-[user function 3]
036, 144	SHIFT-[user function 4]
036, 145	SHIFT-[user function 5]
036, 146	SHIFT-[user function 6]
036, 147	SHIFT-[user function 7]
036, 150	SHIFT-[user function 8]
036, 151	SHIFT-[user function 9]*
036, 152	SHIFT-[user function 10]*
036, 153	SHIFT-[user function 11]*
036, 161	[user function 1]
036, 162	[user function 2]
036, 163	[user function 3]
036, 164	[user function 4]
036, 165	[user function 5]
036, 166	[user function 6]
036, 167	[user function 7]
036, 170	[user function 8]
036, 171	[user function 9]*
036, 172	[user function 10]*
036, 173	[user function 11]*

Additional user-defined codes generated by Screen Management Keypad

036, 001	SHIFT-PRINT**
036, 002	SHIFT-unmarked key
036, 021	PRINT**
036, 022	unmarked key

\*Denotes 6053 Display Terminal only.

\*\*When the display is not equipped with a Printer option.

DG-03828

This page intentionally left blank.

CUT ALONG DOTTED LINE

**DOCUMENT TITLE:** .....

*Your comments, accompanied by answers to the following questions, help us improve the quality and usefulness of our publications. If your answer to a question is "no" or requires qualification, please explain.*

### How did you use this publication?

- ( ) As an introduction to the subject.
- ( ) As an aid for advanced knowledge.
- ( ) For information about operating procedures.
- ( ) To instruct in a class.
- ( ) As a student in a class.
- ( ) As a reference manual.
- ( ) Other.....

**Did you find the material:**

- Useful.....YES ( ) NO ( )
- Complete.....YES ( ) NO ( )
- Accurate.....YES ( ) NO ( )
- Well organized.....YES ( ) NO ( )
- Well written.....YES ( ) NO ( )
- Well illustrated.....YES ( ) NO ( )
- Well indexed.....YES ( ) NO ( )
- Easy to read.....YES ( ) NO ( )
- Easy to understand.....YES ( ) NO ( )

*We would appreciate any other comments; please label each comment as an addition, deletion, change, or error and reference page numbers where applicable.*

## COMMENTS

[illegible]

**From:**

NAME..... TITLE.....  
FIRM..... DIV.....  
ADDRESS.....  
CITY..... STATE..... ZIP.....  
TELEPHONE..... DATE.....

**Data General Corporation**  
ENGINEERING PUBLICATIONS  
COMMENT FORM  
DG-00935

FOLD DOWN

FIRST

FOLD DOWN

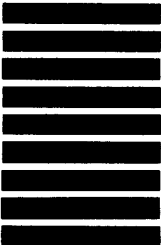
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

**BUSINESS REPLY MAIL**

Postage will be paid by:

**DataGeneral**  
Southboro, Massachusetts 01772

FIRST CLASS  
PERMIT NO 26  
SOUTHBORO  
MASS 01772



**ATTENTION: Engineering Publications**

FOLD UP

SECOND

FOLD UP

STAPLE