SAND80-0538/3
Unlimited Release
UC-32

27
6-23-80 MMB
24 DNTIS

Remote Hard Copy. Volume 3 — Systems Programming Manual

Randall Simons

Prepared by Sandia Laboratories, Albuquerque, New Mexico 87185
and Livermore, California 94550 for the United States Department
of Energy under Contract DE-AC04-76DP00789

Printed March 1980

Sandia National Laboratories

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Issued by Sandia National Laboratories,
operated for the United States Department
of Energy by Sandia Corporation

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Printed in the United States of America

Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Price: Printed Copy ~~\$4.50~~, Microfiche \$3⁵⁰

6.7

SAND80-0538/3
Unlimited Release
Printed March 1980

REMOTE HARD COPY, VOLUME 3;
SYSTEMS PROGRAMMING MANUAL

R. W. Simons
Distributed Processing Systems Design Division 2644
Sandia National Laboratories
Albuquerque, New Mexico 87185

ABSTRACT

The software used to operate and maintain the remote hard copy is described. All operating software that runs in the NOVA minicomputers is covered as are various utility and diagnostic programs used for creating and checking this software.

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Contents

	<u>Page</u>
INTRODUCTION	5
1. SCP Operating Software	6
1.1 SCP	6
1.2 VPLT1	6
1.3 MDEF.	7
1.4 BTOA.	7
1.5 BTOA2	7
1.6 SLA.LB.	7
1.7 RTOS.	7
1.8 MBOOT	8
2. Creating SCP Operating Software.	9
2.1 RDOS.	9
2.2 NSPEED.	9
2.3 MAC	9
2.4 LDRDOS.	10
2.5 WRT66	10
2.6 SCPDECK	11
2.7 RED	11
3. Diagnostic Software	12
3.1 MDIAG	12
3.2 VDIAG	12
3.3 LOADCR,PLT.	12
3.4 COPY,PLT.	12
Appendix	
A. SCP Cross Reference.	14
B. Plot Commands.	20
C. NOVA Disk Organization	32

INTRODUCTION

Scan Converter/Plotter (SCP) hardware and software was developed to provide hard copy graphics at remote terminals to users of the Sandia Laboratories, Albuquerque (SLA) Central Computing Facility. An SCP consists of a Data General NOVA 3/D minicomputer with software to scan convert vector graphics to raster graphics and a Versatec 1110A plotter.

This systems programming manual contains information that should be helpful in maintaining the SCP software. The software that runs in the NOVA when the SCP is operating is described first. Next, we explain the procedures and utilities used to create this operating software. Diagnostic software, plot commands, and disk organization are also documented.

Comments are invited from readers of this manual who have corrections or other suggestions as to how to improve it. Direct comments to Randy Simons, Distributed Processing System Design Division 2644, 4-3080.

1. SCP Operating Software

This section describes the software that runs in the Data General NOVA 3/D minicomputer when the SCP is operating.

All the software is written in the NOVA Macro Assembler language to attain efficiency. Alternatives considered were Data General FORTRAN IV and Data General FORTRAN V. FORTRAN IV is much slower than Macro Assembler, and FORTRAN V requires a floating point processor which costs \$4,000 and is otherwise unneeded.

1.1 SCP (Scan Converter/Plotter)

The main routine of the scan converter/plotter operating software is named "SCP." Throughout Section 1, "SCP" will refer to this main routine, not the scan converter/plotter as a whole.

SCP is divided into modules. Each module handles a certain well-defined function or set of functions, and its listing is usually two pages or less in length. Hopefully, this makes it easier to perceive the structure of the program at a high level. Modules do not necessarily correspond to subroutines since subroutines were usually avoided in favor of macros for the sake of efficiency. A cross reference of modules and macros in SCP is given in Appendix A.

Execution of SCP begins at label START in module START. One-time initializations are done; the first buffer of data is read from the MODCOMP by GETBF; and processing data is begun. FRESH does the initialization required for each job, and CLMEM does the initialization required for each plot. RPT then processes each command by reading it, determining what type of command it is, and transferring control to the module that handles that particular type of command. The commands accepted by SCP are listed in Appendix B. Their descriptions are taken from the document, "Basic Graphics Package Intermediate File Format." Note that the DICOMED Computer Output Microfilm machine accepts almost all the commands in the BGP document, while SCP accepts a much smaller subset. Thus plot files prepared for the SCP can be run on the DICOMED with no errors, but the inverse is not always true.

1.2 VPLT1 (Versatec Plotter)

Versatec supplies a driver for their plotter controller board called VPLT. VPLT1 is a modified version of VPLT which includes changes that guarantee that calls to VPLT1 routines will always return instead of hanging on certain error conditions. Calling sequences for VPLT1 routines are exactly the same as for VPLT routines except that an entry point to the status checking routine has been added. These

calling sequences are documented in Section 3.5 of the "Versatec Controller Operating Manual" for the Model 141 Controller.

The various routines in VPLT1 control operation of the Versatec plotter. Functions include initialization, control, printing, plotting, interrupt handling, status checking, and termination.

1.3 MDEF (MODCOMP-NOVA Interface Definition)

MDEF is the driver for the MODCOMP-NOVA interface. The various routines in MDEF control the functions of initialization, transmitting/receiving data to/from the MODCOMP, interrupting the MODCOMP, managing buffers of input data, and termination. The calling sequences for these routines are documented in the "Remote Hard Copy Volume 4: MODCOMP-NOVA Interface Programming Manual."

1.4 BTOA (Binary to ASCII)

BTOA prints an octal dump on the plotter of NOVA memory from location 0 to part way through the BTOA routine. It is always loaded at location 11000g so that a dump can be taken at any time by setting the front console switches to 11000g and pressing STOP, RESET, and START. SCP also initiates a dump on certain error conditions.

1.5 BTOA2 (Binary to ASCII)

BTOA2 prints an octal dump on the plotter of a specified block of NOVA memory. It is used by SCP to dump the contents of input buffers received from the MODCOMP when switch 2 on the front console is up. It operates much the same as BTOA except for the initial setup.

1.6 SLA.LB (Sandia Laboratories Albuquerque Library)

SLA.LB is a large collection of generally useful routines developed by Data General users at Sandia Laboratories. One of these routines, SIBTOA, is used by SCP. SIBTOA takes a given binary number and converts it to ASCII decimal characters suitable for printing. Dave Ryerson, Division 4733, was the original contact for this software.

1.7 RTOS (Real-Time Operating System)

RTOS is the Data General supplied core resident operating system under which SCP runs. The version currently in use is RTOS REV 6.20. The system is documented in the "Data General Real-Time Operating System Reference Manual," ordering number 093-000056.

1.8 MBOOT (MODCOMP-NOVA Interface Bootstrap)

MBOOT is the bootstrap used to load the SCP operating software from the MODCOMP. MBOOT code is included in the MODCOMP RJE software and is down loaded into locations 0 through 3778 in the NOVA when LOAD,PLT is typed on the MODCOMP console. Once MBOOT is loaded, it begins execution and moves itself to the high end of memory. It then begins input of the SCP operating software (which the MODCOMP receives from the central site) and waits for an interrupt from the MODCOMP to signal completion of the transfer. MBOOT then transfers control to the loaded software.

2. Creating SCP Operating Software

SCP operating software is created and modified using RDOS (Real-Time Disk Operating System). As the name implies, RDOS requires a disk to operate so all this work is done on the NOVA system in Building 880, Room X-32 which has a 10MB disk.

2.1 RDOS (Real-Time Disk Operating System)

To run RDOS on the NOVA mentioned above, follow this procedure:

Set front console switches to 100033g.

Press STOP, RESET, and PR LOAD.

The NOVA terminal should then type FILENAME:

(If not, try switching the disk to LOAD, then READY, and start over.)

To which you respond DP0F:FLPSYS

(This initiates execution of the version of RDOS called FLPSYS which is on the fixed as opposed to removable, disk.)

It then asks for date and time which you enter.

Finally, type DIR SCASMF

(This puts you in subdirectory SCASMF which stands for Scan Conversion in Assembler on Fixed disk.)

RDOS is documented in the "Data General Real-Time Disk Operating System Reference Manual," ordering number 093-000129 and in the "Data General RDOS Command Line Interpreter Reference Manual," ordering number 093-000109.

2.2 NSPEED (NOVA Superedit)

NSPEED is the name of the TECO-like editor available under RDOS. It has been given the shorthand name of ED on this disk. For example, to edit the file SCP, simply type ED SCP. NSPEED is used to edit the source files SCP, VPLTL, MDEF, BTOA, BTOA2, and MBOOT. When a file is edited, the old version is first copied into a backup file identified with an extension of .BU. Thus the files SCP.BU, VPLTL.BU, MDEF.BU, BTOA.BU, BTOA2.BU, and MBOOT.BU contain backups of their respective source files.

NSPEED is documented in the "Data General Superedit Text Editor User's Manual," ordering number 093-000111.

2.3 MAC (Macro Assembler)

MAC is the macro assembler used to assemble source code into NOVA relocatable binaries. A typical assembly is performed by typing MAC SCP \$LPT/L. This says assemble the source code in the file SCP, placing the resultant object code in a file SCP.RB and print a listing on the line printer (on this system LPT = Versatec plotter).

MAC is documented in the "Data General Extended Assembler User's Manual," ordering number 093-000040, and the "Data General Macro Assembler User's Manual," ordering number 093-000081.

2.4 LDRDOS (Load SCP under RTOS)

LDRDOS is a file containing RDOS commands that takes existing relocatable binaries of the SCP operating software routines and loads them into an executable file called SCP.SV. To execute these commands, type @LDRDOS@. This also puts a listing of the load map in the file SCP.LM, which can be listed on the line printer (Versatec plotter) by typing PRINT SCP.LM. LDRDOS uses the RDOS utility RLDR (relocatable loader). RLDR is documented in the "Data General Extended Relocatable Loaders User's Manual," ordering number 093-000130.

To run SCP.SV, type BOOT DP0F:SCP. This will load SCP.SV from the fixed disk and begin executing it. It also means that to get back to RDOS, you must reload it (see Section 2.1). Since the normal mode of operation is to load SCP from the MODCOMP, a few more actions are required to tell the MODCOMP that the NOVA has been loaded. Two flags in the MODCOMP, PLDEAD and DISCARD, need to be set to 0. This is done by:

Set MODCOMP console switches to 78,

On the MODCOMP terminal, type: DIS,nnnn.

Where nnnn is the address of PLDEAD. This address will change from one version of RJE to the next.

If the value displayed (the contents of PLDEAD) does not equal 0, type: PATCH,nnnn,0.

Repeat this procedure for DISCARD.

Now you can type ON,PLT, and the software should operate the same as if it were loaded from the MODCOMP.

2.5 WRT66 (Write a Tape to Go to the 6600)

Once a new version of SCP.SV has been created and tested, it must be moved to the central site so it can be loaded at all remote sites. WRT66 takes a copy of SCP.SV and writes it to tape in a format that can be read by the CDC 6600.

First, the new SCP.SV must be moved from the fixed disk to the removable disk so it can be taken to another NOVA with a tape drive. This is easily accomplished by typing:

INIT DP0:SCASM

MOVE/A/V/R DP0:SCASM,

while running under RDOS in the subdirectory DP0F:SCASMF. This transfers all files that have been changed (or added) since the last transfer from the fixed disk to the removable disk. It also lists the names of all files moved on the NOVA terminal.

Next, take the removable disk with SCP.SV on it to another NOVA system that has both a 10MB disk and a 7-track bipac tape drive. Bipac refers to the fact the NOVA's 16-bit word is packed into two 7-track tape frames throwing out the high order 2-bits of each byte. NOVA systems with these peripherals exist in Buildings 820 and 839. A tape marked PDFTST is used to write the data on. Load the disk and the tape and proceed as follows:

Bootstrap to the disk as in Section 2.1 except,
Type DP0:BOOTSYS in response to the FILENAME: query.

(This brings up the default RDOS system which should be able to access both the disk and the tape.)

Set switches on the tape controller to BIPAC, HI (556 BPI), ODD parity, and MANUAL.

Type INIT MTn

Where n is the number of the tape drive being used

Type DIR DP0:SCASM

It may be necessary to edit WRT66 to make it use the tape drive you have selected. To do this, type

ED WRT66

S*MT\$LT\$\$

change MT number to correspond to the tape drive being used

USH\$\$

MAC WRT66

RLDR WRT66 SLA.LB

Type WRT66

The tape should move and the number of words per block read are listed on the terminal.

Type RELEASE DP0.

2.6 SCPDECK (Scan Converter/Plotter Deck)

Take the tape created by WRT66 and submit it for loading on NOS. SCPDECK is a procedure file on NOS that will read this tape, save it in a file called SCP, and punch cards of the file. To run it, first log on NOS in the batch subsystem, then type

GET,SCPDECK/UN=RWSIMON

CALL,SCPDECK

2.7 RED (Ron Domres' Program to Read Cards)

Finally, take the card deck created by SCPDECK and add it as data on the end of a thin deck of control cards called RED. Ron Domres wrote this program, and he or someone else in the Scientific Computing Systems Design Division 2635 will have to run it since it requires some actions at the CDC 7600 console. Instructions for these actions are output on the 7600 console. RED reads the cards created by SPCDECK and saves them in a file in the 7600. It is this file which is sent to a remote site and loaded into the NOVA when LOAD,PLT is typed. Thus the circle is complete.

3. Diagnostic Software

Software for testing various functions of the SCP is described in this section.

3.1 MDIAG (MODCOMP-NOVA Interface Diagnostic)

MDIAG is a set of routines which test the functions of the MODCOMP-NOVA interface. Functions tested include read and write of status, address, and word count registers, transfer data, interrupt, and receive data. MDIAG is documented in the "Remote Hard Copy Volume 4: MODCOMP-NOVA Interface Programming Manual."

3.2 VDIAG (Versatec Diagnostic)

VDIAG is a set of routines which tests the functions of the Versatec controller and printer/plotter. VDIAG is documented in the "Versatec Controller Operating Manual" for the Model 141 Controller.

3.3 LOADCR,PLT (Load from the Card Reader to the Plotter)

Typing this command on the MODCOMP console will cause the next deck of cards read into the card reader to be down-line loaded from the MODCOMP to the NOVA. The exact sequence of actions is as follows:

On the MODCOMP console, type: OFF,PLT
OFF,CR

Put the card deck to be loaded into the card reader and press the RESET button on the card reader.

On the MODCOMP console, type: LOADCR,PLT

The cards should be read and the NOVA loaded.

Finally, on the MODCOMP console, type: ON,CR
ON,PLT

Loading from cards is not the normal method for loading the NOVA, but it is useful for checking the correctness of a card deck generated by SCPDECK (see Section 2.6).

3.4 COPY,PLT (Copy Data from the Card Reader to the Plotter)

Typing this command on the MODCOMP console will cause the next deck of cards read into the card reader to be sent to the NOVA as plot data. The exact sequence of actions is as follows:

On the MODCOMP console type: OFF,PLT
OFF,CR (Plotter should have
already been loaded.)

Put the card deck to be copied into the card reader and press the reset button on the card reader.

On the MODCOMP console, type: COPY,PLT
The cards should be read and plots output on the plotter.

Finally, on the MODCOMP console, type: ON,CR
ON,PLT

Copying from cards is not the normal method of plotting, but it is useful for separating the functions of data transmission and plotting for the purpose of locating where errors are occurring. Creating a deck of cards to be used in this manner is simple. Run a job as if it were going to produce plots at a remote site but replace the COMQ card with one that punches cards.

Appendix A

SCP Cross Reference

All macros and program modules of SCP are listed along with everything they reference. These references are of three types: macros, internal modules, or external routines. When a macro is referenced, the macro code physically replaces the macro call in the calling routine. As a general rule, internal calls do not return but are simply straight-line transfers of control. The only exceptions to this rule are ERO and GETBF which return to the caller in certain cases. External routines always return to the calling routine.

SCP Cross Reference--Macros

<u>Macro</u>	<u>References</u>
GET16	internal: GETBF
TOVER	internal: ER0 external: .REMAP
TUNDR	internal: ER0 external: .REMAP
INCX	
INCY	
DECX	
DECY	
IMAP	
MUL60	
GETBY	macros: GET16
STOBY	
FSTPT	macros: IMAP internal: ER0 external: .REMAP
REDID	macros: GET16 internal: BADOP
LOCK	external: .REC
UNLOCK	external: .XMT

SCP Cross Reference--Program Modules

<u>Module</u>	<u>References</u>
ERO	macros: LOCK, UNLOCK internal: EODAT(EODT1), NPAGE(OUTP0) external: CSTAT, .VRMV, .VCTR, .MRMV, BTOA, .MINT
GETBF	internal: FRESH external: .NQFRE, .MINT, .DQFUL, BTOA2
START	macros: LOCK, UNLOCK internal: ERO, GETBF(FSTBF), FRESH external: .MDEF, .VMEM, .MEM, .MEMI, .MAPDF, .VDEF, .VCTR, .VPRW
FRESH	macros: MUL60, GET16 internal: ERO, CLMEM external: .GTOD
CLMEM	internal: ERO, RPT(RPT, RPT3) external: .REMAP
RPT	macros: GET16 internal: (MCOPO--COORD, COPOS(IRUNC)), BADOP, SKPWD, ESCP, FILID, NPAGE, EODAT, COPOS(RUNC, SELPX), SETMD(S2D, S3D, SDRAW, SMARK, BPOLY, EPOLY), PMARK
COORD	macros: GET16 internal: MOVE1, (MDRAW--DRAW, MARK)
MARK	macros: FSTPT internal: MOVE1
DRAW	macros: FSTPT internal: HORZ, VERT, DEG45, XLINE, YLINE
YLINE	macros: INCY, DECX, TOVER, IMAP, INCX internal: MOVE1
XLINE	macros: INCX, DECX, TUNDR, IMAP, INCY, TOVER internal: MOVE1
DEG45	macros: INCX, INCY, TOVER, IMAP, DECX, DECX, TUNDR internal: MOVE1
VERT	macros: INCY, TOVER, IMAP internal: MOVE1
HORZ	macros: INCX, IMAP internal: MOVE1

<u>Module</u>	<u>References</u>
MOVE1	macros: GET16 internal: (M2D--RPT,MOVE1(I3D)),RPT
NPAGE	macros: GET16, MUL60, LOCK, UNLOCK internal: BADOP, ER0, PRNT(PRNT, PRNT1) external: .GTOD, .REMAP, .VPLW
PRNT	macros: GET16, MUL60, LOCK, UNLOCK internal: ER0, EODAT(EODT2), CLMEM external: .VPRW, SIBTOA, .GTOD, .VCTR
ESCP	macros: GET16 internal: SKPWD, RAST
FILID	macros: LOCK, UNLOCK, GET16, REDID internal: ER0, SKPWD external: .VCTR
EODAT	macros: LOCK, UNLOCK, GET16 internal: NPAGE(OUTPT), ER0, GETBF, RPT(RPT3) external: .VPRW
BADOP	macros: LOCK, UNLOCK, GET16 internal: ER0, RPT(RPT3) external: SIBTOA, .VPRW
SETMD	internal: SKPWD
PMARK	macros: FSTPT internal: SKPWD
COPOS	macros: GET16 internal: SKPWD, (MRUNX--RPT, COPOS(IRUN1)), RPT
SKPWD	macros: GET16 internal: RPT
RAST	macros: GETBY, STOBV internal: RPT

NOT SHOWN: MOST MODULES REFERENCE GETBF THROUGH GET16

FIGURE 1.

HIERARCHY OF SCP MODULES

FIGURE 2.

HIERARCHY OF SCAN CONVERSION MODULES IN SCP

Appendix B

Plot Commands

The following is a list of all commands accepted by the SCP. Any commands not listed will be considered as bad op codes. The error message "BADOP nnn AT mmmmm" will be output on the plotter where nnn is the op code in decimal, and mmmmm is the decimal number of data words read since the beginning of the current plot (modulo 32,768). After this message, a dump of NOVA memory (in octal) will be output on the plotter if switch 3 is set.

Of the commands listed, all are valid but some indicate operations which are meaningless to the SCP (for example, color commands). These commands do not cause an error message but are simply ignored.

The format is based on multiples of 16-bit words to facilitate processing on most mini and microprocessors as well as many of the larger processors. This choice will also facilitate the processing of 8-bit ASCII characters. There are basically three command formats: coordinate positioning commands, run codes, and OP code commands. Each of these is described below:

Coordinate Positioning Command Format

In most graphics applications, the majority of graphics data consists of coordinate positioning information. In order to minimize the amount of graphics data, the coordinate positioning commands have been kept to a minimum size within the limitations of a 16-bit word. Each coordinate positioning command will consist of an x,y, and optional z coordinate. Analysis of "typical" graphics output indicates that no significant size reduction results in eliminating either the x or the y coordinate when it is unchanged from the previous position. In fact, because of the extra bits necessary to identify whether a coordinate consists of x or y or both instead of a "simple" x,y pair, the total size may actually increase. A basic assumption is that, although graphics output files may be quite large, they are in general short-lived so that a small increase in size to simplify structure and processing logic is justified.

Another crucial choice is the number of bits to maintain for each coordinate in order to adequately accommodate "most" graphics devices. A minimum is 10 bits. With 12 bits, many devices can be accommodated, and with 14 bits, most devices can be accommodated without loss of resolution when going from device to device. Since at least one 16-bit word is required and only a limited amount of control information is necessary, 15 bits per coordinate has been chosen. The format for coordinate positioning commands is as follows:

0	z-coordinate (15 bits)														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Word 2 (3D mode only)

WORD 0--BIT 15 0 implies coordinate positioning command
 1 implies op code command format (see below).

WORD 1--BIT 15 0 implies move beam to x,y.
 1 implies draw line to x,y or draw marker at x,y
 depending on the mode.

After completion of any of the above commands, the current position is at the specified x,y coordinate. For 3D applications, a mode is set, and the coordinate command consists of an x, y, and z coordinate. The z coordinate is ignored by the SCP. For 2D applications, a mode is set, and the coordinate command consists of an x and y coordinate only.

Run Code Command Format

These commands are currently ignored by the SCP. In solid-image graphics applications, the majority of graphics data consists of sequences of pixel locations along a scan line. These sequences can usually be compressed into run codes which define an entire section of a scan line by alleviating the necessity of defining the characteristics of each pixel location individually. Three basic forms of the run code are available. Commands for specifying which form is to be used are described under Op Code Command Class D. The basic format of the run codes is as follows:

0	Run Ending Location with Scan Line (15 bits)														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Word 0

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

Word 1

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

Word 2 (Run Code Format 2 Only)

Word 0--BIT 15 0 implies run code command format.
 1 implies Op Code Command Format (see below).

Run code command format or coordinate positioning format are entered when bit 15 of word 0 is zero. Which format is to be used depends on Op Code Command 87--select Coordinate Position Command Format and Op Code Command 88--select Run Code Command Format. The type of run code command format is specified by Op Code Command D0--Select Run Code Format. After completion of any run code command, the current

position is at the pixel location following the ending location specified in the Word 0 of the run code. This allows one run to be followed by another with no intermediate positioning. If the specified ending location is less than the current position, the run extends to the end of the current scan line, an automatic retrace takes place, and the run is continued on the next scan line to the specified ending location. A run may be started at the position specified by a previous run, a coordinate positioning command, or a TEXT primitive. In 3D mode, runs are created in the z-plane specified by the current position.

Op Code Command Format

The op code command format will accommodate the remainder of the device-independent commands. Although this format will allow most of the possible device capabilities, it will comprise a very small portion of the total graphics data. The majority will consist of coordinate positioning data. Since this is the case, more attention has been given to providing a simple and uniform format rather than complicating the format by trying to keep it short. Each op code will contain a count of the following words to be associated with the op code. This will simplify searching the file for particular op codes and will also allow device drivers to skip unknown op codes since the number of words to be associated with a particular op code is immediately available. The op code format is as follows:

The 7-bit op code will be subdivided into a 3-bit class and a 4-bit subclass. This will allow the various op codes to be defined in an organized fashion with 8 major classes each consisting of up to 16 subclasses. A procedure exists to extend either a class or subclass in the event additional op codes are needed. By dividing the op code in this fashion, jump tables can more easily be used to implement the processing of the class and subclass operations. The format for the op code is as follows:

1	class				subclass				op word count (0-255)							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
op code									word count							

The following are the major class divisions with proposed subclass operations in each division. The number represents the 8-bit op code field in hexadecimal, with x representing the subclass field.

8x16 Control--commands related to the control and operation of the system.

Ex., screen size, new page, escape, etc.

9x16 Global attributes--commands defining attributes applying to both graphics and text.

Ax16 Graphics primitives--commands setting graphics modes and graphics attributes for graphics primitives.

Ex., 2D, or 3D, mark or draw, line style, etc.

Bx16 Undefined

Cx16 Segmentation/Grouping--commands related to segment and group definition and control.

Ex., begin segment n, end current segment, etc.

Dx16 Raster--commands which are peculiar to raster scan display devices.

Ex., Send scan line, etc.

Ex16 Undefined.

Fx16 Extended class--additional classes, if necessary, will use the subclass field for up to 16 additional class divisions.

Op Code Commands

This section describes the specific class and subclass op codes. The notation AAA/BBB indicates that AAA occurs in the left 8 bits of a 16-bit word and that BBB occurs in the right 8 bits of a 16-bit word.

CONTROL (8x16)

- 81₁₆ No operation (NOOP)
81nn₁₆ Op code/count (0-255)
Padding
Note: It may be necessary to pad a buffer to word boundary with NOOPs before writing for overlay support, buffer sharing, etc.
- 82₁₆ Escape
82nn₁₆ Op code/count (2-155)
word 1 Function number/count
Function parameters
Note: This command allows for nonstandard or device-dependent capabilities to be defined. A valid implementation may ignore all such escape sequences.
- 8202₁₆ Op Code/count(2)
word 1 0/1--Security of current page/count (1)
This command is currently ignored by the SCP.
word 2 Classification level of current page
0 Unsecured--not labeled unclassified
1 Unclassified--labeled unclassified
2 OUO--labeled Official Use Only
3 PARD--Protect as Restricted Data
4 Confidential
5 Confidential RD
6 Confidential FRD
7 Confidential NSI
8 Secret
9 Secret RD
10 Secret FRD
11 Secret NSI
12 Top Secret
13 Top Secret RD
14 Top Secret FRD
15 Top Secret NSI
- 8201₁₆ Op code/count(1)
word 1 1/0--Specify raster format
Note: This raster format is different from the run code command format. Code to process this format has been written but not debugged so executing this command will probably eventually result in an error.

820₁₆ Op code/count (1)
word 1 44₁₆/0--Leave out extra page between plots.

Note: All other escape function numbers are ignored by the SCP.

83₁₆ Aspect ratio (normalized device coordinate limits)

This command is ignored by the SCP.

8303₁₆ Op code/count (3)
word 1 x-dimension (15 bits)
word 2 y-dimension (15 bits)
word 3 z-dimension (15 bits)

Note: One of the limits (x, y, or z) must be maximum with the other limits less than or equal to the maximum.

84₁₆ Intermediate file identifier

84nn₁₆ Op code/count (nn)
word 1 Format (BGP=0)/Origination code
0/0 BGP format, unspecified origin
0/1 Los Alamos Scientific Laboratory origin
0/2 Sandia Laboratories, Albuquerque origin
0/3 Air Force Weapons Labaoratory KAFB origin

word 2-5 Date Field:
Stored in form-YY/MM/DD, two characters per word,
high order character first.

word 6-9 Time Field:
Stored in form HH.MM.SS, two characters per word,
high order character first.

word 10-iJob ID Field:
First Word: JOB ID length in words
Following "Length" words:
JOB ID stored two characters per word, high order
character first.

word i+1 -j User ID Field:
First Word: USER ID length in words.
Following "Length" words:
USER ID stored two characters per word, high
order character first.

word j+1 -k Routing Information Field:
First Word: Routing Information length in
words.
Following "Length" words:
Routing Information stored two characters per
word, high order character first.

word k+1 -n Local environment parameters specific to
originating laboratory (ASCII characters).

Note: This must be the first command.

85₁₆ New page
8505₁₆ Op code/count (5)
word 1 FFFF₁₆
word 2 FFFF₁₆ Sync pattern
word 3 FFFF₁₆
word 4 Page number
word 5 Page ID (supplied by user if desired)

Note: The sync pattern only occurs on the page command and provides for a mechanism to synchronize the interpreter with the intermediate file if necessary.

86₁₆ End of Data
8600₁₆ Op code/count (0)
Note: Data between this command and the next
84₁₆ command is ignored.

87₁₆ Select coordinate positioning command format
8700₁₆ Op code/count (0)
Note: This means that if the first bit of a command is zero, the command is assumed to specify coordinates.

88₁₆ Select run code command format
8800₁₆ Op code/count (0)
Note: This means that if the first bit of a command is zero, the command is assumed to specify run codes.

GLOBAL ATTRIBUTES--GRAPHICS AND TEXT (9x₁₆)

90₁₆ Define color index
This command is ignored by the SCP.
9005₁₆ Op code/count (5)
word 1 Color index number (0-255)
word 2 0/red
word 3 Green/yellow
word 4 Blue/magenta
word 5 Cyan/white
Note: Words 2-5 are the intensity settings (8-bit) of the primary and secondary colors defining the color index. Each color definition is modified by current intensity setting when selected.

91₁₆ Select color
 This command is ignored by the SCP.
 9101₁₆ Op code/count (1)
 word 1 Color index number
 0--black
 1--red
 2--green
 3--yellow
 4--blue
 5--magenta
 6--cyan
 7--white
 8--255 remainder of system-defined color table.

92₁₆ Intensity
 This command is ignored by the SCP.
 9201₁₆ Op code/count (1)
 word 1 Intensity (15 bits)
 Note: Intensity is mapped from lowest intensity
 to highest intensity of the device.

GLOBAL ATTRIBUTES--GRAPHICS AND TEXT (A_{x16})

A0₁₆ Set 2D mode-clear 3D mode (default)
 A000₁₆ Op code/count (0)

A1₁₆ Set 3D mode-clear 2D mode
 A100₁₆ Op code/count (00)

A2₁₆ Set draw mode-clear mark code (default)
 A200₁₆ Op code/count (0)

A3₁₆ Set mark mode-clear draw mode
 A300₁₆ Op code/count (0)

A4₁₆ Plot marker at the current position
 A400₁₆ Op code/count (0)

A5₁₆ Select marker symbol
 This command is ignored by the SCP.
 A501₁₆ Op code/count (1)
 Marker symbol number:
 0--point (default)
 1--9 system-defined symbols
 10--65535 installation-defined symbols

A6₁₆ Begin polyline or polymarker
 A600₁₆ Op code/count (0)

A7₁₆ End polyline or polymarker
 A700₁₆ Op code/count (0)

A8₁₆ Line style
 This command is ignored by the SCP.
 A80₁₆ Op code/count (1)
 word 1 Line style (15 bits)
 Note: Line style varies from dotted (zero) to solid (max) lines. Solid lines are the default.

A9₁₆ Line width
 This command is ignored by the SCP
 A90₁₆ Op code/count (1)
 word 1 Line width (15 bits)
 Note: Line width is mapped from thinnest line (default) to widest line the device is capable of drawing. For simulation, the widest line is defined to be 1 percent of the smallest dimension of the screen aspect ratio in x and y.

SEGMENTATION/GROUPING (Cx₁₆)

C7₁₆ Begin batch update
 This command is ignored by the SCP.
 C700₁₆ Op code/count (0)
 C8₁₆ End batch update
 This command is ignored by the SCP.
 C800₁₆ Op code/count (0)

RASTER (DX₁₆)

D0₁₆ Select Pixel index format
 D00₁₆ Op code/count (1)
 word 1 Pixel Index Format Indicator
 0--index 1 specifies intensity index (default)
 1--index 1 specifies color table index
 2--index 1 specifies intensity index and index 2 specifies color table index
 Note: Pixel index format 2 requires a three-word run code. See description of run codes on page 23.

Appendix C

NOVA Disk Organization

The following is a list of all files on the fixed disk on the NOVA System located in Building 880, Room X-32. The files are organized into logical groupings called subdirectories. This information is backed up on the removable disk. To get an up-to-date list of files in a subdirectory plus information on file length, creation date, and last use, enter that subdirectory and type LIST/A/S/L/E.

DP0F System Routines and Libraries

\$LPT.	Line printer
\$TTI.	Teletype input
\$TTO.	Teletype output
\$TTP.	Teletype paper tape punch
\$TTR.	Teletype paper tape reader
ASM.SV	Extended assembler
BOOT.SV	CLI command--boot to new system
BOOTSYS.OL	Default RDOS version
BOOTSYS.SV	
BREAK.SV	Core image saved when terminate DEBUG
CLI.CM	Command Line Interpreter
CLI.ER	
CLI.OL	
CLI.SV	
COM.CM	Communication area between CLI commands
DKINIT.SV	Disk initializer
DKPDR4234.LB	Disk driver, Model 4234
DSKED.SV	Disk editor
ED.SV	Pseudonym for NSPEED, the TECO-like editor
EDIT.SV	Standard editor
FLPSYS.OL	RDOS version including floppy disk, hard disk, and line printer
FLPSYS.SV.	
FOR4LIB.	List of FORTRAN IV libraries
LFE.SV	CLI command--library file editor
LISTDIRS.	Indirect CLI file--list files in all directories
LISTDIRS.BU	
LITTLESYS.SV	RDOS version for Bob Little's NOVA in Bldg. 880, Room C45
MAC.PS	Macro assembler
MAC.SV	
MACXR.SV	Macro assembler cross-reference generator
MAP.DR	Disk map directory
MDIAG.SV	MODCOMP-NOVA interface diagnostic
MPARSAN.RP	Added parameters for Sandia version of macro assembler
MTRX.SR	Versatec plotter stand-alone driver
NBID.SR	Macro assembler basic instruction definition
NFIXLIB.MC	RDOS library fix-up routine

NMAP.LB	RTOS library
NRTOS1.LB	RTOS library
NRTOS2.LB	RTOS library
NSKID.SR	Macro assembler skip instruction definition
NSPEED.SV	TECO-like editor
NSYS.OL	RDOS version including hard disk and line printer
NSYS.SV	
NSYSGENF.DR	Directory
NUMAC.	Create a new macro assembler
NUMAC.BU	
OEDIT.SV	Octal editor
OSID.SR	Macro assembler operating system instruction definition
OVLDR.SV	Overlay loader
PARSAN.RP	Added command definitions for Sandia version of macro assembler
PATCH.SV	Apply patches to a file
PLTSYS.SV	RDOS version including hard disk and plotter
RLDR.OL	CLI command--relocatable loader
RLDR.SV	
RTOS.RB	RTOS (version 6)
RTOS6.PF	RTOS version 6 patch file
RTOS64234.PF	RTOS version 6 disk driver, Model 4234
RTOS6F.DR	Directory
SCASMF.DR	Directory
SCP.SV	Scan converter/plotter program
SLA.LB	Sandia Labs utilities library
SPEED.ER	Editor error file
SYS.DR	System directory
SYS.LB	System library
SYS.OL	RDOS version, same as NSYS
SYS.SV	
SYS000.RB	SYSGEN output
SYS24K.OL	RDOS version, same as BOOTSYS
SYS24K.SV	
UFLIB.	List of FORTRAN V libraries
VDIAG.SV	Versatec plotter diagnostic

NSYSGENF	RDOS Systems and Files Used to Generate Them
\$LPT.	Line printer
\$TTI.	Teletype input
\$TTO.	Teletype output
\$TTP.	Teletype paper tape punch
\$TTR.	Teletype paper tape reader
CLI.CM	Command Line Interpreter
COM.CM	Communication area between CLI commands
FLPSYS.LS	RDOS version including floppy disk, hard disk, and line printer
FLPSYS.OL	
FLPSYS.SG	
FLPSYS.SV	
LFE.SV	CLI command--library file editor
LITTLESYS.LS	RDOS version for Bob Little's NOVA in Bldg. 880, Room C45
LITTLESYS.OL	
LITTLESYS.PD	
LITTLESYS.SG	
LITTLESYS.SV	
MAP.DR	Disk map directory
NFIXLIB.MC	RDOS library fix-up routine
NRDOS.PF	RDOS patch file
NRDOSA.LB	RDOS system libraries
NRDOSB.LB	
NRDOSC.LB	
NRDOSI.LB	
NRDOSO.LB	
NSYS.OL	RDOS version including hard disk, TTY, and line printer
NSYS.PD	
NSYS.SV	
NSYS.TU	
NSYSANS.	SYSGEN dialogue for NSYS
NSYSGEN.SV	System generation routine
NSYSMAP.	Load map of NSYS
PATCH.SV	Apply patches to a file
PLTSYS.LS	RDOS version including hard disk and plotter
PLTSYS.OL	
PLTSYS.PD	
PLTSYS.SV	
RLDR.OL	CLI command--relocatable loader
RLDR.SV	
SYS.DR	System directory

RTOS6F

RTOS Systems and Files Used to Generate Them

\$LPT.	Line printer
\$TTI.	Teletype input
\$TTO.	Teletype output
\$TTP.	Teletype paper tape punch
\$TTR.	Teletype paper tape reader
CASDR.LB	Cassette tape driver?
COM.CM	Communications area between CLI commands
DKPDR4047.LB	Disk drivers
DKPDR4234.LB	
DKPDR6030.LB	
DKPDR6038.LB	
DKPDR6060.LB	
DSKDR.LB	
GSUBR.RB	
MAP.DR	Disk map directory
MRTOS1.LB	RTOS system libraries, mapped version
MRTOS2.LB	
MTADR.LB	
NMAP.LB	
NRTOS1.LB	RTOS system libraries, unmapped version
NRTOS2.LB	
PARR.SR	List of RTOS parameters
PATCH.SV	Apply patches to a file
RLDR.SV	CLI command--relocatable loader
RTOS.RB	RTOS version in use
RTOS5.PF	RTOS version 5 patch files
RTOS54047.PF	
RTOS54234.PF	
RTOS6.PF	RTOS version 6 patch files
RTOS64047.PF	
RTOS64234.PF	
RTOSGEN.PD	RTOS system generation routine
RTOSGEN.RB	
RTOSGEN.SV	
RTOSGEN5.PF	RTOS version 5 patch file
RTOSGEN6.PF	RTOS version 6 patch file
SYS.DR	System directory
UPDATE.	Release notice, RTOS 6.00 update 1

SCASMF	Scan Conversion Programs Written in Assembly Language
\$LPT.	Line printer
\$TTI.	Teletype input
\$TTO.	Teletype output
\$TTP.	Teletype paper tape punch
\$TTR.	Teletype paper tape reader
ASM.SV	Extended assembler
BTOA.	Binary to ASCII conversion, dump NOVA memory
BTOA.BU	
BTOA.RB	
BTOA.SV	
BTOA2.	Binary to ASCII conversion, dump input buffers
BTOA2.BU	
BTOA2.RB	
COM.CM	Communication area between CLI commands
DEMO.SV	Early versions of SCP
DEM01.SV	
DEMO2.SV	
ED.SV	Pseudonym for NSPEED, the TECO-like editor
HEADER.	Source code documentation header
LDRTOS.	Indirect CLI file--load everything needed for RTOS version of SCP
LDRTOS.BU	
MAC.PS	Macro assembler
MAC.SV	
MACXR.SV	Macro assembler cross-reference generator
MAP.DR	Disk map directory
MBOOT.	MODCOMP-NOVA interface downline load bootstrap
MBOOT.BU	
MBOOT.LS	
MBOOT.RB	
MBOOT.SV	
MDEF.	MODCOMP-NOVA interface driver
MDEF.BU	
MDEF.RB	
MDIAG.	MODCOMP-NOVA interface diagnostic
MDIAG.BU	
MDIAG.LM	
MDIAG.PD	
MDIAG.RB	
MDIAG.SV	
MDIAGRDOS.SV	
MDIAGRtos.SV	
MEMSZ.	Determine how much memory is available
MEMSZ.RB	
MEMSZ.SV	
NMAP.LB	RTOS system libraries
NRTOS1.LB	
NRTOS2.LB	
NSYS.SV	RDOS version including hard disk, TTY, and line printer
OEDIT.SV	Octal editor
PATCH.SV	Apply patches to a file

PDFTS0.	Raster plot file--GCS village
PDFTS2.	Vector plot file--GCS, 3 plots, 15-bit resolution
PDFTS4.	Vector plot file--DISSPLA, 8 plots, SCORMON output
PDFTS6.	Vector plot file--GCS line options, etc.
PDFTS7.	Vector plot file--10 minute, 20,000 vector plot
PDFTSA.	Vector plot file--SCORS, maps zooming in on Albuquerque
PDFTSB.	Vector plot file--DISSPLA through SCORS, 5 plots
PDFTSC.	Vector plot file--SCORS, grids and characters, 23 plots
PDFTSD.	Vector plot file--SCORS, grids and characters
PLTSYS.SV	RDOS version including hard disk, TTY, and plotter
RDT66.	Read a tape from the CDC 6600 to NOVA
RDT66.BU	
RDT66.LS	
RDT66.RB	
RDT66.SV	
RLDR.OL	CLI command-relocatable loader
RLDR.SV	
RTOS.RB	RTOS version in use
RTOS6.PF	RTOS version 6 patch file
SCP.	Scan converter/plotter program
SCP.BU	
SCP.LM	
SCP.RB	
SCP.SV	
SLA.LB	Sandia Labs utility library
SPEED.ER	Editor error file
SYS.DR	System directory
SYS.LB	System library
VDIAG.SV	Versatec plotter diagnostic
VPLT.	Versatec plotter driver
VPLT.BU	
VPLT.RB	
VPLTL.	Versatec plotter driver, modified version
VPLTL.BU	
VPLTL.RB	
WRT66.	Write a tape to the CDC 6600 from NOVA
WRT66.BU	
WRT66.RB	
WRT66.SV	

RWS:drs:4434A:P

Distribution:

2635 P. A. Lemke
2635 R. E. Domres
2635 J. Brabson
2640 J. L. Tischhauser
2644 D. M. Darsey
2644 C. D. Brown
2644 R. W. Simons (10)
2648 D. H. Schroeder
2648 L. Pierson
3141 T. L. Werner (5)
3151 W. L. Garner (3)
For DOE/TIC
DOE/TIC (25)
(R. P. Campbell, 3154-3)
8266 E. A. Aas (1)

[illegible]

*Recipient must initial on classified documents.