



GDS II

GRAPHIC DESIGN SYSTEM USER'S OPERATING MANUAL

FIRST EDITION

Data in this manual, because of the program (software), is subject to change during future development. Descriptions for all of the available commands are not yet included in Section 4.

NOTICE

The information contained herein is proprietary in nature and is not to be disclosed, used, or duplicated, in whole or in part, for any purpose whatsoever without prior written permission of CALMA, 527 Lakeside Drive, Sunnyvale, California 94086. Receipt of this document shall be deemed to be an acceptance of the conditions specified herein.

NOVEMBER 1978

TABLE OF CONTENTS

| SECTION | | PAGE |
|---------|---|------|
| 1 | GENERAL DESCRIPTION | 1-1 |
| | 1.1 System Configuration | 1-1 |
| | 1.2 CPU | 1-1 |
| | 1.3 Disk Drive | 1-1 |
| | 1.4 Magnetic Tape Drive | 1-1 |
| | 1.5 DECWRITER II | 1-3 |
| | 1.6 Design Station | 1-3 |
| | 1.6.1 Digitizer Table | 1-3 |
| | 1.6.2 Digitizer Talos Table | 1-3 |
| | 1.6.3 Digitizer Tablet | 1-5 |
| | 1.6.4 GDS II Keyboard | 1-5 |
| | 1.6.5 Alphanumeric QTY Display | 1-5 |
| 2 | OPTIONS | 2-1 |
| | 2.1 Hardware Options | 2-1 |
| | 2.1.1 Plotters | 2-1 |
| | 2.1.2 CRT Hard Copy | 2-1 |
| | 2.1.3 Optical Pattern Generator | 2-1 |
| | 2.1.4 E-Beam Pattern Generator | 2-1 |
| | 2.1.5 Line Printer | 2-1 |
| 3 | SYSTEM OPERATION | 3-1 |
| | 3.1 Introduction | 3-1 |
| | 3.2 Controls and Indicators | 3-2 |
| | 3.2.1 CPU Controls and Indicators | 3-2 |
| | 3.2.2 Disk Drive Controls and Indicators | 3-6 |
| | 3.2.2.1 Century 114 Disk Drive | 3-6 |
| | 3.2.2.2 Control Data 9762 and 9766 Disk Drives | 3-8 |
| | 3.2.3 Tape Drive Controls and Indicators | 3-10 |
| | 3.2.3.1 800 and 1600 BPI Wangco Tape Drive | 3-10 |
| | 3.2.3.2 800/1600 BPI Cipher Tape Drive | 3-13 |
| | 3.2.4 System Console - DECWRITER II Controls and Indicators | 3-15 |
| | 3.2.5 Design Station Alphanumeric Keyboard | 3-18 |
| | 3.2.6 Design Station Digitizer Tables | 3-21 |
| | 3.2.6.1 48" x 60" Digitizer Table | 3-21 |
| | 3.2.6.2 44" x 60" Talos Table | 3-23 |
| | 3.2.7 Digitizer Styluses | 3-25 |
| | 3.2.7.1 7-Key Programmable Stylus | 3-25 |
| | 3.2.7.2 12-Key Programmable Free Stylus or "Puck" | 3-27 |

TABLE OF CONTENTS Cont.

| SECTION | PAGE |
|---------|---|
| 3.2.8 | Design Station Digitizer Tablet and Pen Stylus 3-28 |
| 3.2.9 | Design Station Display Screens 3-29 |
| 3.2.9.1 | QTY Screen 3-29 |
| 3.2.9.2 | Interactive CRT 3-29 |
| 3.2.10 | X-Y Coordinate Displays 3-30 |
| 3.3 | GDS II System Power-On and Power-Off Procedures 3-30 |
| 3.3.1 | Applying Power to the System 3-30 |
| 3.3.2 | Loading and Unloading the Century Disk Pack 3-31 |
| 3.3.3 | Loading and Unloading a CDC Disk Pack 3-32 |
| 3.3.4 | Loading and Unloading the Wangco Magnetic Tape Drive 3-33 |
| 3.3.5 | Loading and Unloading the Cipher Magnetic Tape Drive 3-35 |
| 3.3.6 | System Bootstrap Procedure 3-36 |
| 3.3.7 | Disk Pack Initialization and Formatting 3-36 |
| 3.3.8 | Creating and Loading a BACKUP Tape 3-41 |
| 3.3.9 | Reformatting a Disk Pack 3-44 |
| 3.3.10 | RDOS Initialization and Shut Down 3-44 |
| 3.3.11 | GDS II Start Up and Shut Down 3-45 |
| 3.3.12 | System Generation Dialog 3-46 |
| | |
| 4 | GDS II COMMANDS |
| 4.1 | General 3-46 |
| 4.2 | Command Syntax 3-46 |
| 4.2.1 | Command Names and Arguments 3-46 |
| 4.2.2 | Coordinate Entry (CE) 3-46 |
| 4.2.3 | Command Prompt 3-46 |
| 4.2.4 | Input Corrections 3-46 |
| 4.2.5 | Console Abort 3-46 |
| 4.2.6 | Station Release 3-46 |
| 4.2.7 | Calculator Mode 3-46 |
| 4.3 | Commands 3-46 |
| 4.3.1 | Absolute Versus Relative 3-46 |
| 4.3.2 | Command List by Function 3-46 |
| 4.3.3 | Command Descriptions 3-46 |

FIGURES

| Figure # | Titles | Page # |
|----------|--|--------|
| 1-1 | Eclipse S/230 CPU | 1-1 |
| 1-2 | Century and CDC Disk Drives | 1-2 |
| 1-3 | Wangco and Cipher Tape Drives | 1-2 |
| 1-4 | DECWRITER II | 1-3 |
| 1-5 | Design Station | 1-3 |
| 1-6 | Digitizer Table | 1-4 |
| 1-7 | Digitizer Talos Table | 1-4 |
| 1-8 | Digitizer Tablet | 1-5 |
| 1-9 | Special Function Keyboard | 1-5 |
| | | |
| 3-1 | Eclipse S/230 CPU Power Switches | 3-1 |
| 3-2 | Eclipse S/230 Control Panel | 3-2 |
| 3-3 | Century 114 Disk Drive Controls and Indicators | 3-6 |
| 3-4 | Control Data 9762 and 9766 Disk Drive Controls and Indicators | 3-8 |
| 3-5A | Wangco Tape Drive Controls and Indicators | 3-10 |
| 3-5B | Cipher Tape Drive Controls and Indicators | 3-13 |
| 3-6 | DECWRITER II Controls and Indicators | 3-15 |
| 3-7 | Design Station Alphanumeric Keyboard | 3-18 |
| 3-8 | Digitizer Table Controls and Indicators | 3-21 |
| 3-9 | Talos Table Controls and Indicators | 3-23 |
| 3-10 | 7-Key Stylus Controls and Indicators | 3-25 |
| 3-11 | 12-Key "Puck" Controls and Indicators | 3-27 |
| 3-12 | Digitizer Table Controls and Indicators | 3-28 |
| 3-13 | CRT Controls | 3-29 |
| 3-14 | Disk Pack | 3-31 |
| 3-15A | Removing Tape Seal from Reel | 3-33 |
| 3-15B | Location of Write Enable Ring | 3-34 |
| 3-16 | Loading the Supply Reel | 3-35 |
| 3-17 | Wangco Tape Loading Diagram | 3-35 |
| 3-18 | Typical Disk Initializer Dialog | 3-38 |
| 3-19 | Typical Dialog to Create a BACKUP Tape | 3-41 |
| 3-20 | Typical Dialog to Load BACKUP Tape | 3-43 |
| 3-21 | Typical MSYSGEN Dialog for Systems with Eclipse CPUs and Multiground RDOS | 3-47 |

TABLES

| Table # | Titles | Page # |
|---------|---|--------|
| 3-1 | CPU Cabinet Controls | 3-3 |
| 3-2 | CPU Controls and Indicators | 3-3 |
| 3-3 | Century Model 114 Disk Drive Controls and Indicators | 3-7 |
| 3-4 | Control Data 9762 and 9766 Disk Drive Controls and Indicators | 3-9 |
| 3-5A | Wangco Tape Drive Controls and Indicators | 3-11 |
| 3-5B | Cipher Tape Drive Controls and Indicators..... | 3-14 |
| 3-6 | DECWRITER II Controls and Indicators | 3-16 |
| 3-7 | Alphanumeric Keyboard Control and Indicators..... | 3-19 |
| 3-8 | Digitizer Table Controls and Indicators | 3-22 |
| 3-9 | Talos Table Controls and Indicators | 3-24 |
| 3-10 | 7-Key Programmable Stylus Controls and Indicators..... | 3-26 |
| 3-11 | 12-Key "Puck" Controls and Indicators | 3-27 |
| 3-12 | Digitizer Tablet and Pen Stylus Controls and Indicators..... | 3-28 |
| 3-13 | CRT Controls | 3-29 |

SECTION 1

General Description

1.1 System Configuration

The standard GDS II graphic design system consists of the following components:

- Central Processing Unit with 96K to 256K word memory; with Expansion Frame
- Disk Drive (25, 80, or 300 megabyte)
- 9-track Magnetic Tape Drive
- DECWRITER II
- 3 Design Stations, each including:
 - Digitizer Table or Tablet
 - Programmable Stylus or Pen Stylus
 - Alphanumeric Keyboard and Display Screen
 - Interactive CRT

The maximum GDS II system can expand to consist of the following:

- CPU, with Expansion Frame
- Modem for CPU/CPU link
- 4 Disk Drives
- Magnetic Tape Drive
- DECWRITER II
- Up to 8 Design Stations
- Up to 3 Plotters
- Line Printer

See Section 2 for a list of optional equipment.

1.2 CPU

The Eclipse S/230 CPU, illustrated in Figure 1-1 has 96K core memory. This configuration supports one design station and one plotter. (Core can be expanded to 256K by adding the desired number of 16K boards, in an expansion chassis, for up to 6 Design Stations and Plotters.) Also included are a Floating Point Board and a QTY Board that supports four design stations. A synchronous Line Adapter (SLA) is also available. The CPU processes and controls data flow between all attached peripheral components.



Figure 1-1. Eclipse S/230 CPU

1.3 Disk Drive

The disk drive, either 25, 80, or 300 megabytes (Figure 1-2), stores the system software programs and the GDS II data currently being processed at the Design Station(s).

1.4 Magnetic Tape Drive

The 9-track tape drive, either 800, 1600, or 800/1600 BPI (Figure 1-3), retains user software programs and GDS II data as a backup in the event of an inadvertent data loss from the disk memory.

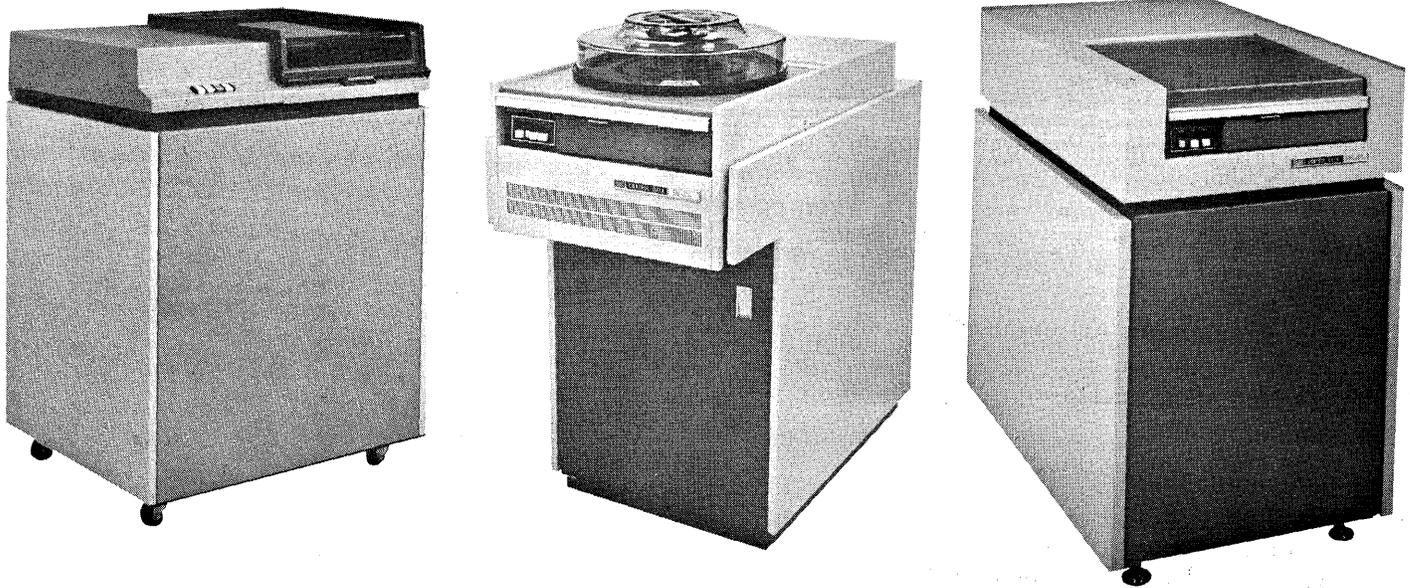


Figure 1-2. Century and CDC Disk Drives

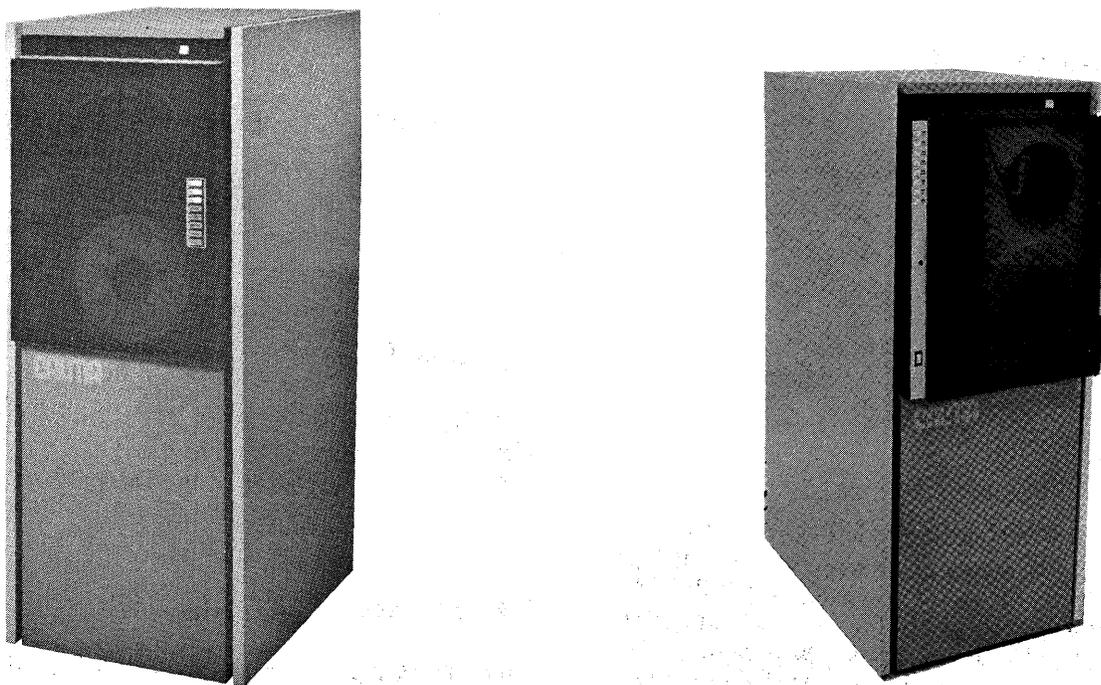


Figure 1-3. Wangco and Ciper Tape Drives

1.5 DECWRITER II

The DECWRITER II, illustrated in Figure 1-4, prints out CPU-generated output messages and lists for a permanent record. It also serves as an input device for initializing the system.

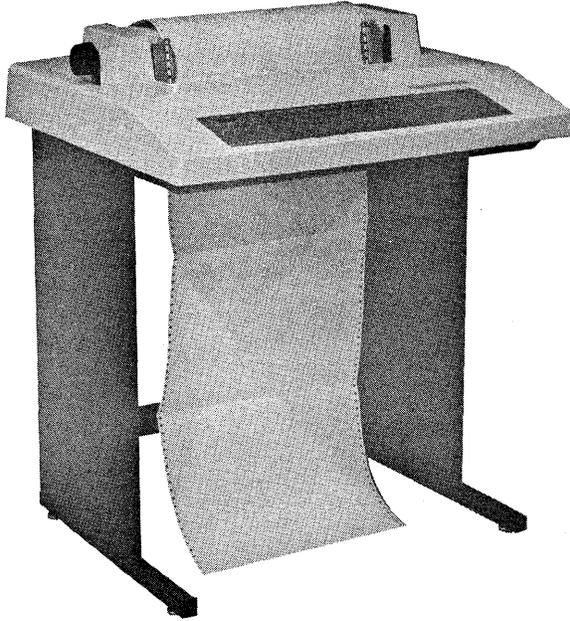


Figure 1-4. DECWRITER II

1.6 Design Station

The Design Station, as shown in Figure 1-5, is used by the operator to communicate with the system. The operator uses input commands to create and edit data and to operate the GDS II system. Three types of stations are available with the system, one using the backlit Digitizer Table and Constrained Stylus, one using the Talos Table or "Tabletizer" and Free Stylus, and one using the Digitizer Tablet and Pen. All types of Design Stations also include: an Alphanumeric Keyboard with a QTY display, an interactive CRT for graphics display, and optionally, a Special Functions Keyboard. Each of the Design Station components is described below.

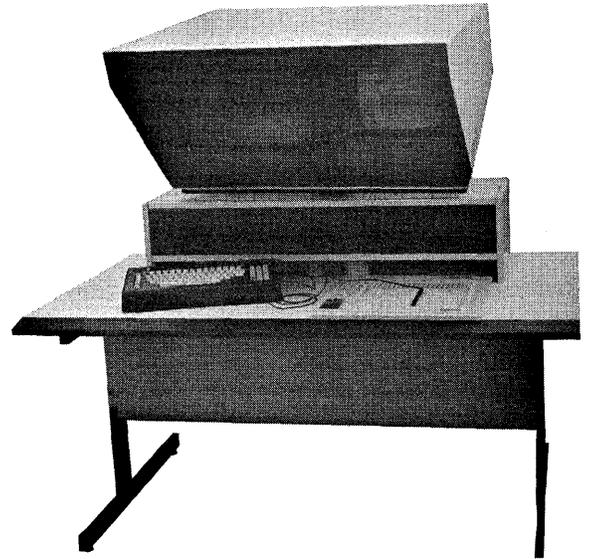


Figure 1-5. Design Station

1.6.1 Digitizer Table

The backlit Digitizer Table, Figure 1-6, is 48 x 60" with a 7button constrained stylus. Four of the stylus buttons can be programmed for the most frequently used commands.

1.6.2 Digitizer Talos Table or Tabletizer

The Talos Table, 44 x 60", shown in Figure 1-7, is essentially a large tablet. Data is input to it from a 12-button free Stylus or from a pen.



Figure 1-6. Digitizer Table

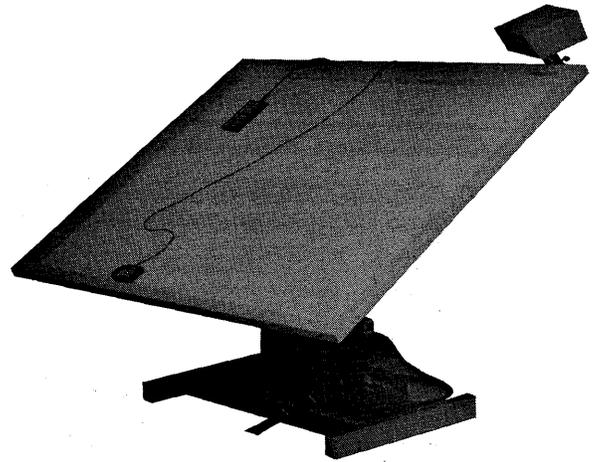


Figure 1-7. Digitizer Talos Table

1.6.3 Digitizer Tablet

The 11 x 11" Tablet, Figure 1-8, can be placed on the Design Station table. Data is entered by means of a pen. A copy of the command menu in use can be taped to the Tablet for convenient use.

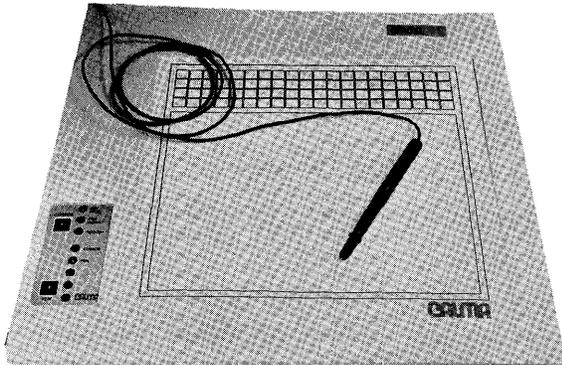


Figure 1-8. Digitizer Tablet

1.6.4 GDS II Keyboard

Alphanumeric data and control information are entered into the system at the keyboard, shown in Figure 1-5.

1.6.5 Alphanumeric QTY Display

An 11" QTY (Figure 1-5) displays alphanumeric user input commands, results of calculations, and other system information, and it responds with prompts to the user.

1.6.6 CRT Display Monitor

A 19" CRT (Figure 1-5) displays all graphics digitized in the system. The movement of the stylus as it digitizes on the Table or Tablet is represented on the screen by a cursor, a flashing cross. A hard-copy-compatible CRT is available for use with a Hard Copy (see Section 2, Options).

1.6.7 Special Function Keyboard

A 32-button keyboard shown in Figure 1-9 can be used to supplement the command repertoire. The buttons can be programmed for frequently used commands, DFUNCTION commands 01-32.

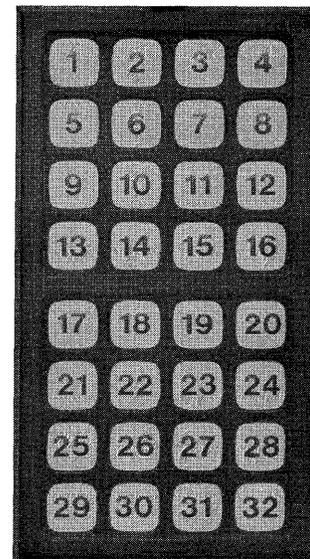


Figure 1-9. Special Function Keyboard

SECTION 2

Options

2.1 Hardware Options

In addition to the components standard to the GDS II graphic design system, as described in Section I, there are several kinds of optional hardware that can be driven by the system. The following subsections describe this equipment. (For complete descriptions and operating procedures, see the individual manufacturer's manuals.)

2.1.1 Plotters

The following online plotters can be used with the GDS II system:

- Calcomp 936 Plotter
- Calcomp 960 Plotter
- Calcomp 1055 Plotter
- Xynetics 1100 Plotter
- Xynetics 1200 Plotter

2.1.2 CRT Hard Copy

The Hard Copy provides a paper copy of the graphics from the CRT at the Design Station. The following models can be used with the GDS II system:

- Tektronics Model 4631
- Versatec Model 1200 Plotter Hard Copy w/C-Tex-2 Controller, standard configuration (requires hard copy compatible CRT).
- Versatec Model 1200 Plotter Hard Copy w/C-Tex-5 Controller—Rack mounted, Multiple Configuration (requires hard copy compatible CRT).

2.1.3 Optical Pattern Generator

The OPG outputs the final design and photostats it at a very fine scale. The following OPG's are compatible with the GDS II system to be used with either metric or English software, using 9-track magnetic tape.

- Mann 1600 Pattern Generator
- Mann 2600 Pattern Generator
- Mann 3000 Pattern Generator
- Mann 3600 Pattern Generator
- Electromask Pattern Generator

2.1.4 E-Beam Pattern Generators

The Electron Beam Pattern Generator uses a laser as its method of controlling precision. Like the OPG, it outputs the final design and photostats it. The following E-Beam Pattern Generators are used with the GDS II system:

- ETEC (MEBES)
- ETEC (LEBES)

2.1.5 Line Printer

The Line Printer provides a hard copy of data from the CPU. The Data Products Line Printer, which prints out 300 lines per minute, is used in the GDS II system.

The following printer/plotters are also used:

- Versatec Model 1200A Printer/Plotter Hard Copy w/C-Tex-2 Controller, CALMA Line Printer Controller, Standard Configuration.
- Versatec Model 1200A Printer/Plotter Hard Copy w/C-Tex-5 Controller—Rack Mounted, CALMA Line Printer Controller, Multiple Configuration.

SECTION 3

System Operation

3.1 Introduction

The standard GDS II Interactive Graphic System consists of a CPU; a disk drive; a tape drive; a system console (DECWRITER II); and one to three design stations, which have a digitizer table or tablet, a stylus, an alphanumeric keyboard, a QTY for user input, an interactive CRT, and a plotter.

This section presents the information needed to operate the GDS II system. First, it describes the controls and indicators for each component of the standard system. Second, it presents the power-up and power-down procedures for this system, in the order they should be accomplished.

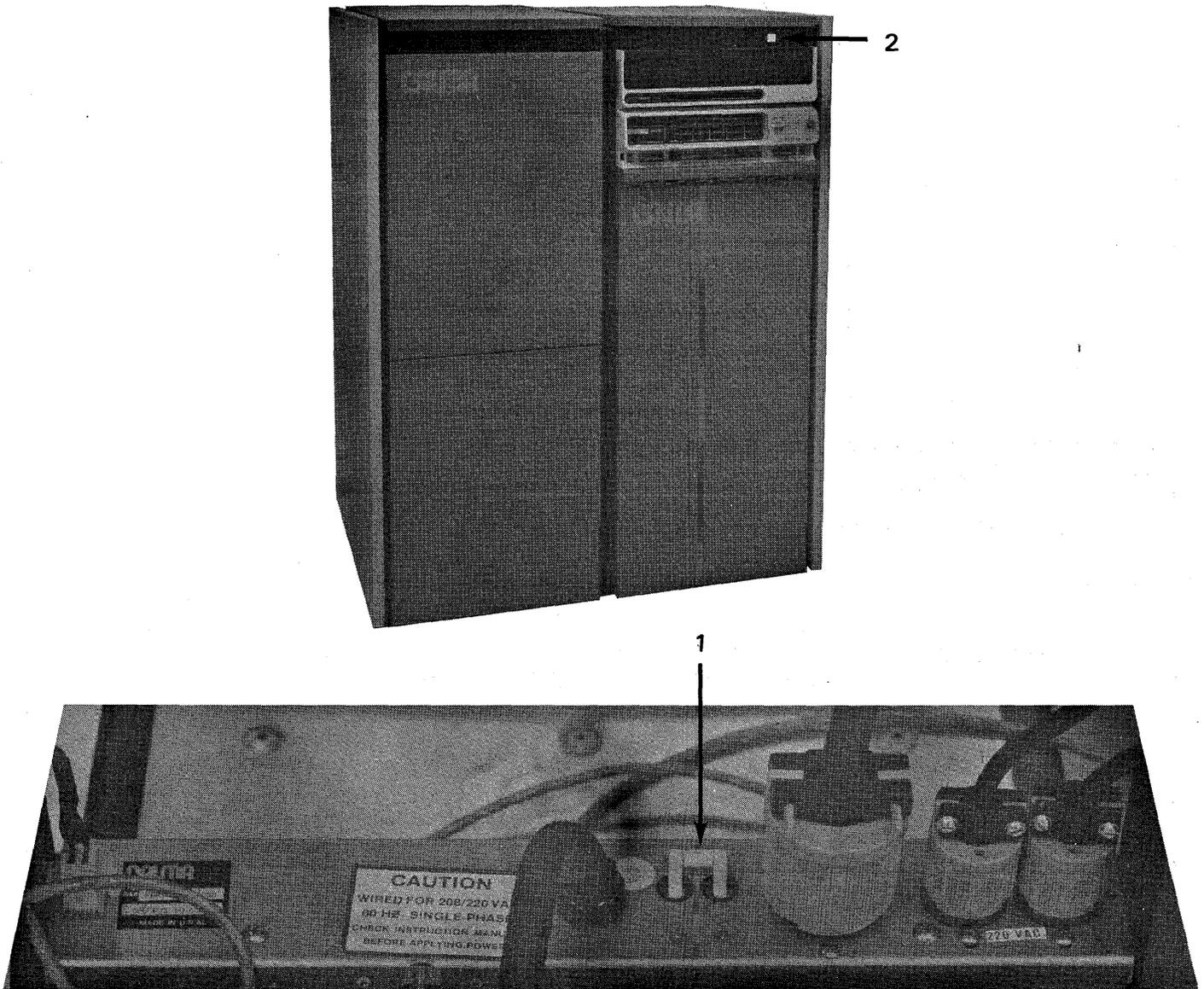


Figure 3-1. Eclipse S/230 CPU Power Switches

3.2 Controls and Indicators

A description of the controls and indicators for each component in the GDS II system is presented in the following subsections.

3.2.1 CPU Controls and Indicators

The Data General Eclipse S/230 CPU accommodates up to 256 words of core memory. The controls and indicators on the Eclipse are shown in Figures 3-1 and 3-2. Tables 3-1 and 3-2 identify the function of each control or indicator.

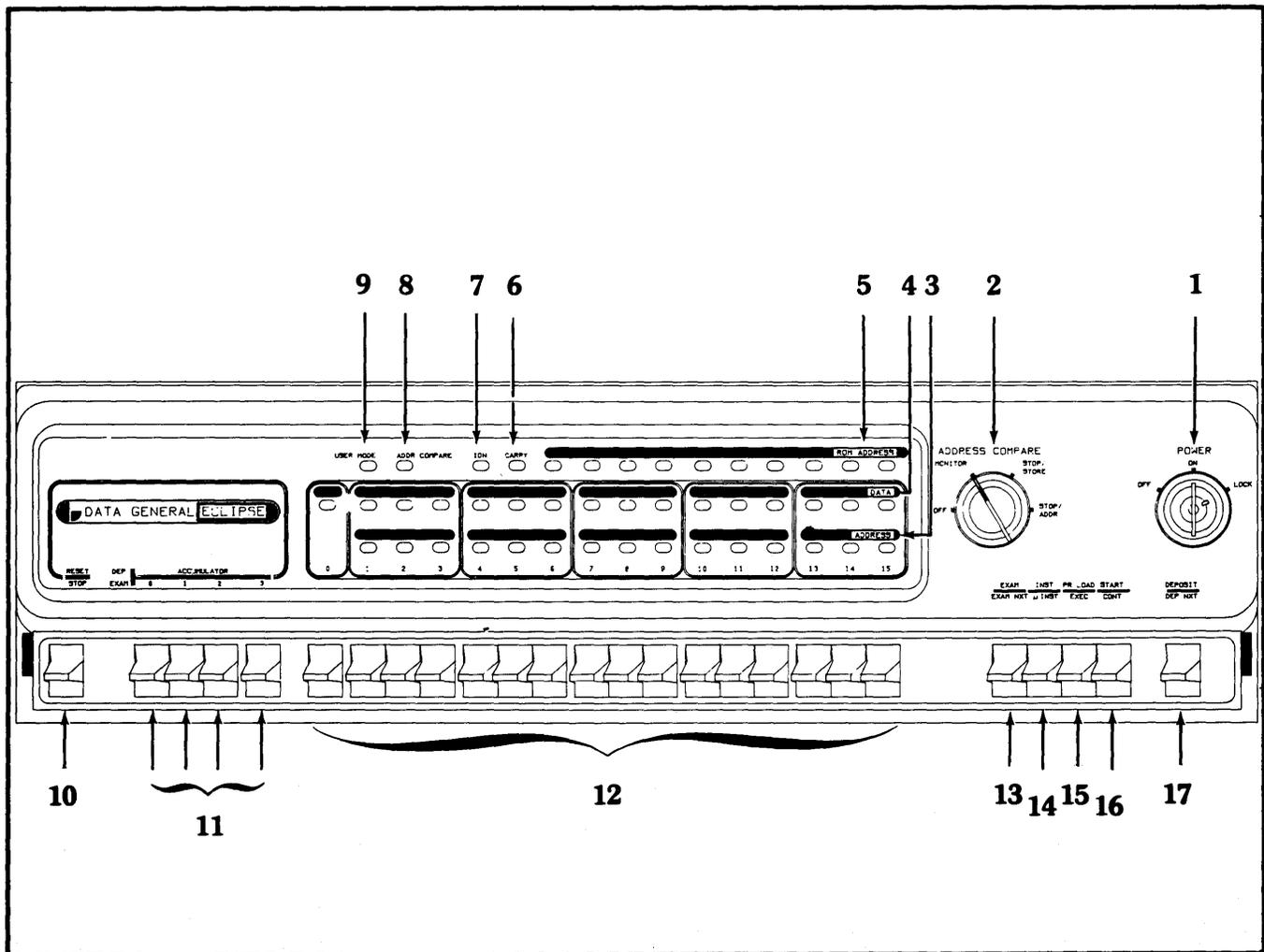


Figure 3-2. Eclipse S/230 Control Panel

Table 3-1. CPU Cabinet Controls

| ID NO. | NAME | FUNCTION |
|--------|---|--|
| 1 | Main Power Circuit Breaker (at rear of cabinet) | Controls power to the main power switch. |
| 2 | Main Power Pushbutton (white button at front of cabinet) | Controls power to computer, and to power outlets, at rear of cabinet, which provides power to all other components in the system, the card cage, if one is in use, the disk drive, the DECWRITER II, the tape drive and the design stations. |

Table 3-2. CPU Controls and Indicators

| ID NO. | NAME | FUNCTION |
|--------|--|---|
| 1 | POWER (Key Switch) | Regulates power to CPU. When switch is OFF, no power goes to CPU. When switch is ON, power is applied to CPU, enabling all computer switches. |
| 2 | ADDRESS COMPARE (4-position rotary switch): | |
| | OFF | Address compare feature is disabled. |
| | MONITOR | Permits examination and monitoring of memory locations while CPU is running. |
| | STOP/STORE | Suspends CPU operation if an attempt is made to alter address location specified by DATA switches. |
| | STOP/ADDR | Suspends CPU operation if an attempt is made to access location specified by DATA switches. |
| 3 | ADDRESS (16 Indicators) | Display current content of memory address bus. |
| 4 | DATA (16 Indicators) | Display binary data located in general register 0. |
| 5 | ROM ADDRESS (10 Indicators) | Display, in microcode, the address of next micro instruction to be fetched. |
| 6 | CARRY (Indicator) | Lights when carry bit is a logical 1. |
| 7 | ION (Indicator) | Lights when interrupts are enabled. |
| 8 | ADDR COMPARE (Indicator) | Lights to indicate that a comparison exists when the Address Compare switch is positioned to request an address comparison. |
| 9 | USER MODE | Lights when computer is operating. |

| ID NO. | NAME | FUNCTION |
|--------|---|--|
| 10 | RESET/STOP (Spring-loaded switch; returns to neutral "OFF" when released) | |
| | Up = RESET | I/O reset instruction is executed. CPU is stopped after completing current processor cycle. |
| | Down = STOP | Stops CPU after current instruction is completed. |
| 11 | ACCUMULATOR (0, 1, 2, 3, Spring-loaded switches) | These four accumulator deposit switches are sequentially numbered from left to right and are mutually exclusive. Each switch affects only its corresponding accumulator. |
| | | When switched to DEP (up) the contents specified by the DATA switches are deposited in the corresponding accumulator. When switched to EXAM (down) the contents of the corresponding accumulator are displayed by the DATA indicators. |
| 12 | DATA (16 two-position toggle switches) | Represent a 16-bit binary word, either data or an address. |
| 13 | EXAM/EXAM NXT (Spring-loaded switch) | |
| | Up = EXAM | Address specified by DATA switches is loaded into the Program Counter; this address is displayed by ADDRESS indicators, and addressed word is displayed by DATA indicators. |
| | Down = EXAM Next Position | Program counter is incremented by one; new address is displayed by ADDRESS indicators, and addressed word contents are displayed by DATA indicators. |
| 14 | INST/ μ INST (Spring-loaded switch) | |
| | Up = INST | Instruction at current address specified in contents of program counter is executed. Processing stops. |
| | Down = μ INST | Next sequential micro-instruction is executed. Microcode processor is stopped. |
| 15 | PR LOAD/EXEC (Spring-loaded switch) | |
| | Up = Program LOAD | Contents of bootstrap read-only memory are loaded into memory location 0 through 378. JUMP 0 instruction is executed. |
| | Down = EXEC | Instruction specified by contents of DATA switches is executed. |
| 16 | START/CONT (Spring-loaded switch) | |
| | Up = START | Address contents specified by DATA switches are placed in program |

| ID NO. | NAME | FUNCTION |
|-----------|--|--|
| | Down = CONTInue | counter. Execution resumes with instruction at address location specified by contents of program counter. Execution resumes with instruction at address location specified by ADDRESS indicators. |
| 17 | DEPOSIT/DEP NXT (Spring-loaded switch) | |
| | Up = DEPOSIT | Contents set by DATA switches are loaded into address location specified by current value in program counter; updated value is displayed by DATA indicators. |
| | Down = EXAM NeXT Position | Program counter is incremented by one. Contents of DATA switches displayed by DATA indicators are loaded into address location indicated by ADDRESS indicators. |

3.2.2 Disk Drive Controls and Indicators

Three disk drives of three different capacities can be used in the GDS II system:

- 25-megabyte Century 114 Disk Drive
- 80-megabyte Control Data 9762 Disk Drive
- 300-megabyte Control Data 9766 Disk Drive

The standard GDS II system uses just one of these drives.

3.2.2.1 Century 114 Disk Drive

The controls and indicators for this 25-megabyte drive are shown in Figure 3-3 and described in Table 3-3.

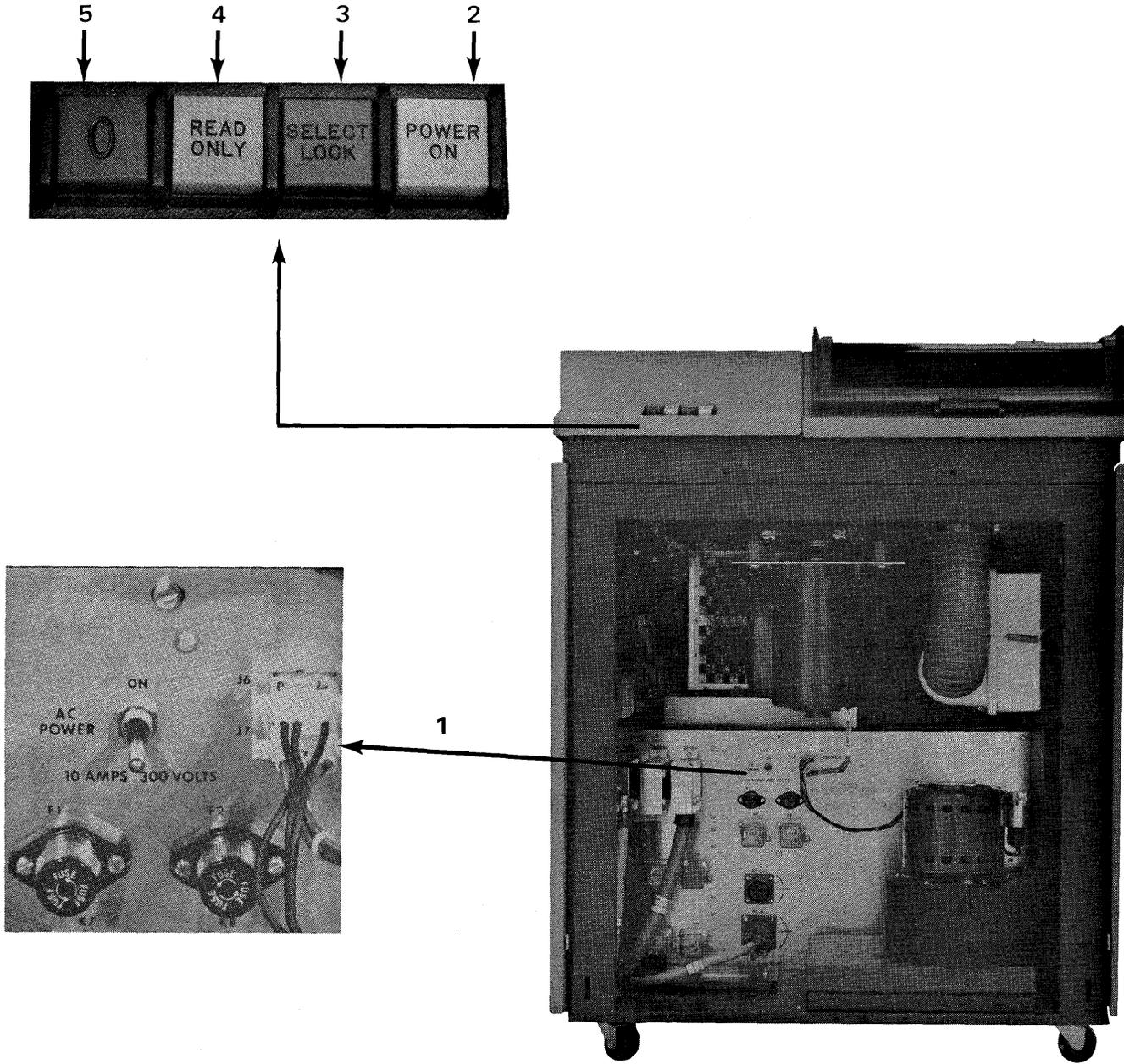


Figure 3-3. Century 114 Disk Drive Controls and Indicators

Table 3-3. Century Model 114 Disk Drive Controls & Indicators

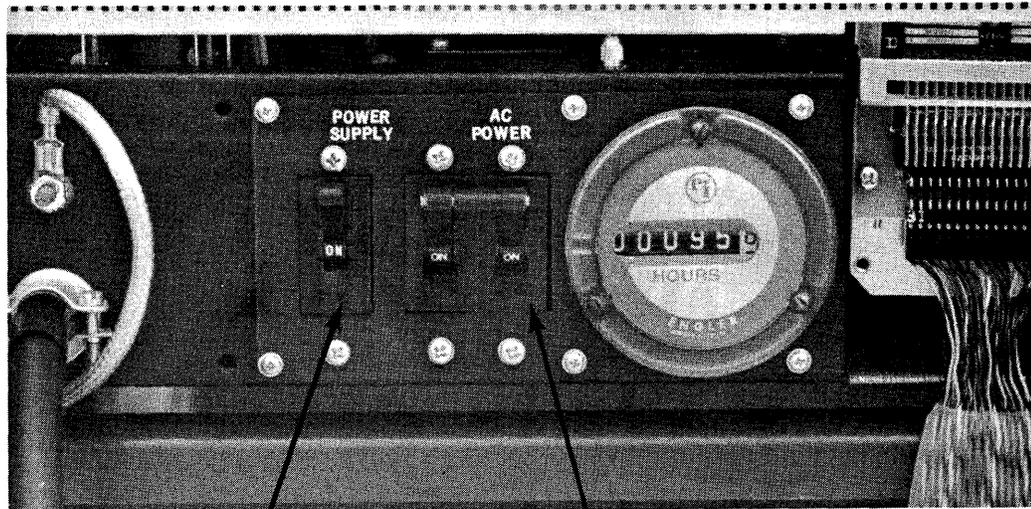
| ID NO. | NAME | FUNCTION |
|--------|---|--|
| 1 | AC POWER ON/OFF (Switch located behind front maintenance panel) | Normally left in the ON position. When the switch is set to the ON position, power is routed to the POWER ON switch. |
| 2 | POWER ON (Pushbutton) | <p>After a disk pack has been properly loaded onto the spindle and the access door has been closed, the POWER ON pushbutton switch is depressed to turn on the drive motor and the disk pack brush cleaning motor. The POWER ON pushbutton lights to indicate that power is applied to the drive. As soon as power is applied, the disk pack starts to rotate and a cleaning brush sweeps across the disk surfaces. After approximately 14 seconds, the 0 (ready) indicator lights.</p> <p>The POWER ON pushbutton is also depressed to turn power off. Turning power off the drive causes the disk pack to stop rotating within approximately 14 seconds.</p> |
| 3 | SELECT LOCK (Indicator) | Lights to indicate that an internal failure has occurred and that the disk drive requires maintenance. The disk drive may not be placed on line. |
| 4 | READ ONLY (Pushbutton) | Depressed to inhibit writing to the disk. Indicator lights when writing to the disk inhibited. Normally, this pushbutton should not be used (the light should remain out) to enable writing on the disk pack. |
| 5 | 0 (Ready Indicator) | Lights to indicate that the drive is ready to be exercised by the computer. |

NOTE

If the indicator does not light within 20 seconds after the POWER ON pushbutton is depressed, depress the POWER ON pushbutton again and check the light bulb.

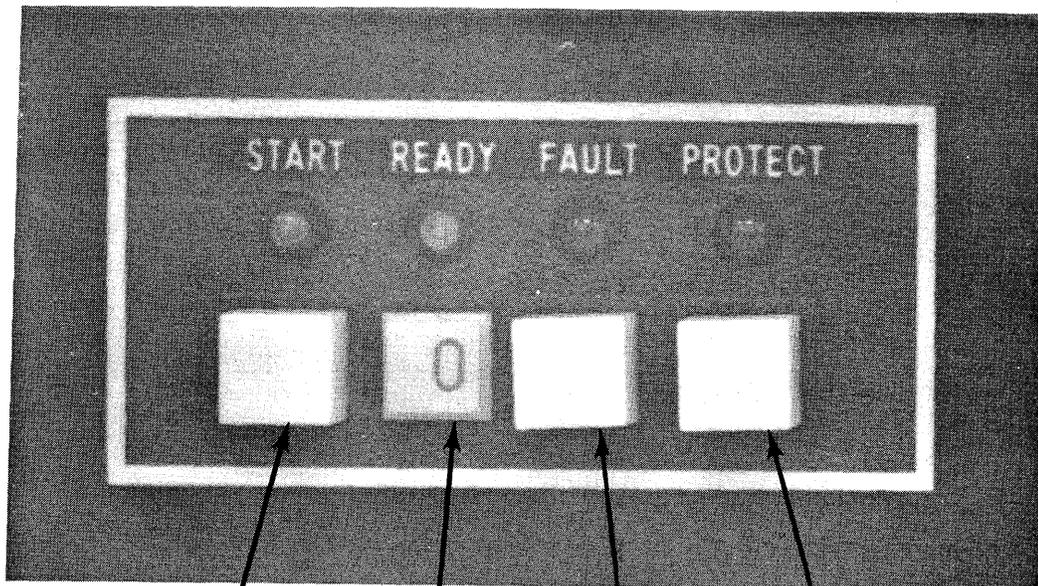
3.2.2.2 Control Data 9762 and 9766 Disk Drives

The controls and indicators of the two Control Data drives are identical; they are illustrated in Figure 3-4 and described in Table 3-4.



2

1



3

4

5

6

Figure 3-4. Control Data 9762 and 9766 Disk Drive Controls and Indicators

**Table 3-4. Control Data 9762 and 9766
Disk Drive Controls and Indicators**

| ID NO. | NAME | FUNCTION |
|--|--|--|
| 1 | AC POWER (Double switch at rear of drive) | Supplies AC power to the drive; this switch is turned on first. |
| 2 | POWER (Single switch at rear of drive) | Turns on the internal power to operate the drive; turns on the air blower to purge dust from the drive. |
| NOTE | | |
| These switches must be turned on before loading the disk pack to allow purge time. | | |
| 3 | START (Pushbutton and light indicator at front of drive) | After the disk pack has been properly loaded and the drive cover closed, the START button can be depressed. The START light indicator will light. The disk pack will start spinning and will take several seconds to get up to speed. The START button is also used to stop the drive. |
| 4 | READY (Pushbutton and light indicator) | While the drive pack is coming up to speed, the READY light indicator will blink. Once the drive is at speed and the heads are loaded, the indicator will light in a steady state. The READY light indicator also blinks when the pack is slowing down when it is being stopped. The number on the READY pushbutton indicates the drive number. The pushbutton is set in the depressed position, indicating the drive is in use on the system. |
| 5 | FAULT (Pushbutton and light indicator) | The FAULT pushbutton is to be pressed when a fault has occurred and the drive must be reset. The light indicator lights when the drive is reset. Then the START button can be pressed to restart the drive. |
| 6 | PROTECT (Pushbutton and light indicator) | The PROTECT button is used to inhibit writing on the disk pack and to Read Only. The light indicator lights when the drive is in Read Only. Normally, the PROTECT button is not used. Therefore the indicator normally will remain unlit. |

3.2.3 Tape Drive Controls and Indicators

There are three magnetic tape drives available for use in the GDS II system, an 800 BPI Wangco drive, 1600 BPI Wangco drive, and an 800/1600 BPI Cipher drive.

3.2.3.1 800 and 1600 BPI Wangco Tape Drive

The controls and indicators for the Wangco drives are shown in Figure 3-5A and described in Table 3-5A.

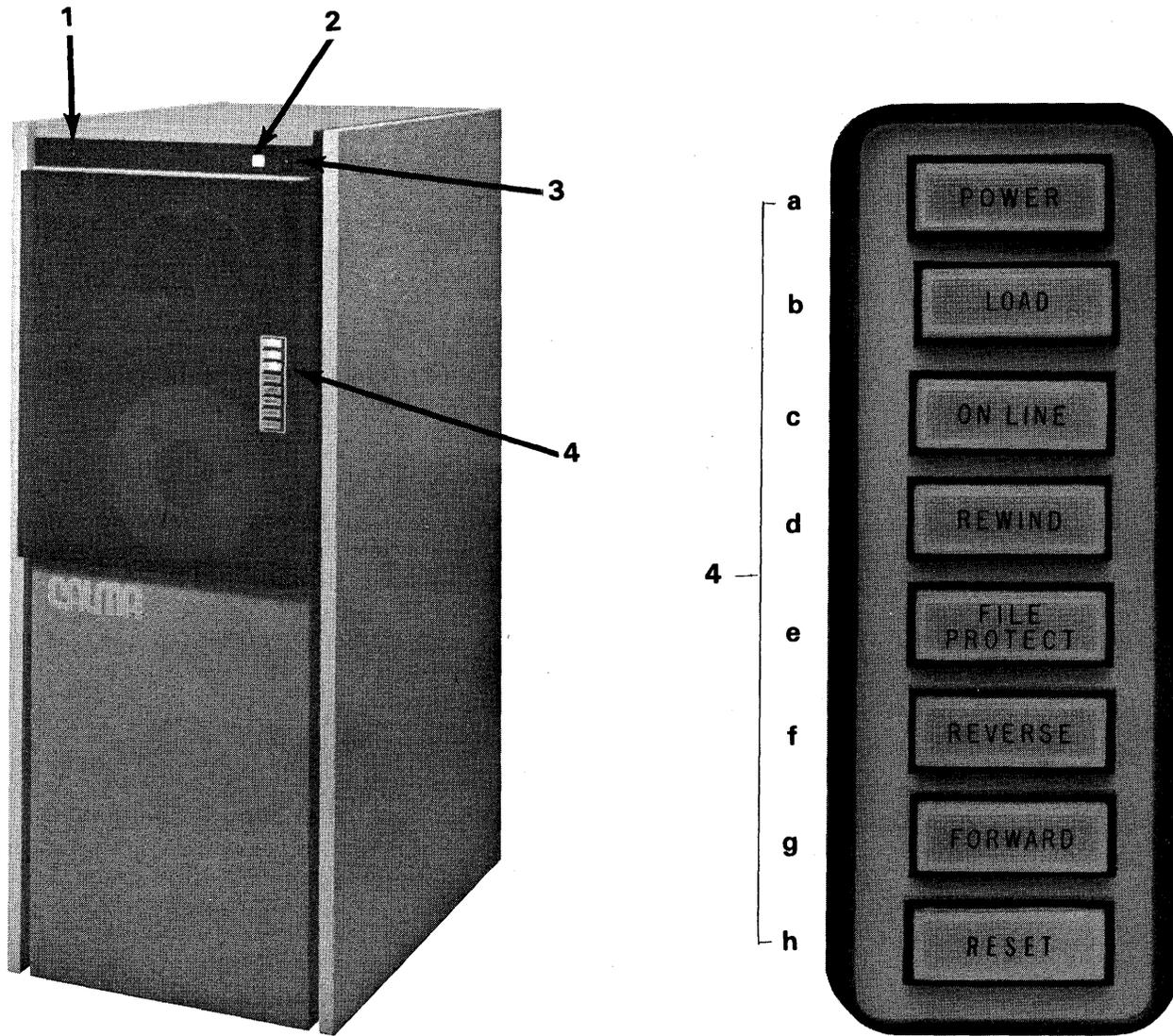


Figure 3-5A. Wangco Tape Drive Controls and Indicators

Table 3-5A. Wangco Tape Drive Control and Indicators

| ID NO. | NAME | FUNCTION |
|-----------|-----------------------------|---|
| 1 | SYSTEM SELECT (Switch) | When the system has dual computers, this thumbwheel switch is set to the address of the controlling computer. If two computers are used, one is system 0; the other is system 1. |
| 2 | POWER (Pushbutton) | This pushbutton controls power to the tape drive POWER ON switch. The indicator lights when power is applied to the tape drive. |
| 3 | UNIT SELECT (Switch) | When more than one tape drive is used by the system, this thumbwheel switch is set to the number (address) of the tape drive assigned to be used by the system. If only one tape drive is used, the thumbwheel is set to 0. |
| 4 | CONTROL PANEL | |
| | a. POWER (Pushbutton) | The POWER pushbutton controls power to the tape drive. The pushbutton lights when power is on. |
| | b. LOAD (Pushbutton) | <p>After the tape has been threaded, the LOAD pushbutton is depressed to complete the tape load operation. This causes the tape to advance to the load point or to the beginning-of-tape (BOT) marker. The LOAD pushbutton lights when the tape load operation is completed or when the tape is positioned to the load point.</p> <p>The LOAD pushbutton light goes out when the tape is advanced from the point or is rewound. At the completion of the tape load operation, the LOAD pushbutton is disabled. It is reenabled by the loss of tape tension.</p> |
| | c. ON LINE (Pushbutton) | The ONLINE pushbutton is depressed to place the tape drive on line to the computer. It lights when the tape drive is on line after the tape load operation is completed. |
| | d. REWIND (Pushbutton) | The REWIND pushbutton is depressed to rewind the tape at high speed when the tape drive is off line. It lights when the rewind operation is in process. The tape rewinds beyond the BOT marker stops, and returns to the load point. Depressing the REWIND a second time rewinds the tape off the supply reel. The REWIND pushbutton is disabled when the tape drive is on line. This safety feature prevents accidental tape damage. |
| | e. FILE PROTECT (Indicator) | <p>The FILE PROTECT indicator lights to indicate that a write enable ring is not installed on the tape reel. When this indicator is lighted, data cannot be written on the tape.</p> <p>When the write enable ring is installed on the tape reel, the indicator is not lighted and data can be written on the tape or read from the tape.</p> |
| | f. REVERSE (Pushbutton) | This pushbutton is depressed to rewind the tape at normal tape speed |

| ID NO. | NAME | FUNCTION |
|-----------|----------------------|--|
| | | when the tape drive is off line. The pushbutton lights when the rewind operation is in progress. |
| g. | FORWARD (Pushbutton) | When the tape drive is on line, the FORWARD pushbutton is disabled. |
| h. | RESET (Pushbutton) | This pushbutton is depressed to: (1) stop all tape motion except unloading, (2) place the tape drive in the off line mode (ON LINE indicator goes out); and (3) clear all read, write, and control functions. It lights when the reset function occurs, and remains lighted until another operation commences. |

3.2.3.2 800/1600 BPI Cipher Tape Drive

The controls and indicators for the Cipher Tape Drive are shown in Figure 3-5B and described in Table 3-5B.

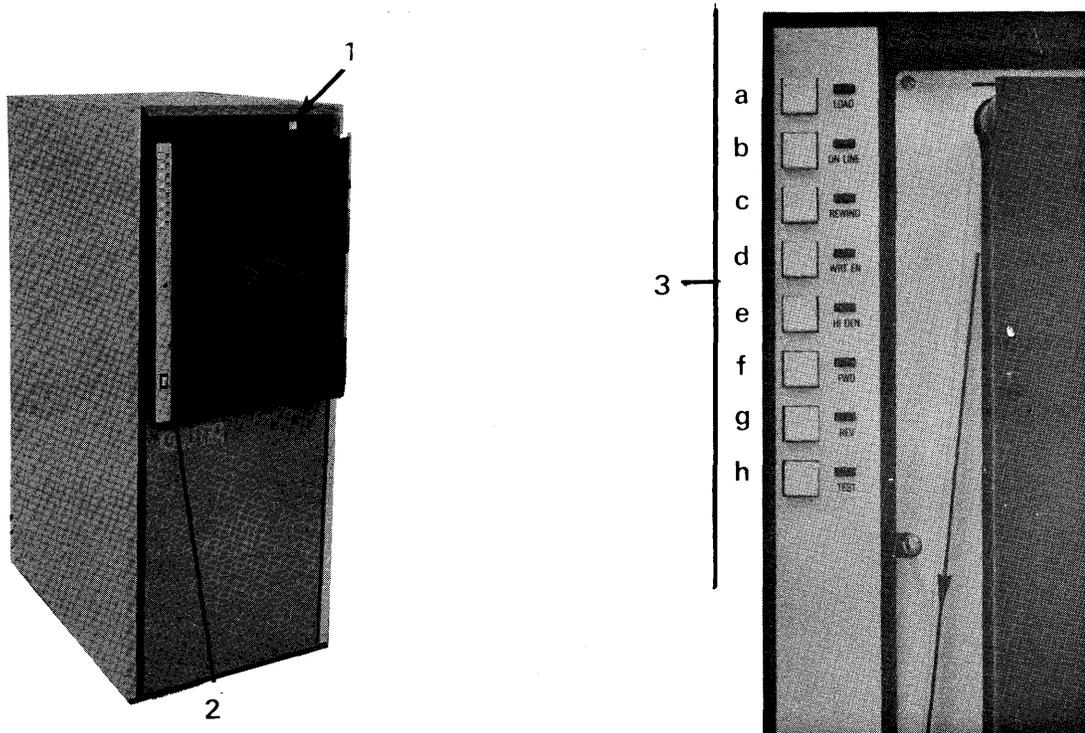


Figure 3-5B. Cipher Tape Drive Controls and Indicators

Table 3-5B. Cipher Tape Drive Controls and Indicators

| ID NO. | NAME | FUNCTION |
|-----------|---|--|
| 1 | White Pushbutton/Indicator (at top, right front of drive) | Supplies power, through the CPU, to the drive; turns on blower. The button lights when the drive is powered on. |
| 2 | ON/OFF Toggle Switch | Supplies power internally to the drive. |
| 3 | Operator Control Pushbuttons and Light Indicators | |
| | a. LOAD | Loads tape into vacuum until it reads Load Point on the tape. Then it stops and the indicator lights. The tape is now ready for reading or writing. |
| | b. ON LINE | Once the tape is loaded and ready, the online pushbutton can be depressed to provide the CPU access to the tape. The indicator lights. When the online button is pressed again, the tape cannot be accessed by the CPU. The light is off. |
| | c. REWIND | When this button is depressed, the tape rewinds and stops at the Load Point. The indicator lights. When pressed again, the button enables the dismount of the tape so that it can be removed from the drive. The indicator light is off. |
| | d. WRT EN (Write Enable) | Enables writing on the tape. Can be used only if the Write Ring is in the tape. Write Enable will not work if the Write Ring is absent. |
| | e. HI DEN (High Density) | Switches the drive from 800 BPI to 1600 BPI operation. |
| | f. FWD (Forward) | An ON/OFF switch which, when depressed, starts the drive moving forward. When depressed again, the drive stops. |
| | g. REV (Reverse) | An ON/OFF switch which, when depressed, moves the drive in reverse. When depressed again, the drive stops. |
| | h. TEST | Used by the Customer Service Engineer, for operations internal to the tape drive. |

3.2.4 System Console - DECWRITER II Controls and Indicators

The controls and indicators of the DECWRITER II are illustrated in Figure 3-6 and described in Table 3-6.

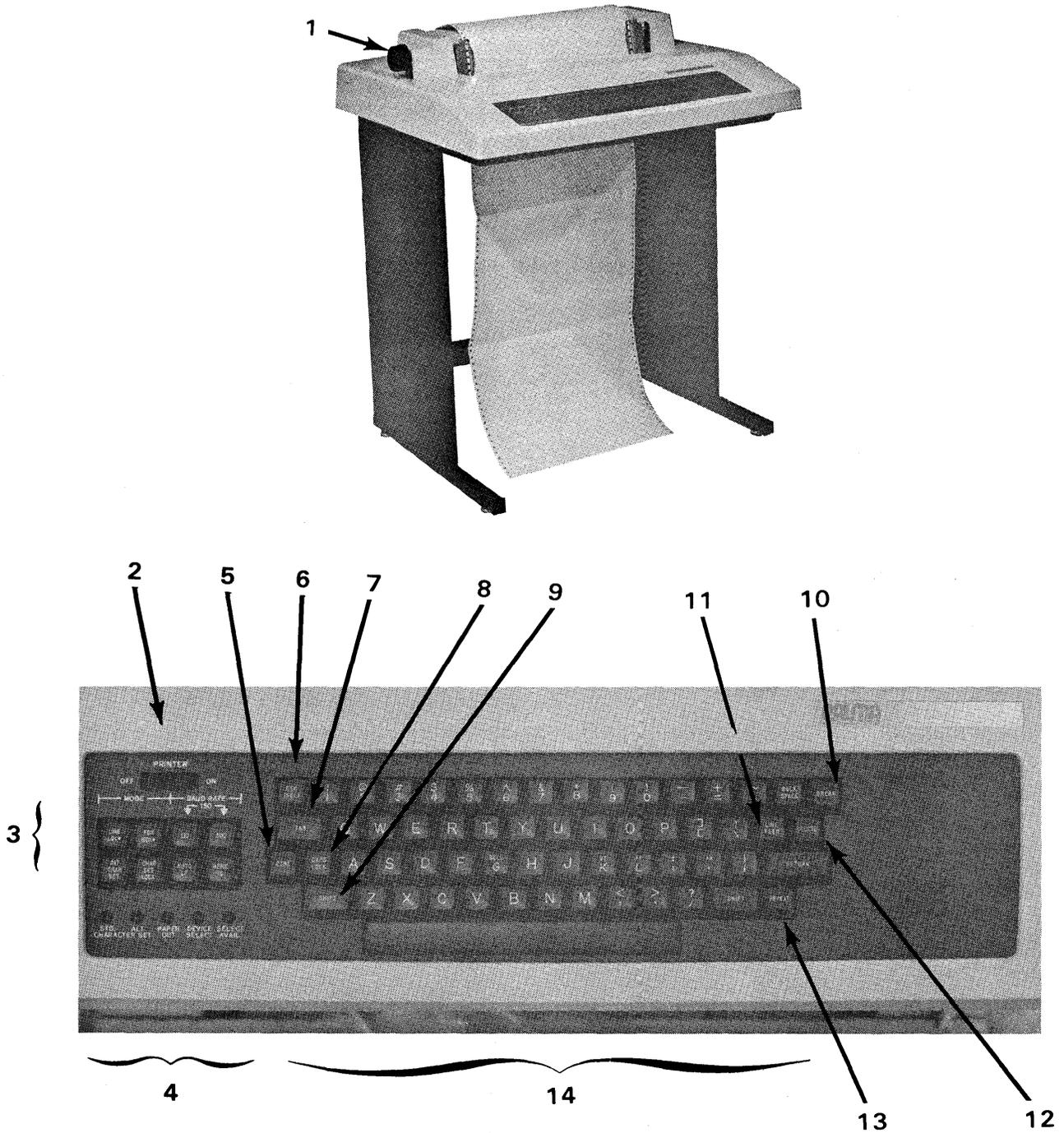


Figure 3-6. DECWRITER II Controls and Indicators

Table 3-6. DECWRITER II Controls and Indicators

| ID NO. | NAME | FUNCTION |
|--------|----------------------------------|--|
| 1 | Paper Advance Knob | Allows manual paper feed and adjustment. |
| 2 | PRINTER OFF/ON Switch | Controls application of power to the DECWRITER II. |
| 3 | Operator Control Switches | <p style="text-align: center;">NOTE</p> <p>A dot on a switch indicates the function of the switch when depressed.</p> |
| | LINE/LOC (2-position switch) | When in LINE position, the DECWRITER II can transmit and receive data. When in LOC position (depressed), the receive/transmit lines are disabled, and only local operation can occur. |
| | FDX/HDX (2-position switch) | Controls the printing of transmitted keyboard characters. When in FDX position, characters typed on keyboard are transmitted; only received characters are printed. When in HDX position, keyboard characters are both transmitted and printed; received characters are also printed. Do not attempt to transmit data when receiving data in HDX mode. |
| | BAUD RATE Switches | Select rate at which characters are transmitted and received over the communication line. |
| | Switch 110 | Transmits 10 characters/second. |
| | Switch 300 | Transmits 30 characters/second. |
| | Switches 110 and 300 | Transmits 15 characters/second. |
| | ALT CHAR SET Switch | When an alternate character set is installed, this switch allows selection of either ASCII characters (switch up) or the alternate set (switch down). |
| | CHAR SET LOCK Switch | Allows selection of manual (switch down) or automatic (switch up) character set switching. |
| | AUTO LF Switch | Not used in GDS II. |
| | HERE IS Switch | Not used in GDS II. |
| 4 | Light Indicators | |
| | STD/ALT CHARACTER SET Indicators | When the standard character set is installed, the STD indicator lights. When an alternate character set is in use, the ALT indicator lights. |
| | PAPER OUT Indicator | When illuminated, indicates that the DECWRITER II is out of paper; printing stops. |
| | DEVICE SELECT Indicator | When illuminated, indicates that the DECWRITER II has been selected as a slave terminal and can transmit data back to the transmitting unit. |

| ID NO. | NAME | FUNCTION |
|------------------------|---------------------------|--|
| | SELECT AVAIL Indicator | Indicates that data communications channel is available and DECWRITER II can initiate data transmission. |
| Special Keyboard Keys: | | |
| 5 | CTRL Key | Provides method of transmitting ASCII control codes. Holding the CTRL down and depressing any alphanumeric key or control key changes the standard ASCII code for that key to a control code. |
| 6 | ESC (SEL) Key | Generates an escape command code sequence. |
| 7 | TAB Key | Generates a tab. |
| 8 | CAPS LOCK Key | The 26 letter key transmits only upper case when this key is down. No other keys are affected. |
| 9 | SHIFT Key | Momentary switch, which when depressed, allows printable characters to be in uppercase. |
| 10 | BREAK Key | For use in the half-duplex mode of transmission. Interrupts incoming data flow by forcing communication line from a mark mode into a space mode, until BREAK key is released. |
| 11 | LINE FEED Key | Generates the code that causes the printer to advance the paper one line. |
| 12 | DELETE Key | When this key is depressed, a code is sent to the CPU to delete the last character on the current line. The printer does not respond to the DELETE key. |
| 13 | REPEAT Key | When the REPEAT key is held down and any key is depressed, the ASCII code for that character is transmitted and printed at a repetition rate of approximately 15 words/second until the key is released. |
| 14 | Standard Keyboard | The keys of the standard keyboard perform the same functions of a standard typewriter with the following exceptions: |
| | BELL (Uppercase on G key) | Not used in GDS II. |
| | VT (Uppercase on K key) | Not used in GDS II. |
| | FF (Uppercase on L key) | Not used in GDS II. |

3.2.5 Design Station Alphanumeric Keyboard

The controls and indicators on the Alphanumeric Keyboard are shown in Figure 3-7 and described in Table 3-7.

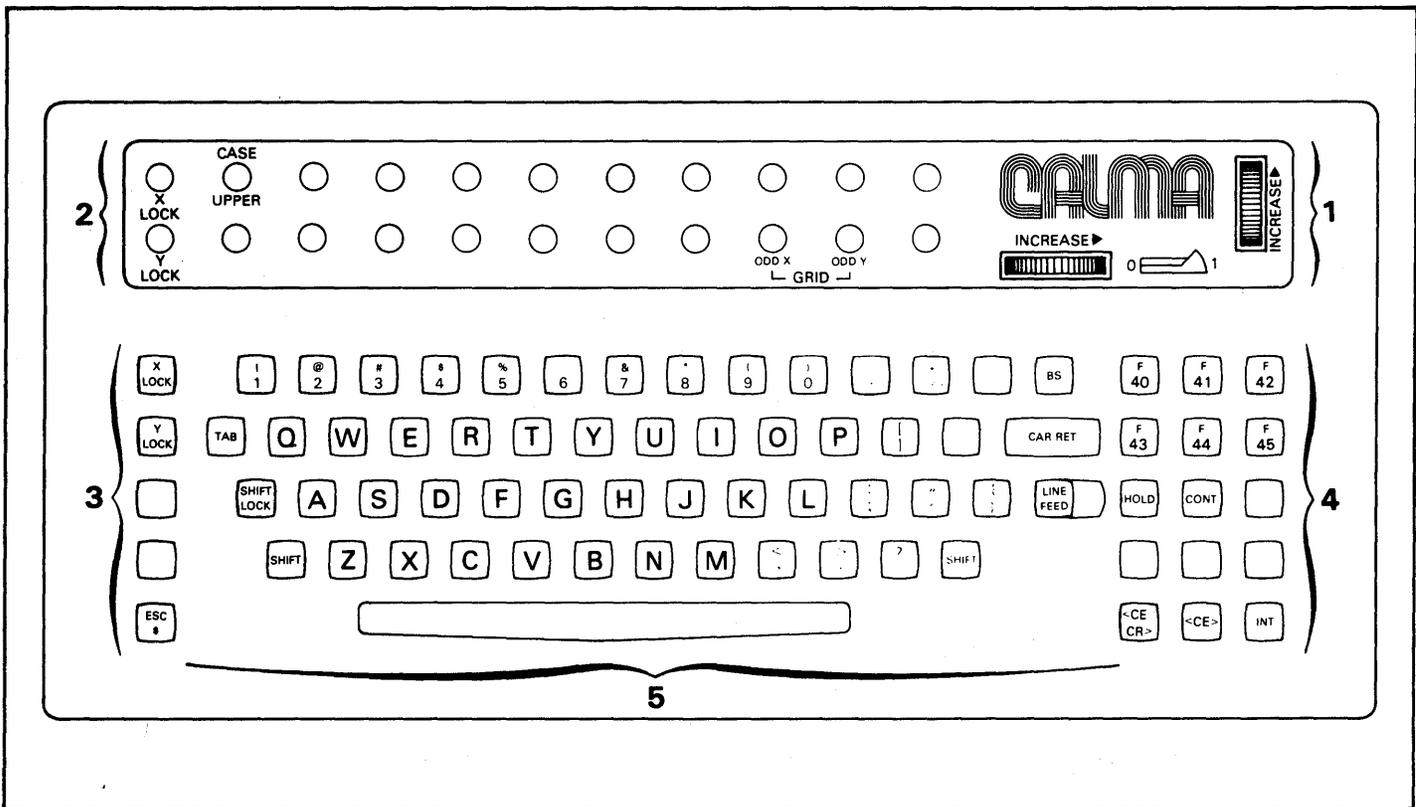


Figure 3-7. Design Station Alphanumeric Keyboard

Table 3-7. Alphanumeric Keyboard Controls and Indicators

| ID NO. | NAME | FUNCTION |
|-----------|--------------------------------|---|
| 1 | INCREASE Rotary Dials (2) | The vertical dial turns the keyboard and QTM ON and OFF. The horizontal dial is not used. |
| | System Select Switch - 0 and 1 | Selects one of two CPU's, 0 or 1, when two CPU's are used in the system. Use 0 when only CPU 0 is used. |
| 2 | Indicators | Only the indicators listed are used in GDS II. |
| | X LOCK | Lights when the stylus X axis lock is on. |
| | Y LOCK | Lights when the stylus Y axis lock is on. |
| | CASE UPPER | Lights when SHIFT key is depressed. |
| | GRID ODD X | Indicates that stylus or pen is at an odd multiple of the X grid in the scaled area. |
| | ODD Y | Indicates that stylus or pen is at an odd multiple of the Y grid in the scaled area. |
| 3 | X LOCK Key | Used to lock and unlock stylus in the X axis. When X LOCK is on, the X LOCK indicator lights and the stylus cannot be moved in the X axis. Not used on Tablet Station and Tabletizer. |
| | Y LOCK Key | Used to lock and unlock stylus in the Y axis. When Y LOCK is on, the Y LOCK indicator lights and the stylus cannot be moved in the Y axis. Not used on Tablet Station and Tabletizer. |
| | ESC (Escape) | Used to terminate a DFUNCTION command. |
| 4 | Special Function Keys | |
| | F 40 - 45 | DFUNCTION Keys. |
| | HOLD | Holds the scroll on the QTY. |
| | CONT | Continues the scroll on the QTY. |
| | <CE/CR> | Enters a coordinate and does a carriage return. |
| | <CE> | Enters a coordinate. |
| | INT | Interrupts the program so that it is terminated. |
| | CTRL | Control Function key, usually used with other keyboard keys. |

| ID NO. | NAME | FUNCTION |
|-----------|---------------------------|--|
| 5 | Typical Keyboard Controls | These keys perform the same function as the keys on a standard typewriter, with the following exception: |
| | BS (Upper right) | The backspace key deletes entered characters one at a time to the system and on the QTY screen so that changes or corrections can be made. |

3.2.6 Design Station Digitizer Tables

There are two types of digitizer tables in the standard GDS II system: the 48 x 60" Digitizer Table with backlighting and a constrained stylus; and the 44 x 60" Talos Table or

"Tabletizer" with no backlighting and with a free stylus.

3.2.6.1 48 x 60" Digitizer Table

The controls and indicators of this Table are illustrated in Figures 3-8A and 3-8B and are described in Table 3-8.

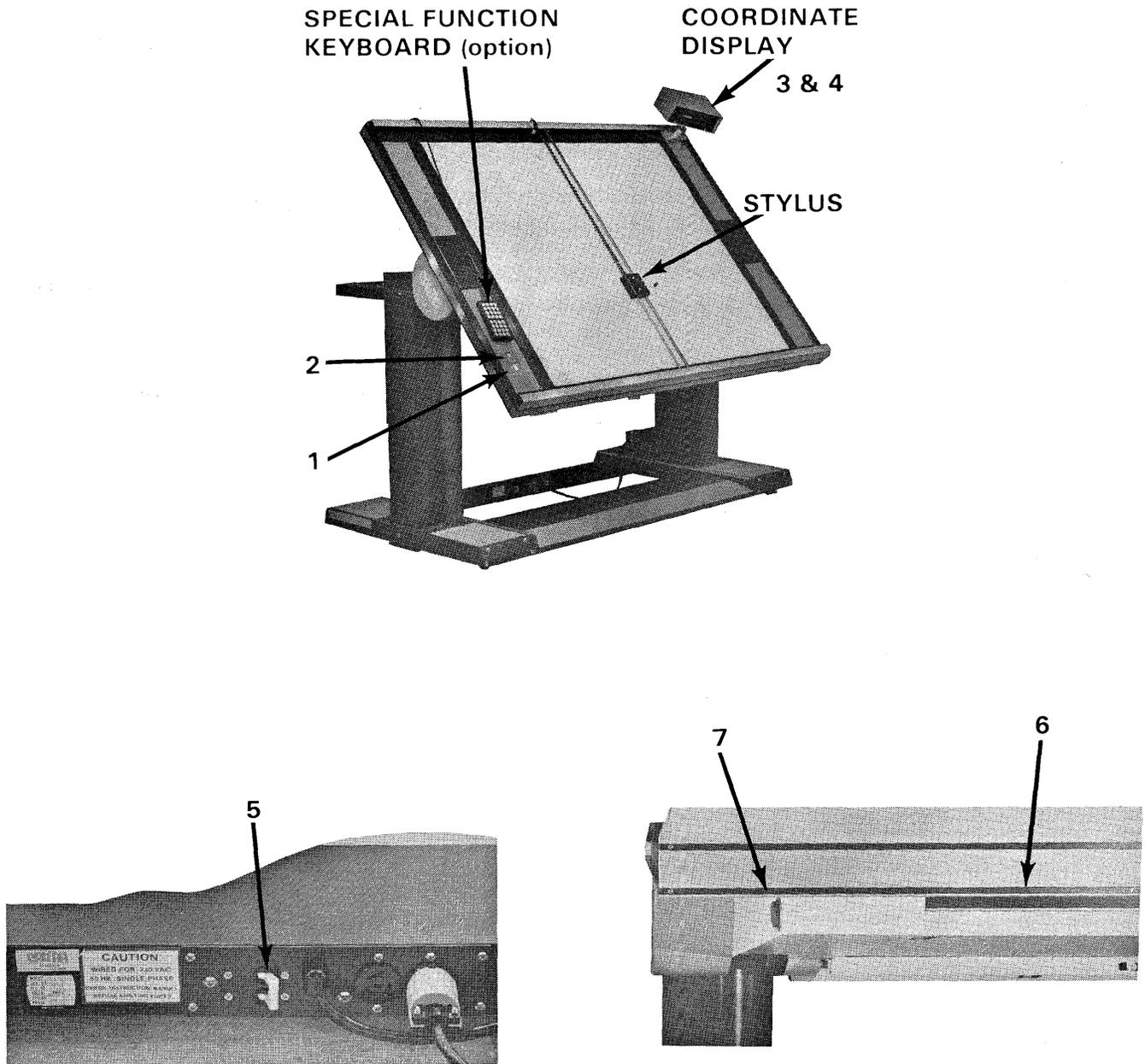


Figure 3-8. Digitizer Table Controls and Indicators

Table 3-8. Digitizer Table Controls and Indicators

| ID NO. | NAME | FUNCTION |
|-----------|-------------------|---|
| 1 | LAMP POWER Switch | This switch controls power to the table to turn backlighting on and off. |
| 2 | LAMP INTENSITY | This control increases or decreases the intensity of the backlighting. |
| 3 | X COORD | Not used in GDS II. |
| 4 | Y COORD | Not used in GDS II. |
| 5 | Circuit Breaker | This circuit breaker controls power to the table. It is normally left on so that the keyboard switch may be used to control power to the table. |
| 6 | Height Adjustment | This level is used to raise or lower the table and is located under the front edge of the table. |
| 7 | Tilt Control | This red button is used to control the tilt (angle) of the table from horizontal to 80 degrees. It is located at the side edge toward the front of the table. |

3.2.6.2 44" x 60" Talos Table or "Tabletizer"

The controls and indicators of the Talos Table are shown in Figure 3-9 and described in Table 3-9.

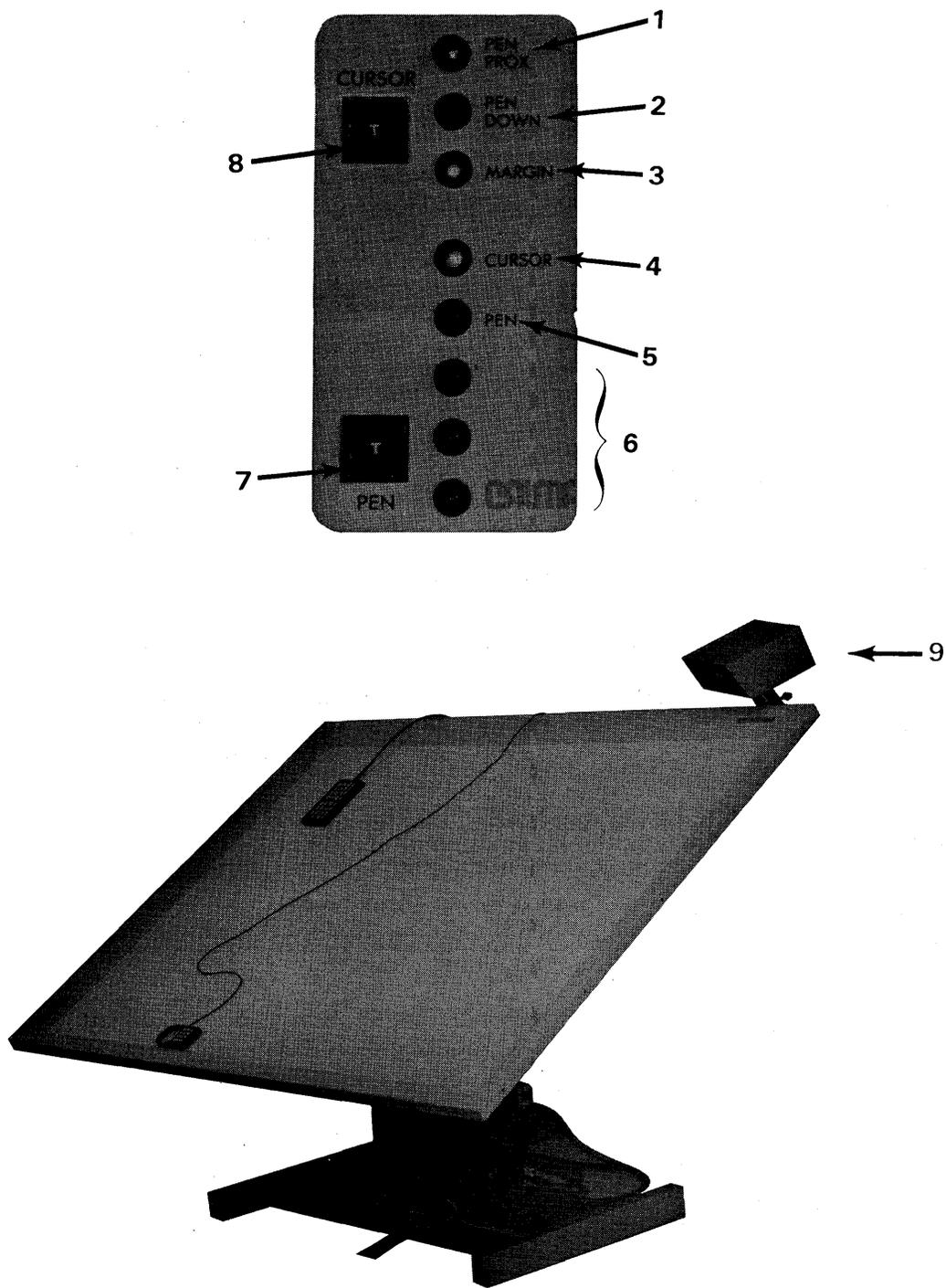


Figure 3-9. Talos Table Controls and Indicators

Table 3-9. Talos Table Controls and Indicators

| ID NO. | NAME | FUNCTION |
|--|------------------------|--|
| 1 | PEN PROX (Indicator) | Senses the proximity of the pen to the tablet surface. The pen must be within $\frac{1}{4}$ " distance from the surface in order for the Tabletizer to work. Indicator lights when proximity is correct. |
| 2 | PEN DOWN (Indicator) | Lights when a coordinate is being entered on the Tabletizer with the Puck or the Pen. |
| 3 | MARGIN (Indicator) | Lights when the Puck or Pen contracts the Tabletizer margin area, where data cannot be entered. |
| 4 | CURSOR (Indicator) | Lights when the Puck, or Free stylus (cursor) is selected for use. |
| 5 | PEN (Indicator) | Lights when the Pen is selected for use. |
| 6 | 3 Unlabeled Indicators | Not used in GDS II. |
| 7 | PEN (Pushbutton) | Selects the Pen for digitizing on the Tabletizer. |
| 8 | CURSOR (Pushbutton) | Selects the Puck, or Free stylus, for digitizing on the Tabletizer. |
| 9 | X-Y Coordinate Display | Not used in GDS II. |
| Tilt and Brake Lever (underneath the table at the right) | | Releases or sets tilt brake to hold table at desired tilt angle. To release brake, move lever to extreme right. To lock brake when table is at desired angle, move lever to extreme left. |
| Circuit Breaker (at the rear foot of the table) | | Turns power to the table on and off. |
| Foot Control (at base of table stand) | | Raises and lowers table. |

3.2.7 Digitizer Styluses

There are two programmable styluses available in the standard GDS II system: a 7-key constrained stylus, which is used with the 48 x 60" Digitizer Table ; and a 12-key free stylus, called a Puck, which is used with the 44 x 60" Talos

Digitizer Table, or "Tabletizer".

3.2.7.1 7-Key Programmable Stylus

The controls and indicators of the 7-key stylus are illustrated in Figure 3-10 and described in Table 3-10.

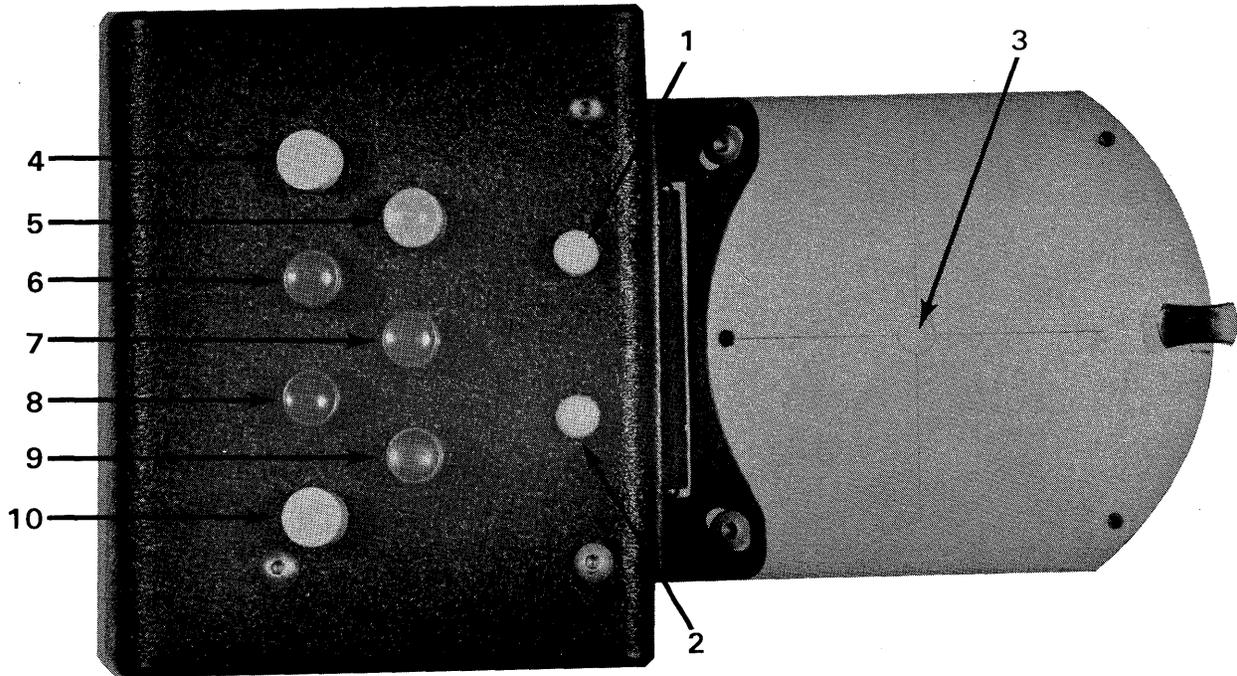


Figure 3-10. 7-Key Stylus Controls and Indicators

Table 3-10. 7-Key Programmable Stylus Controls and Indicators

| ID NO. | NAME | FUNCTION |
|--------|------------------|--|
| 1 | Yellow Indicator | Indicates that stylus is at an odd multiple of the X grid in the scaled area (corresponds to the Odd X LED on the Alphanumeric Key Board). |
| 2 | White Indicator | Indicates that stylus is at an odd multiple of the Y grid in the scaled area (corresponds to the Odd Y LED on the Alphanumeric Keyboard). |
| 3 | Crosshairs | Pinpoint location of coordinate entry. |
| 4 | White button | Coordinate Entry button; inputs a command or enters a digitized point located at the crosshairs. This button is equivalent to the <CE> button on the keyboard. |
| 5 | Blue button | Coordinate Entry and Carriage Return; inputs a command or enters a digitized point located at the crosshairs, and automatically does a carriage return on the QTY. This button is equivalent to the <CE/CR> key on the keyboard. |
| 6 | Red button | DFUNCTION button 35. |
| 7 | Green button | DFUNCTION button 36. |
| 8 | Black button | DFUNCTION button 37. |
| 9 | Amber button | DFUNCTION button 38. |
| 10 | Yellow button | Lock override button; releases both the X and Y lock so that the stylus can be moved freely on the Digitizer Table. |

3.2.7.2 12-Key Programmable Free Stylus or "Puck"

The Free Stylus or "Puck" is used with the 44 x 60" Talos Table (see Section 3.2.6.2). The controls and indicators on the Puck are shown in Figure 3-11 and described in Table 3-11.

Table 3-11. 12-Key "Puck" Controls and Indicators

To Be Supplied

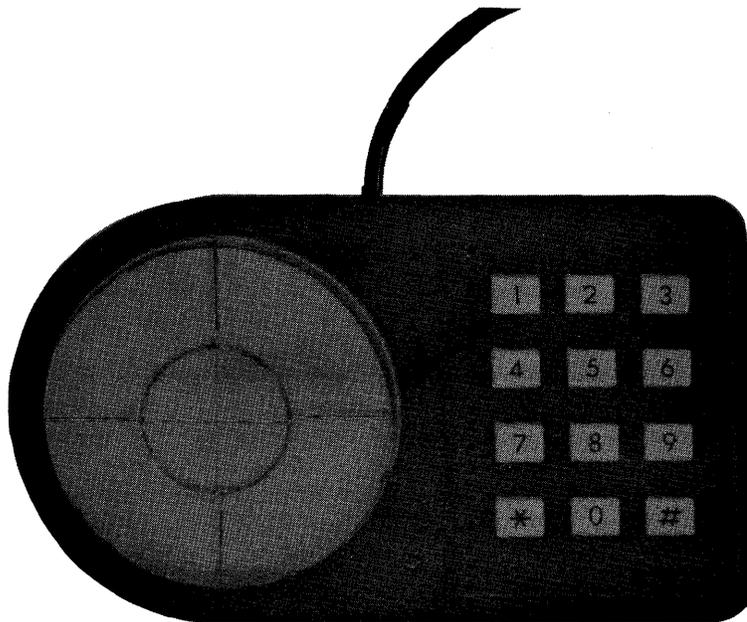


Figure 3-11. 12-Key "Puck" Controls and Indicators

3.2.8 Design Station Digitizer Tablet and Pen Stylus

An 11 x 11" Digitizer Tablet and Pen Stylus can be used instead of a digitizer table. The Tablet and Pen are shown in Figure 3-12, and its controls and indicators are described in Table 3-12.

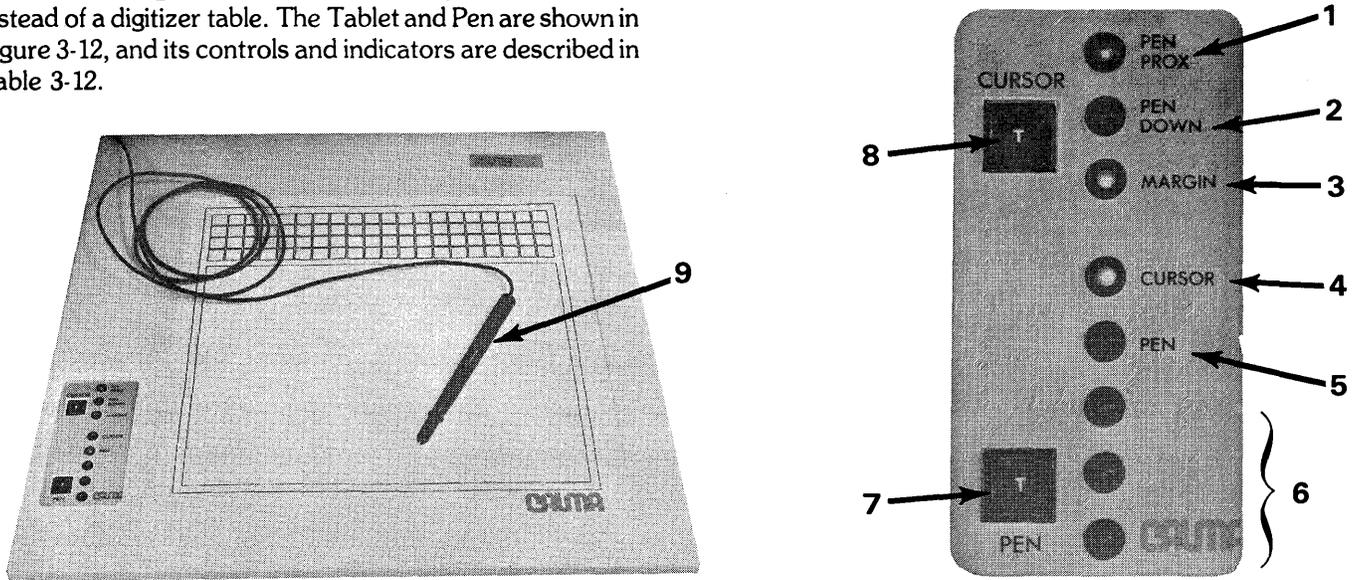


Figure 3-12. Digitizer Tablet Controls and Indicators

Table 3-12. Digitizer Tablet and Pen Stylus Controls and Indicators

| ID NO. | NAME | FUNCTION |
|--------|----------------------|---|
| 1 | PEN PROX (Indicator) | Senses the proximity of the pen to the Tablet surface. Pen must be within 1/4" proximity in order for the Tablet to work. Indicator lights when proximity is correct. |
| 2 | PEN DOWN (Indicator) | Lights when a coordinate is being entered on the Tablet. |
| 3 | MARGIN (Indicator) | Lights when the pen contacts the Tablet margin area, where data cannot be entered. |
| 4 | CURSOR (Indicator) | Lights when Cursor mode (an option) is in use. |
| 5 | PEN (Indicator) | Lights when the Pen is selected for use. |
| 6 | Unmarked Indicators | Not used in GDS II. |
| 7 | PEN (Pushbutton) | Selects the pen for digitizing on the Tablet. |
| 8 | CURSOR (Pushbutton) | Selects the Cursor mode (an option). |
| 9 | Pen Stylus | Digitizes coordinate on the Tablet when pressed onto the Tablet surface. |

3.2.9 Design Station Display Screens

The GDS II design stations each have two display screens. One is an erasable alphanumeric display (QTY) for user communications, and the other is a CRT for graphics.

3.2.9.1 QTY Screen

There are no controls and indicators on the QTY cabinet itself (see Figure 3-13). All controls and indicators are on the Alphanumeric Keyboard, which resides on the table surface just below the screen (see Section 3.2.5). This screen displays input from the operator, prompts for commands, reports status, and provides help files to aid in use of the system.

3.2.9.2 Interactive CRT

The 19" CRT has three controls on the cabinet itself, as shown in Figure 3-13 and described in Table 3-13.

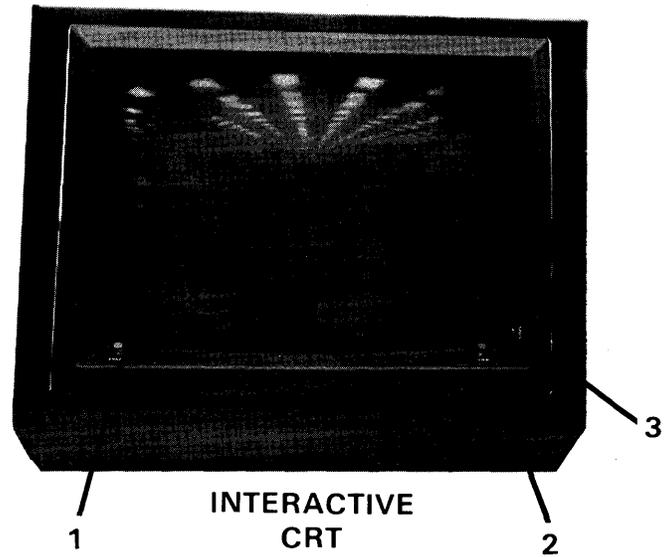


Figure 3-13. CRT Controls

Table 3-13. CRT Controls

| ID NO. | NAME | FUNCTION |
|--|------------------|--|
| 1 | ERASE Pushbutton | Erases the data displayed on the screen; the data stored in memory, both core and disk, is not affected. |
| 2 | VIEW Pushbutton | Causes the geometry to be displayed again after it has been automatically dimmed. |
| NOTE | | |
| (1) The view displayed on the CRT is automatically dimmed after 90 seconds of inactivity. | | |
| (2) The CRT is automatically erased after 4 minutes of inactivity. The geometry can be restored with a REDRAW command from the keyboard. | | |
| 3 | COPY Pushbutton | Used with hard copy unit (optional). |

3.2.10 X-Y Coordinate Displays

X-Y Coordinate displays are part of the Digitizer Tables and the Design Stations, but they are not used in the GDS II system. For display of X and Y coordinates, see the top of the QTY screen where they are displayed once the system is initialized.

3.3 GDS II System Power-On and Power-Off Procedures

The following procedures power on and power off the GDS II graphics design system. Refer to Subsection 3.2 for description and illustration of the controls and indicators for each component in the system.

3.3.1 Applying Power to the System

First, verify that all system component circuit breakers are on:

- a. CPU
- b. Disk Drive
- c. Tape Drive
- d. Design Station

To apply power to the system, depress the main power switch located on the upper right hand corner of the CPU cabinet. This switch is a white pushbutton which is illuminated when the power is ON. Provided that no other switches have been turned off, and the operating system has already been installed on disk, the system may be started up as described in Subsection 3.3.11.

Prior to turning off the system, verify that the operating system has been shut down as described in Subsection 3.3.11. Turn power off by depressing the main power switch.

CAUTION

Failure to shut down the operating system before turning the system off may result in the loss of critical data files stored on disk.

WARNING

The main power switch DOES NOT remove all power from all system components. The location of auxilliary power switches is described below.

Auxilliary power switches are:

1. **Computer power.** The ECLIPSE computer has its own power switches located on the right hand side of the front panel. This switch is normally left in the ON position when the system is shut down by the main power switch. In case of difficulty, verify that the POWER key is turned to ON and the ADDRESS COMPARE switch is OFF. Moving the POWER switch to the LOCK position disables all other front panel switches except the MONITOR function of the ADDRESS COMPARE switch. Therefore, the POWER switch must be ON in order to initialize the system. Moving the ADDRESS COMPARE switch to STOP/STORE or STOP/ADDR enables a debugging feature that makes normal system operation impossible.

The presence of the power at the CPU may be verified as follows:

- A. Turn the POWER switch ON.
- B. Turn the ADDRESS COMPARE switch OFF.
- C. Move the RESET/STOP switch up momentarily.
- D. Set the 16 data switches up.
- E. Move the EXAM/EXAM NXT switch up momentarily.
- F. Verify that the 16 data indicators are lit.

If no indicators are lit, it is likely that there is no power at the CPU. If a few data lights remain off, it is likely that a bulb has burned out and normal system operation will not be affected.

2. **Disk Power.** The disk drives contain power supplies that are controlled by switches located inside the drive cabinet. These switches are normally left in the ON position when the main power switch is turned off.

CDC Drives. Two switches are located inside the back panel of CDC disk drives. BOTH switches must be OFF to remove all power from the drive. BOTH switches must be turned ON for normal system operation. These switches may be turned off or on in either order.

Century Drives. A switch is located inside the front cover of Century drives. The switch must be OFF to remove all power from the drive and ON for normal system operations.

NOTE

If two Century drives are connected to a system, the power supplies in BOTH drives must be ON for the system to operate. This is not necessary with the CDC drives, which can run independently.

The presence of power at a disk drive and the correct setting of internal power supply switches can be verified if the disk drive blowers start up when the main power switch is turned ON.

3. **Magnetic Tape Drive Power.** Each magnetic tape drive has a power switch located on the front panel. The power switches are normally left ON when the main power switch is turned OFF. While the placement of the magnetic tape drive power switch depends on the model, the power switch is clearly labeled in all cases.
4. **System Console Power.** The power switch on the system console, a DECWRITER II, is to the left of the keyboard. The switch is normally left on when the main power is turned off.
5. **Design Station Power.** The master power switch for each station is a thumbwheel switch located on the upper right hand side of the keyboard. This

switch is normally left ON when the main power switch is turned OFF. This is a dual purpose switch which also controls the volume of certain tone generators built into the station. However, GDS II does not presently support this tone. This switch applies power to all station components including the graphics CRT, the alphanumeric QTY, the coordinate displays, and the tablet or digitizer. For stations equipped with a Tektronix 2 graphic CRT, an additional power switch may be found by opening the front panel.

3.3.2 Loading and Unloading the Century Disk Pack

Loading the Pack

To load the disk pack into the disk drive, proceed as follows:

1. Invert the disk pack and rotate the handle counter-clockwise until it stops. See Figure 3-14.
2. Remove the bottom cover.
3. Open the disk drive access door.
4. Insert the disk pack into position over the disk spindle.

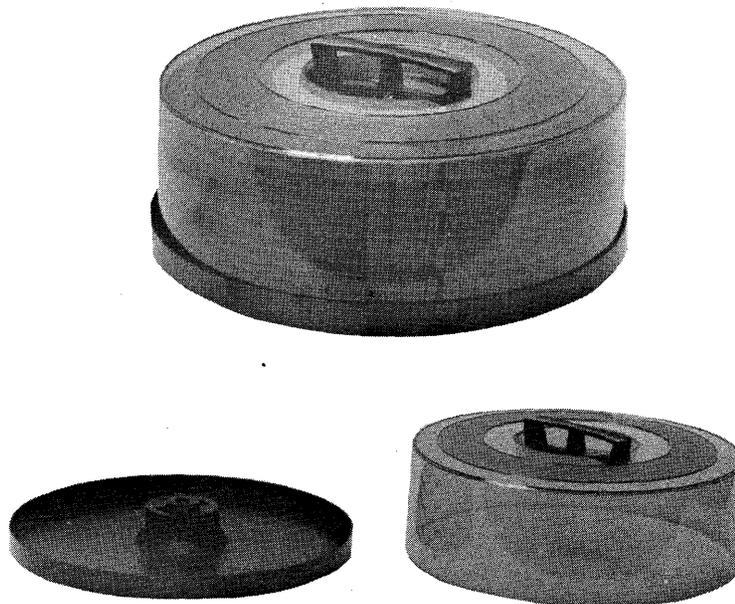


Figure 3-14. Disk Pack

5. Rotate the handle clockwise until it stops; then lift the top cover from the disk pack. The disk pack is now loaded.
6. Close the disk drive access door.
7. Depress the POWER ON pushbutton.
8. Allow approximately 14 seconds before proceeding. (Audible clicks may be heard as the read/write elements (heads) are positioned.) The 0 (green indicator) pushbutton will light in a steady state.
9. Place the top and bottom covers together and store in a dust-proof environment.

Unloading the Pack

To remove the disk pack from the disk drive unit, proceed as follows:

1. Verify that the disk has been released by the operating system. (Release disk from system by typing "Release DC0" on the system console. Refer to Section 3.3.12, No. 33.)
2. Depress POWER ON pushbutton.
3. Verify that the 0 (green lamp) and the POWER ON lamps are off.

NOTE

This action occurs in approximately 14 seconds after depressing the POWER ON pushbutton.

4. Open the access door.

NOTE

The door should not be opened until the disk has stopped rotating.

5. Place the top disk cover over the disk pack and rotate handle counterclockwise until it clicks and turns freely.
6. Lift cover and disk pack up from drive cavity.
7. Place disk pack over bottom cover and rotate the handle clockwise until it stops.
8. Close disk drive access door.

9. Store disk pack in the appropriate area.

3.3.3 Loading and Unloading a CDC Disk Pack

Loading the Disk Pack

To load the disk pack into the CDC disk drive, proceed as follows:

1. Invert the disk pack and squeeze the clamps at the center of the bottom cover.
2. Remove the bottom cover.
3. Open the disk drive access door.
4. Insert the disk pack into position over the disk spindle.
5. Rotate the handle of the top cover clockwise until it is tight. Then lift the top cover from the disk pack. The disk pack is now loaded.
6. Close the disk drive access door.
7. Press the START button on the drive control. The READY light will flash until the drive is up to speed. Then the READY indicator will light in a steady state.
8. Place the top and bottom covers together and store in a dust-proof environment.

Unloading the Disk Pack

To remove the disk pack from the CDC drive, proceed as follows:

1. Release the disk from the system by typing "RELEASE BD0" on the system console (DECWRITER II).
2. Press the START button on the disk drive to turn the drive off. The READY light will flash until the disk stops spinning; then the light will go out.
3. Open the disk drive access door.
4. Place the top cover over the disk pack, and rotate the handle counterclockwise until it clicks.
5. Lift the cover and disk pack up from the drive cavity.

6. Place the disk pack over the bottom cover by squeezing and then releasing the clamps so that the cover locks onto the pack.
7. Store the disk pack in an appropriate place.

3.3.4 Loading and Unloading the Wangco Magnetic Tape Drive

Loading the Tape Reel

1. Depress POWER ON pushbutton on the Tape Drive if the drive has been turned off. POWER ON pushbutton should light.
2. Open the front door of the Tape Drive.

3. Remove Tape seal from around the reel. (See Figure 3-15A.) If the reel is to be read only, remove the Write Enable ring. The FILE PROTECT button will light to indicate that the tape is write protected. If the tape is to be written, install a Write Enable ring if necessary. (The Write Enable ring should be placed in a groove in the back of the tape reel. See Figure 3-15B.)

CAUTION

Pressure on the outer reel flange can deform the tape edges and dislodge oxide from the tape. The dislodged oxide and deformed tape edges can lead to read/write errors.

4. Press on end of the hold down marked PRESS.

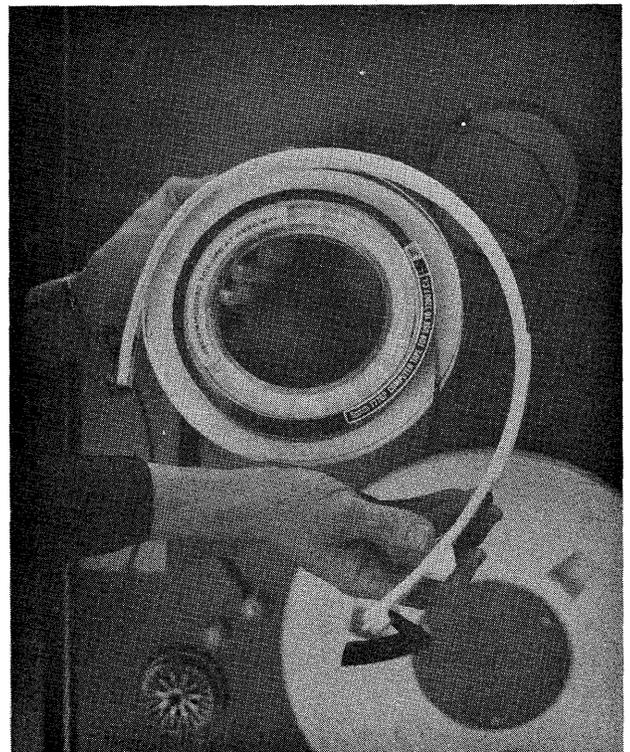


Figure 3-15A. Removing Tape Seal from Reel

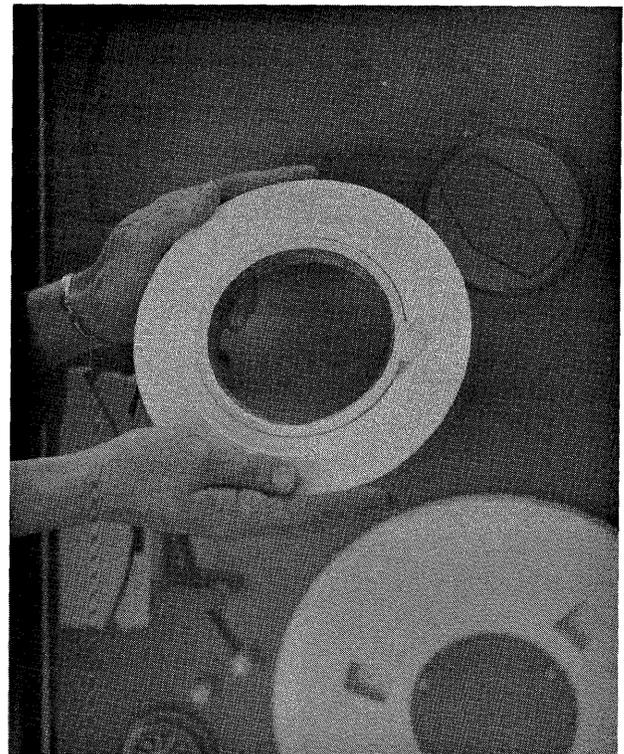


Figure 3-15B. Location of Write Enable Ring

5. Guide tape reel onto the hold down ring and gently press into place. Press hold down until it is flush with the supply knob locking the tape reel into place. (See Figure 3-16.)
6. Thread the tape in accordance with the tape loading diagram shown in Figure 3-17.
7. Place finger through one of the openings in the take up reel, press end of tape against the take up reel hub, and rotate the hub until the end of the tape is overlapped and held securely by the next layer of tape.
8. Manually rotate the takeup reel hub until five layers of tape are wound onto the hub.
9. Depress LOAD pushbutton. The buffer arms automatically move to their normal operating position and tape is wound onto the takeup reel hub until the BOT (beginning of tape) marker is reached.
10. Close front door.

NOTE

Be sure the online button is lit, or the tape drive cannot be INIT'ed (addressed).

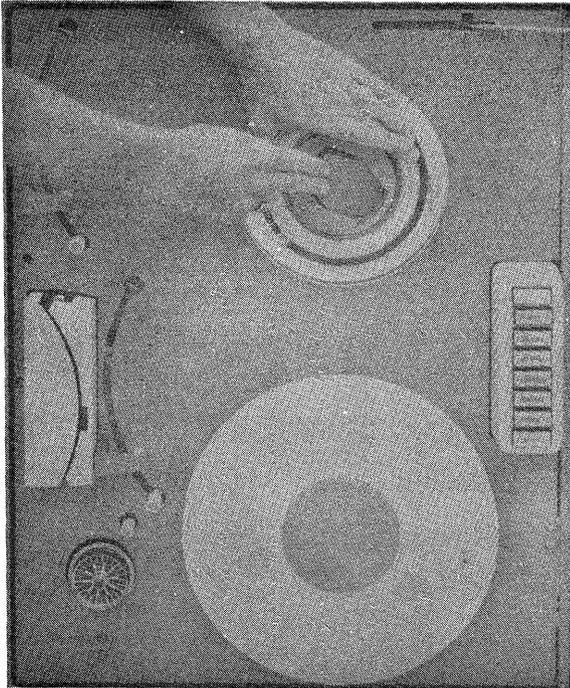


Figure 3-16. Loading the Supply Reel

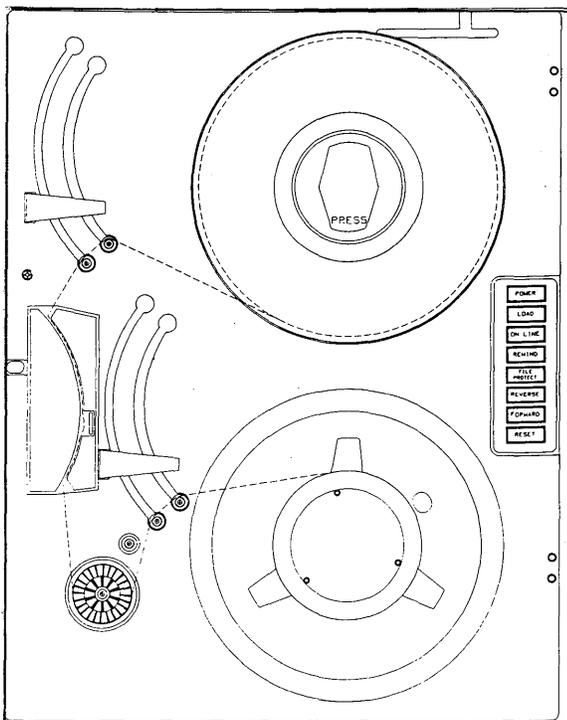


Figure 3-17. Wangco Tape Loading Diagram

Unloading the Tape Reel from the Wangco Tape Drive

To unload the magnetic tape from the tape drive, proceed as follows:

1. Verify that the magnetic tape drive has been released by the operating system. (To use magnetic tape in RDOS, the tape must be INIT'ed before any data can be transferred and RELEASED when all data transfers are complete.)
2. Depress RESET pushbutton. This takes the system off line. The on line light should go out.
3. Press REWIND pushbutton. (If tape is not at the BOT load point, the tape will be rewound. Press REWIND again.) Tape is pulled completely through and returned to supply reel.
4. Open front door.
5. Press on the end of the supply reel marked PRESS.
6. Ease the tape reel from the supply hub.
7. Place tape seal around the supply reel.
8. Close front door.
9. Label the tape and remove the Write Enable ring to ensure that the tape is not accidentally written over.
10. Store the tape in the designated area.

3.3.5 Loading and Unloading the Cipher Magnetic Tape Drive

Loading the Tape Reel

(Figures illustrating loading procedures will be supplied at a later date.)

1. Verify that the ON/OFF toggle switch on the lower left corner of the drive is turned ON.
2. Open the front door of the tape drive.
3. On the tape reel, remove the tape seal from around the reel. If the reel is to be read only, remove the Write Enable ring; if writing is to be done, retain or install the ring in the groove located on the back of the tape reel.

The disk is initialized and booted from a program located on a mag tape that is supplied with the system. The following procedure describes how the magnetic tape files **TBOOT.SV** and **DKINIT.SV** are used to start and to initialize the disk. Unused or crashed disks should be fully initialized before loading an RDOS/GDS II system. Partial initialization is also available to check for bad disk blocks (one block contains 1024₁₀ words) and is shown at the end of this procedure. This procedure is applicable to a CENTURY or CDC disk drive. However, the CENTURY disk drive is used in the examples. See Figure 3-18 for a copy of a typical disk initializer dialog.

1. Turn on system power and ensure that all units of the system are receiving power (see Subsection 3.3.1).
2. Lift and release the computer RESET switch to halt the computer.
3. Install a disk pack on disk drive unit 0 to receive **TBOOT.SV** and **DKINIT.SV** from tape (see Subsections 3.3.2 or 3.3.3), turn on the disk drive, and wait for the disk to get up to speed.
4. Mount the tape labelled RDOS BOOTTAPE (see Subsection 3.3.4 or 3.3.5). Be sure the Write Enable ring is removed.
5. Set the tape drive for power on, load, and on line. If the system has a TAPE DENSITY switch (the two densities are 800 and 1600 BPI), the switch setting must match the tape being used.
6. Install the bootstrap from tape (see Subsection 3.3.6, setting switches 0, 11, and 14 only).
7. The system reads **TBOOT.SV** file from magnetic tape, executes the **TBOOT.SV** program, which enables **DKINIT** to be read in and executed, and then prints MT0:.

NOTE

All responses to **MT0** refer to the list in Step 9. For example, a 3 response calls the tape file with the **DKINIT.SV** Program. Different tapes have different tape file numbers for this program so the user must refer to the list supplied with the applicable RDOS BOOTTAPE tape.

8. Type the appropriate response for the **DKINIT.SV** program (on the same line as the **MT0**: print out) and then depress the RETURN key. A typical list of responses is given below.

| MAG TAPE UNIT:FILE | PROGRAM NAME | DESCRIPTION |
|--------------------|-------------------|-------------------------------------|
| MT0:0 | TBOOT.SV | Tape Bootstrap Program |
| MT0:1 | CLI | CLI, BOOT, and LOGON in a dump file |
| MT0:2 | BOOT.SV | RDOS Disk Bootstrap |
| MT0:3 | DKINIT.SV | RDOS Disk Initializer |
| MT0:4 | 24KNOVA.SV | 24K Century Disk System |
| MT0:6 | ECLIPSE.SV | Mapped Eclipse Century System |

9. The system then reads the selected tape file and prints:

CALMA DISK INITIALIZER - REV xx.xx
DISK DRIVE MODEL NUMBER?

Here xx.xx indicates the revision level.

Type **CENTURY** or **CDC** (depending on type of disk drive) and press the RETURN key.

10. The system then prints:

CENTURY 114 DISKPACK DRIVE TYPE
(for Century disks)
DISK UNIT?

NOTE

Typing an illegal response to any **DKINIT.SV** query causes an error message or a repeat query to print out.

Type **DC0** for a Century disk unit or **BD0** for a CDC disk unit and press the RETURN key. (**DC1** or **BD1** may also be used on systems equipped with multiple disk drives.)

- 10A. The system then prints:

ENTER INTERLEAVE FACTOR (<CR> FOR 1)?

Press the CR key to enter this interleave factor.

11. The system then prints:

COMMAND?

Type **FULL** for full disk initialization and press the RETURN key.

FROM MT0:3

CALMA DISK INITIALIZER - REV 03.00

DISK DRIVE MODEL NUMBER? CENTURY

CENTURY 114 DISKPACK DRIVE TYPE

DISK UNIT? **DC0**

ENTER INTERLEAVE FACTOR (<CR> FOR 1)? **CR**

COMMAND? **FULL**

COMMAND DESTROYS ANY PREVIOUS RDOS DISK STRUCTURE
RDOS INIT/F MUST BE DONE ON DISK AFTER COMMAND
TYPE CONTROL-A NOW TO ABORT WITHOUT LOSS

NUMBER OF PATTERNS TO RUN (1-5) ? **2**

*** PATTERN # 1 (125252) ***

*** PATTERN # 2 (052525) ***

*** ALL PATTERNS RUN ***

DO YOU WISH TO DECLARE ANY BLOCKS BAD
THAT ARE NOT ALREADY IN THE BAD BLOCK TABLE?

ANSWER YES OR NO

DO YOU WISH TO DECLARE ANY BLOCKS BAD
THAT ARE NOT ALREADY IN THE BAD BLOCK TABLE? **NO**

DEFAULT REMAP AREA SIZE IS 12 BLOCK(S) LONG
IT NEEDS TO BE AT LEAST 0 BLOCK(S) LONG

REMAP AREA SIZE (TYPE RETURN FOR DEFAULT) ? **CR**

REMAP AREA START BLOCK NUMBER (TYPE RETURN FOR DEFAULT) ? **CR**

DEFAULT FRAME SIZE IS 83,
MIN IS 1, AND MAX IS 4060

DISK FRAME SIZE (TYPE RETURN FOR DEFAULT) ? **CR**

FULL DISK INIT COMPLETE

COMMAND? **STOP**

NOTE

Operator responses are shown in bold and must be followed by pressing the RETURN key.

The **CR** indicates the RETURN key was pressed to accept the default parameters.

(Carriage return was wrong response.)

Figure 3-18. Typical Disk Initializer Dialog

12. The system then prints:

COMMAND DESTROYS ANY PREVIOUS
RDOS DISK STRUCTURE
RDOS INIT/F MUST BE DONE ON DISK
AFTER COMMAND
TYPE CONTROL-A NOW TO ABORT
WITHOUT LOSS

NUMBER OF PATTERNS TO RUN (1-5)?

NOTE

This command checks the disk quality by writing and then reading number patterns over the entire disk. This destroys any data that previously existed on the disk. To save the data, depress the CONTROL and A keys at the same time. Then change the disk and restart the initialization. (Pressing CTRL and A keys starts the program over from the beginning.)

Type a number from 1 thru 5 and press the RETURN key to select the number of patterns to be run. A 1 response runs only the first pattern, a 2 response runs the first two patterns, etc.

13. On a Century drive each pattern requires approximately 6 minutes running time which starts when the system prints:

PATTERN #x (yyyyyy)

(On an 80 megabyte CDC drive, 10 minutes are required; on a 300 megabyte CDC drive, 40 minutes.)

Here x is the pattern number being run and yyyyyy is the pattern in octal:

| PATTERN# | PATTERN |
|----------|---------|
| 1 | 125252 |
| 2 | 052525 |
| 3 | 155555 |
| 4 | 177777 |
| 5 | 000000 |

14. When the pattern(s) has been run, the system prints one of the following messages:

| MESSAGES | ACTION REQUIRED |
|--|--|
| ILLEGAL NUMBER OF PATTERNS | Type a response between 1 and 5. |
| CRITICAL DISK BLOCKS ARE BAD RDOS CANNOT BE BUILT, ABORTING TOO MANY DISK ERRORS TO COMPLETE | Reformat the disk using Data General disk formatter program Part No. 095-000071 or 095-000241 as applicable. |
| ***ALL PATTERNS RUN*** | Normal return after all selected patterns are run. |

DO YOU WISH TO DECLARE ANY
BLOCKS BAD THAT ARE NOT
ALREADY IN THE BAD BLOCK
TABLE?

Type NO and press the RETURN key to advance the program.

15. If the RETURN key is pressed before a response is typed, the system prints:

ANSWER YES OR NO and repeats the previous message.

When YES is typed, the system prints:

BAD BLOCK NUMBER xxxx

Here xxxx represents the number in octal.

Then, type the number of the bad block and press the RETURN key. The system then prints:

BAD BLOCK ENTERED xxxx
BAD BLOCK NUMBER zzzz

Here xxxx and zzzz represent the numbers, in octal.

This exchange continues for each block number entered. To exit from this routine, press the RETURN key. The system then prints:

DO YOU WISH TO DECLARE ANY MORE
BLOCKS BAD?

If YES and the RETURN key is pressed, the routine continues as before.

16. When NO is typed and the RETURN key is pressed, the program advances, and the system prints:

DEFAULT REMAP AREA SIZE IS xx
BLOCK(S) LONG
IT NEEDS TO BE AT LEAST yy
BLOCK(S) LONG

REMAP AREA SIZE (TYPE RETURN FOR
DEFAULT)?

Here xx and yy vary depending on the type of system disk in use.

Press the RETURN key to select the default remap parameters and advance the program. To change the remap area size, type the new length and press the RETURN key.

17. The system then prints:

REMAP AREA START BLOCK NUMBER
(TYPE RETURN FOR DEFAULT)?

Make one of the following three responses:

- Press the RETURN key to select the default location.
- Type the physical block number (octal) and then press the RETURN key.
- Type the head, sector, and cylinder locations (separated by commas) and then press the RETURN key.

18. After the return key is pressed, the system prints one of the following messages. Determine which message is printed and perform the action required.

MESSAGES

THERE IS NO CURRENT REMAP
AREA, RUN FULL INIT TO
ESTABLISH ONE **COMMAND**?

ILLEGAL DISK BLOCK NUMBER

RDOS WILL NOT RUN WITH THIS
BAD BLOCK, BLOCK NOT
ACCEPTED

NO MORE ROOM FOR BAD
BLOCKS IN REMAP AREA DUE
TO TOO MANY BAC BLOCKS-
ABORTING

BAD BLOCK LIST IS FULL,
UNABLE TO ENTER ANY MORE
BAD BLOCKS

ILLEGAL REMAP AREA START
BLOCK NUMBER

BAD BLOCK CONTAINED IN
REMAP AREA SPECIFIED, PLEASE
SPECIFY ANOTHER AREA

DEFAULT FRAME SIZE IS 83
MIN IS 1, MAX IS 4060

ACTION REQUIRED

Type **FULL** is response to
COMMAND?

Retype entry for remap area
start block number.

Reformat disk.

Remap again with larger remap
area.

Select another area for remap.

Select an area with contiguous
good blocks.

Normal program advance.
Printout indicates the default
frame size and limits. Default
and maximum size vary with
disk type.

19. The system then prints:

DISK FRAME SIZE (TYPE RETURN FOR
DEFAULT)?

Press the RETURN key to accept the default frame size and limits. Frame size determines the number of disk blocks assigned to the RDOS system directory (**SYS.DR**). RDOS runs more efficiently when all file names for a directory fit within one frame.

To select a different frame size than the default parameters, type any number from 1 thru 4060 (disk blocks) and then press the RETURN key.

20. If the default parameters were not selected and the number typed was not from 1 thru 4060, the system prints:

ILLEGAL FRAME SIZE

and a different selection must be typed.

When a legal selection is made, the system indicates the disk is fully initialized by printing:

FULL DISK INIT COMPLETE
COMMAND?

Type **STOP** and press the **RETURN** key to terminate the **DKINIT** routine.

21. The system then does a carriage return and halts the computer. In addition to **STOP**, several alternate **DKINIT** commands can be typed in response to **COMMAND?** These commands are described in the following list.

| ALTERNATE COMMANDS | DESCRIPTION |
|-----------------------|--|
| DISK | Restarts the DKINIT program at the beginning (Step 10). |
| ENTER | Enters bad blocks in the bad block accounting table. |
| FRAME | Allows selection of a different frame size. The message "COMMAND DESTROYS etc." prints out but is automatically by-passed without user response. |
| LIST | Lists the disk type and block accounting information. |
| PARTIAL | Provides partial disk initialization by checking for bad blocks without over writing existing data. |
| REMAP | Allows selection of a different remap area. The message "COMMAND DESTROYS etc." prints out but is automatically by-passed without user response. |

At this point, a backup tape can be loaded on disk if no bad blocks have been detected.

3.3.8 Creating and Loading a BACKUP Tape

A BACKUP tape is a magnetic tape containing an RDOS disk image. Whenever the disk contains information that is difficult or time consuming to recreate, a BACKUP tape should be made. This BACKUP tape is then loaded to reestablish system operation in the event of a head crash or other disk failure. Separate procedures are provided for creating and loading BACKUP tapes.

Creating a BACKUP Tape

A BACKUP magnetic tape is created any time the disk holds data that is critical to operation of the system. This procedure creates a complete copy of the disk sector-by-sector without interpretation or validation of the contents. Any working disk can be dumped onto a magnetic tape to create a BACKUP tape. A typical dialog is shown in Figure 3-19.

```

FILENAME? ECLIPSE

MAPPED ECLIPSE MULTIGROUND RDOS REV 6.01.03
DATE (M/D/Y) ? 5 11 78
TIME (H:M:S) ? 7 51 00

CALMA REV6 CLI
R
INIT MT0
R
BACKUP
CALMA RDOS CENTURY BACKUP 11FEB78
BACKUP OF CENTURY MADE 07:51:53 05/11/78
R
RELEASE DC0

MASTER DEVICE RELEASED

NOTE

Operator responses are shown in
bold and must be followed by
pressing the RETURN key.

```

Figure 3-19. Typical Dialog to Create a BACKUP Tape

NOTE

This procedure assumes that only one CLI program is running. The first three steps below establish the necessary conditions.

1. Type **GDIR** and press the RETURN key to get the name of the current directory.

2. The system prints:

```
DCO (or BDO for CDC disk)  
R
```

If **DCO** is not the current directory, type **DIR DCO** (or **DIR BDO** for CDC disk) and press the RETURN key.

3. Continue when the system prints an R. Type **DIRECTORY** and press the RETURN key to get the names of all directories that have been initialized. For each directory except the master disk directory, type **RELEASE**, one or more spaces, the directory name, and a RETURN.

NOTE

Releasing a directory which has been created as a partition automatically releases any sub-directories.

4. Type **SYSTAT** and press the RETURN key to get the names of all programs running in all grounds. For each ground running the CLI, type **POP** and RETURN at the appropriate console. For each ground running a user program, either terminate the program or, as a last resort, abort it by using **ABGROUND**.

Type **BACKUP** and press the RETURN key to copy the disk image onto magnetic tape.

5. The system prints the following message and then performs the copy operation. The amount of time required to copy the disk depends upon the amount of information on the disk.

```
CALMA RDOS CENTURY BACKUP 11FEB78  
BACKUP OF CENTURY MADE 08:12:00 05/03/78  
R
```

The name of the disk (**CENTURY**) and dates and time vary.

6. When the copy operation is complete, the system prints an R and automatically rewinds the tape.

Continue with programming operations or boot **GDS II RELEASE DCO** (or **BDO** for CDC disks) and press the RETURN key to release the master device.

7. When the master device is released, the system halts the computer after printing:

```
MASTER DEVICE RELEASED
```

Remove the **BACKUP** tape from the tape drive after pressing the **RESET** and **REWIND** switches, label the tape with the information in Step 5, remove the Write-Enable ring, and store the tape in a safe place.

Loading a BACKUP Tape

A **BACKUP** tape created with the procedures described above can be loaded onto disk any time the system is up and running and an initialized disk is available. However, loading the **BACKUP** tape destroys any data that was previously on the disk. If a disk crash has occurred or other failure that may have damaged the disk, the disk initializing procedures should be performed (refer to Subsections 3.3.6 and 3.3.7).

All **BACKUP** tapes contain the disk bootstrap and an image of the disk at the time that tape was made. Depending on the amount of data that was stored, a **BACKUP** tape may consist of more than one physical tape. When more than one tape is loaded, the number 1 tape containing the disk bootstrap is loaded first and remaining tapes are loaded sequentially using the same procedure. A typical dialog for loading a **BACKUP** tape is shown in Figure 3-20.

To load the **BACKUP** tape on an initialized disk, proceed as follows:

1. Install the disk pack that will receive data from the **BACKUP** tape on disk drive unit 0. Turn on the disk drive and wait for the disk to get up to speed.
2. Mount and thread the **BACKUP** tape on tape drive unit 0 and set the tape drive for power on, load, and on line.

3. Boot the tape (100022a). The system reads a special program from tape, which identifies the date the BACKUP tape was created:

BACKUP OF CENTURY MADE 07:52:02 05/12/78

Then the tape is used to copy the BACKUP data to the disk. Mount additional reels when the system prompts for them.

4. Boot the desired RDOS system from the disk (proceed to Subsection 3.3.10).

NOTE

After loading a BACKUP tape, the tape is automatically released.

NOTE

The first time RDOS is booted from a disk created by loading a backup tape, the PARTITION IN USE error message is normal.

5. The system responds to the boot by reading the BACKUP tape and printing:

FILENAME?

Then the system automatically rewinds the tape.

```
BACKUP OF CENTURY MADE 07:50:02 05/12/78

FILENAME? ECLIPSE

PARTITION IN USE - TYPE C TO CONTINUE C
CONTINUE

MAPPED ECLIPSE MULTIGROUND RDOS REV 6.01.03
DATE (M/D/Y) ? 5 12 78
TIME (H:M:S) ? 8 02 10

CALMA REV6 CLI

USE COUNTS INCORRECT - ALL DIRECTORIES BEING CLEARED
SYSGEN.DR CLEARED

DC0 CLEARED

R

RELEASE DC0

MASTER DEVICE RELEASED

NOTE
Operator responses are shown in
bold and all responses except C are
followed by pressing the RETURN
key. Dialog shows automatic
clearing of use counts.
```

Figure 3-20. Typical Dialog to Load BACKUP Tape

3.3.9 Reformatting a Disk Pack

In the event that RDOS has crashed or the system has not been properly shut down, reinitializing the RDOS system will cause all directories to be cleared as shown in Subsection 3.3.10. In some cases, however, the disk data file that defines what disk sectors are available is partially destroyed. This condition is indicated either by a MAP.DR error or, in severe cases, by random failures of standard system utilities and application programs (caused by one or more sectors of the file being overwritten). If such an event should ever occur, it is necessary to reformat the pack as soon as possible to prevent the loss of additional data.

It is also recommended that the pack be reformatted on a periodic basis as an alternative to the faster backup procedure since this will disclose any undetected MAP.DR errors. Reformatting on a weekly basis is suggested.

The basic procedure for reformatting a pack is to execute the AFDUMP system utility while working in the master directory. Once all data has been written onto tape, delete all files except the CLI, the operating system, and AFLOAD, and reload the data. It is strongly recommended, however, that a second pack be used to reload the data as a precaution against magnetic tape errors or tape drive failures.

3.3.10 RDOS Initialization and Shut Down

As a general rule, several versions of RDOS systems are available on disk. When RDOS is initialized as described below, the name of the operating system version to be used must be given if a default name has not previously been assigned. Two of these standard systems are BDCNOVA and 32KNOVA. The system named BDCNOVA will run on any Data General computer that has at least 32K of core memory and a Century disk. Once either of these systems is booted, it becomes possible to use the standard CLI utilities to discover what other systems are available.

An RDOS system must have two files located in the master disk directory, a save file and an overlay file. For a system named USER, these file names would be USER.SV and USER.OL. Most systems have an initializer file, in this case USER.IN, which contains a sequence of CLI commands that are automatically executed whenever the system is booted. A default system can be selected by linking SYS.SV and SYS.OL to the respective system files desired.

A custom RDOS system is supplied to each user. By convention, the default name SYS is always linked to the custom system name. For GDS II systems, an initializer file is supplied which causes the GDS II system to be initialized automatically whenever RDOS is initialized. There are cases, however, where some other system is desired.

Two more standard systems that are supplied to all users with a Century disk are ECLIPSE and ZECLIPSE. These systems exist because some system utilities, AFDUMP and AFLOAD in particular, can be executed only on ECLIPSE computers. The difference between the two systems is that ECLIPSE will run only on S/200 model computers, while ZECLIPSE will run only on S/230 models. If the wrong system is chosen, the system will identify itself and then hang.

RDOS Initialization Procedure

See Subsection 3.3.6, System Bootstrap Procedure, Steps 1-6, for instructions on how to begin initializing the system.

When the system prints on the console:

FILENAME?

1. Type ECLIPSE for the S/200 CPU, or ZECLIPSE for the S/230 CPU, and press the RETURN key.

2. The system then prints:

MAPPED ECLIPSE MULTIGROUND RDOS REV
x.xx.xx
DATE(M/D/Y)?

Here x.xx.xx indicates the current RDOS revision.

Type the actual date. For example, 5 11 78 is May 11, 1978. Press the RETURN key.

3. The system then prints:

TIME(H:M:S)?

Type the actual time based on a 24-hour clock. For example, 8 17 45 is 8:17 AM and 45 seconds while 20 17 45 is 8:17 PM and 45 seconds. Press the RETURN key.

4. The system prints:

```
CALMA REVx CLI  
R
```

Here x indicates the current CLI revision and R is the CLI prompt.

The system is now ready to accept CLI commands so the system generation, described in the next paragraph, or other CLI procedures can be performed.

3.3.11 GDS II Start Up and Shut Down

To initialize GDS II, perform the following:

1. Boot from the disk.
2. In response to FILENAME?, enter a carriage return or appropriate system name.
3. If the previous user of the system crashed or did not release the disk, the system will respond PARTITION IN USE - TYPE C TO CONTINUE. Enter C and the system will respond by typing the remainder of the word CONTINUE.
4. The system will then identify the version of RDOS being executed and the type of computer which is assumed:

```
MAPPED ECLIPSE MULTIGROUND (S/230,  
C/330) RDOS REV 6.01.07
```

5. The system will ask for the current date. Respond appropriately, terminating the response with a carriage return (NOTE: Spaces may be substituted for slashes):

```
DATE (M/D/Y)? 09 24 78
```

6. The system will ask for the time. Respond appropriately, terminating the response with a carriage return (NOTE: Spaces may be substituted for colons; seconds and even minutes are optional):

```
TIME (H:M:S)? 17 11
```

CAUTION

Reasonable care should be used in giving the correct date and time since this information is associated with all files created by RDOS and GDS II.

7. The system will then clear all directories if the PARTITION IN USE message occurred in Step 3, and then execute the initializer program associated with the current system. The system prints on the console:

```
CALMA REV6 CLI
```

```
GDS II RELEASE 00.05.01
```

```
!!BG7 ***** JOB SYSTEM INITIALIZATION *****
```

```
!!BG7 JOB MONITOR DIRECTORY IS GDS II
```

```
!!BG7 1 ON-LINE PLOTTER
```

```
!!BG7 WILL USE BG3
```

```
!!BG7 WILL USE BG4
```

```
!!BG7 WILL USE BG5
```

```
!!BG7 WILL USE BG6
```

```
!!BG 7 LOWEST JOB PRIORITY IS C
```

```
!!BG7 **** JOB SYSTEM INITIALIZATION COMPLETE ***
```

To shut down GDS II, perform the following:

1. Enter QUIT at each station.
2. Release the master disk, i.e., RELEASE %MDIR%
3. The system will respond with a message indicating the master device has been released:

```
MASTER DEVICE RELEASED
```

The computer will halt.

WARNING

The system should never be turned off without shutting down GDS II as described above. Failure to heed may result in a catastrophic loss of data.

3.3.12 System Generation Dialog

The system generation dialog is activated by a CLI command, **MSYSGEN**, which is the filename for systems with ECLIPSE computers and multiground RDOS. The dialog is listed in Figure 3-21.

The following procedure describes the **MSYSGEN** dialog. To generate an RDOS system, ensure that the last print out was the CLI prompt (R) and then perform the following steps:

1. Type **GDIR** and press the RETURN key to get the name of the current directory.

2. The system prints:

DCO (or **BDO** for CDC disks)
R

If the current directory is not **DCO** (or **BDO**), type **DIR DCO** (or **DIR BDO**) and press the RETURN key.

3. When the system prints an R continue to the next step.

Type **DIR SYSGEN** and press the RETURN key to make **SYSGEN** the current directory.

4. The system prints an R.

Type **MSYSGEN** and press the RETURN key to call the system generation routine. For global and local options, refer to the **SYSGEN** description under CLI commands in Section III of the CALMA RDOS manual.

5. The system prints:

MULTIGROUND ECLIPSE SYSTEM REV x.xx.xx
VALID ANSWERS ARE IN PARENTHESIS
RESPOND ACCORDINGLY

S/230 or C/330 MAP? ("0"=NO "1"=YES)

NOTE

The system may skip some system generation queries depending upon the response typed to a previous query.

Type 1 if the system has a 256K or greater memory with memory protection option, or type 0 if the system has a 128K memory with memory protection option, and press the RETURN key.

6. The system prints:

NO. OF ADDITIONAL BACKGROUNDS (0-14)

Type a number from 0 thru 14 and press the RETURN key to select the number of additional backgrounds needed for multiground operation. If unsure, select 2 for a total of four grounds since the basic system has 1 foreground and 1 background.

The system prints:

NO. BACKGROUNDS SWAPPABLE (0 OR 2-MAX)

Type 0 if no grounds are to be swapped, or type 2 so that two grounds can be swapped to disk if memory is full, and press the RETURN key.

The system prints:

NO. BACKGROUND CHANNELS (15-63)

Type a number from 15 thru 63 and press the RETURN key to select the maximum number of hardware data I/O channels needed to execute programs in the main background. CLI runs in the main background and requires 15 channels. RDOS will not run a program that requests more channels than the number selected. If unsure, select 50.

The system prints:

NO. FOREGROUND CHANNELS (0-63)

Type a number from 0 thru 63 and press the RETURN key to select the maximum number of hardware data I/O channels needed to execute programs that requests more channels than the number selected. If unsure, select 0.

The system prints:

NO. BG 03 CHANNELS (1-63)

Type a number from 1 thru 63 and press the RETURN key to select the number of hardware data I/O channels available to the first additional background. If unsure, select 17.

R
MSYSGEN

MULTIGROUND ECLIPSE SYSGEN REV 6. 01. 05
VALID ANSWERS ARE IN PARENTHESIS RESPOND ACCORDINGLY

S/230 OR C/330 MAP? ("0"=NO "1"=YES) 0
NO. OF ADDITIONAL BACKGROUNDS (0-14) 7
 NO. BACKGROUNDS SWAPPABLE (0 OR 2-MAX) 6
 NO. BACKGROUND CHANNELS (15-63) 50
 NO. FOREGROUND CHANNELS (0-63) 0
 NO. BG 03 CHANNELS (1-63) 17
 NO. SWAPPING BG CHANNELS (1-63) 17
NO. OF CENTURY DISKS (0-2) 2
NO. OF BIG DISKS (0-4) 1
DUAL PROCESSORS? ("0"=NO "1"=YES) 0
NO. OF STACKS (1-20) 10
NO. OF EXTRA CELLS (0-64) 6
TUNING? ("0"=NO "1"=YES) 0
NO. OF EXTRA BUFFERS REQUIRED (0-63) 6
NO. OF DISK DIRECTORY DCB'S (0-64) 10
NO. OF CALMA STATIONS (0-8) 2
 SYSTEM 6? ("0"=NO "1"=YES) 0
NO. OF CALMA VMDS (0-8) 0
NO. OF CONTROLLERS FOR MTA (0-2) 1
 PRIMARY ("0") OR SECONDARY ("1")? 0
 NO. OF DEVICES FOR CONTROLLER #1 (1-8) 1
AUTO RESTART ON POWER FAIL? ("0"=NO "1"=YES) 0
LOGON? ("0"= NO "1"=YES) 0
RTC REQ (1=10HZ 2=50HZ 3=60HZ 4=100HZ 5=1000HZ) 4
NO. OF PTR (0-2) 1
NO. OF PTP (0-2) 1
NO. OF LPT (0-1) 1
 SMART SPOOLER ("0"=NO "1"=YES) 1
 COLUMN SIZE FOR LPT (80 OR 132) 132
 LOWER CASE PRINTER ("0"=NO "1"=YES) 0
NO. OF CDR (0-2) 0
NO. OF PLT (0-3) 1
QTY? ("0"=NO "1"=YES) 1
SIZE OF PRIMARY TTY (80 OR 132) 132
SECOND TTY? ("0"=NO "1"=YES) 0
CORE DUMP FACILITY? ("0"=NO "1"=YES) 1

Figure 3-21. Typical MSYSGEN Dialog for Systems with Eclipse CPUs and Multiground RDOS

The system repeats the last query for each additional background and following the last response, prints:

NO. SWAPPING BG CHANNELS (1-63)

Type a number from 1 thru 63 and press the RETURN key to select the number of channels to be swapped in the swapping backgrounds. If unsure, select 17.

7. The system prints:

NO. OF CENTURY DISKS (0-2)

Type a number from 0 thru 2 and press the RETURN key to indicate the number of CENTURY disk controller boards in the system.

8. The system prints:

NO. OF BIG DISKS (0-4)

Type a number from 0 thru 4 and press the RETURN key to indicate the number of CDC disk controller boards in the system.

9. The system prints:

DUAL PROCESSORS? ("0"=NO "1"=YES)

Type 0 and press the RETURN key for systems with one computer. If the system has two computers, type a 1 and then press the RETURN key.

10. When a 1 response is typed, the system prints:

OTHER CPU RDOS-IPB ("0") OR GDS ("1")?

Then, type a 0 and press the RETURN key to select the interprocessor buffer (IPB) or type a 1 and press the RETURN key to select the CALMA CPU-CPU handler.

11. The system then prints:

NO. OF STACKS (1-20)

Type a number from 1 thru 20 and press the RETURN key to indicate the number of I/O devices that will operate concurrently. One stack is required for each ground, one stack for spooling, and an additional stack for each task running concurrently. If unsure, select 10.

12. The system prints:

NO. OF EXTRA CELLS (0-64)

Type a number from 0 thru 64 and press the RETURN key to select the number of extra cells (data buffers with 16 memory words) required. Two extra cells are recommended for each spooling device in the system such as a line printer or magnetic tape controller. If unsure, type 6.

13. The system prints:

TUNING? ("0"=NO "1"=YES)

Type 0 and press the RETURN key if no tuning is desired. Type 1 and press the RETURN key to request tuning. When tuning is requested, RDOS records how often it needed a system stack, cell, or data buffer that was unavailable. This information is kept in a disk file named **RDOS.TU** and is used at a later time to reconfigure the RDOS system for more efficient operation.

If a 1 response is typed, the system prints:

WITH("1") OR WITHOUT("0") OVERLAY REPORT

Then, type a 1 or 0 and press the RETURN key to indicate whether or not an overlay report is desired. The overlay report records the number of times each overlay is transferred from disk because it was not core resident. This information is appended to the **RDOS.TU** file and is used to determine if more buffer space should be allocated to the overlay.

14. The system then prints:

NO. OF EXTRA BUFFERS REQUIRED (0-63)

Type a number from 0 thru 63 and press the RETURN key to select the number of extra buffers required. Each extra buffer requires an additional 270 words and reduces the memory available for programs. Extra buffers are used to reduce the disk swapping time so that RDOS runs more efficiently. If unsure, select 6.

15. The system prints:

NO. OF DISK DIRECTORY DCBs (0-64)

Type a number from 0 thru 64 and press the RETURN key to select the number of system directories that can be initialized at the same time. If unsure, type 10.

16. The system prints:

NO. OF CALMA STATIONS (0-8)

Type a number from 0 thru 8 and press the RETURN key to indicate the number of graphic display stations in the system.

The system then prints:

SYSTEM 6? ("0"=NO "1"=YES)

Type a 1 or 0 and press the RETURN key to indicate whether or not the system is system 6.

17. The system then prints:

NO. OF CALMA VMDs (0-8)

Type a number from 0 thru 8 and press the RETURN key to indicate the number of Vector Memory Displays in the system.

18. The system prints:

NO. OF CONTROLLERS FOR MTA (0-2)

Type a number from 0 thru 2 and press the RETURN key to specify the number of magnetic tape controllers (not drives) in the system.

If a 1 is typed for one controller, the system prints:

PRIMARY ("0") OR SECONDARY ("1")

This specification is set at the time the system is ordered (usually 0); type the applicable response and press the RETURN key.

For each controller specified, the system prints:

NO. OF DEVICES FOR CONTROLLER (#1 (1-8)

Type a number from 1 thru 8 and press the RETURN key to indicate the number of tape drives connected to each controller.

19. The system prints:

AUTO RESTART ON POWER FAIL? ("0"=NO "1"=YES)

Type 0 and press the RETURN key if auto restart hardware is not in the system. If the system has auto restart on power fail hardware, type a 1 response and press the RETURN key.

20. The system prints:

LOGON? ("0"=NO "1"=YES)

Type 0 and press the RETURN key to prevent Logon. To enable the Logon feature, type 1 and press the RETURN key. When the Logon feature is selected, each time the system is booted through the computer switches, the program requires a user ID and password before CLI is allowed to run.

21. The system prints:

RTC FREQ(1=10HZ 2=50HZ 3=60HZ 4=100HZ 5=1000HZ)

Type a number from 1 thru 5 and press the RETURN key to select the Real-Time Clock frequency. Higher clock frequencies cause slower program execution times, so unless there are special program requirements, the 1 should be typed. Typing 2 or 3 allows the program to be synchronized to the applicable line frequency.

22. The system prints:

NO. OF PTR (0-2)

Type a number from 0 thru 2 and press the RETURN key to indicate the number of high speed paper tape readers in the system.

23. The system prints:

NO. OF PTP (0-2)

Type a number from 0 thru 2 and press the RETURN key to indicate the number of high speed paper tape punches in the system.

24. The system prints:

NO. OF LPT (0-2)

Type a number from 0 thru 2 and press the RETURN key to indicate the number of line printers in the system. If a 1 or 2 response is typed, the system then prints:

SMART SPOOLER ("0"=NO "1"=YES)

If the smart spooler will not be used, type 0 and press the RETURN key, and if the smart spooler will be used, type 1 and press the RETURN key.

For each line printer indicated the system also prints:

COLUMN SIZE FOR LPT (80 OR 132)

To respond to this query, type either 80 or 132 depending upon the column size for the line printer and then press the RETURN key.

The system then prints:

LOWER CASE PRINTER ("0"=NO "1"=YES)

Type a 1 or 0 and press the RETURN key to indicate whether or not the printer is lower case.

25. Then the system prints:

NO. OF CDR (0-2)

Type a number from 0 thru 2 and press the RETURN key to indicate the number of punched or mark-sense card readers in the system.

26. The system prints:

NO. OF PLT (0-3)

Type a number from 0 thru 3 and press the RETURN key to indicate the number of plotters in the system.

27. The system prints:

QTY? ("0"=NO "1"=YES)

Type a 0 or 1 to indicate whether or not (respectively) the system is equipped with a 4060 asynchronous data multiplexer and then press the RETURN key.

28. The system prints:

SIZE OF PRIMARY TTY (80 or 132)

Type 132 and then press the RETURN key.

29. The system then prints:

SECOND TTY? ("0"=NO "1"=YES)

Type 0 if only one keyboard will be used or type a 1 if the system uses a second keyboard and then press the RETURN key.

If a 1 was typed the system prints:

SIZE OF SECOND TTY (80 or 132)

Type 132 and then press the RETURN key.

30. The system then prints:

CORE DUMP FACILITY? ("0"=NO "1"=YES)

Type a 0 to inhibit core dump or a 1 to allow core dump and press the RETURN key. The core dump facility requires approximately 100 memory locations and allows selected portions of machine-code or compiler language programs to be listed as a debugging aid.

31. The system then prints:

SYS000.SV LOADED BY CALMA RLDR REV
05:03 AT 08:45:19 05/11/78

The revision level, time, and date will vary.

Unless interrupted, the system proceeds to print the load map and any load errors. Save the dialog just produced and the load map for future reference when making revisions to the system generation.

To interrupt the listing, press the CTRL (Control) and A keys at the same time. When the listing is complete the system prints an R or, if interrupted, it prints INT and then R. System generation is complete and control is returned to CLI.

Type **DIR DCO** to initialize the primary partition and make **DCO** the current directory.

32. The system then prints an R.

Type BOOT ECLIPSE (or ZECLIPSE or GDS II) and press the RETURN key to load the appropriate file containing SYSGEN information into core.

33. The system then prints:

MASTER DEVICE RELEASED (DC0 automatically released at BOOT complete.)

```
MAPPED ECLIPSE MULTIGROUND RDOS REV
x.xx.xx
CALMA REVn CLI
R
```

Here x.xx.xx indicates the current RDOS revision and n indicates the current CLI revision.

The generated RDOS system is now in core and the system is ready to receive overlay programs or program instructions, or to run GDS II.

SECTION 4
GDS II COMMANDS

4.1 GENERAL

This section contains information about the GDS II Commands including the command syntax and command descriptions. The command syntax describes how the commands are used and the command descriptions tell how to use the commands. Command descriptions are also contained on disk and may be displayed on the QTY by use of the HELP command.

NOTE:
Descriptions for all of the available
commands are not yet included in this
manual.

4.2 COMMAND SYNTAX

The GDS II command syntax defines the rules necessary to enter and use commands. These rules are established by the program being used. Most of the rules are internal to GDS II, but some depend on the operating system (RDOS). For example, an RDOS filename may have up to 10 alphanumeric characters and a dollar sign (\$) in any sequence. Since GDS II libraries are RDOS files, a library name must conform to the rules for RDOS files. Structure names are internal to the GDS II program and may have up to 32 alphanumeric characters back arrow (\leftarrow), dollar sign (\$), and question mark (?).

4.2.1 Command Names and Arguments

Command names are also established by the program and only certain words (usually mnemonics) are accepted by the program. The command arguments (variables) become part of the command and are subject to the command syntax rules. Two types of commands are recognized by the program. They are NILADIC (no arguments) and MONADIC (arguments). Most GDS II commands can use either type to produce different results. For example, the GSCALE command with no arguments will clear the working grid while GSCALE 0 0 1 1 establishes a new working grid with an offset of 0 0 and the increments between points of one user unit in both x and y axes. User units are specified with an INITLIB command. There are also three methods of entering command arguments. These methods are called:

1. Command Input
2. Literal Input
3. Evaluated Input

These methods are displayed on the QTY to indicated the type of input that the program expects. Command input is always displayed until a command name is entered and followed by a carriage return <CR>. If the command returns a query such as "Enter Data:," the display changes to literal or evaluated input. When literal input is displayed the program is looking for an alphanumeric input such

as a file name which is acceptable as a command argument. An evaluated input display indicates that a numeric argument such as a coordinate is expected.

MONADIC commands can have their arguments given by the command input method or by either of the other two methods depending upon the type of argument required. For example, the BSTRUCT command can be given in either of the following two ways:

1. BSTRUCT "structure name" <CR>
2. BSTRUCT <CR>
Enter Structure: structure name <CR>

The first way uses the command input method and the structure name must be enclosed in double quotes. In the second way, the literal input method is used. After the program outputs the query "Enter Structure:," the structure name is typed immediately following the colon. No quotes are used and the second <CR> enters the command. Both ways perform the same operation.

Another example is the LAYER command which is given in either of the following two ways:

1. LAYER n <CR>
2. LAYER <CR>
Enter Layer: n <CR>

The expected response is a number (n) from 0 to 63. The first way uses the command input method and a space is input between the command and the number. When the evaluated input method is used, the program outputs the query "Enter Layer:" and the number is typed immediately following the colon. Double quote marks are not required for numerical arguments. Both ways shown for this command perform the same function.

For the command input method when arguments are used on the same line as the command, the following syntax rules apply:

1. Literal arguments are enclosed in double quotes.
2. Evaluated arguments are preceded by a space.
3. Arguments are separated by spaces or semicolons.
4. Command entry is terminated and execution begun by a <CR>.

For literal or evaluated input methods, a <CR> is input after the command name. Then, the system prompts with one or more queries and waits for the proper response to continue. After each response is entered, <CR> is input again, and the program either outputs another query or terminates command entry and begins execution.

All of the command input methods and syntax are specifically shown in the help files. These files can be accessed at any time by using the HELP command in either of the following forms:

```
HELP"command name"<CR>  
or  
HELP<CR>  
Command Name: commandname<CR>
```

When entering negative numbers as a command argument, such as the scaling commands, a pound sign (#) is used in place of the minus sign (-) whenever there are negative and positive numbers on the same line. If a minus sign (-) is used preceding the first number on a line, all numbers on the line are read as negative numbers. Refer to operation in calculator mode for more information on inputting negative numbers.

4.2.2 Coordinate Entry (CE)

Coordinate entry refers to entering x and y axis data points or vertices using the keyboard, digitizer table, or tablet. The x and y axis coordinate values are displayed on the QTY and are given in system hardware values until a library is initialized and opened to establish user units. A GSCALE command is then given to provide greater precision and ease of reading the x and y coordinates in user units. Scaling commands establish the

coordinate range in minimum and maximum values of x and y. These commands require the entry of specific coordinates.

Where a coordinate entry is required, it is shown in the command syntax by <CE>. A coordinate entry followed by a carriage return is shown as <CE/CR>.

How the coordinates are entered depends upon the input device being used and the selected input mode. When data is being entered using the CE command with GEDMODE off, the system prompts for coordinates until a <CR> is entered before a <CE>. As long as the CE command is in use, nothing but coordinates can be entered. With the graphic editor mode on (GEDMODE command), coordinates are entered at any time and intermixed with other commands. To enter coordinates from the keyboard, the coordinate numbers are typed with the x value given first, then a space and the y value. Each point must have an x value and a y value. For example: 0 0 5 2 7 4 9 8 would represent four points with values of x=0 y=0; x=5 y=2; x=7 y=4; and x=9 y=8.

Digitizer table coordinates (<CE>) are entered by first positioning the stylus to the desired location and then pressing the top-left (white) button on the stylus. To enter <CE/CR> from the digitizer table, position the stylus and then press the top-right (blue) button on the stylus. These same functions can be performed by special keys on the station keyboard. The special keys are located in the 3 by 5 group of keys at the right side of the standard keyboard. A <CE> is entered by pressing the bottom-center key and a <CE/CR> is entered by pressing the bottom-left key.

Tablet coordinates are entered using the pen-type stylus or the station keyboard. The pen-type stylus provides a <CE/CR> input each time the point is pressed on the tablet surface. To enter only a <CE> or a <CR>, use the station keyboard.

4.2.3 Command Prompt

The GDS II program prompts for a new command with a line feed and question mark (?). The operator then types the command starting on the same line as the question mark. Typing a command on any other line or following a different symbol than a question mark may result in an error.

4.2.4 Input Corrections

To correct typing errors while entering commands and before a carriage return (<CR>) is entered to terminate the command input, two keyboard keys can be used. The backspace (BS) key deletes one character each time it is pressed and erases the character from the QTY. If the last entry was a coordinate, the whole coordinate (both x and y values) are deleted and erased. The other key is the back slant (↵) located next to the carriage return (CAR RET) key. Pressing the back slant key erases the entire line.

For example:

```
?BSTRUCT<CR>  
Structure Name: anyname↵
```

In this example the back slant erases the entire line. A new prompt will not be given. Retype the structure name "anyname".

After a <CR> is typed the input is processed and the back space (BS) and back slant (↵) keys are no longer valid.

4.2.5 Console Abort

Most GDS II commands are console abortable so that they can be terminated before or during execution without causing a system failure. Following a console abort the system outputs the message ***INTERRUPT*** on the QTY, and then a line feed, carriage return and the command prompt (?). To activate a console abort, press the INT (interrupt) key on the keyboard. The INT key is located on the right side of the keyboard in the 3 x 5 group of special keys. It is the bottom-right key.

4.2.6 Station Release

To release a GDS II station, it is necessary to issue a QUIT command that releases the station for other use. The QUIT command implies a CLOSELIB that closes any open libraries at the station and a TSTRUCT that closes any open structures at the station. If any items in a structure were not complete, data is lost when the QUIT command is issued. It is usually best to issue an ISTAT command first to see if there is any data in ITEM. An ITEM with data is closed by a NULL command if GEDMODE is off or a <CR> if the GEDMODE is on. When an ITEM is closed, the data is entered into the data base files and becomes permanent data.

4.2.7 Calculator Mode

Calculator mode is a design tool that provides six functions for performing arithmetic operations. The available functions are:

- + plus sign for addition
 - minus sign for subtraction of all values in a group
 - # pound sign for subtraction of a single value
 - * asterisk for multiplication
 - % percent sign for division
- E or e as Exponent for scientific notation.

The following examples show the use of minus (-) and pound (#) for subtraction.

1. In a group of values preceded by a minus sign, all values are minus regardless of other signs in the group. $-10\ 5\ +6\ \#7 = -10\ -5\ -6\ -7$
2. A pound sign applies to one value and is not distributed to all other values in a group. $\#10\ 5\ +6\ \#7 = -10\ +5\ +6\ -7$

Calculator mode is always active, whether or not GEDMODE is on, so command arguments are affected by the calculator functions. For example: with GEDMODE on, an entry of $2 + 3$ is treated as 5 and changes the ITEM number to 5. Also, with GEDMODE on or during coordinate entry with the CE command, an entry of $2+4\ 5+2$ is treated as a coordinate entry with $x=6$ and $y=7$.

4.3 COMMANDS

The commands used by GDS II are described in separate command files arranged alphabetically at the end of this section. In order to use these files accurately it is necessary to know the conventions used to describe the commands. Most of the symbols used such as semicolon (;) and double quotes (" ") are part of the command syntax and are required for the command to work properly.

Spaces are also part of the required command syntax. However, angle brackets (< >) and braces ({ }) are used to describe command syntax and are not part of the command. The angle brackets are used to enclose single functions such as carriage return (<CR>), coordinate entry (<CE>), and coordinate entry with carriage return (<CE/CR>). Braces are used to show information that is useful for example but is not entered by the operator such as {min x, min y} in the DISPLAYSTAT command listing. In this case, the command is listing values that were entered by a previous command.

Also, the commands are shown in all upper case letters (all capital letters) and other text is shown in both upper and lower case. Usually it does not matter in actual command input. There are commands that change the input to all upper-case (UPPERCASE), lower case (LOWERCASE), or upper and lower case (UPLOWCASE), which is the default condition. It is necessary to observe strict upper or lower case conventions for entering structure names. Structure names are read and retained exactly as they are entered so that PARIS, paris, and Paris, are all unique structure names.

4.3.1 Absolute Versus Relative

The command descriptions refer to commands as being absolute and relative. This is extremely important when placing structures in relation to other structures. Absolute indicates the value is fixed and is independent of other structures and values. Relative indicates that a value is dependent and remains constant in relation to another value. If the reference value changes the relative value changes accordingly so the same spacing remains between the two values. Also, if the reference value rotates, the relative value also rotates, but an absolute value remains where it was placed.

4.3.2 Command List By Function

The following list separates the commands alphabetically within functional groups.

1. Station Configuration Control Commands

| | | |
|-------------|-----------|-----------|
| AUTOHOLD | LOWERCASE | TMON |
| BUTTONBEGIN | MEMUMOVE | UPLOWCASE |
| BUTTONEND | RMENU | UPPERCASE |
| DFUNCTION | SETTAB | |
| HARDLOCKS | TMOFF | |

2. View Control Commands

| | | | |
|--------------|----------|-------------|-----------|
| AREFMARK | FSETA | VDTYPE | VTTYE |
| BOUNDARYMARK | FSETB | VDTYPEOFF | VTTYE OFF |
| BROKEN | FSETC | VDTYPEON | VTTYE ON |
| DASHED | FSETD | VIEW | ZOOM |
| DISPLAYSTAT | GRID | VIEWMOVE | |
| DOTTED | LEVEL | VKIND | |
| ERASE | MARKSIZE | VKINDOFF | |
| FILLA | MASKS | VKINDON | |
| FILLB | PATHMARK | V LAYER | |
| FILLC | PAN | V LAYER OFF | |
| FILLD | REDRAW | V LAYER ON | |
| FLAYER | SETVIEW | V LEVELS | |
| FLAYER OFF | SOLID | VON | |
| FLAYER ON | SREFMARK | VOF | |

3. Scale Commands

| | | |
|--------|--------|--------|
| ASCALE | SSTAT | ZSCALE |
| BSCALE | WSCALE | |
| GSCALE | XSCALE | |
| RSCALE | YSCALE | |

4. Library Access Commands

| | | |
|-----------|------------|---------|
| BSTRUCT | LISTLIB | REF2LIB |
| CLOSELIB | LISTGEN | REVERT |
| CLOSEREF1 | LSTAT | RSTRUCT |
| CLOSEREF2 | MERGELIB | SSTRUCT |
| CSTRUCT | OPENLIB | SUMMARY |
| DSTRUCT | OPENREF1 | TREE |
| ESTRUCT | OPENREF2 | TSTRUCT |
| EXPUNGE | OSTRUCT | VSTRUCT |
| INITLIB | RECOVERLIB | |
| INPLACE | REF1LIB | |

5. Data Entry Commands

| | |
|----|------|
| R3 | SEGS |
| RT | |

6. Data Movement Commands

| | |
|--------------|-----------|
| DATACOPY | SDTYPE |
| DATADELETE | SDTYPEOFF |
| DATAMAGNIFY | SDTYPEON |
| DATAMOVE | SKIND |
| DATAOVERSIZE | SKINDOFF |
| DATAROTATE | SKINDON |
| DATAREFLECT | SLAYER |
| EXPLODE | SLAYEROFF |
| NEWDATATYPE | SLAYERON |
| NEWLAYER | STTYPE |
| NEWWIDTH | STTYPEOFF |
| | STTYPEON |

7. Identified Group Commands

| | |
|------------|------------|
| ID | IDONLY |
| IDCLEAR | IDPATHEXP |
| IDCOPY | IDSUMMARY |
| IDDATATYPE | IDWIDTH |
| IDDELETE | IDWINDOW |
| IDEXPLODE | UNID |
| IDLAYER | UNIDWINDOW |
| IDMARK | |
| IDMOVE | |
| IDOVERSIZE | |

8. Window Option Commands

| | |
|-----------|------------|
| WINCOPY | WINMOVE |
| WINDELETE | WINOPTIONS |
| WINFENCE | WINSTRETCH |

9. Graphic Editor Commands

| | | |
|------------|------------|--------------|
| ABSANGLE | GEDMODE | REVERSE |
| ABSMAG | GET | SETDEFAULTS |
| ABSREFL | HORIZFIRST | SHOWDEFAULTS |
| ABSWIDTH | INTRT | SNAME |
| ADDRT | ISTAT | SREF |
| ANGLE | ITEM | ST |
| AREF | LAYER | STRAIGHT |
| ASHAPE | LINES | SUBRT |
| BINDFONTS | LINKS | TEXT |
| BOUNDARY | MAG | TEXTFONT |
| CE | MANDALA | TEXTTYPE |
| COORDS | MOVE | TFTST |
| CUTPATHIN | NULL | TJUST |
| CUTPATHOUT | NULLALL | TRANS |
| DATATYPE | OCTAGONAL | UNDO |
| DEFMODE | ORTHINT | UNIONRT |
| EDGE | PATH | VERTFIRST |
| ENTERCSD | PUT | WIDTH |
| ENTERTEXT | PUTALL | WIPE |
| FIXPOLY | REFL | WITEM |

10. Text Editor Commands

| | | | |
|-----|-------|-------|---------|
| A | ESTAT | IMOVE | SPLIT |
| B | G | J | T |
| CH | I | JOIN | TEDEBUG |
| DB | IB | K | W |
| DE | ICOPY | L | Z |
| DLN | IE | S | |
| DS | IFILE | SB | |

11. Background Job Control Commands

| | | |
|-----------|-------------|-------|
| JOBABORT | JOBPRIORITY | PAUSE |
| JOBCREATE | JOBSTART | RS |
| JOBINFO | JOBSUSPEND | |

12. System Utility Commands

| | | |
|-----------|----------|--------|
| ABGROUND | FCREATE | HELP |
| ABSTATION | FDELETE | LOG |
| COMMANDS | FINIT | QUIT |
| DEBUG | FLINK | START |
| DIRECTORY | FLIST | SYSTAT |
| DISK | FMOVE | TYPE |
| ENDLOG | FPRINT | |
| FCDIR | FRELEASE | |
| FCHATR | FRENAME | |
| FCOPY | FUNLINK | |

13. GPL Input Operator Commands

| | |
|----------|-----------|
| EXPINPUT | TEXTINPUT |
|----------|-----------|

14. GPL Commands

| | |
|-------|--------|
| CLEAR | SYSN |
| FETCH | TIME |
| GPL | TXGPL |
| LOAD | VARS |
| NAMES | WASTAT |
| SUBS | |

4.3.3 Command Descriptions

The remainder of this section contains command descriptions for each command used in the GDS II system. These command descriptions start with the command given first in all its various forms. Then separate paragraphs are given for prerequisites, purpose, classification, restrictions, and examples.

1. Prerequisites, if any, list the commands that should have been performed first or conditions that must exist for the command to work.
2. Purpose indicates what the command does and how it is used.
3. Classification describes what functional group of commands it belongs to.
4. Restrictions indicate ways to avoid problems using the command.
5. Examples are given to show how the command is used.

The command descriptions are given in alphabetic sequence for ease of referencing to a particular command. A list of commands by classification (functional group) is given in Paragraph 4.3.2.

```
A<CR>
:textline1<CR>
:textline2<CR>
:textline3<CR>
:.<CR>
or
A"text"<CR>
```

Prerequisite: None required.

Purpose: Appends or inserts lines of text below the current line. The current line is always shown in bright characters on the QTY. After issuing the A command and <CR>, the program prompts for lines of text with a colon (:). Each line of text is terminated with <CR> and the program then prompts for the next line. A text line may consist of only a <CR> or up to 132 characters and spaces including the colon. To terminate the append routine, type a single period (.) following the colon (:) prompt and then <CR>. Also, the ESC (escape) key on the lower-left of the keyboard can be pressed to terminate the append command. When the ESC key is pressed the program echos with a dollar sign (\$). The terminating dollar sign (\$) or period (.) are not stored in the data file. The same procedure may be used to create a new text file in the edit buffer.

Using the second form of the A command allows text to be appended to the beginning of the current line. Text to be appended is enclosed in double quotes and followed by <CR>, but the double quotes and <CR> are not transferred to the display or file.

Classification: Text editor command.

Restrictions: The QTY displays only 80 characters and spaces per line and then automatically skips to the next line. However, the system does not recognize a new line until a <CR> is input, so that when a file is output to the Decwriter up to 132 characters and spaces are printed on each line.

Examples:

```
?A<CR>
:Lines of text can be added<CR>
:<CR>
:below the current line<CR>
:<CR>
:anywhere in a file.<CR>
:.<CR>
.
.(text lines are output to QTY)
.
?
```

Three lines of text, with blank lines between the text lines, are added to a file. The new lines are added starting on the next line below the current line and any existing lines following the current line are moved to make room. Existing lines are not deleted.

```
?A"text is added to"<CR>  
.  
.(text lines output to QTY)  
.  
?
```

In this example, the current line was changed from "the beginning of the current line" to "text is added to the beginning of the current line."

ABSANGLE n<CR>
or
ABSANGLE<CR>
Enter Angle: n<CR>

Prerequisites: SNAME. Must be in SREF, AREF, or TEXT.

Purpose: Sets the angle mode for a structure reference to the angle specified, 0-360°. The angle is absolute, which means that the angle remains the same as it was when created even though other conditions may be changed. See also ANGLE, for an explanation of the relative condition.

Classification: Graphic Editor

Restrictions: None

ABSMAG n<CR>

or

ABSMAG<CR>

Enter Magnification: n<CR>

Prerequisites: Check ITEM. SNAME must be entered. Must be in SREF, AREF, or TEXT.

Purpose: Sets the magnification mode for a structure reference to the number specified. The number must be positive and not zero. The magnification is absolute, which means that it will not change even though associated conditions may change. See MAG for an explanation of the relative condition.

Classification: Graphic Editor

Restrictions: Must be a number greater than 0.

Examples:

?ABSMAG 2<CR>

?

Sets the absolute magnification to twice the size.

?ABSMAG<CR>

Enter Magnification: .5<CR>

?

Sets the absolute magnification to half the size.

ABSREFL "string"<CR>
or
ABSREFL<CR>
Reflection (YES/NO): "string"<CR>

Prerequisites: Check ITEM. Must be in AREF, SREF, or TEXT.

Purpose: Sets the reflection mode of the current structure or array reference in item and makes it absolute, which means that even though associated conditions may change, the reflection will not change. The "string" may be "Y", "YES", "y", "yes" or just <CR> to specify the reflection. "N", "NO", "n", or "no" means no reflection is desired. The reflection is always in the y axis. See REFL for an explanation of the relative condition.

Classification: Graphic Editor

Restrictions: ABSREFL relates to SREF, AREF, or TEXT only.

ABSWIDTH n<CR>
or
ABSWIDTH<CR>
Enter Width: n<CR>

Prerequisites: PATH or TEXT.

Purpose: Sets the width of the current item to the width specified. The width is in absolute user units and is visible in ITEM when the current item type is PATH or TEXT. This command also sets the default value for the width property. See DEFSTAT for the current default value. (Absolute means that the condition remains the same as it was when input regardless of whether or not associated conditions are changed.)

Classification: Graphic Editor

Restrictions: Width cannot be smaller than one database user's unit.

?ABSWIDTH 2<CR>

?

Sets the absolute width of the current item to 2.

?ABSWIDTH<CR>

Enter Width: 4 CR

?

Sets the absolute width of the current item to 4.

ADDRT<CR>
Enter RT Point:<CE/CR>
Enter RT Point:<CE/CR>

Prerequisites: Must have data in current Item.

Purpose: Adds a rectangular area to the data in the current item in order to create or modify a new geometry. This new data is put in the database, and data in the current item remains unaffected. The rectangular area created by ADDRT is added to the exterior of the original geometry, like an appendage. For the first argument, enter the lower left point of the rectangular area. For the second argument, enter the upper right point. See also SUBRT.

Classification: Graphic Editor

Restrictions: The new rectangular area must intersect at least one side of the geometry in the current item.

ANGLE n<CR>
or
ANGLE<CR>
Enter Angle: n<CR>

Prerequisites: Check ITEM. Must be in SREF, AREF, or TEXT. Set SNAME before setting ANGLE.

Purpose: Sets the angle for a structure reference to the degree specified, 0° to 360°. The angle is relative, which means that if subsequent conditions are changed, the angle will also be changed to relate with those new conditions.

Classification: Graphic Editor

Restrictions: None

AREF<CR>

Prerequisites: A library and a Structure must both be open. Check ITEM or ISTAT for status. SNAME and ASHAPE must also be input.

Purpose: Declares that the current item is an AREF (Array Reference). If the current item already has data that has not yet been NULLed (or PUT), an error occurs. Check ITEM first to avoid this problem. Go to the next available item. ASHAPE, which sets the array shape, is also associated with AREF. Like SNAME, it is normally input before AREF. See example of command chain below.

Classification: Graphic Editor

Restrictions: Cannot create an array within an array.

Examples: This example shows the normal command sequence for setting up an AREF. ITEM then lists the conditions that have been set.

```
?SNAME<CR>
Enter Structure for Reference: NEW
?ASHAPE<CR>
Enter the Number of Columns: 5
Enter the Number of Rows: 3
Enter the X-Spacing: 3.5
Enter the Y-Spacing: 4
?AREF<CR>
?ITEM<CR>
You Are Editing Item # 0
It is an Aref of "NEW" (0 Points So Far).
Relative Angle of 0. Degrees,
Relative No Reflection.
5 Columns, 3 Rows.
X-Spacing of 3.5 Units, Y-Spacing of 4. Units.
0 Links Allocated
?
```

The conditions for angle, magnification, and reflection for this AREF were set up by prior commands, as illustrated:

```
?SNAME<CR>
Enter Structure for Reference: NEW
?ANGLE<CR>
Enter Angle: 0
?MAG<CR>
Enter Magnification: 1
?REFL<CR>
Reflection (YES/NO): NO
?
```

If absolute, rather than relative, conditions are desired, the following command chain is used, changing the values; ITEM lists the new conditions:

```
?SNAME<CR>
Enter Structure for Reference: NEW
?ABSANGLE<CR>
Enter Angle: 90
?ABSMAG<CR>
Enter Magnification: 2.5
?ABSREFL<CR>
Reflection (YES/NO): YES
?ASHAPE<CR>
Enter the Number of Columns: 5
Enter the Number of Rows: 3
Enter the X-Spacing: 3.5
Enter the Y-Spacing: 4
?AREF<CR>
?ITEM<CR>
You Are Editing Item # 0
It is an Aref of "NEW" (0 Points So Far).
Absolute Angle of 90. Degrees,
Absolute Magnification of 2.5X,
Absolute Reflection.
5 Columns, 3 Rows.
X-Spacing of 3.5. Units, Y-Spacing of 4. Units.
0 Links Allocated
?
```

AREFMARK<CR>

Prerequisites: AREF

Purpose: Causes a mark to be placed at the origin of each selectable AREF so that it is apparent which geometries are ARefs and that they are placed at level 0, the level at which data is created. The ARefmark is erased when the array is redrawn.

Classification: View Control

ASCALE <CR>
Digitizer Coordinates: CE1 <CR>
or
ASCALE <CR>
Digitizer Coordinates: CE1 CE2 <CR>
or
ASCALE <CR>
Digitizer Coordinates: CE1 CE2 CE3 <CR>
or
ASCALE <CR>
Digitizer Coordinates: CE1 CE2 CE3 CE4 <CR>

Prerequisites: TMOFF

Purpose: Aligns the x and y axes of the digitizer table to a drawing attached to the table. A coordinate entry is done with up to four points. If only one point is used, the x and y axes are forced to be aligned with the digitizer axes (no rotation). Two points force the y axis to be orthogonal to the x axis and rotates the digitizer table alignment. Three points align the x axis with points CE1 and CE2 and aligns the y axis to points CE1 and CE3. Four points align the x axis with points CE1 and CE2 and aligns the y axis to points CE3 and CE4.

Classification: Scale command.

Restrictions: Tablet mode must be off.

Examples:

```
?ASCALE <CR>  
Digitizer Coordinates: <CE> <CE> <CE> <CR>  
?
```

Three points were entered. Usually, the three points are the lower-left, lower-right, and upper-left points on the drawing.

```
?ASCALE <CR>  
Digitizer Coordinates: <CE> <CE> <CE> <CE> <CR>  
?
```

Four points were entered. Usually, the four points are center-left, center-right, bottom-center, and top-center of the drawing. The origin (rotation point) is the drawing center.

```
ASHAPE<CR>
Enter Number of Columns: n<CR>
Enter Number of Rows: n<CR>
Enter x-spacing: xs<CR>
Enter y-spacing: ys<CR>
```

Prerequisite: SNAME should be defined. AREF should be set.

Purpose: Sets the AREF (Array Reference) parameters--columns, rows, x and y spacing. The number of columns and the number of rows must both be whole, non-zero integers. The stepping distance is the width of the structure plus the spacing to the next structure.

Classification: Graphic Editor

Restrictions: None

Examples:

```
?ASHAPE<CR>
Enter number of Columns: 10<CR>
Enter Number of Rows: 2<CR>
Enter x-spacing: 3<CR>
Enter y-spacing: 4.5<CR>
?
```

Sets an AREF that has 10 columns and 2 rows. There are 3 user units for the x-spacing and 4.5 user units for the y-spacing.

AUTOHOLD <CR>

Prerequisite: None required

Purpose: Works with the QTY to stop text scrolling past the top of the display. The command operates like a toggle switch to change states between on and off each time the command is issued. If AUTOHOLD is on and text scrolls to the top of the display and stops without returning the command prompt (?), the QTY display COMMAND INPUT changes to AUTOHOLD. The CONT key is then pressed to continue scrolling. The CONT key is located in a special 3 by 5 group of keys on right side of the keyboard, and is the third key down in the center column.

Classification: Station configuration control command.

Restrictions: AUTOHOLD should not be left in the on state during structure creation and edit since it may cause some operator inconvenience.

Example:

```
?AUTOHOLD<CR>  
Now On  
?
```

The AUTOHOLD was off and is now on.

```
?AUTOHOLD<CR>  
Now Off  
?
```

The AUTOHOLD was on and is now off.

B<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Searches to the beginning of the edit buffer and displays the first three lines on the QTY. The first line becomes the current line and is displayed in bright characters. To insert text above the current line, use the I command; and to add text below the current line, use the A command.

Classification: Text editor command.

Restrictions: None

Example:

```
?B<CR>
.
.(text lines)
.
?
```

The first three lines of a file in the edit buffer are displayed.

BINDFONTS<CR>
Font Binding Utility -
Font File: name CR
Font Number: (0,1,2,3): n<CR>
Font File: name<CR>
Font Number: (0,1,2,3): n<CR>

Prerequisites: Library must be open. Must have a TEXTFONT definition file.

Purpose: Calls the font definition file to be used. Currently only one definition file can be used at a given time, but files can be reassigned. Each library has access to up to four different fonts. There is no restriction on the number of different fonts that may be defined.

BOUNDARY<CR>

Prerequisite: A Library and a Structure must both be open. Check ITEM or ISTAT for status.

Purpose: Declares the type of the current item to be BOUNDARY. If the current item already has data that has not yet been NULled (or PUT), an error occurs. Check ITEM first to avoid this problem. Go to the next available item.

Classification: Graphic Editor

Restrictions: None.

Examples: This example shows the normal command sequence for setting up a BOUNDARY, in this case in EDGE mode. ITEM then lists the conditions that have been set:

```
?LAYER<CR>
Enter Layer: 1
? DATATYPE<CR>
Enter Datatype: 0
?BOUNDARY<CR>
?EDGE<CR>
?ITEM<CR>
You Are Editing Item # 0.
It is a BOUNDARY (0 Points So Far).
Layer 1, Datatype 0.
You Are In Edge Mode (Horizontal-First).
0 Links Allocated
?
```

BOUNDARYMARK<CR>

Prerequisites: BOUNDARY

Purpose: Causes a mark to be placed at the origin point of each selectable boundary so that it is apparent which geometries are boundaries and that they are at level \emptyset , the level at which data is created. The Boundarymark is erased when the geometry is redrawn.

Classification: View Control

Restrictions: None

BROKEN<CR>
or
BROKEN n n n...n<CR>
or
BROKEN -n<CR>

Prerequisites: None required.

Purpose: Specifies layers that are to be presented as broken lines. Each entered number specifies a layer, 0-63. BROKEN with "-n" (minus sign plus a number) indicates that layers 0 through n are to be broken lines. See example below. Check MASKS for status.

Classification: View Control

Restrictions: None

Examples:

?BROKEN<CR>
?
Specifies that all layers are to be in broken lines.

?BROKEN 1 3 5<CR>
?
Specifies that layers 1, 3, 5 are to be broken lines.

?BROKEN -5<CR>
?
Specifies that layers 0 - 5 are to be broken lines.

```
BSCALE <CR>
x Scale Values: x1 x2 ... x11 <CR>
y Scale Values: y1 y2 ... y11 <CR>
Digitizer Coordinates: CE1 CE2 ... CE11 <CR>
```

Prerequisite: TMOFF, ASCALE

Purpose: Provides linear incremental scaling in both x and y axes. Enter from 2 through 11 scale values for x and y, then do a coordinate entry (CE) for each value. One set of values and coordinates is used to translate previous coordinates to a new location.

Classification: Scale command.

Restriction: Enter coordinates in increasing order. Use pound (#) in place of minus (-). Tablet mode must be off.

Examples:

```
?BSCALE <CR>
x Scale Values: #2 #1 0 1 2 <CR>
y Scale Values: #2 #1 0 1 2 <CR>
Digitizer Coordinates: <CE> <CE> <CE> <CE> <CE> <CR>
```

Five x and y values from -2 (#2) to +2 were selected and digitized.

```
?BSCALE <CR>
x Scale Values: 1
y Scale Values: 1
Digitizer Coordinates: <CR> <CR>
```

One value was selected to translate previously entered values.

```
?BSCALE <CR>
x Scale Values: 1, 2, ... 11 <CR>
y Scale Values: 1, 2, ... 11 <CR>
Digitizer Coordinates: <CE/CR>
Digitizer Coordinates: <CE/CR>
.
.
.
Digitizer Coordinates: <CE/CR>
?
```

Up to 11 values are selected for x and y. When <CR> is input with each <CE>, the program prompts until there is a <CE> for each value.

BSTRUCT"structure name"<CR>
or
BSTRUCT<CR>
Enter Structure: structure name<CR>

Prerequisites: OPENLIB

Purpose: Reserves a structure name in the current working library and opens the file to build a new structure. The new structure name is added to the library listing with one generation and any structure previously open for editing is closed. A structure name may contain up to 32 characters consisting of upper and lower case alphabet, 0 thru 9, back arrow (←), dollar sign (\$), and question mark (?).

Classification: Library access command.

Restrictions: Two stations cannot create the same structure if both stations have the same working library.

Example:

```
?BSTRUCT"MEMORYBUS1"<CR>  
?
```

A structure named "MEMORYBUS1" is opened for building a new structure.

```
?BSTRUCT<CR>  
Enter Structure: 1230THERNAME<CR>  
?
```

A structure named "1230THERNAME" is opened for building a new structure.

| | |
|--------------------------|------------------------|
| CE<CE/CR> | or , when in GED mode: |
| or | <CE/CR> |
| CE<CR> | <CE/CR> |
| ce 1:<CE/CR> | <CE/CR> |
| ce 2:<CE/CR> | . |
| ce 3:<CE/CR> | . |
| . | . |
| . | <CE/CR> |
| . | or |
| ce n:<CE/CR> | <CE><CE><CE>...<CE/CR> |
| or | |
| CE<CE><CE><CE>...<CE/CR> | |

Prerequisite: BSTRUCT or ESTRUCT. Structure must be open for editing.

Purpose: Allows coordinates to be input into the current item. CE can be used when Gedmode is on, but it must be used when Gedmode is off. Data is input in CE mode more quickly than in Gedmode (when Gedmode is on), because in CE mode, the system automatically evaluates the data as coordinates. Moreover, in CE, the number of coordinates is indicated, so there is no chance of accidental input of data, as might occur in Gedmode.

A listing of coordinates entered can be obtained by using the COORDS command (see Help File).

Classification: Graphic Editor

Restrictions: When coordinates are input as multiple arguments on the same line using the CE command, the geometry does not appear on the CRT as it is being input. It is viewable only after a carriage return.

Examples:

?CE<1.242 #.402 CR>

?

Inputs the x and y coordinates and brings an origin point marker on the CRT, if this is the first coordinate of a geometry, or continues a geometry already started.

?CE<CR>

ce 1: <1.243 #.402/CR>

ce 2: <.173 #.659/CR>

ce 3: <#1.077 #.38/CR>

.

.

.

ce n: <1.320 #.525/CR>

?

Inputs n coordinates and brings the lines of the geometry in the CRT as they are being input.

?CE <1.242 #.402><.173 #.659><#1.077 #.38>...<1.320 #.525/CR>

?

Inputs coordinates (up to 132 characters per line), but input is not viewable until the carriage return has been performed.

CH"text1";"text2"<CR>

Prerequisite: A or I create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Changes "text1" to "text2" in the current line only. The current line is always shown in bright characters on the QTY. If the character string given in "text1" is not found in the current line, an error occurs. Following an error, a few text lines of the file are displayed, but the current line is not changed.

Classification: Text editor command.

Restrictions: The character string given in "text1" must match the characters and spaces to be changed in the current line. Do not exceed the maximum line length of 132 characters and spaces.

Examples:

```
?CH"NOW is the time";"The time is now"<CR>
```

```
.(A few lines showing the change.)  
:  
?
```

The text "NOW is the time" in the current line is changed to read "The time is now".

```
?CH"programmre";"programmer"<CR>
```

```
.(A few lines showing the change.)  
:  
?
```

The misspelled word "programmre" in the current line is change to "programmer."

CLOSELIB<CR>
or
CLOSEREF1<CR>
or
CLOSEREF2<CR>

Prerequisite: None required.

Purpose: Closes currently open user libraries. The CLOSELIB command closes the working library and any opened reference libraries, and also implies a TSTRUCT if any structures are in the creation/edit mode. A QUIT also implies a CLOSELIB. Close libraries prior to system shutdown to prevent accidental loss of data.

The CLOSEREF1 and CLOSEREF2 commands close the reference 1 library and reference 2 library respectively. These commands have no affect on the working library.

Classification: Library access command.

Restrictions: Only libraries opened at the commanding station are closed.

Examples:

?CLOSELIB<CR>
?

All opened libraries are closed.

?CLOSEREF1<CR>
?

Reference 1 library is closed.

?CLOSEREF2<CR>
?

Reference 2 library is closed.

COMMANDS<CR>
or
COMMANDS"patterns";"filename"<CR>

Prerequisite: None required.

Purpose: Outputs an alphabetic ordered list of all command names for GDS II system or outputs a partial list of command names specified by "patterns." The list is output to the QTY by default or to any file specified by "filename."

A command name is specified if "patterns" matches the name. Groups of similar command names are specified by using the asterisk and dash convention. This convention allows an asterisk (*) to represent any single character and a dash (-) to represent any number of consecutive characters in a name. More than one pattern is specified if "patterns" are separated by spaces or commas.

Classification: System utility command.

Restrictions: The first portion of the command name list will scroll off of the QTY unless AUTOHOLD is on.

Examples:

```
?COMMANDS<CR>  
.  
.(command list)  
.  
?
```

The list of commands is displayed on the QTY.

```
?COMMANDS;"commands.DC"<CR>  
?  
?G"commands.DC"<CR>  
?  
?I<CR>  
:  
:Release 00.05.04<CR>  
:  
:Date 09 06 78<CR>  
:  
:$  
?  
?W"Commands.DC"<CR>  
?  
?W"$TTO"<CR>  
?  
?
```

This example shows a typical use for the command COMMANDS and text editor commands G(get), I(insert), and W(write). First, COMMANDS outputs the list of commands to a file named "commands.DC." The G command gets the file for the text editor and the I command is used to insert a release number and date. Pressing the ESC (escape) key on the keyboard (left side, bottom key), outputs a dollar sign(\$) and stops the insert routine. The W command then writes the file into an output buffer, and then to the Decwriter for printing a hard copy of the file.

```
?COMMANDS"AB- FILL*"<CR>  
?
```

This lists all commands starting with the characters "AB" and names with five characters that start with "FILL."

COORDS<CR>
or
COORDS n<CR>

Prerequisites: Must have data in the current Item.

Purpose: COORDS with no argument prints out all the coordinates of the current item. COORDS with a positive argument prints out the first indicated number of coordinates of the current item. With a negative argument, COORDS prints out the last indicated number of coordinates in the current item.

Classification: Graphic Editor

Restrictions: None

Examples:

?COORDS<CR>

?

Prints out all coordinates of the current item.

?COORDS 10<CR>

?

Prints out the first 10 coordinates of the current item.

?COORDS -6<CR>

?

Prints out the last 6 coordinates of the current item.

CSTRUCT"source name";"destination name"<CR>

or

CSTRUCT"source name"<CR>

or

CSTRUCT<CR>

Enter the source structure name: source name<CR>

Enter the destination structure name: destination name<CR>

Prerequisites: A structure with the source name must exist in a currently open library.

Purpose: Copies a structure (source name) and the current generation to a new structure (destination name). A source name in the working library has the highest priority. The program searches for the source name in the working library first, then the REF1 library, and finally the REF2 library if they are open. The first structure name found that matches "source name" is copied into the current working library and becomes the new structure name. If a destination name was not given, it is the same as the source name.

Classification: Library access command.

Restrictions: Only the current generation of the source structure is copied.

Examples:

?CSTRUCT"grdbus";"grdbus2"<CR>

?

The current generation of an existing structure named "grdbus" is copied from any open library to the working library where it is renamed "grdbus2".

?SSTRUCT<CR>

Enter the source structure name: grdbus<CR>

Enter the destination structure name: grdbus2<CR>

?

This example does exactly the same as the first example.

?CSTRUCT"grdbus"<CR>

?

In this example, source file "grdbus" is copied from an open reference library into the working library (default destination file) and the file is not renamed. If a file "grdbus" exists in the working library, no transfer occurs.

CUTPATHIN<CR>

Enter RT Point: <CE/CR>

Enter RT Point: <CE/CR>

Prerequisites: Must have data in current Item. Must be in PATH mode.

Purpose: Creates a new geometry within the path in the current item. This new data is input to the database, and the data in item remains unaffected. The geometry created by CUTPATHIN is placed in an internal portion of the current geometry, making, in effect, the new geometry from a section of the current geometry.

For the first argument, enter the lower left point of the rectangular area.

For the second argument, enter the upper right point.

See also CUTPATHOUT.

Classification: Graphic Editor

Restrictions: The rectangular window must intersect at least one side of the temporary geometry.

CUTPATHOUT<CR>

Enter RT Point: <CE/CR>

Enter RT Point: <CE/CR>

Prerequisites: Must have data in current Item. Must be in PATH mode.

Purpose: Cuts out a rectangular area from the path in the current item in order to create or modify a new geometry. This new data is put in the database, and the data in the current item remains unaffected. The rectangular area taken out of path creates, in effect, two new geometries, one on either side of the new rectangular area.

For the first argument, enter the lower left point of the rectangular area.

For the second argument, enter the upper right point. See also CUTPATHIN.

Classification: Graphic Editor

Restrictions: The rectangular window must intersect at least two sides of the temporary geometry.

DASHED<CR>
or
DASHED n n n...n<CR>
or
DASHED -n<CR>

Prerequisites: None required.

Purpose: Specifies layers that are to be presented as dashed lines. Each entered number specifies a layer, 0-63. DASHED with "-n" (minus sign plus a number) indicates that layers 0 through n are to be dashed lines. See example below. Check MASKS for current status.

Classification: View Control

Restrictions: None

Examples:

?DASHED<CR>
?
Specifies that all layers are to be in dashed lines.

?DASHED 1 3 5<CR>
?
Specifies that layers 1,3,5 are to be dashed lines.

?DASHED -10<CR>
?
Specifies that layers 0-10 are to be dashed lines.

DATATYPE n<CR>

or

DATATYPE<CR>

Enter Datatype: n<CR>

Prerequisites: Check Item for current status.

Purpose: Sets the Datatype of the current item and the default value to the number specified (0-63). When the current item type is SREF or AREF, Datatype is not visible in ITEM. To check the default status, see DEFSTAT.

Classification: Graphic Editor

Restrictions: None

DB n <CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Deletes a number (n) of characters and spaces at the beginning of the current line. The current line is always shown in bright characters. If n is equal to or greater than the total number of characters and spaces in the current line, only the <CR> remains. The current line number is not changed.

Classification: Text editor command.

Restrictions: The argument n may be any number. Command IB is used to add characters to the beginning of the current line.

Example:

?DB 17<CR>

·
·(A few lines showing the deletion)

·
?

With this command, the line "Delete beginning characters on this line." is changed to read "characters on this line."

DE n <CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Deletes a number (n) of characters and spaces at the end of the current line. The current line is always shown in bright characters. If n is equal to or greater than the total number of characters and spaces in the current line, only the <CR> remains. The current line number is not changed.

Classification: Text editor command.

Restrictions: The argument n may be any number. Command IE is used to add characters to the end of the current line.

Example:

?DE 26<CR>

·
·(A few lines showing the deletions)

·
?

With this command, the line "characters on this line were deleted from the end" is changed to read "characters on this line."

DEBUG<CR>

Prerequisite: PROGRAMMER USE ONLY, SUSPENDS ALL OTHER SYSTEM USE.

Purpose: Starts the debugger program and suspends all other activity on the system. The debugger is a programmer's tool for finding and correcting program errors. To exit from the debugger program, press the CTRL key and then the R key at the same time.

Classification: System utility command.

Restrictions: Do not issue this command during system operation.

Example:

```
?DEBUG<CR>
.
.debugger commands
.
<CTRL/R>
?
```

The debugger is started, work performed, and the debugger exited from to return to GDS II operation.

DEFMODE<CR>
or
DEFMODE 1<CR>
or
DEFMODE 0<CR>

Prerequisites: SETDEFAULT. Check SHOWDEFAULT and ITEM.

Purpose: This command puts the system in or out of Default Mode, which returns the notions of the current item to the default properties set by SETDEFAULTS (notions are coordinate property mode, digitizing mode, layer, datatype, relative width, and texttype). DEFMODE with 1 as the argument turns the default on; with 0 as the argument turns the default off. When no argument is given, the Default Mode is toggled. When the Default Mode is on, a PUT (NULL) returns the notions in the current item to the defaults established earlier by SETDEFAULTS. If the Default Mode is off, a PUT (NULL) does not affect these notions. See also SETDEFAULT and SHOWDEFAULT Help Files.

Classification: Graphic Editor

Restrictions: None

DFUNCTION<CR>

Depress Function Button to be Defined: (Dfunction Button)

Enter Contents of Function Button

: Command Name<CR>

: Command Name<CR>

:

.

.

.

:\$

Prerequisites: None

Purpose: Used to define a function button on the alphanumeric keyboard, the 7-key stylus, or the Special Function keyboard. The Special Function keyboard contains buttons F1 - F32; the 7-key stylus contains F35 - F38; the alphanumeric keyboard contains F40 - F45. Each function button can have a string of commands, up to 64 characters, assigned to it.

Classification: Station Configuration Control

Restrictions: Commands are limited to 64 characters, which include spaces and the<CR>.

Examples:

?DFUNCTION<CR>

Depress Function Button to be Defined: (F41)

Enter Contents of Function Button

: ITEM 4<CR>

: ITEM<CR>

: \$

?

The first argument is entered by pressing function button to be defined, in this case, F41. A carriage return automatically occurs when this is pressed, and the next prompt appears on the QTY, "Enter Contents of Function Button" with an automatic carriage return and a colon to prompt for the first command name, in this case, ITEM 4. Enter a carriage return and a second command is prompted with a colon, and so on until all commands to be assigned to the function button have been entered. (It is also all right to assign just one command per button.) An ESCAPE (\$) terminates this command.

DISPLAYSTAT <CR>

Prerequisite: None required.

Purpose: Lists the status of view control (display) commands.

Classification: View control command.

Restrictions: None.

Example:

```
?DISPLAYSTAT <CR>
```

```
View Window
```

```
{min x, min y}
```

```
{max x, max y}
```

```
Grid Multiplier Count is {n}
```

```
Tablet Mode is {off or on}
```

```
Viewing Level Limits are {0 to 10}
```

```
Maximum Explosion Level:{10}
```

```
?
```

The values enclosed in brackets depend upon data entered by the operator.

DLN<CR>
or
DLN 1<CR>
or
DLN Ø<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Controls the display of line numbers for the file being edited. The DLN command without an argument acts as a toggle switch to change states of the line number display. If the display was off, it is turned on; and if the display was on, it is turned off. When a 1 argument is used, it ensures that the line number display is turned on; and when the Ø argument is used, it ensures that the line number display is turned off.

Classification: Text editor command

Restriction: The file for text editing should be displayed on the QTY.

Examples:

?DLN<CR>
?

If the line number display was off it is now on.

?DLN 1<CR>
?

Ensures that the line numbers are displayed.

?DLN Ø<CR>
?

Ensures that the line number display is suppressed.

DOTTED<CR>
or
DOTTED n n n...n<CR>
or
DOTTED -n<CR>

Prerequisites: None required.

Purpose: Specifies layers that are to be presented as dotted lines. Each entered number specifies a layer, 0-63. DOTTED with "-n" (minus sign plus a number) indicates that layers 0 through n are to be dotted lines. See example below. Check MASKS for current status.

Classification: View Control

Restrictions: None

Examples:

?DOTTED<CR>
?
Specifies that all layers are to be in dotted lines.

?DOTTED 3 8 22<CR>
?
Specifies that layers 3, 8, and 22 are to be dotted lines.

?DOTTED -20<CR>
?
Specifies that layers 0 through 20 are to be dotted lines.

DS"text"<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Deletes the first occurrence of a given "text" string from the current line only. The current line is always shown in bright characters. If "text" is a copy of all characters and spaces in the current line, only the <CR> remains. The current line number is not changed.

Classification: Text editor command.

Restrictions: The argument "text" must exactly match the characters and spaces to be deleted from the current line.

Example:

?DS"deleted"<CR>

·
·(A few lines showing the deletion)

·
?

With this command, the line "characters deleted on this line" is changed to read "characters on this line."

DSTRUCT"structure names or patterns<CR>
or
DSTRUCT<CR>
Enter structure name: structure names or patterns<CR>

Prerequisite: BSTRUCT using structure names.

Purpose: Deletes structure names or patterns of structure names from the working library and from the LISTLIB listing. However, the structures are still on disk and may be listed with a LISTGEN. While a structure is deleted, BSTRUCT can be used to create a new structure and generation with the deleted name.

To delete more than one structure name in the same library, the structure names and patterns are separated by spaces or commas. Patterns match groups of similar structure names by using the asterisk and dash convention. This convention allows an asterisk (*) to represent any single character and a dash (-) to represent any number of consecutive characters in a name.

Deleted structures can be restored with REVERT or permanently removed with EXPUNGE.

Classification: Library access command.

Restrictions: Deleted structures remain on disk unless removed by EXPUNGE.

Examples:

```
?DSTRUCT"minipads"<CR>  
?
```

A structure named "minipads" is deleted.

```
?DSTRUCT<CR>  
Enter structure name: minipads<CR>  
?
```

This does the same as the previous example.

```
?DSTRUCT"-bus inter-"<CR>  
?
```

Deletes all structures that have names ending in "bus" or starting with "inter." These could include controlbus, databus, graphicbus, interface, interconnect, and internal if these structure names were contained in the working library.

EDGE<CR>

Prerequisites: HORIZFIRST or VERTFIRST. The current item must be either PATH or BOUNDARY.

Purpose: Sets the current digitizing mode to EDGE. The mode affects points entered only when digitizing into a polygonal type item (PATH or BOUNDARY). EDGE mode follows the horiz/vert first convention. All sides input are composed of two orthogonal components connected by a 90 degree angle. Horizontal and vertical sides alternate until all sides of the geometry have been input. EDGE mode requires that a point be digitized for each side of the geometry. A 90 degree angle between sides is included in each input.

Classification: Graphic Editor

Restrictions: No 45 degree angles are input. Right angles only are input.

ENTERTEXT "string"<CR >
or
ENTERTEXT<CR>
Text: string<CR>

Prerequisites: BINDFONTS and TEXT.

Purpose: Allows a primitive form of text entry into the text item. Currently, no <CR>s can be placed in a text string.

Classification: Graphic Editor

Restrictions: None

ESTAT<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Lists the editing status of a file. The list is output to the QTY and contains the following data.

1. Name of the file in the edit buffer.
2. Number of lines in the file.
3. The current line number.
4. Number of characters in the file.
5. Whether or not the last line ends with <CR>.

Classification: Text editor command.

Restrictions: None

Example:

```
?ESTAT<CR>
Editing File="commands.dc"
Lines in File= 59
Current Line= 5
Chars in File= 4435
?
```

This is a typical listing for a file in the edit buffer.

ESTRUCT"structure name"<CR>
or
ESTRUCT<CR>
Enter Structure: structure name<CR>

Prerequisite: BSTRUCT

Purpose: Calls a structure by name and opens the file for editing the structure. Editing includes creating, adding to, deleting, or otherwise modifying the existing structure. Data is edited in the data base file and display file simultaneously. Each time a file is opened and subsequently closed by a TSTRUCT, the generation number and modification time for the structure are stored until an EXPUNGE or DSTRUCT is used to delete them. Previous generations that have not been deleted can be restored with REVERT.

Classification: Library access command.

Restrictions: Two stations cannot edit the same structure if both stations have the same working library.

Examples:

```
?ESTRUCT"minipads"<CR>  
Number of Vectors: 7  
?
```

The structure named "minipads" is opened for editing. The program also prints out the number of vectors (segments) already contained in the structure. A segment is a line between two points of a structure.

```
?ESTRUCT CR  
Enter Structure: minipads CR  
Number of Vectors: 7  
?
```

Does the same as the previous example.

EXPUNGE"library name";generations<CR>
or
EXPUNGE
Enter Library Name: library name<CR>
Enter Number of Generations to Save: generations<CR>

Prerequisite: CLOSELIB.

Purpose: Deletes the specified number of generations from all structures in a given library. If zero generations is specified, only the currently active generation of each structure is retained and all other generations are deleted. When a structure was previously deleted (DSTRUCT) and zero generations is specified, the structure name and all generations are deleted and cannot be recovered (REVERT). Inactive generations of structures are deleted to recover space in the specified library for building new structures or adding to existing structures.

Classification: Library access command.

Restrictions: A backup tape should be made to save all existing data before an EXPUNGE is issued. A library cannot be accessed during an expunge execution.

Example:

```
?EXPUNGE"lib32";4<CR>  
?
```

Retains and renumbers the last four generations of all structures in a library named "lib32". If the structure had less than four generations, the last generation was retained.

```
?EXPUNGE<CR>  
Enter Library Name: 27LIB<CR>  
Enter Number of Generations to Save: 0<CR>  
?
```

Retains and renumbers the currently active generation of all structures in a library named "27LIB". If the library contained any deleted structures, the structure names and all generations of the structures were deleted.

FILLA n<CR>
or
FILLA n n...n<CR>
or
FILLA-n; n<CR>

Prerequisite: FLAYER or FLAYERON

Purpose: Fills all boundaries on specified layers (n) with hatch lines. These hatch lines indicate the boundaries and their respective layers on the CRT display. The rotation angle and the number of lines displayed on the CRT are set using the FSETA command. To specify the layers to be indicated by FILLA, the following command forms are used:

1. One argument specifies a particular layer and excludes all others. There are 64 layers numbered 0 through 63.
2. Several arguments separated by spaces specify the listed layers and do not affect unlisted layers.
3. An argument preceded by a dash (-) specifies 0 thru n layers, where n is any number from 2 through 63.

Classification: View control command.

Restrictions: The FILLA command only affects boundaries on the specified layers.

Examples:

```
?FILLA 2<CR>  
?
```

All boundaries on layer 2 will be filled with the hatch lines defined by the FSETA command.

```
?FILLA 2 5 16<CR>  
?
```

All boundaries on layers 2, 5, and 16 will be filled with the hatch lines defined by the FSETA command.

```
?FILLA -12;<CR>  
?
```

Layers 0 thru 12 are specified.

FILLB n<CR>
or
FILLB n n...n<CR>
or
FILLB -n; n<CR>

Prerequisite: FLAYER or FLAYERON

Purpose: Fills all boundaries on specified layers (n) with hatch lines. These hatch lines indicate the boundaries and their respective layers on the CRT display. The rotation angle and the number of lines displayed on the CRT are set using the FSETB command. To specify the layers to be indicated by FILLB, the following command forms are used:

1. One argument specifies a particular layer and excludes all others. There are 64 layers numbered 0 thru 63.
2. Several arguments separated by spaces specify the listed layers and do not affect unlisted layers.
3. An argument preceded by a dash (-) specifies 0 thru n layers, where n is any number from 2 thru 63.

Classification: View control command.

Restrictions: The FILLB command only affects boundaries on the specified layers.

Examples:

```
?FILLB 2<CR>  
?
```

All boundaries on layer 2 will be filled with the hatch lines defined by the FSETB command.

```
?FILLB 2 5 16<CR>  
?
```

All boundaries on layers 2, 5, and 16 will be filled with the hatch lines defined by the FSETB command.

```
?FILLB -12;<CR>  
?
```

Layers 0 thru 12 are specified.

FILLC n<CR>
or
FILLC n n...n<CR>
or
FILLC -n; n<CR>

Prerequisite: FLAYER or FLAYERON

Purpose: Fills all boundaries on specified layers (n) with hatch lines. These hatch lines indicate the boundaries and their respective layers on the CRT display. The rotation angle and the number of lines displayed on the CRT are set using the FSETC command. To specify the layers to be indicated by FILLC, the following command forms are used.

1. One argument specifies a particular layer and excludes all others. There are 64 layers numbered 0 thru 63.
2. Several arguments separated by spaces specify the listed layers and do not affect unlisted layers.
3. An argument preceded by a dash (-) specifies 0 thru n layers, where n is any number from 2 thru 63.

Classification: View control command.

Restrictions: The FILLC command only affects boundaries on the specified layers.

Examples:

```
?FILLC 2<CR>  
?
```

All boundaries on layer 2 will be filled with the hatch lines defined by the FSETC command.

```
?FILLC 2 5 16<CR>  
?
```

All boundaries on layers 2, 5, and 16 will be filled with the hatch lines defined by the FSETC command.

```
?FILLC -12;<CR>  
?
```

Layers 0 thru 12 are specified.

FILLD n<CR>
or
FILLD n n...n<CR>
or
FILLD -n; n<CR>

Prerequisite: FLAYER or FLAYERON

Purpose: Fills all boundaries on specified layers (n) with hatch lines. These hatch lines indicate the boundaries and their respective layers on the CRT display. The rotation angle and the number of lines displayed on the CRT are set using the FSETD command. To specify the layers to be indicated by FILLD, the following command forms are used.

1. One argument specifies a particular layer and excludes all others. There are 64 layers numbered 0 thru 63.
2. Several arguments separated by spaces specify the listed layers and do not affect unlisted layers.
3. An argument preceded by a dash (-) specifies 0 thru n layers, where n is any number from 2 thru 63.

Classification: View control command.

Restrictions: The FILLD command only affects boundaries on the specified layers.

Examples:

```
?FILLD 2<CR>  
?
```

All boundaries on layer 2 will be filled with the hatch lines defined by the FSETD command.

```
?FILLD 2 5 26<CR>  
?
```

All boundaries on layers 2, 5, and 16 will be filled with the hatch lines defined by the FSETD command.

```
?FILLD -12;<CR>  
?
```

Layers 0 thru 12 are specified.

FLAYER<CR>
or
FLAYER n<CR>
or
FLAYER n n ... n<CR>
or
FLAYER -n; n<CR>

Prerequisite: None required.

Purpose: Enables layers (n) to receive hatch lines so that all boundaries on the enabled layers are indicated on the CRT display. When a layer is enabled, one of four fill commands (FILLA thru FILLD) specifies the type of hatch lines that are used. The default condition, when the system is first turned on, specifies the FILLA command with 127 hatch lines at 0 degrees rotation (horizontal lines). Initial default conditions for the other fill commands are 127 lines at 45 degrees for FILLB, 127 lines at 90 degrees (vertical) for FILLC, and 127 lines at 135 degrees for FILLD. The fill commands specify which layers are to receive each of the four types of hatch lines. The MASKS command is used to check for enabled layers and which fill command was used. To enable certain layers to receive fill commands, the following FLAYER command forms are used.

1. No argument enables all layers. There are 64 layers numbered 0 thru 63.
2. One argument enables a particular layer and disables all others.
3. Several arguments separated by spaces and no semicolons enables the listed layers and disable all others.
4. An argument preceded by a dash (-) enables layers 0 thru n, where n is any number from 2 thru 63.
5. Several arguments separated by semicolons enable the first argument or arguments separated by spaces, disables the next argument or arguments separated by spaces, and alternately enables and disables the remaining arguments at each semicolon in the list.

Classification: View control command.

Restrictions: Only boundaries on the enabled layers are filled with hatch lines.

Examples:

```
?FLAYER<CR>  
?
```

Enables all layers to use fill commands.

```
?FLAYER 3<CR>  
?
```

Enables layer 3 to use fill commands and disables all other layers.

```
?FLAYER -10; 1 5 9<CR>  
?
```

Enables layers 0 thru 10 and disables layers 1, 5, and 9 so that layers 0, 2 thru 4, 6 thru 8, and 10 are enabled.

FLAYEROFF<CR>
or
FLAYEROFF n<CR>
or
FLAYEROFF n n...n<CR>
or
FLAYEROFF -n; n<CR>

Prerequisite: None required.

Purpose: Disables layers (n) from receiving hatch lines so that boundaries on disabled layers are not filled when displayed on the CRT. Refer to FLAYER command for information on enabling layers to receive fill commands. The MASKS command is used to check for disabled layers. To disable certain layers from receiving fill commands, the following command forms are used.

1. No argument disables all layers. There are 64 layers numbered 0 thru 63.
2. One argument disables a particular layer but does not affect other layers.
3. Several arguments separated by spaces and no semicolons disables the listed layers but does not affect other layers.
4. An argument preceded by a dash (-) disables layers 0 thru n, where n is any number from 2 thru 63.
5. Several arguments separated by semicolons disable the first argument or arguments separated by spaces, enable the next argument or arguments separated by spaces, and alternately disables and enables the remaining arguments at each semicolon in the list.

Classification: View control command

Restrictions: Only boundaries on the disabled layers are affected by this command.

Examples:

```
?FLAYEROFF<CR>  
?
```

All layers are disabled from using fill commands.

```
?FLAYEROFF 3<CR>  
?
```

Disables layer 3 but does not affect other layers.

```
?FLAYEROFF -10; 1 5<CR>  
?
```

Disables layers 0 thru 10 except layers 1 and 5 so that layers 0, 2 thru 4, and 6 thru 10 are disabled, layers 1 and 5 are enabled, and all other layers are not affected.

FLAYERON<CR>
or
FLAYERON n<CR>
or
FLAYERON n n...n<CR>
or
FLAYERON -n; n<CR>

Prerequisites: None required.

Purpose: Enables layers (n) to receive hatch lines so that all boundaries on the enabled layers are indicated on the CRT display. Refer to the FLAYER command for information on enabling layers to receive fill commands. The difference between the FLAYER and FLAYERON commands is that FLAYERON will enable one or more specified layers without affecting any other layer. The MASKS command is used to check for enabled layers and which fill command was used. To enable certain layers with the FLAYERON command, the following command forms are used:

1. No argument enables all layers. There are 64 layers numbered 0 thru 63.
2. One argument enables a particular layer but does not affect any other layers.
3. Several arguments separated by spaces and no semicolons enable the listed layers but do not affect other layers.
4. An argument preceded by a dash (-) enables layers 0 thru n, where n is any number from 2 thru 63.
5. Several arguments separated by semicolons enable the first argument or arguments separated by spaces, disables the next argument or arguments separated by spaces, and alternately enables and disables the remaining arguments at each semicolon in the list.

Classification: View control command.

Restrictions: Only boundaries on the enabled layers are affected by this command.

Examples:

```
?FLAYERON<CR>  
?
```

All layers are enabled to use the fill commands.

```
?FLAYERON 3<CR>  
?
```

Enables layer 3 but does not affect other layers.

```
?FLAYERON -10; 1 5<CR>  
?
```

Enables layers 0 thru 10 except layers 1 and 5 so that layers 0, 2 thru 4, and 6 thru 10 are enabled, layers 1 and 5 are disabled, and the other layers are not affected.

```
FLIST "arg"<CR>
or
FLIST
Filename or Pattern:arg<CR>
Next Filename or Pattern:arg<CR>
```

```
.
.
.
Next Filename or Pattern:<CR>
```

Prerequisite: None required.

Purpose: Lists filenames or patterns on the QTY that match the argument (arg) parameters. The argument form is:

filename.extension

Filename matches any file name or pattern in the GDS II program that contains up to any 10 alphanumeric characters or dollar sign (\$) in any sequence. Extension is any two alpha characters that represent an existing extension. The hyphen (-) and asterisk (*) conventions from CLI may be used to represent characters. A hyphen may be substituted for any number of characters in the filename and the extension, and the asterisk is substituted for any single character. Thus, the argument -A*.S* would cause all filenames with an A as the next-to-last character and an extension starting with S to be listed.

If a /N is added at the end of any argument, filenames or patterns matching that argument are not listed. The filename "NO LINKS" inhibits any links from being listed.

Classification: System utility command.

Restrictions: If a long list is expected, the AUTOHOLD should be issued first.

Examples:

```
?FLIST "-.DB"<CR>
.
.(list of filenames matching -.DB)
.
?
```

This example causes all filenames with a .DB extension to be listed. Library names are automatically given .DB extensions.

```
?FLIST<CR>
Filename or Pattern:-.DB<CR>
Next Filename or Pattern:<CR>
.
.(list of filenames matching-.DB)
.
?
```

This example uses the literal input method to do the same thing as the previous example.

```
?FLIST<CR>
Filename or Pattern:--.<CR>
Next Filename or Pattern:NO LINKS<CR>
Next Filename or Pattern:<CR>
.
.(list of filenames)
.
?
```

All filenames except links are listed.

```
?FLIST<CR>
Filename or Pattern:A-.*W<CR>
Next Filename or Pattern:***GATE.-<CR>
Next Filename or Pattern:--.SS<CR>
Next Filename or Pattern:<CR>
.
.(list of filenames)
.
?
```

This example provides three lists as follows:

1. Filenames starting with A and having a W as the second character of the extension.
2. All filenames with seven characters that end with GATE.
3. All filenames with a .SS extension (.SS is the extension that is automatically assigned to scale files stored on disk).

FSETA<CR>
or
FSETA n; m<CR>

Prerequisite: None required.

Purpose: Lists or sets the rotation angle (n) and number of hatch lines (m) used to fill boundaries. These hatch lines are used to indicate boundaries and their respective layers on the CRT display. If the command is used without an argument, the program lists the angle and number of lines that were last set. When arguments are used, the first argument (n) sets the angle starting with 0 degrees at horizontal and 90 degrees at vertical. The second argument (m) sets the number of hatch lines from 1 through 255 that appear on the CRT display. Four commands (FSETA through FSETD) are available to set different values. The default values for FSETA, when the system is first turned on, are 0 degrees and 127 hatch lines.

Classification: View control command.

Restrictions: Any rotation angle from 0 to 360 degrees is valid but the maximum number of hatch lines is 255.

Examples:

```
?FSETA<CR>  
    0 degrees, 127 hatch lines  
?
```

Lists the values previously set at 0 degrees of rotation and 127 hatch lines.

```
?FSETA 135; 64<CR>  
?
```

Sets the rotation angle at 135 degrees and hatch lines at 64. This provides hatch lines slanting downward from top-left to bottom-right.

FSETB<CR>
or
FSETB n; m<CR>

Prerequisite: None required.

Purpose: Lists or sets the rotation angle (n) and number of lines (m) used to fill boundaries. These hatch lines are used to indicate boundaries and their respective layers on the CRT display. If the command is used without an argument, the program lists the angle and number of lines that were last set. When arguments are used, the first argument (n) sets the angle starting with 0 degrees at horizontal and 90 degrees at vertical. The second argument (m) sets the number of hatch lines from 1 thru 255 that appear on the CRT display. Four commands (FSETA through FSETD) are available to set different values. The default values for FSETB, when the system is first turned on, are 45 degrees and 127 hatch lines.

Classification: View control command.

Restrictions: Any rotation angle from 0 to 360 degrees is valid but the maximum number of hatch lines is 255.

Examples:

```
?FSETB <CR>  
    45 degrees, 127 hatch lines  
?
```

Lists the values previously set at 45 degrees of rotation and 127 hatch lines.

```
?FSETB 135; 64 <CR>  
?
```

Sets the rotation angle at 135 degrees and hatch lines at 64. This provides hatch lines slanting downward from top-left to bottom-right.

FSETC <CR>

FSETC n; m<CR>

Prerequisite: None required.

Purpose: Lists or sets the rotation angle (n) and number of hatch lines (m) used to fill boundaries. These hatch lines are used to indicate boundaries and their layers on the CRT display. If the command is used without an argument, the program lists the angle and number of lines that were last set. When arguments are used, the first argument (n) sets the angle starting with 0 degrees at horizontal and 90 degrees at vertical. The second argument (m) sets the number of hatch lines from 1 thru 255 that appear on the CRT. Four commands (FSETA through FSETD) are available to set different values. The default values for FSETC when the system is first turned on, are 90 degrees and 127 hatch lines.

Classification: View control command.

Restrictions: Any rotation angle from 0 to 360 degrees is valid but the maximum number of hatch lines is 255.

Examples:

```
?FSETC<CR>
    90 degrees, 127 hatch lines
?
```

Lists the values previously set at 90 degrees of rotation and 127 hatch lines.

```
?FSETC 135; 64<CR>
?
```

Sets the rotation angle at 135 degrees and hatch lines at 64. This provides hatch lines slanting downward from top-left to bottom-right.

FSETD<CR>
or
FSETD n; m<CR>

Prerequisite: None required.

Purpose: Lists or sets the rotation angle (n) and number of hatch lines (m) used to fill boundaries. These hatch lines are used to indicate boundaries and their respective layers on the CRT display. If the command is used without an argument, the program lists the angle and number of lines that were last set. When arguments are used, the first argument (n) sets the angle starting with 0 degrees at horizontal and 90 degrees at vertical. The second argument (m) sets the number of hatch lines from 1 thru 255 that appear on the CRT display. Four commands (FSETA through FSETD) are available to set different values. The default values for FSETD, when the system is first turned on, are 135 degrees and 127 hatch lines.

Classification: View control command.

Restrictions: Any rotation angle from 0 to 360 degrees is valid but the maximum number of hatch lines is 255.

Examples:

```
?FSETD<CR>  
    135 degrees, 127 hatch lines  
?
```

Lists the values previously set at 135 degrees of rotation and 127 hatch lines.

```
?FSETD 135; 64<CR>  
?
```

Sets the rotation angle at 135 degrees and hatch lines at 64. This provides hatch lines slanting downward from top-left to bottom-right.

G"filename"<CR>

Prerequisite: Legal RDOS filename.

Purpose: Gets a file for editing and places it in the edit buffer. Any data previously contained in the edit buffer is destroyed. A back up file is not made for the contents of the edit buffer until a W command is used to write the edit buffer into permanent file.

After the file is in the edit buffer, the following commands are used to examine and modify the file:

| | |
|---------|---|
| A | Adds lines following the current line. |
| B | Search to beginning of edit buffer. |
| CH | Change text on current line. |
| DB | Delete characters from beginning of current line. |
| DE | Delete characters from end of current line. |
| DLN | Controls display of line numbers. |
| DS | Delete text string from the current line. |
| ESTAT | List status of file in the edit buffer. |
| I | Inserts lines of text before the current line. |
| IB | Inserts "text" at beginning of current line. |
| ICOPY | Copies lines that were previously input. |
| IE | Inserts "text" at end of current line. |
| IFILE | Insert contents of another file. |
| IMOVE | Move lines to a new location. |
| J | Search to a given text line. |
| JOIN | Connects two lines to make one line. |
| K | Delete (kill) given text lines. |
| L | Search ahead for a given number of lines. |
| S | Search for a given text string. |
| SB | Search backward for a given text string. |
| SPLIT | Divide one line into two lines. |
| T | Output (type) the edit file to the QTY. |
| TEDEBUB | For programmer use only to debug text editor. |
| W | Write the edit buffer into a given file. |
| Z | Search to end of the edit buffer. |

Classification: Text editor command.

Restrictions: The "filename" must be a legal RDOS file.

Examples:

```
?G"commands.dc"<CR>  
?
```

A file named "commands.dc" is retrieved and copied into the edit buffer. The existing file is not affected.

GEDMODE<CR>
or
GEDMODE 1<CR>
or
GEDMODE 0<CR>

Prerequisites: None required.

Purpose: Puts the system in Graphic Editor mode. If GEDMODE 1 is specified, Gedmode is turned ON. If GEDMODE 0 is specified, Gedmode is turned OFF. If GEDMODE with no argument is entered, the current state is changed, from Gedmode OFF to ON, or from Gedmode ON to OFF. If coordinates are input when Gedmode is turned OFF, they cannot be evaluated as coordinates; no data is digitized. The CE command must be used to input data when Gedmode is OFF. When Gedmode is turned on, each coordinate input is interpreted as data. CE can also be used when Gedmode is ON. See CE command

Classification: Graphic Editor

Restrictions: None

GET<CE/CR>
or
GET<CR>
Coordinate:<CE/CR>

Prerequisites: Library and structure must be open. Data must be selectable.

Purpose: Gets an element--BOUNDARY, PATH, SREF, AREF, or TEXT--from a currently open structure and restores it in the current item, if empty, for editing. All properties of that element are reset into the item. A <CR> in Gedmode, or a NULL or PUT when Gedmode is off, returns the element to the database file. Check ITEM for new status.

Classification: Graphic Editor

Restrictions: The coordinate must be at the nearest vertex, not segment, of the element to be gotten.

Examples:

?GET<CR>
Coordinate: <CE/CR>
?

The coordinate is input at the nearest vertex of the element to be gotten.

GRID n <CR>
or
GRID <CR>
Enter Grid Multiplier Count: n <CR>

Prerequisite: GSCALE

Purpose: Selects every n-th grid point to turn on. The n indicates a numerical input. If n=0 the displayed grid is turned off; n=1 turns on all grid points; n=2 turns on every other grid point; n=3 turns on every third grid point; etc. Numbers up to 32767 can be entered. Select an n value that provides 200 or less points across the CRT display. Any more than 200 points are not readily seen so the system automatically turns off the points display. Also, to see the grid display, a REDRAW is issued each time GRID is issued.

Classification: View control command.

Restrictions: Negative n values are illegal. Reissue GRID if GSCALE is changed.

Examples:

```
?GRID 2 <CR>  
?
```

Turns on every other grid point.

```
?GRID <CR>  
Enter Grid Multiplier Count: 2 <CR>  
?
```

Also turns on every other grid point.

GSCALE <CR>

or

GSCALE xbase ybase xdelta ydelta <CR>

Prerequisite: INITLIB, OPENLIB

Purpose: Establishes the working grid with an offset at xbase for the x axis and ybase for the y axis and increments (space between points) are xdelta ydelta. Points on the CRT display are located at the coordinates nearest a multiple of xdelta ydelta from the starting point of xbase ybase.

The xdelta ydelta arguments are expressed in user units which are then displayed as x and y coordinates on the QTY. This makes it easier to resolve specific coordinates and sets the precision used for entering data.

A command issued without arguments clears the working grid.

Classification: Scale command.

Restrictions: Resolution established by INITLIB is not changed. The GSCALE increments cannot be smaller than the user unit established by INITLIB.

Examples:

```
?GSCALE 0 0 1 1 <CR>  
?
```

The working grid offset is x0 y0 (frequently used) and increments are one user unit between points.

```
?GSCALE <CR>  
?
```

Any previously entered working grid is cleared.

HARDLOCKS<CR>

Prerequisites: Must be in BOUNDARY or PATH. Must be in STRAIGHT mode. X and Y locks on the keyboard must be off to start with.

Purpose: Sets the current digitizing mode to HARDLOCKS. This means that the locks on the table stylus will toggle with each point entered. Either the Horizontal or the Vertical component is output first in HORIZFIRST or VERTFIRST modes, respectively. To get out of HARDLOCKS mode, simply enter another mode, such as STRAIGHT, EDGE, ORTHINT, or OCTAGONAL.

Classification: Graphic Editor

Restrictions: Must be in STRAIGHT mode.

HELP"commandname"<CR>
or
HELP<CR>
Command Name: commandname<CR>

Prerequisite: GDS II command prompt (?).

Purpose: Outputs a command description to the QTY. Any "commandname" is valid. Some command descriptions require more than the number of text lines available on the QTY. If the top lines of the command description scroll off of the display, the HOLD key on the keyboard can be pressed to stop the scrolling or an AUTOHOLD command can be issued before the HELP command. The HOLD key is located in the special 3 by 5 group of keys on the right side of the keyboard, and is the third key down the left column. The CONT key is located next to the HOLD key and is used to continue scrolling until the GDS II command prompt (?) is returned.

The AUTOHOLD command works like a toggle switch to change states between off and on each time the command is issued. When AUTOHOLD is on and there are more than 21 lines of text, the text scrolls to the top of the QTY and stops. The CONT key is pressed as required to scroll the text until the command prompt (?) is returned. AUTOHOLD should not be left on during structure creation and edit operation.

Classification: System utility command.

Restrictions: A command prompt must be returned following the command description before another command can be issued.

Example:

```
?HELP"BSTRUCT"<CR>  
.  
.(command description)  
.  
?
```

The BSTRUCT command description (HELPfile) is output to the QTY.

```
?HELP<CR>  
Command Name:BSTRUCT<CR>  
.  
.(command description)  
.  
?
```

Performs the same operation as the previous example.

HORIZFIRST<CR>

Prerequisites: ORTHINT or EDGE. The current item must be either BOUNDARY or PATH.

Purpose: Sets the current ORTHINT or EDGE operation so that the horizontal segment of each digitized input is output first. See also VERTFIRST.

Classification: Graphic Editor

Restrictions: None

```
I <CR>
:textline1<CR>
:textline2<CR>
:textline3<CR>
:.<CR>
or
I"text"<CR>
```

Prerequisite: None Required

Purpose: Inserts lines of text above the current line. The current line is always shown in bright characters on the QTY. After issuing the I command and <CR>, the program prompts for lines of text with a colon (:). Each line of is terminated with a <CR> and the program then prompts for the next line. A text line may consist of only a <CR> or up to 132 characters and spaces including the colon. To terminate the insert routine, type a single period (.) following the colon (:) and then <CR>. Also, the ESC (escape) key on the lower-left of the keyboard can be pressed to terminate the insert command. When the ESC is pressed the program exhos with a dollar sign (\$). The terminating dollar sign (\$) or period (.) are not stored in the data file. The same procedure may be used to create a file in the edit buffer.

Using the second form of the I command allows text to be inserted at the end of the current line. Text to be inserted is enclosed in double quotes and followed by <CR>, but the double quotes and <CR> are not transferred to the display or file.

Classification: Text editor command.

Restrictions: The QTY displays on 80 characters and spaces per line and then automatically skips to the next line. However, the system does not recognize a new line until a <CR> is input, so that when the file is output to a Decwriter up to 132 characters and spaces are printed on each line.

Examples:

```
?I<CR>
:Lines of text can be inserted<CR>
:<CR>
:above the current line<CR>
:<CR>
:anywhere in a file.<CR>
:.<CR>
.
.(text lines output to QTY)
:
?
```

Three lines of text, with blank lines between the text lines, are inserted in a file. The new lines are inserted starting on the line just above the current line and the current line, along with any following lines, are moved to make room. Existing lines are not deleted.

```
?A"of the current line" <CR>
.
.(text lines output to the QTY)
:
?
```

In this example, the current line was changed from "text is added to the end" to "text is added to the end of the current line."

IB"text"<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Inserts "text" at the beginning of the current line. The "text" and current line are connected and the current line number does not change. The current line is always shown in bright characters.

Classification: Text editor command.

Restrictions: With "text" inserted, the total number of characters and spaces cannot exceed 132. Command DB is used to delete characters at the beginning of a line.

Example:

?IB"Insert beginning"<CR>

.(A few lines showing inserted text.)

?

With this command, the text "characters on this line." is changed to read "Insert beginning characters on this line."

ICOPY n; m<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Copies existing lines, from line number n to line number m, and inserts the copied lines just before the current line. The original lines are also retained where they were copied from. The current line is always shown in bright characters on the QTY.

Classification: Text editor command.

Restrictions: The argument n must be less than or equal to m.

Examples:

```
?ICOPY 12; 16<CR>  
?
```

Original lines 12 through 16 are copied and inserted before the current line. If the current line was 65, then the copy of line 12 becomes line 65 and the remaining lines of the file are renumbered accordingly. The text of the current line remains the same, but the line number becomes 70. Original lines 12 through 16 remain unchanged.

```
?ICOPY 72; 72<CR>  
?
```

Line 72 is copied and inserted before the current line. If the current line number is 10, then the copy of line 72 becomes line 10 and the remaining lines in the file are renumbered. The original line 72 becomes line 73.

IE"text" <CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Inserts "text" at the end of the current line. The "text" and current line are connected and the current line number does not change. The current line is always shown in bright characters.

Classification: Text editor commands.

Restrictions: With "text" inserted, the total number of characters and spaces cannot exceed 132. Command DE is used to delete characters at the end of a line.

Example:

```
?IE"were inserted at the end"<CR>
.
.(A few lines showing inserted text.)
.
?
```

With this command, the text "characters on this line" is changed to read "characters on this line were inserted at the end."

IFILE"filename"<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Inserts the contents of another file (filename) **just** before the current line of the file being edited. Lines are renumbered as necessary in the file being edited. The current line is always shown in bright characters on the QTY.

Classification: Text editor command.

Restrictions: The "filename" must be an existing RDOS file name.

Examples:

```
?IFILE"command.dc"  
?
```

The contents of a file named "command.dc" are inserted into the edit buffer **just** before the current line.

IMOVE n; m<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Moves existing lines, specified by line number n through line number m, to just before the current line. Moved lines are deleted from their original location, and the remaining lines are renumbered. The current line is always shown in bright characters on the QTY.

Classification: Text editor command.

Restrictions: The argument n must be less than or equal to m.

Examples:

```
?IMOVE 8; 12<CR>  
?
```

Lines 8 through 12 are moved to just before the current line and all lines are then renumbered.

```
?IMOVE 32; 32<CR>  
?
```

Line 32 is moved to just before the current line and all lines are then renumbered.

```
INITLIB "filename"; "filetype"; filesize; divisor; "unit size" <CR>
or
INITLIB <CR>
Enter Library Name: filename <CR>
Reinitialize Existing Library ('YES' to Reinitialize)? Y or N <CR>
Random or Contiguous? R or C <CR>
Maximum Filesize (Pages <=50): 50 or greater <CR>
Data Base Units per User Unit: user unit divisor <CR>
User Unit Size (M,CM,MM,MICRON,A,INCH,MIL): unit size <CR>
```

Prerequisites: Library name does not exist or library is closed and data is to be destroyed.

Purpose: Creates a library with a new name or recreates a library with an existing name but destroys any data contained in the library. Filename is up to 10 alphanumeric characters and a dollar sign in any combination. If an existing file name is used, the program asks if reinitialization is desired. When using command input, the reinitialization query prints out following the first <CR>. Filetype is either random (R) or contiguous (C) file organization. Maximum filesize is given in pages (2048 bytes) with a minimum of 50. Data base units is the divisor for the user unit size so that selecting 1000 (frequently used) provides a data base resolution of 1/1000 of the user unit-size. User unit size is meters (M), centimeters (CM), millimeters (MM), microns (MICRONS), Augstroms (A), inches (INCH), or mils (MIL). The user unit can be prefixed with a number and a space if desired.

Classification: Library access command.

Restrictions: A library is required to contain the data used to build structures. The reinitialize statement is printed only if an existing library name is used. Reinitializing an existing library destroys all data contained in the library.

Example:

```
?INITLIB "123$"; "R"; 50; 1; "A" <CR>
Reinitialize Existing Library ('YES' to Reinitialize)? Y <CR>

Initialization Complete
?
```

A library named 123\$ was reinitialized using the command input method.

```
?INITLIB <CR>
Enter Library Name: $78YR <CR>
Reinitialize Existing Library ('YES' to Reinitialize)? Y <CR>
Random or Contiguous? R <CR>
Maximum Filesize (Pages >=50): 72<CR>
Data Base Units per User Unit: 1000 <CR>
User Unit Size (M,CM,MM,MICRON,A,INCH,MIL): INCH <CR>

Initialization Complete
?
```

A library named \$78YR was reinitialized using the literal input method.

INTRT<CR>

Enter RT Point:<CE/CR>

Enter RT Point:<CE/CR>

Prerequisites: Must have data in the current Item.

Purpose: Creates a new polygon within the geometry in the current item. This new data is input to the database, and the data in item remains unaffected. The rectangular area created by INTRT is placed in an internal portion of the current geometry, using at least two sides of that geometry, making, in effect, the new geometry from a section of the current geometry. For the first argument, enter the lower left point of the rectangular area. For the second argument, enter the upper right point.

Classification: Graphic Editor

Restrictions: The new rectangular area must intersect at least two sides of the geometry in the current item.

ISTAT<CR>
or
ISTAT n<CR>

Prerequisites: None required.

Purpose: ISTAT with no arguments lists in full all information in all items that have data, as contrasted with ITEM, which lists only the current Item, and not in full. ISTAT lists the current Item, if it has data, in bright. If no items have data, no listing occurs in response to this command. If a listing of the default notions is desired when no items have data, ISTAT followed by the item number desired (0-9) will produce such a listing.

Classification: Graphic Editor

Restrictions: Data must be in an Item in order for ISTAT with no argument to list data.

Examples:

```
?ISTAT<CR>
Item # 0:
Boundary
Relative Width 0. (1 Point So Far).
Layer 0, Datatype 0.
You Are In Straight Mode.
" "Relative Angle of 0. Degrees,
Relative Magnification of 1.X,
Relative No Reflection.
0 Columns, 0 Rows.
X-Spacing of 0. Units, Y-Spacing of 0. Units.
Text Type 0.
Text String:
Top Left Justification.
0 Links Allocated
```

```
Item # 1:
Path
Relative WIDTH 50. (3 Points So Far).
Layer 7, Datatype 0.
You Are In Straight Mode.
" "Relative Angle of 0. Degrees,
Relative Magnification of 1.X,
Relative No Reflection.
0 Columns, 0 Rows.
X-Spacing of 0. Units, Y-Spacing of 0. Units.
Text Type 24.
Text String:
Top Left Justification.
0 Links Allocated
```

```
?ISTAT 0<CR>
This argument displays on the QTY the notions for Item 0, as above.
```

ITEM<CR>
or
ITEM n<CR>
or
ITEM -1<CR>
or, when in Gedmode,
n<CR>
or
-1<CR>

Prerequisites: Check to see if DEFMODE is on or off. See SHOWDEFAULTS for the current defaults.

Purpose: An Item is a temporary area where data is created and modified. There are 10 items, 0 - 9.

ITEM<CR> displays the current type, data properties, and data modes of the current item and gives the number of points generated. The types are: Boundary, Path, Sref, Aref, and Text. The data properties are: Layer, Datatype, Width, and Texttype. The data modes are: Straight, Edge, Octagonal, Orthint, Horizfirst, Vertfirst for Path and Boundary types; Angle, Mag, Refl, and Shape for Sref, Aref, and Text types; and Tjust for Text type only.

ITEM followed by a positive number and a <CR> makes that designated item current. The item number will be displayed at the top of the QTY.

ITEM -1 with a <CR> looks for the first available item, starting with 0, and makes it the current item. The number will be displayed at the top of the QTY.

When in Gedmode, -1 with a <CR> looks for the first available item and makes it current; in Gedmode, a positive number, such as 6, with a <CR> makes Item 6 current. See also DEFMODE and SHOWDEFAULTS Help Files.

Classification: Graphic Editor

Restrictions: None

J n<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Searches for the given line number (n) and makes it the current line. The current line is always shown in bright characters. After command execution, five lines are displayed on the QTY with the current line in the center, unless the current line is at the beginning or end of the edit file. If the current line is at the beginning of the file, it is shown with the next two lines only. When the current line is at the end of the file, it is shown with the previous two lines only.

Classification: Text editor command.

Restrictions: The argument n must be a line number. The ESTAT command lists the number of lines in the edit file.

Example:

```
?J 10<CR>
.
.(displayed text lines)
.
?
```

Text lines number 8 through 12 are displayed on the QTY, with the current line (10) in bright characters.

JOIN<CR>

Prerequisite: A or I create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Combines the current line with the next line, deletes the <CR> in between the combined lines, and then renumbers the remaining lines. The current line is always shown in bright characters on the QTY.

Classification: The combined line cannot exceed 132 characters.

Example:

```
?JOIN<CR>  
?
```

The current line and the next line are joined on the current line and the remaining lines are renumbered.

K n<CR>
or
K -n<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Deletes (kills) the given number (n or -n) of text lines from the current line forward or from the current line backward. After command execution, the new current line is the next text line after the deleted lines, and the remaining text lines are renumbered accordingly. The current line is always shown in bright characters.

Classification: Text editor command.

Restrictions: Arguments of 1 or -1 will both delete the current line only. The argument 0 or no argument are not valid.

Examples:

```
?K 6<CR>
.
.(The new current line
.and the next two lines.)
?
```

The current line and the next five lines were deleted and the remaining lines renumbered.

```
?K -6<CR>
.
.(The new current line
.and the next two lines.)
?
```

The current line and the preceding 5 lines were deleted and the remaining lines renumbered accordingly.

L n<CR>
or
L -n<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Moves the current line a given number of lines (n or -n) either forward or backward from the existing current line. The current line is always shown in bright characters. If the result of (current line -n) is one or less than one, the first line of the file becomes the current line similar to the B command and only the first three lines are displayed on the QTY. When the result of (current line +n) is greater than the number of lines in the file, the last line of the file becomes the current line similar to the Z command and only the last three lines are displayed on the QTY.

Classification: Text editor command.

Restrictions: Zero is a legal argument, but the current line number remains unchanged.

Examples:

```
?L 10<CR>
.
.(displayed text lines)
.
?
```

This command searches 10 lines forward so if the current line was number 6 the new current line is number 16.

```
?L -5<CR>
.
.(displayed text lines)
.
?
```

This command searches 5 lines backward so if the current line was 16 the new current line number is 11.

LAYER n<CR>
or
LAYER<CR>
Enter Layer: n<CR>

Prerequisites: Before using LAYER, check ITEM and DEFSTAT for the current item and default.

Purpose: Sets the layer of the current item to the layer indicated (0-63). The default layer value is also set to the same value. If it is not desired that the default be set, use the ILAYER command instead.

Layer is not visible in ITEM when the current item type is SREF or AREF; it is visible only when the item is in BOUNDARY, PATH, or TEXT.

Classification: Graphic Editor

Restrictions: None

Examples:

?LAYER 25<CR>
?
Sets the layer of the current item to 25.

?LAYER<CR>
Enter Layer: 10 CR
?
Sets the layer of the current layer to 10.

LEVEL n<CR>

or

LEVEL<CR>

Enter Max View Explosion Level: n<CR>

Prerequisites: None required.

Purpose: Sets up the view explosion so that data nested beyond the specified level (n), 0 through 10, is not loaded in the display file with subsequent ESTRUCTS or OSTRUCTS. 10 is the default. Check DISPLAYSTAT for results. Check TREE for nested structures.

Classification: View Control

Restrictions: Can use VLEVEL command to display the selected level(s). LEVEL must be entered prior to an ESTRUCT or an OSTRUCT to have an effect.

Examples:

?LEVEL 3<CR>

?

Specifies that no levels beyond 3 will be viewable.

?LEVEL<CR>

Enter Max View Explosion Level: 8<CR>

?

Specifies that no levels beyond 8 will be viewable.

?LEVEL 10<CR>

?

Specifies that all levels will be viewable.

LINKS n<CR>

Prerequisites: None required.

Purpose: Sets the number of links allocated to the current item. For a new element being created, the number of links is 11, from 0 to 10. If the item was produced by a GET, the number of links cannot be reduced from the number in the original structure; the number can be increased, however, up to the maximum of 11.

Classification: Graphic Editor

Restrictions: None

LISTGEN "structure name"; generation number <CR>
or
LISTGEN <CR>
Structure: name <CR>
Generation: number <CR>

Prerequisites: INITLIB, OPENLIB, BSTRUCT.

^P
Purpose: Lists information about a specified generation of the named structure. Any existing generation of a structure can be listed. The command searches through all open libraries starting with the working library, then reference library 1, and finally reference library 2. The first structure found that agrees with the name and generation is listed.

Classification: Library access command

Restrictions: The generation number is changed after a structure is terminated by a TSTRUCT or the library is closed.

Examples:

?LISTGEN "orthogonal725"; 3 <CR>

| Name | Size | Gen | Created | Modified |
|---------------|------|-----|----------------|---------------|
| orthogonal725 | 6 | 3 | 16:58 07/07/78 | 8:50 07/18/78 |
| ? | | | | |

The third generation of a structure named "orthogonal725" is listed. It's size is 6 pages; it was originally created at 4:58 PM on July 7, 1978, and the third generation was created (original structure modified the second time) at 8:50 AM on July 18, 1978.

?LISTGEN <CR>
Structure: Orthogonal725 <CR>
Generation: 6 <CR>

| Name | Size | Gen | Created | Modified |
|---------------|------|-----|----------------|---------------|
| orthogonal725 | 6 | 3 | 16:58 07/07/78 | 8:50 07/18/78 |
| ? | | | | |

The sixth generation of the structure named "orthogonal725" is listed. It's size has increased to 10 pages and the sixth generation (fifth modification to the original structure) was created at 10:25 AM on August 10, 1978.

LISTLIB<CR>
or
LISTLIB"patterns"<CR>
or
LISTLIB"patterns";code<CR>
or
LISTLIB"patterns";code;"library name"<CR>

Prerequisite: INITLIB

Purpose: Lists all structures, or structures specified by "patterns," that are contained in the working library or in any initiated library specified by "library name." The following information is listed for structures in initiated libraries.

- a. structure name
- b. size of structure in pages
- c. current generation number
- d. time and date structure was created
- e. time and date structure was last modified

A structure is specified if "patterns" matches the structure name. Groups of structures with similar names are specified by using the asterisk and dash convention. This convention allows an asterisk (*) to represent any single character and a dash (-) to represent any number of consecutive characters in a name. More than one pattern is specified if "patterns" are separated by spaces or commas.

Codes 0 thru 3 are used to specify the method of sorting listed structures. A 0 code sorts structure names alphabetically, a 1 code sorts by creation times with the oldest first, a 2 code sorts by modification times with the oldest first, and a 3 code or no code (defaults) does not sort and provides the fastest listing output.

Classification: Library access command.

Restrictions: Libraries are updated after closing or after terminating a structure with a TSTRUCT.

Examples:

?LISTLIB<CR>

| | Working Library | | | |
|-------------|-----------------|-----|----------------|----------------|
| | Size | Gen | Created | Modified |
| 1735294PVA | 17 | 3 | 10:26 07/18/78 | 13:01 08/07/78 |
| MEMORYBOARD | 56 | 10 | 8:42 06/02/78 | 9:45 10/05/78 |
| MEMORYBUS | 33 | 5 | 14:32 06/29/78 | 16:10 09/15/78 |
| ANDORMAYBE | 6 | 2 | 13:26 07/07/78 | 15:15 09/29/78 |
| APPLESWITCH | 13 | 4 | 9:19 08/18/78 | 17:16 10/10/78 |
| SAMS | 11 | 4 | 16:31 07/17/78 | 10:52 08/31/78 |
| SLIM | 21 | 9 | 11:40 06/16/78 | 8:53 07/28/78 |
| SLICE | 14 | 3 | 13:02 06/16/78 | 11:57 10/10/78 |
| SIMULATOR | 46 | 14 | 15:37 08/15/78 | 9:31 10/10/78 |
| ? | | | | |

All structures in the working library are listed without sorting.

?LISTLIB"A- S***";1;"BIGSYSTEM"<CR>

| | Size | Gen | Created | Modified |
|-------------|------|-----|----------------|----------------|
| SLIM | 21 | 9 | 11:40 06/16/78 | 8:53 07/28/78 |
| ANDORMAYBE | 6 | 2 | 13:26 07/07/78 | 15:15 09/29/78 |
| SAMS | | 4 | 16:31 07/17/78 | 10:52 08/31/78 |
| APPLESWITCH | 11 | 4 | 9:91 08/18/78 | 17:16 10/10/78 |
| ? | | | | |

All structure names beginning with A and structure names beginning with S and having four characters are listed by creation dates (1) if they are contained in a closed library named "BIGSYSTEM."

?LISTLIB"MEMORY- S-";2;"BIGSYSTEM"<CR>

| | Size | Gen | Created | Modification |
|-------------|------|-----|----------------|----------------|
| SLIM | 21 | 9 | 11:40 06/16/78 | 8:53 07/28/78 |
| SAMS | 11 | 4 | 16:31 07/17/78 | 10:52 08/31/78 |
| MEMORYBUS | 33 | 5 | 14:32 06/29/78 | 16:10 09/15/78 |
| MEMORYBOARD | 56 | 10 | 8:42 06/02/78 | 9:45 10/05/78 |
| SIMULATOR | 46 | 14 | 15:37 08/15/78 | 9:31 10/10/78 |
| SLICE | 14 | 3 | 13:02 06/16/78 | 11:57 10/10/78 |
| ? | | | | |

All structure names beginning with MEMORY and S* are listed by modification dates (2) if they are contained in a closed library named "BIGSYSTEM."

```

LISTREF1<CR>
or
LISTREF1"patterns"<CR>
or
LISTREF1"patterns";code<CR>

LISTREF1"patterns";code;"library name"<CR>

```

Prerequisite: INITLIB

Purpose: Lists all structures, or structures specified by "patterns," that are contained in reference library 1 or in any initiated library specified by "library name." The following information is listed for structures in initiated libraries.

- a. structure name
- b. size of structure in pages
- c. current generation number
- d. time and date structure was created
- e. time and date structure was last modified

A structure is specified if "patterns" matches the structure name. Groups of structures with similar names are specified by using the asterisk and dash convention. This convention allows an asterisk (*) to represent any single character and a dash (-) to represent any number of consecutive characters in a name. More than one pattern is specified if "patterns" are separated by spaces or commas.

Codes 0 thru 3 are used to specify the method of sorting listed structures. A 0 code sorts structure names alphabetically, a 1 code sorts by creation times with the oldest first, a 2 code sorts by modification times with the oldest first, and a 3 code or no code (default) does not sort and provides the fastest listing output.

Classification: Library access command.

Restrictions: Libraries are updated after closing or after terminating a structure with a TSTRUCT.

Examples:

```
?LISTREF1<CR>
```

| | Reference Library 1 | | Created | Modified |
|-------------|---------------------|-----|----------------|----------------|
| | Size | Gen | | |
| 1735294PVA | 17 | 3 | 10:26 07/18/78 | 13:01 08/07/78 |
| MEMORYBOARD | 56 | 10 | 8:42 06/02/78 | 9:45 10/05/78 |
| MEMORYBUS | 33 | 5 | 14:32 06/29/78 | 16:10 09/15/78 |
| ANDORMAYBE | 6 | 2 | 13:26 07/07/78 | 15:15 09/29/78 |
| APPLESWITCH | 13 | 4 | 9:19 08/18/78 | 17:16 10/10/78 |
| SAMS | 11 | 4 | 16:31 07/17/78 | 10:52 8/31/78 |
| SLIM | 21 | 9 | 11:40 06/16/78 | 8:53 07/28/78 |
| SLICE | 14 | 3 | 13:02 06/16/78 | 11:57 10/10/78 |
| SIMULATOR | 46 | 14 | 15:37 08/15/78 | 9:31 10/10/78 |
| ? | | | | |

All structures in the working library are listed without sorting.

?LISTREF1"A- S***";1;"BIGSYSTEM"<CR>

| | Size | Gen | Created | Modified |
|-------------|------|-----|----------------|----------------|
| SLIM | 21 | 9 | 11:40 06/16/78 | 8:53 07/28/78 |
| ANDORMAYBE | 6 | 2 | 13:26 07/07/78 | 15:15 09/29/78 |
| SAMS | 11 | 4 | 16:31 07/17/78 | 10:52 08/31/78 |
| APPLESWITCH | 13 | 4 | 9:91 08/18/78 | 17:16 10/10/78 |
| ? | | | | |

All structure names beginning with A and structure names beginning with S and having four characters are listed by creation dates (1) if they are contained in a closed library named "BIGSYSTEM."

?LISTREF1"MEMORY- S-";2;"BIGSYSTEM"<CR>

| | Size | Gen | Created | Modification |
|-------------|------|-----|----------------|----------------|
| SLIM | 21 | 9 | 11:40 06/16/78 | 8:53 07/28/78 |
| SAMS | 11 | 4 | 16:31 07/17/78 | 10:52 08/31/78 |
| MEMORYBUS | 33 | 5 | 14:32 06/29/78 | 16:10 09/15/78 |
| MEMORYBOARD | 56 | 10 | 8:42 06/02/78 | 9:45 10/05/78 |
| SIMULATOR | 46 | 14 | 15:37 08/15/78 | 9:31 10/10/78 |
| SLICE | 14 | 3 | 13:02 06/16/78 | 11:57 10/10/78 |
| ? | | | | |

All structure names beginning with MEMORY and S are listed by modification dates (2) if they are contained in a closed library named "BIGSYSTEM."

LISTREF2<CR>
or
LISTREF2"patterns"<CR>
or
LISTREF2"patterns";code<CR>
or
LISTREF2"patterns";code;"library name"<CR>

Prerequisite: INITLIB

Purpose: Lists all structures, or structures specified by "patterns," that are contained in reference library 2 or in any initiated library specified by "library name." The following information is listed for structures in initiated libraries.

- a. structure name
- b. size of structure in pages
- c. current generation number
- d. time and date structure was created
- e. time and date structure was last modified

A structure is specified if "patterns" matches the structure name. Groups of structures with similar names are specified by using the asterisk and dash convention. This convention allows an asterisk (*) to represent any single character and a dash (-) to represent any number of consecutive characters in a name. More than one pattern is specified if "patterns" are separated by spaces or commas.

Codes 0 thru 3 are used to specify the method of sorting listed structures. A 0 code sorts structure names alphabetically, a 1 code sorts by creation times with the oldest first, a 2 code sorts by modification times with the oldest first, and a 3 code or no code (default) does not sort and provides the fastest listing output.

Classification: Library access command.

Restrictions: Libraries are updated after closing or after terminating a structure with a TSTRUCT.

Examples:

?LISTREF2<CR>

| | Reference Library 2 | | | |
|-------------|---------------------|-----|----------------|----------------|
| | Size | Gen | Created | Modified |
| 1735294PVA | 17 | 3 | 10:26 07/18/78 | 13:01 08/07/78 |
| MEMORYBOARD | 56 | 10 | 8:42 06/02/78 | 9:45 10/05/78 |
| MEMORYBUS | 33 | 5 | 14:32 06/29/78 | 16:10 09/15/78 |
| ANDORMAYBE | 6 | 2 | 13:26 07/07/78 | 15:15 09/29/78 |
| APPLESWITCH | 13 | 4 | 9:19 08/18/78 | 17:16 10/10/78 |
| SAMS | 11 | 4 | 16:31 07/17/78 | 10:52 08/31/78 |
| SLIM | 21 | 9 | 11:40 06/16/78 | 8:53 07/28/78 |
| SLICE | 14 | 3 | 13:02 06/16/78 | 11:57 10/10/78 |
| SIMULATOR | 46 | 14 | 15:37 08/15/78 | 9:31 10/10/78 |
| ? | | | | |

All structures in the working library are listed without sorting.

?LISTREF2"A- S***";1;"BIGSYSTEM"<CR>

| | Size | Gen | Created | Modified |
|-------------|------|-----|----------------|----------------|
| SLIM | 21 | 9 | 11:40 06/16/78 | 8:53 07/28/78 |
| ANDORMAYBE | 6 | 2 | 13:26 07/07/78 | 15:15 09/29/78 |
| SAMS | 11 | 4 | 16:31 07/17/78 | 10:52 08/31/78 |
| APPLESEITCH | 13 | 4 | 9:91 08/18/78 | 17:16 10/10/78 |
| ? | | | | |

All structure names beginning with A and structure names beginning with S and having four characters are listed by creation dates (1) if they are contained in a closed library named "BIGSYSTEM."

?LISTREF2"MEMORY- S-";2;"BIGSYSTEM"<CR>

| | Size | Gen | Created | Modification |
|-------------|------|-----|----------------|----------------|
| SLIM | 21 | 9 | 11:40 06/16/78 | 8:53 07/28/78 |
| SAMS | 11 | 4 | 16:31 07/17/78 | 10:52 08/31/78 |
| MEMORYBUS | 33 | 5 | 14:32 06/29/78 | 16:10 09/15/78 |
| MEMORYBOARD | 56 | 10 | 8:42 06/02/78 | 9:45 10/05/78 |
| SIMULATOR | 46 | 14 | 15:37 08/15/78 | 9:31 10/10/78 |
| SLICE | 14 | 3 | 13:02 06/16/78 | 11:57 10/10/78 |
| ? | | | | |

All structure names beginning with MEMORY and S are listed by modification dates (2) if they are contained in a closed library named "BIGSYSTEM."

LOWERCASE<CR>

Prerequisite: None required.

Purpose: Causes all subsequent alphabetic input from the keyboard to be output to the QTY in all lower case (non-capital) letters regardless of the shift key position. The default condition at power turn on is upper and lower case letters exactly as typed. To restore the default condition type UPLOWCASE or to output all capital letters type UPPERCASE.

Classification: Station configuration control command.

Restrictions: Do not use lower case letters where upper case letters are needed to duplicate a structure name.

Example:

```
?LOWERCASE<CR>  
?
```

All typed text is output to the QTY in non-capital letters.

LSTAT <CR>

Prerequisites: INITLIB and OPENLIB

Purpose: Lists the status of all opened libraries. The following information is listed for each library; library type, library name, size of library in total pages, available space in pages, and percentage of pages used.

Classification: Library access command.

Restrictions: Only opened libraries are listed.

Example:

?LSTAT <CR>

| Library | Filename | Total Pages | Free Pages | Used % |
|---------|--------------|-------------|------------|--------|
| WORKING | TCPLOGIC.DB | 478 | 454 | 5% |
| REF1 | TCPLOGIC1.DB | 63 | 56 | 10% |
| REF2 | TCPLOGIC2.DB | 38 | 35 | 6% |
| ? | | | | |

MAG n<CR>
or
MAG<CR>
Enter Magnification: n<CR>

Prerequisites: Check ITEM. SNAME must be entered. Must be in SREF, AREF, or TEXT.

Purpose: Sets the magnification for a structure reference or text nodes to the magnification specified. The number specified must be positive. The magnification is relative, which means that if subsequent conditions are created or changed in the structure, the magnification will also be changed to relate with those conditions. See ABSMAG for an explanation of the absolute condition.

Classification: Graphic Editor

Restrictions: Must be a number greater than 0.

Examples:

?MAG 3<CR>

?

Sets the relative magnification for a structure reference or text node to 3.

?MAG<CR>

Enter Magnification: 2<CR>

?

Sets the relative magnification to 2.

MARKSIZE n<CR>
or
MARKSIZE<CR>
Marker Size (Half-Width): n<CR>

Prerequisites: None assigned.

Purpose: Sets the size of the marker on the CRT for IDWINDOW, IDGET, GET, UNWINDOW, UNGET, and IDMARK to the size specified. The size given is the half-width of the mark used; .01 is the default size. The stepping distance for inputting units on the CRT is 1.0.

Classification: View Control

Restrictions: None

Examples:

?MARKSIZE .005<CR>

?

Sets the marker size to one half the size of the standard marker.

?MARKSIZE<CR>

Marker Size (Half-Width): .02<CR>

?

Sets the marker size to twice that of the standard marker.

MASKS<CR>

Prerequisites: None required.

Purpose: Gives current status of visible and selectable data for layers 0-63. The following symbols are used to indicate information about the data:

- @ = solid lines (also default condition)
 - ! = dashed lines
 - = dotted lines
 - ! = broken lines
 - + = selectable but not visible data
 - = layer, datatype, texttype, fill, and kinds not viewed or selected
- } These three symbols are used only for VLAYER.

Classification: View Control

Restrictions: None

Examples:

| MASKS | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|--------|----------------------------|---------------------------|---------------------------|---------------------------|----------------------|---------------------|---------------------|
| VLAYER | 0123456789 -!@@!@.@@! | 0123456789 @@@@@@@@@@@ | 0123456789 @@@@@@@@@@@ | 0123456789 @@@@@@@@@@@ | 0123456789 @----- | 0123456789 ----- | 0123456789 ---@! |
| SLAYER | -@@@@@@@- | ----- | ----- | ----- | ----- | ----- | ----- |
| VDTYPE | -@@@@@@@@@ | -@@@@@@@@@ | @@@@@@@@@@@ | @@@@@@@@@@@ | @@@@@@@@@@@ | @@@@@@@@@@@ | @@@@@@@@@@@ |
| SDTYPE | +@@@@@@@@@ | +----- | ----- | ----- | ----- | ----- | ----- |
| VTTYPE | @@@@@@@@@@@ | @@@@@@@@@@@ | @@@@@@@@@@@ | @@@@@@@@@@@ | ----- | ----- | ----- |
| STTYPE | @@@@@@@@@@@ | @@@@@@@@@@@ | @@@@@@@@@@@ | @@@@@@@@@@@ | +++++ | +++++ | +++ |
| FLAYER | --B---C-- | ADAAA----- | ----- | ----- | ----- | ----- | ----- |
| VKIND | BD PB TX TO SO AO SN SE AE | | | | | | |
| SKIND | BD (PC) PB SO AO SN SE AE | | | | | | |

Example indicates:

VLAYER: Layers 0, 41-61 are not viewed.
 Layers 1, 9 are broken lines.
 Layer 4 is dashed lines.
 Layer 6 is dotted lines.
 All other Vlayers are viewed solid.

SLAYER: Layers 1-7 are selectable only.

VDTYPE: All datatypes except on layers 0 and 10 are viewable.

SDTYPE: Datatypes on layers 0 through 10 are selectable. Layers 11-63 are not selectable.

VTTYPE: Texttypes on layers 0-39 are viewable; layers 40-63 are not viewable.

STTYPE: All texttypes are selectable.

FLAYER: Only the layers marked A, B, C, and D are filled; the rest are not filled. Defaults for the four types of fill are listed below:

?FSETA
0 degrees, 127 hatch lines
?FSETB
45 degrees, 127 hatch lines
?FSETC
90 degrees, 127 hatch lines
?FSETD
135 degrees, 127 hatch lines

VKIND: All kinds but PC (Path Center) are viewable.

SKIND: BD, PC, PB, SO, AO, SN, SE, and AE are selectable.
PC appears as selectable but not viewable.
TX (Text) and TO (Text Origin) are not selectable.
See VKIND Help File for list of kinds.

Commands that selected the conditions for the layers in the above example are as follows:

?DASHED 4
?BROKEN 1 9
?DOTTED 6
?VLAYER -40;0;62 63
?SLAYER -7;0
?VLAYER -63;0 10
?SDTYPE -10
?VTTYPE -39
?STTYPE
?FLAYER 2 7 10 11 12 13 14
?FILLA 10 12 13 14
?FILLB 2
?FILLC 7
?FILLD 11
?VKINDON; "PC"
?SKINDON; "PC"

MERGELIB <CR>

Prerequisites: Two opened libraries.

Purpose: Duplicates the contents of a reference library in the current working library. As each structure is copied, its name is listed on the QTY and when all structures are copied the program outputs "merge complete". If a console abort is used during a merge, the abort is delayed until the structure being copied is complete. Libraries are merged to organize the contents for optimum storage and retrieval, or to change parameters of the library. The library parameters that can be changed (see INITLIB) are library name, file organization, and filesize. A MERGE will not work if the two libraries have different data base units or user units.

To change library parameters, first initiate a library (INITLIB) with the desired new parameters. Next, open the new library as the first argument to the OPENLIB command and the library to be merged as the second argument. Finally, type MERGELIB and the merging is done automatically.

Classification: Library access command.

Restrictions: If the merging libraries contain duplicate structure names, the duplicate structures are not merged. The working library size must be large enough to accept the merging structures. Both libraries must have the same data base units and user units.

Example:

```
?MERGLIB <CR>
Merging Structure
Microdot Copied
Starsandbars Copied
Linesandcodes Copied
Merge Complete
?
```

Three structures were copied into the working library.

MOVE<CE/CR>
Displacement:<CE/CR>
or
MOVE<CR>
Coordinate(s):<CE/CR>
Displacement:<CE/CR>
or
MOVE
Coordinate(s):<CE><CE/CR>
Displacement:<CE><CE/CR>

Prerequisites: Data must be in current Item.

Purpose: In the current item this command moves a point, or a pair of points, digitized by the first argument to the displacement point computed by the second argument. The CEs in the first argument have two forms:

- One point - Computes the nearest point to the coordinate entered.
- Two points - Computes the points of the current item that are within the window area determined by these two coordinates.

The points identified in the first argument are marked with diamonds before the prompt for the second argument is given.

The displacement is calculated in two ways:

- One point - is moved to the displacement point input in the second argument. If a window of points was input in the first argument, the displacement is computed so that the second point entered in the first argument is moved to the first point entered in the second argument (displacement). The same displacement occurs for all points within the window.

- Two points - If two points were entered in the first argument, the displacement is computed by subtracting the first displacement coordinate from the second displacement coordinate. The new displacement point is the difference between the two.

Classification: Graphic Editor

Restrictions: None

OCTAGONAL<CR>

Prerequisites: The current item must be either PATH or BOUNDARY.

Purpose: Sets the current digitizing mode to OCTAGONAL. In this mode all segments output will consist of an orthogonal component followed by a 45 degree line, in one of four directions. Depending upon whether the distance between the two points is greater in the x axis or in the y axis, the 90 degree angle, from which the 45 degree angle is derived, is entered first in whichever distance is greater.

Classification: Graphic Editor

Restrictions: None

OPENLIB "library name";"library name";"library name" <CR>

or

OPENLIB <CR>

Enter the Working Library Name: library name <CR>

Prerequisites: INITLIB

Purpose: Opens up to three libraries for storing user data. The first argument is always the working library. When a second and third library are opened they are reference libraries 1 and 2 respectively. Any library can be opened as a working library and the current working library receives all the created and modified data at a station.

Classification: Library access command.

Restrictions: Only one working library can be opened with a maximum of two reference libraries. Any previously opened libraries are closed automatically when a new library is opened.

Examples:

```
?OPENLIB "micromem";"buckets";"contacts" <CR>
```

```
?
```

Three libraries were opened with "micromem" as the working library, "buckets" as REF1 library, and "contacts" as the REF2 library.

```
?OPENLIB <CR>
```

```
Enter Working Library Name: contacts <CR>
```

```
?
```

"Contacts" is opened as the working library and all other libraries are closed. Only one library can be opened using this command form.

OPENREF1"library name"<CR>
or
OPENREF1<CR>
Reference 1 Library Name: library name<CR>

Prerequisite: INITLIB

Purpose: Opens any initiated library as the reference 1 library. A reference library is used to contain structures that are referenced from a working library. The same library may be opened at more than one station.

Classification: Library access command.

Restrictions: Structures cannot be created or modified in a reference library.

Example:

```
?OPENREF1"buckets"<CR>  
?
```

A previously initiated library named "buckets" is opened as the reference library.

```
?OPENREF1<CR>  
Reference 1 Library Name: buckets<CR>  
?
```

Performs the same operation as the previous example except that the literal input method is used.

OPENREF2"library name"<CR>
or
OPENREF2<CR>
Reference 2 Library Name: library name<CR>

Prerequisite: INITLIB

Purpose: Opens any initiated library as the reference 2 library. A reference library is used to contain structures that are referenced from a working library. The same library may be opened at more than one station.

Classification: Library access command.

Restrictions: Structures cannot be created or modified in a reference library.

Examples:

```
?OPENREF2"buckets"<CR>  
?
```

A previously initiated library named "buckets" is opened as the reference 2 library.

```
?OPENREF2<CR>  
Reference 2 Library Name: buckets<CR>  
?
```

Performs the same operation as the previous example except that the literal input method is used.

ORTHINT<CR>

Prerequisites: HORIZFIRST or VERTFIRST. The current item must be either PATH or BOUNDARY.

Purpose: Sets the current digitizing mode to ORTHINT. In this mode, all points output to polygonal items are composed of two orthogonal components (two lines at right angles to each other). ORTHINT mode requires that a point be digitized for every other vertex, each at a 90 degree angle, with a 90 degree angle between inputs. Horizontal-first and vertical-first components are entered alternately until all sides of the geometry have been input (the last component is input by a <CR>). Thus, ORTHINT mode inputs a geometry quickly insofar as it enters sides two at a time.

Classification: Graphic Editor

Restrictions: No 45 degree angles are input; only right angles are input.

OSTRUCT"structure name"<CR>
or
OSTRUCT<CR>
Enter Structure: structure name<CR>

Prerequisite: BSTRUCT

Purpose: Calls a structure by name and opens the file for editing the structure. This command is similar to ESTRUCT except that a new generation is not created and so there is no backup for the previous generation. Since this command uses less disk space, it is useful for editing minor structures. Do not use this command where there is a possibility of losing data without a backup in the file. The OSTRUCT and INPLACE commands are identical.

Classification: Library access command.

Restrictions: Two stations cannot edit the same structure if both stations have the same working library.

Examples:

```
?OSTRUCT"minipads"<CR>  
?
```

The structure named "minipads" is open for editing without a backup.

```
?OSTRUCT<CR>  
Enter Structure: minipads<CR>  
?
```

Does the same as the previous example.

PAN<CE><CR>

or

PAN

Coordinate: <CE/CR>

Prerequisite: Data display.

Purpose: Sets the selected coordinate (<CE>) to the center of the CRT display. It has the effect of moving current view window on the CRT display without changing the coordinate location of the data.

Classification: View control command.

Restrictions: None.

Examples:

?PAN<CE><CR>

?

When the GDS II prompt is returned, the coordinate <CE> is centered in the CRT display.

?PAN<CR>

Coordinate: <CE/CR>

?

Does the same as the previous example.

PATH<CR>

Prerequisites: A library and a structure must both be open, and Width must be defined. Check ITEM or ISTAT for status.

Purpose: Declares the type of the current item to be PATH. If the current item already has data that has not yet been NULLED (or PUT), an error occurs. Check ITEM first to avoid this problem. Go to the next available item.

Classification: Graphic Editor

Restrictions: None

Examples: This example shows the normal command sequence for setting up a PATH, in this case in ORTHINT mode. ITEM then lists the conditions that have been set:

```
?LAYER<CR>
Enter Layer: 2
?DATATYPE<CR>
Enter Datatype: 9
?WIDTH<CR>
Enter Width In Natural Units: 1.25
?PATH<CR>
?ORTHINT<CR>
?ITEM<CR>
You Are Editing Item # 0
It is a Path with Relative Width 1.25 (0 Points So Far).
Layer 2, Datatype 9.
You Are In Orthint Mode (Horizontal-First).
0 Links Allocated
?
```

PATHMARK<CR>

Prerequisites: PATH

Purpose: Causes a mark to be placed at the origin point of each selectable path so that it is apparent which geometries are paths and that they are at level 0, the level at which data is created. The Pathmark is erased when the geometry is redrawn.

Classification: View Control

Restrictions: None

PUT<CR>
or, when in Gedmode,
<CR>

Prerequisites: Structure must be open for editing, with data in Item.

Purpose: Outputs the current item to the database and views it on the screen. Lines that were "hidden" while being input are now made solid (or broken, dotted, dashed, as per the condition set). Coordinates must be in the current item in order for a PUT to occur. Once a PUT has been accomplished, the defaultable properties in item are reset to the current default values. See DEFSTAT. Check the default values in item by using ISTAT. See also NULL for the other version of this command.

Classification: Graphic Editor

Restrictions: If there are not enough points to create a BOUNDARY, an error will occur: "Not Enough Data to Make Boundary".

PUTALL<CR>

Prerequisites: Structure must be open for editing, with data in Item.

Purpose: Puts all items that have coordinates into the database. PUTALL is useful when the user is not sure he has PUT individual items in the database. PUTALL will prevent loss of data that otherwise might occur. See PUT; see also ISTAT for a display of all items that have data. See also NULLALL.

Classification: Graphic Editor

Restrictions: None

QUIT<CR>

Prerequisite: TSTRUCT

Purpose: Release the GDS II station for other use or before shutting down the system. If all stations on a system issue a QUIT, the system is shutdown and control returns to the CLI program. The QUIT command implies a CLOSELIB and a TSTRUCT that close all open libraries and structures. A QUIT station can be restarted with a START command.

Classification: System utility command.

Restrictions: Loss of data is possible if any structures have open items.

Example:

```
?QUIT<CR>  
STATION n DOWN
```

The station issuing the command (n=station number) is now released from the GDS II program.

```
R3<CE><CE><CE/CR>
or
R3<CR>
CE:<CE><CE><CE/CR>
CE:<CE><CE><CE/CR>,
.
.
.
CE:<CR>
```

Prerequisites: Check DEFSTAT for current properties. Structure must be open.

Purpose: Using three coordinates, R3 creates a rectangle, or in the interactive form, several rectangles, using the current item's layer and datatype. The rectangle is specified in the following way: The first point inputs one corner, and the second point inputs the next corner, thus inputting the angle at which the rectangle will be drawn. The third point input specifies the line of the edge opposite to the edge specified by the first two points, thus creating the area of the rectangle.

Classification: Graphic Editor

Restrictions: Must enter three coordinates per line (but no more than three), using the CE button alone for the first two and the CE/CR button for the third point. The rectangle cannot have 0 area. If the polygon generated has vertex angles that will not pass the all-angles test for 90 degrees, an error occurs.

```
RECOVERLIB "library name" <CR>
or
RECOVERLIB "library name"; Ø <CR>
or
RECOVERLIB <CR>
Enter Library Name: library name <CR>
```

Prerequisites: None required.

Purpose: Recovers a working library that was not normally closed. A library is normally closed by opening another library, or by issuing a CLOSELIB or QUIT command. RECOVERLIB verifies the internal format of the specified library unless the second argument (Ø) is used. The Ø specifies a fast recovery sequence that clears in-use flags but data is not verified for correctness. Fast recovery is used if an abnormal shut down occurred before any data was last opened. When recovery is complete the program outputs "Library Recovery Completed."

Classification: Library access command.

Restrictions: Only the working library is recovered.

Examples:

```
?RECOVERLIB "micromem" <CR>
Library Recovery Completed
?
```

```
?RECOVERLIB "micromem"; Ø <CR>
Library Recovery Completed.
?
```

```
?RECOVERLIB <CR>
Enter Library Name: micromem <CR>
Library Recovery Completed
?
```

The three ways of recovering the library "micromem" are shown.

REDRAW <CR>

Prerequisite: None required.

Purpose: Erases and redraws the CRT display (view). An erase and redraw is necessary following several commands such as GRID, UNDO, and WIPE in order to see a change in the display. REDRAW and ERASE are identical commands.

Classification: View control command.

Restrictions: Only the CRT display is erased and redrawn.

Example:

```
?REDRAW <CR>  
?
```

When the GDS II prompt (?) is returned the CRT screen is erased and all data contained in the display file is redrawn.

REFL "string" <CR>
or
REFL <CR>
Reflection (YES/NO): "string" <CR>

Prerequisites: Check ITEM. Must be in AREF, SREF, or TEXT.

Purpose: Sets the reflection mode of the current structure or array reference in item. To indicate that a reflection is specified, use "Y", "YES", "y", "yes", or just a <CR> for "string". If no reflection is desired, use "N", "NO", "n", or "no". The reflection is relative, which means that if subsequent conditions are created or changed in the structure, the reflection will be changed also to relate to those conditions. The reflection is always in the y axis.

Classification: Graphic Editor

Restrictions: REFL relates to SREF, AREF, or TEXT.

REVERSE<CR>

Prerequisites: Check COORDS.

Purpose: Reverses the digitizing order in the current item. Useful for digitizing off the other end of a PATH or BOUNDARY. Input REVERSE<CR>again to get out of REVERSE.

Classification: Graphic Editor

Restrictions: None

```
REVERT"structure name";-generations<CR>
or
REVERT<CR>
Structure: structure name<CR>
Generation: -generations<CR>
```

Prerequisite: DSTRUCT

Purpose: Restores previous generations of a structure that were deleted. A minus one (-1) restores the previous generation, minus four (-4) restores the previous four generations, etc. All generations not reverted are lost. LISTGEN can be used to determine how many generations to revert. To avoid the loss of data, a CSTRUCT will copy existing generations into another file before the REVERT is performed.

Classification: Library access command.

Restrictions: Positive generation number is illegal.

Examples:

```
?REVERT"micropin";-1<CR>
?
```

The previous generation of a structure named "micropin" is reverted. If the current generation was 6, then the reverted generation is 5 and generation 6 is lost.

```
?REVERT<CR>
Structure: micropad2<CR>
Generation:Ø <CR>
?
```

The current generation of the structure named "micropad2" is number 6. Using Ø in the command restores all generations to active files.

RSCALE "scalefilename" <CR>
or
RSCALE <CR>
Scale Status File: scalefilename <CR>

Prerequisites: WSCALE

Purpose: Reads the scale files that were stored on disk by a WSCALE command and sets the scale values accordingly. If no other scalefilename extension is given .SS is assumed.

Classification: Scale command.

Restrictions: Tablet mode must be off (TMOFF). After a system shut down or when a drawing has been moved, it is necessary to reissue the rotation and translation commands to correct any scaling errors.

Examples:

?RSCALE "THIRDSCALE" <CR>

Reads a scaling file named "THIRDSCALE.SS" and sets scale values.

?RSCALE <CR>
Scale Status File: FOURTHSCALE.RD <CR>
?

Reads a scaling file named "FOURTHSCALE.RD" and sets the scale values.

```
RSTRUCT"structure name";"new name"<CR>
or
RSTRUCT"structure name"<CR>
Enter new structure name: new name<CR>
or
RSTRUCT<CR>
Enter structure name: structure name<CR>
Enter new structure name: new name<CR>
?
```

Prerequisites: BSTRUCT

Purpose: Renames the specified structure within the working library. Name changes are performed before a structure is placed with an SREF because the name in the data base is not changed if the previous name was referenced into another structure.

Classification: Library access command.

Restrictions: Do not rename after placement.

Examples:

```
?RSTRUCT"databus";"interface"<CR>
?
```

The structure name "databus" was changed to "interface" in the working library.

```
?RSTRUCT"databus"<CR>
Enter new structure name: interface<CR>
?
```

Does the same as the previous example.

```
?RSTRUCT<CR>
Enter structure name: databus<CR>
Enter new structure name: interface<CR>
?
```

Does the same as the previous example.

RT<CE><CE/CR>
or
RT<CR>
CE:<CE><CE/CR>
CE:<CE><CE/CR>
.
.
.
CE<CR>

Prerequisites: Check DEFSTAT for current properties. Structure must be open.

Purpose: Using two coordinates, RT creates a rectangle, or in the interactive form, many rectangles, using the current item's datatype and layer. The rectangles are in diagonal, two-point form.

Classification: Graphic Editor

Restrictions: Must enter two coordinates per line (but no more than two), using the CE button alone for the first point and the CE/CR button for the second point. The rectangle cannot have \emptyset area.

S"text" <CR>
or
S"text";n<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Searches forward for a word or phrase ("text") or moves a specified number of lines forward (n) or backward (-n) and searches that line for the word or phrase ("text"). The new current line is the line containing "text." If "text" is not found, or not found where specified, the current line does not change. The current line is always shown in bright characters. After command execution, the new current line and the two lines above and below the current line are shown on the QTY.

Classification: Text editor command.

Restrictions: The SB command also searches backward for a word or phrase.

Examples:

```
?S"search command"<CR>
.
.(displayed text lines)
.
?
```

The program searches forward through the file for the phrase "search command." When found, the line containing the phrase is output as the new current line.

```
?S"search command";10<CR>
.
.(displayed text lines)
.
?
```

The program moves 10 lines forward and then searches the line for the phrase "search command." When found, the line containing the phrase is output as a new current line.

```
?S"search command";-5<CR>
.
.(displayed text lines)
.
?
```

The program moves 5 lines backward and then searches the line for the phrase "search command." When found, the line containing the phrase is output as a new current line.

SB"text"<CR>
or
SB"text";n<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Searches backward for a word or phrase ("text") or moves a specific number of lines backward (n) and searches that line for the word or phrase ("text"). The new current line is the line containing "text." If "text" is not found, or not found where specified, the current line does not change. The current line is always shown in bright characters. After command execution, the new current line and the two lines above and below the current line are shown on the QTY.

Classification: Text editor command.

Restrictions: Use only positive values of n for the SB command. The S command searches forward for a word or phrase.

Examples:

```
?SB"backward search"<CR>
.
.(displayed text lines)
.
?
```

The program searches backward through the file for the phrase "backward search." When found, the line containing the phrase is output as the new current line.

```
?SB"backward search";10<CR>
.
.(displayed text lines)
.
?
```

The program moves 10 lines backward and then searches that line for the phrase "backward search." When found, the line containing the phrase is output as a new current line.

SEGS<CE><CE/CR>

or

SEGS<CR>

CE:<CE><CE/CR>

CE:<CE><CE/CR>

.

.

.

CE:<CR>

Prerequisites: Check DEFSTAT for current properties. Structure must be open.

Purpose: Creates a segment (a path with two points), or in the interactive form, many segments or open polygons, using the current item's width, layer, and datatype.

Classification: Graphic Editor

Restrictions: None

SETDEFAULTS<CR>

Prerequisites: The following notions must be set before using the SETDEFAULTS command: the coordinate property mode (Boundary, Path, SRef, ARef, and Text), the digitizing mode (Straight, Edge, Orthint, Octagonal, and Hardlocks), the layer (0 - 63), the datatype (0 - 63), the relative width (>or = to the user's smallest DBU), and texttype (0 - 63).

ITEM may be checked for the current notions, but for a new set of defaults, each property must be set and the SETDEFAULTS command must be used.

Purpose: Takes the last set notions in the current item and sets those as the defaults for that item, thereby overriding the previous defaults for this item. When Default Mode is on, and a PUT (NULL) is done, the notions in item return to the defaults set by SETDEFAULTS, thus overriding the notions in the current item. See SHOWDEFAULTS.

Classification: Graphic Editor

Restrictions: None

```
SETTAB <CR>
TABLET AREA
  Coordinate: <CE/CR>
  Coordinate: <CE/CR>
```

Prerequisite: GSCALE, GRID

Purpose: Sets the tablet work area by collecting two diagonal coordinates, fits the work area to the CRT, and turns tablet mode on. The tablet area can be all or any portion of the tablet surface or digitizer table surface.

Classification: Station configuration control command.

Restrictions: Scaling commands are not applicable and cannot be used with tablet mode on.

Example:

```
?SETTAB <CR>
TABLET AREA
  Coordinate: <CE/CR>
  Coordinate: <CE/CR>
?
```

Two diagonal coordinates are selected to set the tablet work area.

SETVIEW <CE> <CE/CR>
or
SETVIEW <CR>
DEFINE DATA BASE WINDOW
Coordinate: <CE/CR>
Coordinate: <CE/CR>

Prerequisites: GSCALE, GRID

Purpose: Sets the view and drawing center so the CRT displays the window specified by the two diagonal coordinates. Coordinates are in user units and may be entered from the tablet, digitizer table or keyboard.

Classification: View control command.

Restrictions: Do not use command input method (arguments on same line as the command) to enter coordinates with the pen on the tablet since it always issues a <CE/CR>.

Examples:

```
?SETVIEW <CE> <CE/CR>  
?
```

Command input method was used to set the view and drawing center.

```
?SETVIEW <CE>  
DEFINE DATA BASE WINDOW  
Coordinate: <CE/CR>  
Coordiante: <CE/CR>  
?
```

Evaluated input method was used to set the view and drawing center.

SHOWDEFAULTS<CR>

Prerequisites: SETDEFAULTS

Purpose: Displays on the QTY the defaults for the current item. The notions defaulted are: coordinate property mode, digitizing mode, horiz/vert-first flag, layer, datatype, texttype, width absolute/relative flag. See also Help Files for DEFMODE and SETDEFAULTS.

Classification: Graphic Editor

Restrictions: None

SNAME "structure name":<CR>
or
SNAME<CR>
Enter Structure for Reference: structure name<CR>

Prerequisites: SREF or AREF

Purpose: Sets the structure name for a structure reference to the name designated. Structure names may be from 1 to 32 characters. The characters may be: "a" through "z", "A" through "Z", "0" through "9", "_", "\$", and "?".

Classification: Graphic Editor

Restrictions: A structure name cannot be recursive: that is, the structure name defined in SNAME cannot be the same name as that of the currently open structure.

Examples:

?NAME "BOX"<CR>
?
Sets the structure reference name to BOX.

?SNAME<CR>
Enter Structure for Reference: 23<CR>
?
Sets the structure reference name to 23.

SOLID<CR>
or
SOLID n n n...n<CR>
or
SOLID -n<CR>

Prerequisites: None required

Purpose: Specifies layers that are to be presented as solid lines. Each entered number specifies a layer 0-63. SOLID with "-n" (minus sign plus a number) indicates that layers 0 through n are to be in solid lines. See example below. Check MASKS for current status.

Classification: View Control

Restrictions: None

Examples:

?SOLID<CR>
?
Specifies that all layers are to be in solid lines.

?SOLID 2 4 6 8<CR>
?
Specifies that layers 2, 4, 6, 8 are to be solid lines.

?SOLID -9<CR>
?
Specifies that layers 0-9 are to be solid lines.

SPLIT"text"<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Divides the current line into two lines so that the portion following "text" is on the next line. The "text" is any word or phrase that occurs on the current line and ends exactly where the line is to be divided. When the division occurs, a <CR> is inserted at the end of both lines and all lines are renumbered as required. The current line is always shown in bright characters on the QTY.

Classification: Text editor command.

Restrictions: If the word or phrase ("text") occurs more than once on the current line, the line is divided following the first occurrence.

Examples:

```
?SPLIT"cat" CRT
the cat
and dog
?
```

The current line was "the cat and dog," but is now split into two lines following the word "cat" and the following lines are renumbered.

SREF<CR>

Prerequisites: A Library and a Structure must both be open. Check ITEM or ISTAT for status. SNAME must also be input.

Purpose: Declares that the current item is an SREF (Structure Reference). If the current item already has data that has not yet been NULLed (or PUT), an error occurs. Check ITEM first to avoid this problem. Go to the next available item.

Classification: Graphic Editor

Restrictions: None

Examples: This example shows the normal command sequence for setting up an SREF. ITEM then lists the conditions that have been set.

```
?SNAME<CR>
Enter Structure for Reference: NEW
?SREF<CR>
?ITEM<CR>
You Are Editing Item # 0.
It is an Sref of "NEW" (0 Points So Far).
Relative Angle of 45. Degrees,
Relative Magnification of .5X,
Relative Reflection.
0 Links Allocated
?
```

The conditions for angle, magnification, and reflection were set up by prior commands, as illustrated:

```
?SNAME<CR>
Enter Structure for Reference: NEW
?MAG<CR>
Enter Magnification: .5
?ANGLE<CR>
Enter Angle: 45
?REFL<CR>
Reflection (YES/NO): YES
?
```

If absolute, rather than relative, conditions are desired, the following command chain is used, also changing the values. ITEM then lists the new conditions.

```
?SNAME<CR>
Enter Structure for Reference: NEW
?ABSMAG<CR>
Enter Magnification: 1
?ABSANGLE<CR>
Enter Angle: 90
?ABSREFL<CR>
Reflection (YES/NO): YES
?SREF<CR>
?ITEM<CR>
You Are Editing Item #0.
It is an Sref of "NEW" (0 Points So Far).
Absolute Angle of 90. Degrees,
Absolute Magnification of 1.X,
Absolute Reflection.
0 Links Allocated
?
```

SREFMARK<CR>

Prerequisites: SREF

Purpose: Causes a mark to be placed at the origin of each selectable SREF so that it is apparent which geometries are SRefs and that they are placed at level 0, the level at which data is created. The SRefmark is erased when the structure is redrawn.

Classification: View Control

Restrictions: None

SSTAT <CR>

Prerequisite: None required.

Purpose: Lists values for the resolution and scale commands. The resolution in data base units/user unit is set with the INITLIB command. Scaling command values that are listed include working grid (GSCALE), axis rotation transfer (ASCALE), and scaling modes (BSCALE or XSCALE and YSCALE).

Classification: Scaling command.

Restrictions: If the INITLIB and scaling commands have not been issued, only system hardware default values are listed.

Examples:

```
?SSTAT <CR>
1000. Data Base Units/User Unit
A user unit is 1. inches or 25400 microns
Working Grid
0., 0., .1, .1,
Axis Rotation Transfer
1., 0
0., 1.
X Scaling Nodes
2075., 0
3068., 1
5080., 2
5574., 3
Y Scaling Nodes
2987., 0
3705., 1
4008., 2
4532., 3
?
```

A typical listing of resolution and scaling values. Do not use numbers for reference.

SSTRUCT"structure name" <CR>
or
SSTRUCT<CR>
Enter Structure: structure name<CR>

Prerequisite: OPENLIB

Purpose: Lists a summary of the data at level 0 for the specified structure. The structure name must be contained in a currently open library. The listing summarizes the number of boundaries or paths with layers, datatypes, and widths; number of array references (AREF) which have row or column specifications; number of boundaries and paths by layer and datatype; total number of boundaries and paths; range of path widths; number of structure references (SREF); number of array references (AREF) with number of times referenced; and range of coordinates.

Classification: Library access command.

Restrictions: Compare with SUMMARY.

Example:

```
?SSTRUCT<CR>  
Structure: micromess<CR>  
10 BOUNDARIES ON LAYER 0  
4 BOUNDARIES WITH DTYPE 0  
14 BOUNDARIES  
0 PATHS  
5 STRUCTURES  
0 ARRAYS WITH 0 REFERENCES  
2 TEXT ELEMENTS  
-100 TO +100 IN X  
-75 TO +75 IN Y
```

A typical print out for a structure with boundaries and structure references (SREF) but without paths or array references (AREF).

ST<CE/CR>>
Move to:<CE/CR>
or
ST<CR>
Point Near Segment to Stretch:<CE/CR>
Move to:<CE/CR>

Prerequisites: Data must be in current Item; if not, use GET.

Purpose: This command takes one segment of a polygon and stretches the attached segments so that they remain adjoined to the polygon. The first argument enters a point near the segment to be stretched, marking both ends of this segment by diamondmarks. The second argument enters the point to which this segment is to be stretched and displays the change on the screen. ST affects only the current item.

Classification: Graphic Editor

Restrictions: Adjoining segments must be orthogonal.

```
START<CR>
STARTING STATION n
INPUT CONSOLE (<CR>FOR STATION):$KEYn<CR>
OUTPUT CONSOLE (<CR>FOR STATION):QTY:n<CR>
CRT UNIT NUMBER (<CR>FOR NONE): m<CR>
```

Prerequisite: At least two stations on line with one station running the GDS II program and one or more stations not running the GDS II program.

Purpose: Issued from any active GDS II station to bring up another on-line station to run GDS II, where n is the number of the station being started. The first station number is 1, and if more than one station is being started the lowest number station is automatically started first. To select the default input and output consoles only a <CR> is required. The CRT unit numbers (m) start with 0 for station 1 so the CRT unit number is always one less than the station number. The responses shown are for standard stations. Other I/O device codes can be used for non-standard stations.

Classification: System utility command.

Restrictions: A station must be running GDS II to use the START command.

Example:

```
?START<CR>
STARTING STATION 3
INPUT CONSOLE (<CR>FOR STATION):<CR>
OUTPUT CONSOLE (<CR>FOR STATION):<CR>
CRT UNIT NUMBER (<CR> FOR NONE): 2<CR>
?
```

The system selected station 3 to start up and the input and output consoles were selected by default. Then CRT unit number 2 was selected for station 3.

STRAIGHT<CR>

Prerequisites: The current item must be defined as either BOUNDARY or PATH.
Check ITEM for status.

Purpose: Sets the current digitizing mode to STRAIGHT. In this mode, all segments are output just as they are input on the CRT, in straight lines.

Classification: Graphic Editor

Restrictions: None

SUBRT<CR>

Enter RT Point:<CE/CR>

Enter RT Point:<CE/CR>

Prerequisites: Must have data in the current Item.

Purpose: Subtracts a rectangular area from the data in the current item in order to create or modify a new geometry. This new data is put in the database, and the data in the current item remains unaffected. The rectangular area created by SUBRT takes one of two forms: (1) a section taken from the interior of the original geometry, like an inset, or (2) a division of the original geometry, which creates, in effect, two new geometries.

For the first argument, enter the lower left point of the rectangular area.

For the second argument, enter the upper right point. See also ADDRT.

Classification: Graphic Editor

Restrictions: The new rectangular area must intersect at least one side of the geometry in the current item.

SUMMARY <CR>

Prerequisites: OPENLIB and BSTRUCT or ESTRUCT

Purpose: Lists a summary of the data at level 0 for the currently opened structure. The structure name must be contained in a currently open library. The list summarizes the number of boundaries or paths with layers, datatypes, and widths; number of array references (AREF) which have row or column specifications; number of boundaries and paths; range of path width; number of structure references (SREF); number of array references (AREF) with number of times referenced; and range of coordinates.

Classification: Library access command.

Restrictions: Structure must be opened to creation and edit. Compare with SSTRUCT.

Example:

```
?SUMMARY <CR>
5 BOUNDARIES ON LAYER 0
2 BOUNDARIES WITH DTYPE 0
10 BOUNDARIES
0 PATHS
2 STRUCTURES
0 ARRAYS WITH 0 REFERENCES
1 TEXT ELEMENT
-100T0+100IN X
-75T0+75IN Y
?
```

A typical print out for a structure with boundaries and structure references (SREF) but without paths or array references (AREF).

SYSTAT<CR>

Prerequisite: None required.

Purpose: Lists the status of all grounds in the system and provides other information about each ground. The information provided includes (column heads in parentheses) whether the ground is active or waiting (Q), ground name (GRD), level (LVL), priority (PRI), allocated memory (TOT MEM), memory in use (USE MEM), number of user file tables (UTF), input console (IN CONS), output console (OUT CONS), initialized directory (DIR DCB), and the number of times a swap from memory to disk has occurred. The asterisk (*) indicates the current ground.

Classification: System utility command.

Restrictions: None required.

Example:

```
?SYSTAT<CR>
```

| Q | GRD | LVL | PRI | PROGRAM | TOT MOM | USE MEM | NO UFT | IN CONS | OUT CONS | DIR DCB | TIME SWAP |
|----|-----|-----|-----|------------|------------|------------|-----------|------------|-------------|------------|--------------|
| | FG | 5 | 1 | *INACTIVE* | 0 | 0 | 0 | \$TTI1 | \$TTO1 | | |
| *A | BG | 2 | 2 | GDSII RUN | 65 | 65 | 50 | \$TTI | \$TTO | GDSII | |
| W | BG2 | 1 | 2 | JOBMON | 11 | 11 | 17 | JMIN | \$JMOU | GDSII | |
| | BG3 | 1 | 2 | *INACTIVE* | 24 | 0 | 17 | CIN | COUT | | 0 |
| | BG4 | 1 | 2 | *INACTIVE* | 24 | 0 | 17 | CIN | COUT | | 0 |
| | BG5 | 1 | 2 | *INACTIVE* | 24 | 0 | 17 | CIN | COUT | | 0 |
| | BG6 | 1 | 2 | *INACTIVE* | 24 | 0 | 17 | CIN | COUT | | 0 |
| | BG7 | 1 | 4 | *INACTIVE* | 24 | 0 | 17 | UTN | UTOUT | BD0 | 0 |

?

A typical listing is shown. Data will vary depending upon system configuration and requirements.

T<CR>
or
T n<CR>
or
T -n<CR>
or
T n;m<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Outputs (types) the contents of the edit buffer to the QTY. The command arguments are used as follows:

| | |
|-------------|--|
| no argument | The whole edit buffer is output sequentially from the beginning. |
| n | The current line plus the number of lines (n) are output. |
| -n | The current line minus the number of lines (-n) through the current line minus one are output. |
| n;m | Outputs a block of lines from the current line plus a number of lines (n) through the current line plus a greater number of lines (m). |

If output, the current line is always shown in bright characters.

Classification: Text editor command.

Restrictions: The T command outputs to the QTY only. To output to another device, such as the Decwriter, use the W command.

Examples:

```
?T<CR>
.
.
.(displayed text lines)
.
?
```

The entire contents of the edit buffer are output sequentially on the QTY.

```
?T 10<CR>
.
.(displayed text lines)
.
?
```

The current line through the next 10 lines are output to the QTY.

?T -10<CR>

·
·(displayed text lines)

·
?

If the current line number is 16, then lines 6 through 15 are output to the QTY.

?T 10;35<CR>

·
·(displayed text lines)

·
?

If the current line number is 40, then lines 50 through 85 are output to the QTY.

TEDDEBUG<CR>

Prerequisite: FOR PROGRAMMER USE ONLY.

Purpose: Debugs the text editor.

TEXT<CR>

Prerequisites: BINDFONTS command must be input first. Check ITEM for status.

Purpose: Declares the type of the current item to be TEXT. If the current item already has data that has not yet been NULLed (or PUT), an error occurs. Check ITEM first to avoid this problem. Go to the next available item.

Classification: Graphic Editor

Restrictions: None

Examples: This example shows the normal command sequence for setting up a TEXT (see Help File for the various commands used). ITEM then lists the conditions that have been set:

```
?BINDFONTS<CR>
Font Binding Utility - 07-27-78
Font File: STK.DF
Font Number (0,1,2,3): 0
Font File: BLK.DF
Font Number (0-1,2,3): 1
Font File:
?LAYER
Enter Layer: 3
?WIDTH
Enter Width in Natural Units: .3
?TEXTTYPE
Enter Texttype: 34
?ANGLE
Enter Angle: 0
?MAG
Enter Magnification: 1
?ABSREFL<CR>
Reflection (YES/NO): YES
?ENTERTEXT<CR>
Text: CALMA
?TJUST<CR>
Vertical Justification: (top, center, bottom) :TOP
Horizontal Justification: (left, middle, right) :LEFT
?TEXT<CR>
?ITEM<CR>
You Are Editing Item #0
It is a Text Node With Relative Width .3 (0 Points So Far).
Layer 3, Text Type 34
Relative Angle of 0. Degrees,
Relative Magnification of 1.X,
Absolute Reflection.
Text String: CALMA
Top Left Justification.
0 Links Allocated
?
```

TEXTFONT <CR>

Enter Font Definition File: filename <CR>

Prerequisites: Must be in Tablet Mode with no defined GSCALE. GRID must be set to 0.

Purpose: Puts system in textfont mode so text can be created. Creates a text definition file which specifies text style, which includes width, height, interspacing between characters, and interspacing between lines. The argument asks for the name of the file that the font definition is to be placed in. When the user creates, he enters the size of the character grid (unitless) and the size of the individual character (in real units). The two points prompted for are the definition points for the screen display of the characters as they are edited and created. There should be no working grid when the user enters TEXTFONT.

The following commands facilitate editing/creating of textfont data:

| <u>Command</u> | <u>Action</u> |
|----------------|--|
| A | adds an Arc to the current character definition. Points entered in this mode are displayed as an X at the spot indicated. When a <CR> is entered instead of a point (in this mode), the command is terminated and the current character definition is redrawn. |
| D | Deletes a point as specified by a coordinate entry. If that point leaves an arc of only one vertex, then the arc is deleted as well. The character is redrawn. |
| F | produces a Full listing of the coordinates in all arcs in the current character definition and redraws. |
| G | wipes the current character definition and Gets a character from the font file currently open. The new definition is shown. |
| M | a point selected by coordinate entry is Moved to a grid point. If this point is moved on top of a neighboring vertex, it is removed by the redundant point check. The same is true of collinear points, like in GED. |
| Q | Quits TEXTFONT, writing the font file out to disk, and escapes to GDS II. No redraw is done. |
| S | prints out the Status of the current editing session. Indicated are the number of arcs in the current definition, the number of characters in the font file defined so far, the percent of free space in the font file that is used up, the last character gotten from, and the last character written to. A redraw is done. |
| W | asks for verification, and if it is given, the current definition is Wiped clean. A redraw is done. |

Command

Action

?

asks for the status for all characters in the font file. For all printable characters, the number of arcs present in the font file is given. A value of zero indicates that the character has not been defined. A redraw of the current definition is done.

<CR>

(null input) asks for the character and writes the current definition to the font page. This definition supercedes the definition, if any, that is there. No wipe is done. A redraw is done.

Classification: Graphic Editor

Restrictions: None

TEXTTYPE n<CR>

or

TEXTTYPE<CR>

Enter Texttype: n<CR>

Prerequisites: ITEM, ISTAT. Check ITEM for current status. Input BINDFONTS, which defines a font definition file.

Purpose: Sets the TEXTTYPE of the current item to the number specified and also sets the default value for texttype. Texttype is visible in item only when the current item is TEXT. The number of texttypes is 64 (0-63). See DEFSTAT to check default value.

Classification: Graphic Editor

Restrictions: None

TJUST<CR>

Vertical Justification : (top, center, bottom): answer 1<CR>

Horizontal Justification: (left, middle, right): answer 2<CR>

Prerequisites: Check ITEM for current status; must be in TEXT.

Purpose: Sets the justification notions for the TEXT item. The proper replies for the first prompt are:

'TOP', 'T', 'top', or 't', signifying the top vertical justification for the text.

'CENTER', 'C', 'center', or 'c', signifying the center vertical justification for the text.

'BOTTOM', 'B', 'bottom', 'b', signifying the bottom vertical justification for the text.

The proper replies for the second prompt are:

'LEFT', 'L', 'left', or 'l', signifying the lefthand horizontal justification for the text.

'MIDDLE', 'M', 'middle', or 'm', signifying the middle horizontal justification for the text.

'RIGHT', 'R', 'right', or 'r', signifying the righthand horizontal justification for the text.

Classification: Graphic Editor

Restrictions: None

TMOFF <CR>

Prerequisites: None required

Purpose: Turns off tablet mode. With tablet mode off, scaling commands are enabled and the tablet can be used like a small digitizer table.

Classification: Station configuration control command.

Restrictions: Scaling does not apply when tablet mode is **on**.

Example:

```
?TMOFF <CR>  
?
```

Tablet mode is turned off.

TMON <CR>

Prerequisites: None required.

Purpose: Turns on tablet mode without changing the tablet work area.
TMON is used if the tablet was temporarily turned off by a TMOFF command.

Classification: Station configuration control command.

Restrictions: Scaling commands are not applicable and cannot be used with tablet mode on.

Example:

```
?TMON <CR>  
?
```

Tablet mode is turned on.

TRANS<CE><CE/CR>
or
TRANS<CR>
Coordinates:<CE><CE/CR>

Prerequisites: Data must be in current Item.

Purpose: Translates, or moves, all points in the current item to a new location specified by the second coordinate.

Classification: Graphic Editor

Restrictions: Two arguments must be given on the same line; the<CE>button, not the <CE/CR> button, must be used.

TREE<CR>

Prerequisite: SREF or AREF

Purpose: Lists reference structures, their levels up to a maximum explosion level of 10, and the number of times referenced. The data on nesting of structures comes from the display file and indicates all levels of a structure up to the maximum level being viewed.

Classification: Library access command.

Restrictions: No listing occurs unless structures have been referenced.

Examples:

```
?TREE<CR>
Micromess level=0 references 9
Micromess 2 level=0 references 4
  Micromess level=1 references 9
  Micropad level=1 references 16
    Databus level=2 references 8
    Vccbus level=2 references 4
      Heatsink1 level=3 references 2
      Micropad level=4 references 4
      Micromess level=5 references 5
?
```

A typical listing showing six levels of referenced structures.

TSTRUCT<CR>
or
TSTRUCT<CE/CR>

Prerequisites: BSTRUCT, ESTRUCT, or INPLACE

Purpose: Terminates or closes the state of structure creation and edit, sets the structure origin, and sets the use count to zero. Only the structure open for creation and edit is affected. If the coordinate (<CE>) argument is used, the data is translated so that point <CE> becomes \emptyset, \emptyset . A TSTRUCT is implied by a CLOSELIB, BSTRUCT, ESTRUCT, or QUIT. Before a direct or implied TSTRUCT, all ITEMS should be closed and the data placed in the data base using a NULL with GEDMODE off or a <CR> with GEDMODE on. ISTAT can be used first to see if any items are open.

Classification: Library access command.

Restrictions: A loss of data is possible if a TSTRUCT is issued with open items.

Examples:

?TSTRUCT<CR>
?

The currently open structure is closed.

?TSTRUCT<CE/CR>

The <CE> point of the currently open structure is translated to point \emptyset, \emptyset and the structure is closed.

UNDO<CR>
or
UNDO n <CR>

Prerequisite: Must have data in current item.

Purpose: Deletes the last data point (as listed with COORDS) or the last n points respectively (where n=any number). If no points were entered or n is greater than the number of existing points, the item is left with no points. Compare with WIPE.

Classification: Graphic editor command (ITEM).

Restrictions: Use REDRAW after UNDO to see the results displayed on the CRT.

Examples:

?UNDO<CR>
?

Deletes the last data point entered.

?UNDO 5<CR>
?

Deletes the last 5 data points entered.

UPLWCASE<CR>

Prerequisite: None required.

Purpose: Causes all subsequent alphabetic input from the keyboard to be output to the QTY in upper and lower case (capital and non-capital) letters exactly as it is typed. Upper and lower case is the default condition at power turn on. All uppercase (captials) or all lower case letters can be selected by typing the UPPERCASE or LOWERCASE command respectively.

Classification: Station configuration control command.

Restrictions: Do not use upper case letters where lower case letters are needed to duplicate a structure name.

Example:

```
?UPLWCASE<CR>  
?
```

All typed text is output to the QTY exactly as it is typed.

UPPERCASE<CR>

Prerequisite: None required.

Purpose: Causes all subsequent alphabetic input from the keyboard to be output to the QTY in all upper case (capital) letters regardless of the shift key position. The default condition at power turn on is upper and lower case letters exactly as typed. To restore the default condition type UPLOWCASE or to output all non-capital letters type LOWERCASE.

Classification: Station configuration control command.

Restrictions: Do not use upper case letters when lower case letters are needed to duplicate a structure name.

Example:

?UPPERCASE<CR>
?

All typed text is output to the QTY in capital letters.

VDTYPE<CR>
or
VDTYPE n<CR>
or
VDTYPE n n n...n<CR>
or
VDTYPE n; n; n;...n<CR>

Prerequisite: DATATYPE

Purpose: VDTYPE with no argument causes all datatypes, 0-63, to be visible. VDTYPE with one argument makes that datatype alone visible, all others invisible. VDTYPE with several arguments makes those specified datatypes visible; with several arguments separated by semicolons, makes the first group of datatypes visible, the second invisible, and subsequent datatypes alternately visible and invisible. VDTYPE with "-n" (minus sign plus a number) indicates that datatypes 0 through n are to be visible. If this designation is followed by a semi-colon and other numbers, those specified datatypes are then to be invisible. See example below. Once datatypes have been selected, check MASKS for results.

Classification: View Control

Restrictions: None

Examples:

?VDTYPE<CR>
?
Causes all datatypes to be visible.

?VDTYPE 2<CR>
?
Makes datatype 2 visible and all others invisible.

?VDTYPE 2 4 6 8<CR>
?
Makes datatypes 2, 4, 6, and 8 visible, all others invisible.

?VDTYPE -10; 4 5<CR>
?
Makes datatypes 0-10 visible; then makes datatypes 4 and 5 invisible.

VDTYPEOFF<CR>
or
VDTYPEOFF n<CR>
or
VDTYPEOFF n n n...n<CR>
or
VDTYPEOFF n; n; n;...n<CR>

Prerequisite: VDTYPE

Purpose: VDTYPEOFF with no argument makes all datatypes invisible. Otherwise VDTYPEOFF makes specified datatypes invisible, or when used with a semi-colon, turns the first group of datatypes off, the second group on, and every other subsequent group off, then on. VDTYPEOFF with "-n" (minus sign plus a number) indicates that datatypes 0 through n are to be invisible. If this designation, if followed by a semi-colon and other numbers, those specified datatypes are then to be visible. See example below. Check MASKS for results.

Classification: View Control

Restrictions: None

Examples:

?VDTYPEOFF<CR>
?
Makes all datatypes invisible.

?VDTYPEOFF 8<CR>
?
Makes datatype 8 invisible.

?VDTYPEOFF 12 18 19<CR>
?
Turns datatypes 12, 18, 19 off; all others remain as they were.

?VDTYPEOFF 12 18; 19 25<CR>
?
Turns datatypes 12 and 18 off; datatypes 19 and 25 on.

?VDTYPEOFF -10<CR>
?
Turns datatypes 0-10 off.

VDTYPEON<CR>
or
VDTYPEON n<CR>
or
VDTYPEON n n n...n<CR>
or
VDTYPEON n; n; n;...n<CR>

Prerequisite: VDTYPE

Purpose: Adds datatypes to the view, and also adds and subtracts specified datatypes for viewing. VDTYPEON with no argument turns all datatypes on. VDTYPEON with one argument makes that datatype alone visible, all other datatypes invisible. VDTYPEON with several arguments makes those specified datatypes visible; with several arguments separated by semicolons, makes the first group of datatypes visible, the second invisible, and subsequent datatypes alternately visible and invisible. VDTYPEON with "-n" (minus sign plus a number) indicates that datatypes 0 through n are to be visible. If this designation is followed by a semi-colon and other numbers, those specified datatypes are then to be invisible. See example below. Once datatypes have been selected, check MASKS for results.

Classification: View Control

Restrictions: None

Examples:

?VDTYPEON<CR>

?

Turn all datatypes on, adding whatever datatypes were not already turned on by VDTYPE.

?VDTYPEON 10<CR>

?

Turns on datatype 10, adding it to visible datatypes.

?VDTYPEON 5 8 20 32<CR>

?

Specifies that datatypes 5, 8, 10, and 32 be added to the visible datatypes.

?VDTYPEON -9; 11 20<CR>

?

Turns on datatypes 0-9; turns off datatypes 11 and 20.

VERTFIRST<CR>

Prerequisites: EDGE or ORTHINT. The current item must be either BOUNDARY or PATH.

Purpose: Sets the current EDGE or ORTHINT operation so that the vertical segment of each digitized input is output first. See also HORIZFIRST.

Classification: Graphic Editor

Restrictions: None

VIEW<CR>
or
VIEW mag<CR>
or
VIEW<CE/CR>
or
VIEW<CE>mag<CR>
or
VIEW<CE><CE/CR>
or
VIEW<CE><CE>mag<CR>

Prerequisites: None required.

Purpose: A combination of ZOOM, PAN, and SETVIEW commands, VIEW centers the view of data at the coordinates specified and zooms in or out at the magnitude specified. A nonpositive value is illegal. A value larger than 1 zooms in; a value smaller than 1 zooms out.

Classification: View Control

Restrictions: There must be data in the structure for VIEW to include it on the screen.

Examples:

?VIEW<CR>
?
Sets the view to include all data.

?VIEW 2<CR>
?
Sets the view to twice the size of the original view window; same as ZOOM.

?VIEW .5<CR>
?
Sets the view to half the size of the original view window; same as ZOOM.

?VIEW<100 100/CR>
?
PANs the picture, centering the drawing at the specified x and y coordinates.

?VIEW<100 100> .5<CR>
?
PANs the picture, centering the drawing at the specified x and y coordinates, and reduces the size of the picture to half the original.

?VIEW<1150 1200> <1350 1300/CR>
?
This is the SETVIEW command. Sets up the view scale factor and drawing center so that the picture in the window specified by the two sets of x and y coordinates fills the viewpoint.

?VIEW<1150 1200> <1350 1300> .5<CR>
?
This is both SETVIEW and PAN. It sets up the view scale and reduces the size of the picture to half the original size.

VIEWMOVE xfraction yfraction<CR>

Prerequisites: SETVIEW

Purpose: Moves the view port by specified fractions of the screen size. The fractions are expressed in decimal and may be any amount, but typical values are .5 or -.5, or whole integers.

Classification: View Control

Restrictions: None

Examples:

?VIEWMOVE 0 .5<CR>

?

Moves the window through which data is viewed one half the screen in the positive Y direction, so that a geometry originally in the center of the CRT will now appear at the lower center of the screen.

?VIEWMOVE #.5 .5<CR>

?

Moves the window one half the screen in the negative x direction and one half the screen in the positive Y direction, so that the upper left corner of a geometry originally in the center of the CRT will now appear in the lower right corner of the screen.

VKIND<CR>
or
VKIND "kinds1";"kinds2"<CR>

Prerequisites: None required.

Purpose: Specifies what kinds are to be visible and invisible, using the following two-character codes:

| | |
|--------------------|----------------------------|
| BD - Boundary | DO - Structure Origin Data |
| PC - Path Center | AO - Array Origin Data |
| PB - Path Boundary | SN - Snap Nodes |
| TX - Text | SE - Structure Extent |
| TO - Text Origin | AE - Array Extent |

Using these codes, VKIND selects kinds of data that subsequently can be edited. VKIND with no argument turns on all kinds. VKIND with an argument turns the specified kinds on, all other kinds off. Check MASKS for results.

Classification: View Control

Restrictions: None

Examples:

?VKIND<CR>
?
Makes all kinds visible; whatever data is available will be visible.

?VKIND "PB TX TO"<CR>
?
Turns on Path Boundary, Text, and Text Origin; all other kinds remain invisible.

VKINDOFF<CR>
or
VKINDOFF "kinds1";"kinds2"<CR>

Prerequisites: VKIND

Purpose: VKINDOFF with no argument turns all kinds off. VKINDOFF with two arguments separated by a semicolon turns the first set of kinds OFF and the second set ON. VKINDOFF uses the following two-character codes:

| | |
|--------------------|----------------------------|
| BD - Boundary | DO - Structure Origin Data |
| PC - Path Center | AO - Array Origin Data |
| PB - Path Boundary | SN - Snap Nodes |
| TX - Text | SE - Structure Extent |
| TO - Text Origin | AE - Array Extent |

After kinds have been selected, check MASKS for results.

Classification: View Control

Restrictions: None

Examples:

?VKINDOFF<CR>

?

Turns all kinds off.

?VKINDOFF "BD TX SO";"PC AO AE"<CR>

?

Turns Boundary, Text, and Structure Origin OFF; turns Path Center, Array Origin, and Array Extent ON. (This would be appropriate if PC, AO, and AE had been turned off originally by VKIND.)

VKINDON<CR>
or
VKINDON "kinds1";"kinds2"<CR>

Prerequisites: VKIND

Purpose: Turns specified kinds on for viewing or specified kinds off, using the following two-character codes:

| | |
|--------------------|----------------------------|
| BD - Boundary | DO - Structure Origin Data |
| PC - Path Center | AO - Array Origin Data |
| PB - Path Boundary | SN - Snap Nodes |
| TX - Text | SE - Structure Extent |
| TO - Text Origin | AE - Array Extent |

VKINDON with no argument turns all kinds on for viewing. VKINDON with an argument turns the specified kinds on. VKINDON with arguments separated by semicolons turns the first kinds on the second kinds off. After kinds have been selected, check MASKS for results.

Classification: View Control

Restrictions: None

Examples:

?VKINDON<CR>

?

Turns all kinds on for viewing.

?VKINDON "PB TX TO SO";"AO AE"<CR>

?

Turns Path, Boundary, Text, Text Origin, and Structure Origin ON; turns Array Origin and Array Extent OFF. (This would be appropriate if AO and AE had originally been turned on by VKIND; all other kinds not already visible will remain invisible.)

V_LAYER<CR>
or
V_LAYER n<CR>
or
V_LAYER n n n...n<CR>
or
V_LAYER n; n; n...n<CR>

Prerequisites: None required

Purpose: V_LAYER command with no argument makes all layers visible. V_LAYER with a single argument makes that layer alone visible, all others invisible. V_LAYER with several arguments makes those layers visible; with several arguments separated by semi-colons makes the first group of layers visible, the second invisible, and every other layer visible and invisible. V_LAYER with "-n" (minus sign plus a number) indicates that layers 0 through n are to be visible. If this designation is followed by a semi-colon and other number(s), those specified layers are then to be invisible. See example below. Enter MASKS for results.

Classification: View Control

Restrictions: None

Examples:

?V_LAYER<CR>

?

Sets all layers to be visible.

?V_LAYER 12<CR>

?

Makes layer 12 visible, all other layers invisible.

?V_LAYER 2 4 6 8<CR>

?

Makes layer 2, 4, 6, 8 visible.

?V_LAYER -10; 0 5<CR>

?

Makes layers 0 through 10 visible; then layers 0 and 5 invisible.

VLAYEROFF<CR>
or
VLAYEROFF n<CR>
or
VLAYEROFF n n n...n<CR>
or
VLAYEROFF n;n;n...n<CR>

Prerequisites: V LAYER

Purpose: V LAYEROFF with no argument makes all layers invisible. Otherwise V LAYEROFF makes specified layers invisible, leaving all other layers in the state selected by V LAYER. When V LAYEROFF uses arguments separated by semicolons, the first group of layers is turned off and the second group of layers is turned on, and every other subsequent group is turned off, then on. When V LAYEROFF is followed by "-n" (minus sign plus a number) layers 0 through n are to be invisible. If this designation is followed by a semi-colon and other numbers, those specified layers are then to be made visible again. See example below. Enter MASKS for results.

Classification: View Control

Restrictions: None

Examples:

?VLAYEROFF<CR>

?

Turns all layers off so that no layers are visible.

?VLAYEROFF 10<CR>

?

Turns layer 10 off.

?VLAYEROFF 1 4 6 9<CR>

?

Turns off layers 1,4,6,9.

?VLAYEROFF 2; 4; 5; 6; 8<CR>

?

Turns layer 2 off, layer 4 on, layer 5 off, layer 6 on, and layer 8 off.

?VLAYEROFF -10; 4 5<CR>

?

Turns layer 0 through 10 off; layers 4 and 5 back on.

V LAYERON<CR>
or
V LAYERON n<CR>
or
V LAYERON n n n...n<CR>
or
V LAYERON n; n; n...;n<CR>

Prerequisite: V LAYER

Purpose: Adds layers to the view, and also adds and subtracts specified layers for viewing. V LAYERON with several arguments makes those layers visible; with several arguments separated by semi-colons makes the first group of layers visible, the second invisible, and every other layer visible and invisible. V LAYERON with "-n" (minus sign plus a number) indicates that layers 0 through n are to be visible. If this designation is followed by a semi-colon and other numbers, those specified layers are then to be invisible. See example below. Check MASKS.

Classification: View Control

Restrictions: None.

Examples:

?V LAYERON<CR>

?

Turns all remaining layers on, adding whatever layers were not already turned on by V LAYER.

?V LAYERON 3<CR>

?

Turns on layer 3, adding it to the visible layers.

?V LAYERON 4 5 6 7<CR>

?

Specifies that layers 4, 5, 6, 7 be added to the visible layers.

?V LAYERON 4 10 15; -3<CR>

?

Specifies that layers 4, 10, and 15 be made visible; layers 0 - 3 be invisible.

VLEVELS firstlevel lastlevel<CR>

or

VLEVELS<CR>

Inclusive Levels to View: firstlevel lastlevel<CR>

Prerequisites: LEVEL

Purpose: Indicates the inclusive levels to be viewed, from lowest to highest, 0 through 10 levels. Check DISPLAYSTAT for results. Check TREE for nested structures.

Classification: View Control

Restrictions: None

Examples:

?VLEVELS 2 8<CR>

?

Specifies that levels 2 through 8 are to be viewed.

?VLEVELS<CR>

Inclusive Levels to View: 3 7<CR>

?

Specifies that levels 3 through 7 are to be viewed.

VSTRUCT"structure name"<CR>
or
VSTRUCT<CR>
Enter structure: structure name<CR>

Prerequisite: OPENLIB

Purpose: Views (displays on CRT) any structure contained in a currently open library. The program searches for the structure name in the working library, then reference library number 1, and finally reference library number 2. The first structure found that satisfies the structure name is displayed in the current window area. A structure does not need to be opened for creation or edit to be displayed.

Classification: Library access command.

Restrictions: This is a view only command and does not allow creation or edit of a structure. Since the current window is used, the structure could be too large or too small for the area. SSTRUCT can be used to determine window size.

Example:

```
?VSTRUCT"micromess"<CR>  
?
```

The open libraries are searched and the structure named "micromess" is displayed.

```
?VSTRUCT<CR>  
Enter structure name: micromess<CR>  
?
```

Does the same as the previous example.

VTTYPE<CR>
or
VTTYPE n<CR>
or
VTTYPE n n n...n<CR>
or
VTTYPE n; n; n;...n<CR>

Prerequisites: TEXTTYPE

Purpose: VTTYPE with no argument causes all texttypes to be made visible. VTTYPE with one argument causes the specified texttype to be visible and all others to be invisible. VTTYPE with multiple arguments makes the specified texttypes visible; with arguments separated by semi-colons makes the first group of texttypes visible, the second invisible, and subsequent groups alternately visible and invisible. VTTYPE with "-n" (minus sign plus a number) indicates that texttypes 0 through n are to be visible. If this designation is followed by a semi-colon and other numbers, those specified texttypes are then to be invisible. See example below. Check MASKS for results of selection.

Classification: View Control

Restrictions: None

Examples:

?VTTYPE <CR>
?
Makes all texttypes visible.

?VTTYPE 4 <CR>
?
Makes texttype 4 visible and all others invisible.

?VTTYPE 4 10 12 <CR>
?
Makes texttypes 4, 10, 12 visible; all others remain as they were.

?VTTYPE -9; 2 <CR>
?
Makes texttypes 0-9 visible; texttype 2 invisible.

VTTYPEOFF<CR>
or
VTTYPEOFF n<CR>
or
VTTYPEOFF n n n...n<CR>
or
VTTYPEOFF n; n; n;...n<CR>

Prerequisites: VTTYPE

Purpose: VTTYPEOFF with no argument makes all texttypes invisible. Otherwise VTTYPEOFF makes specified texttypes invisible, or when used with a semi-colon, turns the first group of texttypes off, the second on, and subsequent groups alternately off and on. VTTYPEOFF with "-n" (minus sign plus a number) indicates that texttypes 0 through n are to be invisible. If this designation is followed by a semi-colon and other numbers, those specified texttypes are then to be visible. See example below. Check MASKS for results of selection.

Classification: View Control

Restrictions: None

Examples:

?VTTYPEOFF<CR>
?
Turns all texttypes off.

?VTTYPEOFF 3 16 22<CR>
?
Turns texttypes 3, 16, 22 off.

?VTTYPEOFF -40; 10 12<CR>
?
Turns texttypes 0 through 40 off; texttypes 10 and 12 back on.

VTTYPEON<CR>
or
VTTYPEON n<CR>
or
VTTYPEON n n n...n<CR>
or
VTTYPEON n; n; n;...n<CR>

Prerequisites: VTTYPE

Purpose: Adds texttypes to the view, and also subtracts texttypes. VTTYPEON with no argument makes all texttypes visible; with one argument makes that texttype visible; with several arguments makes those texttypes visible and all other invisible. VTTYPEON with several arguments separated by semi-colons turns on the first group of texttypes, turns off the second group, and turns subsequent groups alternately on and off. VTTYPEON with "-n" (minus sign plus a number) indicates that texttypes 0 through n are to be visible. If this designation is followed by a semi-colon and other numbers, those specified texttypes are then to be invisible. See example below. Check MASKS for results of selection.

Classification: View Control

Restrictions: None

Examples:

?VTTYPEON<CR>

?

Turns on all texttypes to add to whatever texttypes were turned on by VTTYPE.

?VTTYPEON 8<CR>

?

Turns on texttype 8.

?VTTYPEON 2 4 6 8<CR>

?

Adds texttypes 2, 4, 6, and 8 to visible texttypes.

?VTTYPEON -10; 2 3<CR>

?

Turns texttypes 0 through 10 on; turns texttypes 2 and 3 off.

W<CR>
or
W"filename"<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Writes the contents of the edit buffer into a file specified by "filename" or, if a G command was used to get a file for editing, the same "filename" used to get the file is assumed. File names for I/O devices, such as \$TTO for the Decwriter, are valid. A backup file is created when the W command is used.

Classification: Text editor command.

Restriction: The edit buffer and the current line remain unchanged.

Examples:

?W<CR>
?

The contents of the edit buffer are written back into the file with the "filename" used to get the file for editing.

?W"\$TTO"<CR>
?

The contents of the edit buffer are output to the Decwriter.

WIDTH n<CR>
or
WIDTH<CR>
Enter Width: n<CR>

Prerequisites: Check DEFSTAT for default status and ITEM for current status. Must be in PATH mode.

Purpose: Sets the width of the current item to the value indicated and makes this the default value. The width is relative, which means that if subsequent conditions are created or changed in the structure, the width will also be changed to relate with those conditions. The width is visible in Item only when the current item is PATH or TEXT.

Classification: Graphic Editor

Restrictions: Width cannot be smaller than one database user's unit.

WIPE<CR>

Are You Sure (Y/<CR>): response<CR>

Prerequisite: Must have data in current item.

Purpose: Deletes all data points in current ITEM if response begins with a Y or y. After all points are deleted, the defaultable properties in ITEM are reset to their default values (see DEFSTAT). To abort the WIPE command and save the data points, a <CR> is given without a response.

Classification: Graphic editor command (ITEM).

Restrictions: Data is not recoverable after the WIPE command is executed.

Example:

?WIPE<CR>

Are You Sure (Y/<CR>): y<CR>

?

All data points in the current item are deleted.

WSCALE"scalefilename" <CR>

or

WSCALE <CR>

Scale Status File: scalefilename <CR>

Prerequisites: Established scaling values.

Purpose: Stores scaling values on disk to protect against loss from system shut down or unintended use of TMON, SETTAB, QUIT, or ZSCALE commands. Stored scaling values are recoverable with an RSCALE command. Use any scalefilename with up to 10 characters. A .SS extension is assumed unless another extension is given.

Classification: Scaling command

Restrictions: Tablet mode must be off (TMOFF).

Examples:

```
?WSCALE"THIRDSCALE" <CR>
```

```
?
```

A scale status file named "THIRDSCALE.SS" is stored on disk.

```
?WSCALE <CR>
```

```
Scale Statue File: 4THSCALE.RD<CR>
```

```
?
```

A scale status file named "4THSCALE.RD" is stored on disk.

XSCALE<CR>
X Scale Values: x1 x2 ... x11<CR>
Digitizer Coordinates: CE1 CE2 ... CE11<CR>

Prerequisite: ASCALE, TMOFF

Purpose: Provides linear incremental scaling in x axis only. The y axis is scaled separately to provide better compensation for drawing areas subject to stretch or distortion. Enter from 2 to 11 scale values for the x axis, then do one coordinate entry (CE) for each value. A special form of this command uses one x scale value and one coordinate to translate previously given coordinates to a new location.

Classification: Scale command.

Restrictions: Enter coordinates in increasing order. Use pound (#) in place of minus (-). Tablet mode must be off.

Examples:

```
?XSCALE<CR>  
X Scale Values: #2 #1 0 1 2 <CR>  
Digitizer Coordinates:<CE><CE><CE><CE><CE><CR>  
?
```

Five x values from -2 (#2) to +2 were selected and digitized.

```
?XSCALE<CR>  
X Scale Values: 1  
Digitizer Coordinates:<CE><CR>
```

One value was selected and digitized to translate previously entered coordinates.

```
?XSCALE<CR>  
X Scale Values: 1 2 ... 11<CR>  
Digitizer Coordinates:<CE/CR>  
Digitizer Coordinates:<CE/CR>  
.  
.  
.  
Digitizer Coordinates:<CE/CR>  
?
```

Up to 11 values are selected for x. When <CR> is input with each <CE>, the program prompts until there is a <CE> for each value.

YSCALE<CR>
Y Scale Values: y1 y2 ... y11<CR>
Digitizer Coordinates: CE1 CE2 ... CE11<CR>

Prerequisite: ASCALE, TMOFF

Purpose: Provides linear incremental scaling in y axis only. The x axis is scaled separately to provide better compensation for drawing areas subject to stretch or distortion. Enter from 2 to 11 scale values for the y axis, then do one coordinate entry (CE) for each value. A special form of this command uses one y scale value and one coordinate to translate previously given coordinates to a new location.

Classification: Scale command

Restrictions: Enter coordinates in increasing order. Use pound (#) in place of minus (-). Tablet mode must be off.

Examples:

```
?YSCALE<CR>  
Y Scale Values: #2 #1 0 1 2 <CR>  
Digitizer Coordinates:<CE><CE><CE><CE><CE><CR>  
?
```

Five y values from -2 (#2) to +2 were selected and digitized.

```
?YSCALE<CR>  
Y Scale Values: 1  
Digitizer Coordinates:<CE><CR>
```

One value was selected and digitized to translate previously entered coordinates.

```
?YSCALE <CR>  
Y Scale Values: 1 2 ... 11 <CR>  
Digitizer Coordinates: <CE/CR>  
Digitizer Coordinates: <CE/CR>  
.  
.  
.  
Digitizer Coordinates: <CE/CR>  
?
```

Up to 11 values are selected for y. When <CR> is input with each <CE>, the program prompts until there is a <CE> for each value.

Z<CR>

Prerequisite: A or I to create a file in the edit buffer, or G to get an existing file for editing.

Purpose: Searches to the end of the edit buffer and outputs the last lines on the QTY. The last line becomes the current line and is shown in bright characters. To insert text above the current line, use the I command; and to add text below the current line, use the A command.

Classification: Text editor command.

Restrictions: None.

Example:

```
?Z<CR>  
.  
.(text lines)  
.  
?
```

The last three lines of the file in the edit buffer are displayed.

ZOOM n<CR>
or
ZOOM<CR>
Relative Magnification: n <CR>

Prerequisite: Any display on CRT.

Purpose: Changes the view magnification factor by value n. If n is a decimal, the window area is smaller (zoom out) and if n is a whole number the window area is larger (zoom in). Zoom always works from the center of the display screen and decreases or increases the number of points that fill the window area. Data does not change.

Classification: View control command.

Restrictions: Negative n values are illegal.

Examples:

```
?ZOOM .5<CR>  
?
```

The window now has twice as many points so the displayed data appears half the previous size.

```
?ZOOM<CR>  
Relative Magnification:2<CR>  
?
```

The window area now has half as many points so the data appears twice the previous size.

ZSCALE <CR>

Prerequisites: TMOFF

Purpose: Deletes (zaps) previously entered scaling commands. This clears the working grid and realigns the x and y axis to the digitizer table. Scaling is set to 1 data base unit per digitizer count and the digitizer table default coordinates are set. This does not change the resolution (data base units) established by initiating a library.

Classification: Scale command.

Restrictions: Do not issue a ZSCALE during data collection.

Example:

```
?ZSCALE <CR>  
?
```

All scaling commands are deleted.