TECHNICAL ORDER

NSP SYSTEM OPERATION AND MAINTENANCE MANUAL

VOLUME 3 OF 3

NSP SOFTWARE USER'S DOCUMENT

Volume 3 Part Number: 901181-118A

Prepared by

EVANS & SUTHERLAND COMPUTER CORPORATION
580 Arapeen Drive
Salt Lake City, Utah 84108

NSP SOFTWARE USER'S DOCUMENT

CONTENTS

ILLUSTRATIONS

TABLES

SECTION 1

INTRODUCTION

1.0 OBJECTIVE OF DOCUMENT

This document contains information on the basic operation of the operating system, utilities, and modeling programs for the NSP system. Section 2 contains an overview and operating instructions on the monitor and associated utility programs. section 3 covers the operating instructions for modeling program NSPBLD and Section 4 covers the operating instruction for the model merge program NSPMER.

SECTION 2

MONITOR AND UTILITY PROGRAMS

2.1 BASIC OPERATING SYSTEM - NSPMON

The basic operating system, NSPMON, provides for simplified operation and control of the programs provided with the NSP system. NSPMON is provided as a custom configuration of the TI IOP8 basic system software and provides the following facilities:

- System loader for operating system and program loading
- Computer functions for power fail/restart processing, memory protect/privileged instruction monitoring, memory address mapping and protection, and control of program execution
- I/O processing functions and association of logical units to physical devices
- Operator communication functions for program loading, assignment of logical/physical units, and program execution
- File management for allocation and maintenance of disk files

The following list is a brief description of the console commands that are available to the operator. Refer to the Model 980 Computer, Basic System Use and Operation Manual for a complete description of those functions.

```
//ASSIGN,lun,pdev,filename.
//RELEASE,lun.
//ENDFILE,lun.
//EXECUTE,pdev,filename.
//LOAD,pdev,filename.
//REWIND,lun.
//SWAPCS,lun.
//SKIPCS,nn.
//DEFINE,pdev,filename,recsiz,type,extent.
//DELETE,pdev,filename.
//RENAME,pdev,filename1,filename2.
//COMPRESS,pdev,X,X.
```

where:
- lun is a logical unit number
- pdev is a physical device name (see table 2-1)
- filename is a one to six character filename
- filename 1 is the file name being changed
- filename 2 is a new file name
- recsiz is the file record size
- type is the file type code
- extent is the file length extent

Device Name	Device Description
KEY	Console keyboard
CS1	Cassette tape, unit one
CS2	Cassette tape, unit two
FDØ	Floppy disk, unit zero
FD1	Floppy disk, unit one
TEK	Tektronics 4010 or 4010-1 terminal
DMY	Dummy device

TABLE 2-1
PHYSICAL DEVICE NAMES

System Error Messages

Error messages output by NSPMON are divided into general system errors and abort errors.

System errors are caused by improper use of a logical unit or one of the physical devices. When such an error occurs, an error message in the following format is output:

LUN xxxx ERR yyyy AT zzzz LL ssss

where: xxxx is the logical unit number
yyyy is one of the error codes shown in table 2-2
zzzz is the memory location of the request which resulted in the error
ssss is the contents of the lower limit register or Ø if the error occurred in the monitor.

The user should correct the cause of the error and execute the program again.

Error Code	Description
0000	Undefined operation code
0001	File not opened
0002	File not defined
0003	End of file encountered
0004	Internal file management buffers full
0005	Hardware error
0006	Disk volume full
0007	Illegal operation code
0008	Directory full
0009	Duplicate file name
0010	Logical unit not defined
0011	Buffer outside user memory space
0040	EIA interface error

TABLE 2-2
GENERAL SYSTEM ERROR CODES

Abort error messages are output on the system logging device on the occurrence of one of the events described in table 2-3. The message output is of the form:

ABORT ERR yyyy AT zzzz U ssss

where: yyyy is one of the error codes shown in table 2-3
zzzz is the memory location where the event occurred
ssss is the contents of the lower limit register or Ø if the event occurred in the monitor

Error Code	Description
0000	Power failure (printed on power restart)
0001	Illegal operation
0002	Privileged instruction violation
0003	Memory protect violation

TABLE 2-3
ABORT ERROR CODES

2.2 BOOTSTRAP INITIALIZATION PROGRAM - FDBOOT

The program FDBOOT is used to define and install the system loader file, and attach the specified system file to the loader. An example of the procedure used to establish a new system on an initialized diskette is as follows:

A) Mount a system diskette in drive Ø and an initialized diskette in drive 1

B) Boot the operating system

C) Allocate the bootstrap file and system file on drive 1 using the following commands:

```
//DEFINE,FD1,SYSLOD,64,1,12.
//DEFINE,FD1,NSPMON,64,3,78.
```

D) Using the program CPYOBJ, copy the file NSPMON from diskette Ø to 1.

E) Boot the FOBASR program using the procedure for loading operating system independent programs. When the program begins execution, the loader file name will be requested by the message:

LOADER FILE NAME =

Remove the system diskette from unit Ø and replace it with the new diskette being generated. The operator should then enter the six character file name, SYSLOD in this example. The program then requests the system file name by the message:

SYSTEM FILE NAME =

to which the operator should enter the six character system file name. In this example, NSPMON should be entered. The system loader will then be written onto the diskette. Bootstrap initialization is now complete and the system can be loaded from the new diskette.

NOTE: If file name is less than 6 characters the space bar must be used to complete the 6 character name.

2.3 INITIALIZATION PROCEDURE PROGRAM - INITLZ

The program INITLZ provides the user the facility to format and initialize new diskettes. In order to run the program, assign logical unit 5 to the input control device, assign logical unit 32 to the disk drive to be used for initialization, load the program into memory, enable the initialization switch on the disk drive, load the diskette to be initialized, and execute the program. The program will request input of 12 characters which will be used as the volume name and serial number then proceed with the formatting and initialization process. For example:

```
//ASSIGN,5,KEY.
//ASSIGN,32,FDØ.
//LOAD,FDØ,INITLZ
    ... load the diskette to be initialized and
        and enable the initialization switch
//EXECUTE.
SYSTEMDISKØ1
    ... when complete, disable initialization
        switch
```

In this example, the volume name will be SYSTEM and the serial number will be DISKØ1.

2.4 VOLUME LISTING PROGRAM - CATLOG

The program CATLOG provides the user the facility to list the directory of a diskette volume. In order to run the program, assign logical unit 6 to the printing device, assign logical unit 32 to the diskette volume, then load and execute the program. For example:

```
//ASSIGN,6,KEY.
//ASSIGN,32,FDØ.
//EXECUTE,FDØ,CATLOG.
```

2.5 DISK COPY PROGRAM - CPYDSK

The program CPYDSK provides the user the facility to copy one diskette volume to another diskette volume. Copies are done on a track by track basis with a write check on the output diskette volume. In order to run the program, assign logical unit 4 to the input control device then load and execute the program. The following messages and directions will be output:

```
CPYDSK  VØ2-Ø5

SOURCE DISKETTE IN DRIVE Ø
DESTINATION DISKETTE IN DRIVE 1
ENTER 6 CHARA VOL:
```

At this point the user should mount the diskette volumes as directed. If the system is a single drive system, sense switch 1 should be set to alternate reads and writes on the one available disk drive. After setting up the disk drives and diskette volumes, enter the six character volume name to be used followed by a carriage return. The program will respond by typing:

```
ENTER 6 CHARA SN:
```

The user should enter the six character serial number to be used followed by a carriage return. Disk reads and writes will then proceed until the copy operation is complete. For single drive systems, the user will be prompted by the message READ when the input diskette is to be mounted and the drive select set to unit zero and WRITE when the output diskette is to be mounted and the drive select set to unit one. After each prompt, the user should type a carriage return to indicate that the appropriate actions have been completed.

2.6 TERMINAL SOURCE EDITOR - TSE980

TSE980 is a terminal source editor that has been provided for the movement of source format files. Refer to the Model 980 Computer, Basic System Use and Operation Manual for a complete description of its operation.

2.7 GENERALIZED OBJECT COPY PROGRAM - CPYOBJ

CPYOBJ is a general purpose program for moving and concatenating object format files and has been provided for the movement of object format files. Refer to the Model 980 Computer, Basic

System Use and Operation Manual for a complete description of its operation.

2.8 SYMBOLIC ASSEMBLER PROGRAM - SAPG

The TI SAPG assembler accepts TI 980 Assembly Language Files as source files and outputs machine acceptable object code as destination files. These can then be executed immediately on the TI 980 or linked with others by LINKG. See TI Basic System Use & Operation for the 980 Comp. T.O. 31S5-4-421-1

2.9 OVERLAY LINK EDITOR - LINKG

LINKG is a one-pass editor that links or merges separately assembled subprograms into one that can be loaded and executed. This allows modification of one subprogram without reassembling the entire program. See TI Basic System Use & Operation for the 980 Comp. T.O. 31S5-4-421-1

SECTION 3

MODEL BUILD PROGRAM NSPBLD

3.1 INTRODUCTION

Models are created for the NSP system by operator interaction with the modeling program NSPBLD. This program provides prompts to the user that lead through the definition of the various data elements required to define a model. Commands are also provided which allow the user to examine and modify data definitions, store and retrieve partial or completed models, and list the data of the various model data elements. The following sections provide the user an overview of the entire model organization, information on general operating conventions, and a detailed discussion of the prompts and replies for all of the model data elements.

3.2 OVERVIEW OF MODEL ORGANIZATION

An NSP model consists of a collection of several modules which provide the data to define light strings, surfaces, coordinate system relationships, visual priority relationships, processing control, data base management relationships, and the association of one set of data to another. These modules and the general purpose for each of them is as follows:

- Light Modules — Each model may contain up to four light modules which provide the definitions for various types of light strings.
- Surface Modules — Each model may contain up to four active surface modules which provide the definitions and relationships for surfaces. The number of surface modules is directly related to the number of hardware surface processor options that exist in the system.
- Environment Module — Each model must contain at least one environment module for associating the light and surface modules and to provide global information about the model.
- Group Module — At least one group module is required to provide information to the real-time system about the available models.

Each of the various types of modules is defined and manipulated by a corresponding section of NSPBLD. The user will select which type of module is to be manipulated, then proceed to work on that module (see the NSPBLD flowchart illustration, figure 3-1). This approach provides a structured organization to both the model and the commands needed to manipulate the data.

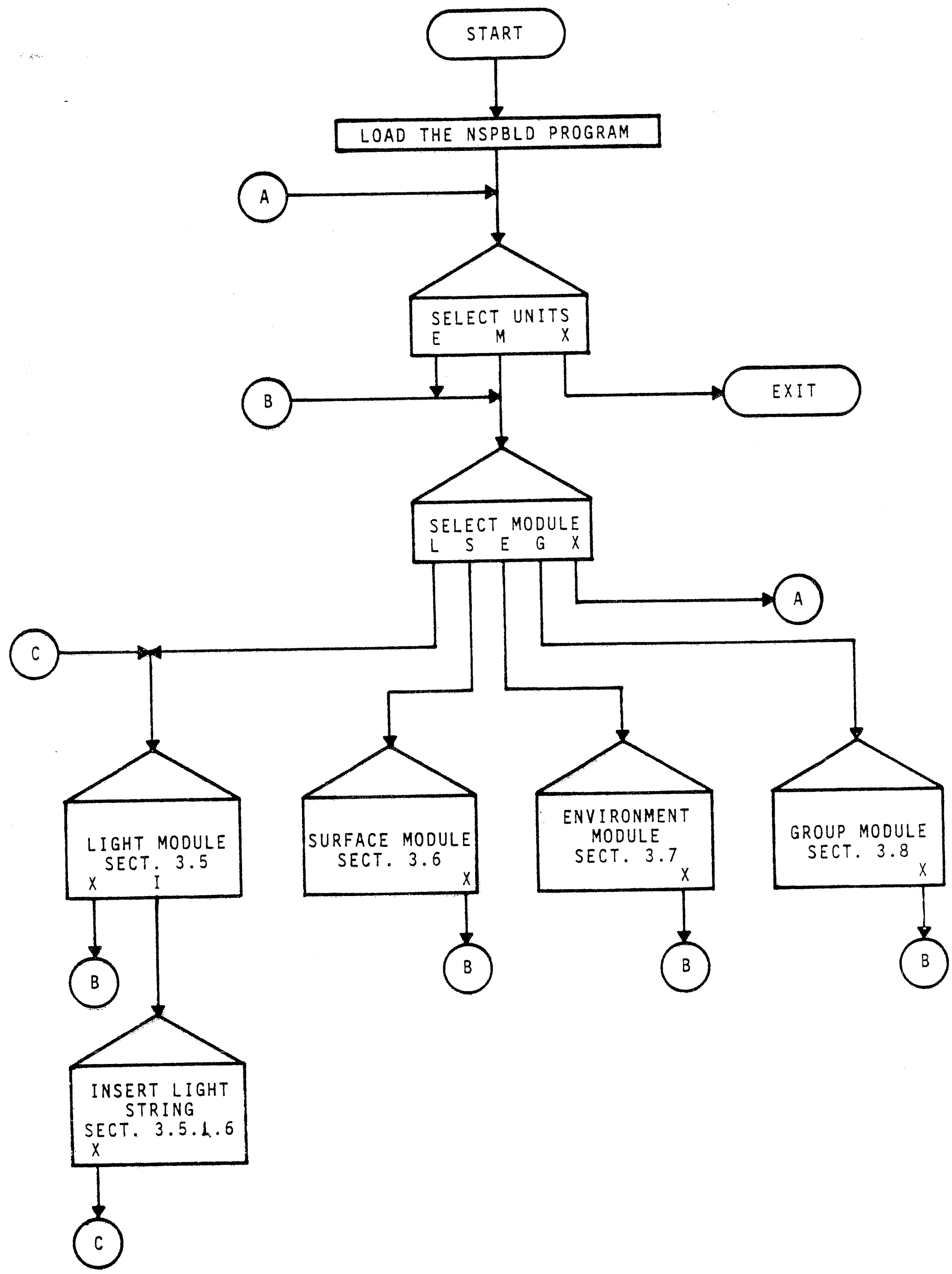


FIGURE 3-1
NSPBLD FLOWCHART ILLUSTRATION

3.3 GENERAL CONVENTIONS

The dialogue between the modeling program NSPBLD and the user is basically that of the user responding to a prompt message from the program. User responses to these prompts are such things as single alphabetic characters, short text strings, names, integer numbers, or real or fractional numbers. Appendix A provides a summary of all prompts and the type of response that is expected. Each user reply to a prompt is terminated by either a tab character (control-I) or a carriage return. In some instances, a tab or carriage return alone is an appropriate response. When such inputs are legal, entering a tab in response to a prompt will result in the program using the existing definition, echoing that definition as the response to the prompt, and continuing; a carriage return will do the same function without echoing the existing definition. This facility is particularly useful when entering many similar definitions and/or making minor changes to existing definitions.

Names that are used by the system are intended to help the user in associating and identifying various data elements. Names that are used in each type of module, the interpretation of that name, and any special characteristics are as follows:

Light Modules

P1 thru P8	Entry point names
R1 thru R110	Red light string names
A1 thru A110	Amber light string names
O1 thru O110	Orange light string names
W1 thru W110	White light string names
G1 thru G110	Green light string names.

Since light strings must be maintained in color order and sequential locations in the light module, specification of a string name that is higher than currently exists will result in the next available string name being used. For example, inserting R110 will always result in the definition of the next red string. If the user specifies an existing string name when doing an insert, it will be inserted as requested and all string names for that color incremented by one.

Surface Module

P1 thru P8	Entry point names. There are eight entry points each for edges and faces.
E1 thru E64	Edge names. Since edge definitions must be maintained in sequential locations in the surface module, specification of an edge name that is higher than currently exists will result in the next available edge name being used. For example, inserting E64 will always result in the definition of the next edge.
F1 thru F64	Face names. Since face definitions must be maintained in sequential locations in the surface module, specification of a face name that is higher than currently exists will result in the next available face name being used. For example, inserting F64 will always result in the definition of the next face.
N1 thru N5	Priority tree node names.
I1 thru I16	Command list instruction names. Since command list instructions must be maintained in sequential locations in the surface module, specification of an instruction name that is higher than currently exists will result in the next available instruction being used. For example, inserting I16 will always result in the definition of the next instruction.

Environment Module

M1 thru M11	Light and Surface module names.
R1 thru R4	Runway offset names.
N1 thru N6	Priority tree node names. Priority processing begins with node N1.
I1 thru I64	Command list instruction names. Since command list instructions must be maintained in sequential locations in the environment module, specification of an instruction name that is higher than currently exists will result in the next available instruction being used. For example, inserting I64 will always result in the definition of the next instruction.

C1 thru C8 — Coordinate system names.

T1 thru T7 — Transformation names.

S1 thru S4 — Hardware surface processor option names.

Group Module

GØ thru G127 — Group number names. Group numbers are used to associate the model select number received from the host computer with the actual data on the diskette.

M1 thru M32 — Environment module names. These module names are used to order the environment module references used for data base management.

3.4 PROGRAM EXECUTION

Program loading and execution is accomplished in the following manner where 4=CONTROL, 5=GRAPHICS, 6=LISTING, F=CURSOR FIND:

```
//ASSIGN,4,KEY.     OR //ASSIGN,4,TEK (CURSOR NOT AVAIL)
//ASSIGN,5,TEK.     OR //ASSIGN,5,DMY.
//ASSIGN,6,KEY.     OR //ASSIGN,6,TEK (CURSOR NOT AVAIL)
//ASSIGN,F,KEY.     OR //ASSIGN,F,TEK (CURSOR NOT AVAIL)
//EXECUTE,FDØ,NSPBLD.
```

NOTE: If TEK is selected, cursors are not available to use.

The program will be loaded and start execution. The program name and version number will be output as follows:

```
NSPBLD VØ2-Ø1
NOVO-SP MODELING SYSTEM
```

This version number should be referenced when reporting any error in the modeling system.

The first question asked is UNITS= to which the user should reply with E for english, M for metric, or X to exit the program. After specifying the units, the prompt LIT,SUR,ENV,GRP>> will be output to request the type of module to be input or edited. User responses to this prompt are: L for light modules, S for surface modules, E for environment modules, G for group modules, and X to return to the unit selection. After entering the desired module selection, the prompt for that module will be output. Refer to the editing function sections for each module for a description of the available commands.

3.5 LIGHT MODULE

The light module file contains all of the information that is required to define the light strings in a model. A light module contains entry points that group light strings and the actual light string defintions.

This section describes the commands provided by NSPBLD to define and manipulate the data contained in a light module.

3.5.1 EDITING FUNCTIONS

Editing functions provided for defining and manipulating light strings are selected by single character codes in response to the prompt LIT>> and are terminated by either a tab or carriage return. Functions that are available and the associated character code are as follows:

E = edit light module
R = read light module
Z = zero light module
X = return to module selection
P = define entry point
I = insert light string
C = change light string
K = kill light string
L = list light string
S = select first/last or first/delta
    entry mode
W = write module
B = blank screen (Tektronix 4010 option)
D = display string (Tektronix 4010 option)
V = set view position (Tektronix 4010 option)
F = find light string (Tektronix 4010 option)
A = angle rotation
T = translation
N = print statistical information

Each of these functions and its relationship to generating the light module is discussed in 3.5.1.1 through 3.5.1.18.

3.5.1.1 Edit Light Module (E)

The edit module function provides the user the ability to declare the current module in memory to be active without destroying the data. This may be used after a program abort or a module write function to gain access to the light module data. It is assumed by the system that the user is enabling access to valid light module data.

3.5.1.2 Read Light Module (R)

Previously generated light modules are read into the modeling system using the read function. In response to the selection of the read function, the system will prompt the user by typing:

FILE NAME=

The user reply is either CS1: or CS2: (in foreground mode only) if the module is on cassette tape or FDØ: or FD1: followed by a file name from one to six characters in length if the module is on floppy disk. If input is from cassette tape, the user is expected to have rewound the tape if necessary. If a file name is entered and the device name is omitted, FDØ: is assumed. If an active module currently exists, the system will ask for verification of the read command by typing VER> to which the user must respond with either "Y" to complete the read command or "N" to abort it.

3.5.1.3 Zero Light Module (Z)

The zero function clears all string definitions and resets all of the pointer structures for the light module. This should be the first command executed when defining a new light module. If an active module currently exists, the system will ask for verification of the zero command by typing VER> to which the user must respond with either "Y" to complete the zero command or "N" to abort it.

3.5.1.4 Exit (X)

When the user has completed light module editing the exit function is used to allow selection of the editing mode for another type of module. If an active module still exists when the exit function is selected, the system will respond with the message:

ACT MOD
VER>

to which the user must reply with either a "Y" to exit or a "N" to abort the command.

3.5.1.5 Entry Point Definition (P)

Entry points are used to associate groups of light strings that are to be processed together by the hardware. Each light module must have at least one defined entry point and may have up to eight entry points. After selecting the entry point definition function the user will receive the prompt:

ENTRY=

to which the user should respond by entering the name of the entry point to be defined, P1 through P8, terminated by either a tab or a carriage return. If a legal entry point name was entered, the system will request the first string by typing STR #1= and waiting for a string name. After a legal string name has been entered, a second string will be requested by the message STR #2= typed on the console. The user should respond by entering the second string number. When a second legal light string name has been entered, this will result in the entry point defining a group of strings from the first through the second string names inclusively. If the user wishes to delete an entry point definition, define the entry point with the first string name higher than the second string name. For example,

ENTRY= P3 STR #1= R2Ø STR #2= R1

would cause entry point P3 to be deleted.

Since the entry points are entered independently and define a contiguous group of light strings from the first to the second light string specified, care should be exercised not to overlap entry point specifications since this will result in incorrect light string groupings. The user also needs to be aware that entry points define contiguous light string groups but cannot change the color order of the light strings. In cases where the user would need to rearrange the color order to achieve the proper contiguous groups, it must be accomplished by the use of more entry point groups and proper references to those entry points.

3.5.1.6 Insert Light String (I)

The insert light string function provides the user the ability to define new data base light strings in the active light module. After selecting the insert function, the user will be prompted to select the type of light string as follows:

FLS,ROT,VASI,SFL,NML>>

User responses to this prompt and the type of light string selected are:

F - flashing light string
R - rotating light string
V - VASI (Visual Approach Slope Indicator)
S - strobe light string (sequential flashing

lights)
N - normal light string
X - return to command selection

After selecting the type of light string to be inserted, the system will request the string number to be inserted by typing the message STR #=. The user should enter the light string number terminated by either a tab or carriage return. If the user is inserting a series of light strings of the same color, the next sequential light string of that color will be used if the user responds to the STR #= prompt by typing a table for strings after the first string. If the user specifies the string number for an existing string, the new string will be inserted as designated and all of the strings numbers of that color following the new string will be incremented by one. Once the string number has been selected, any special characteristics required by the light string are requested followed by the actual light string characteristics. Special characteristics are discussed by light type and the basic string light characteristics are discussed under the normal light string specification.

Flashing Light Strings (F)

Flashing light strings require the user to specify flash pattern characteristics in the form of light on time, period, and phase. The on time is requested by the message ONT= to which the user should enter the integer number of frames that the light string should be on. Flash period is requested by the message PER= and is the flash period in frames. If the value input is less than the on-time, it is set equal to the on-time. Flash phase is requested by the message PHA= and is the phase delay in frames for this light string. Figure 3-2 shows two light strings set up to flash in alternating sequence. After the flash pattern characteristics have been specified, the light characteristics discussed under normal light strings are entered.

Note: 30 frames = 1.0 second

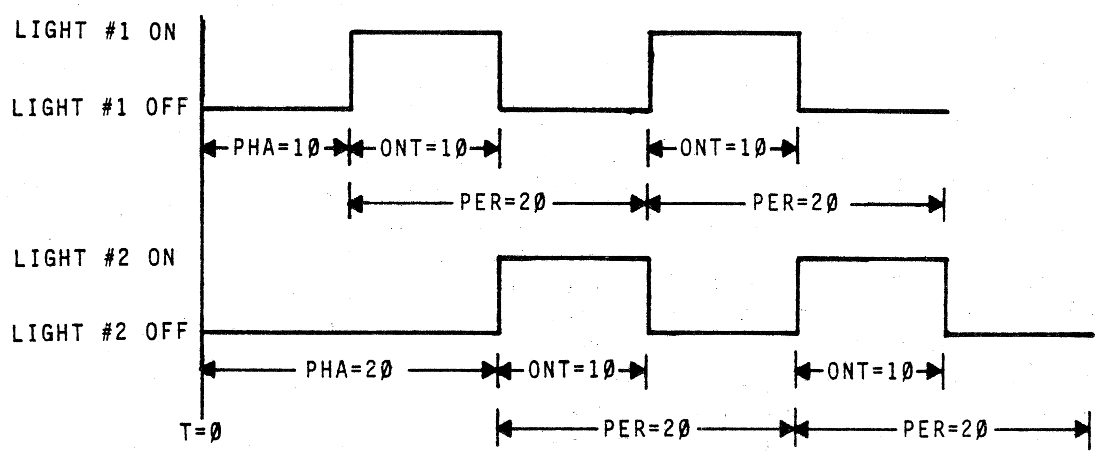


FIGURE 3-2
FLASHING LIGHTS

Rotating Light String (R)

Rotating light strings require the user to specify update characteristics. The user will be asked for the per frame update angle by the prompt UP ANG= and the initial heading angle by the prompt HDG=. After the update characteristics have been specified, the light characteristics discussed under normal light strings are entered.

VASI Light Strings (V)

Definition of the VASI (Visual Approach Slope Indicator) light strings requires that the user define an appropriate set of red and white light string pairs for the entire VASI set. The user will be asked to specify the number of master light strings by the prompt # MASTERS = to which a reply of either 2 or 3 should be made. The user will next be asked to specify the number of repeat strings for each master string by the prompt # REPEATS =. An appropriate response is an integer value between Ø and 7. After specifying the number of light strings in the VASI set, the prompt ELEV= will request the glide slope angle for the furthest master in the set. Successive masters will be assumed to have an angle of approximately 1/2^{o} less than the previous master. The user will then be asked for the red string number by the prompt STR #1= and the white string number by the prompt STR #2=. User response should be R1 through R11Ø and W1 through W11Ø respectively. Definition of successive master strings and associated repeat strings will then proceed in the manner described for normal light strings with the user specifying a total of (M + M*R) light strings where M is the number of masters and R is the number of repeats for each master. See figure 3-3.

In addition to the requirements and constraints described above the user must be aware of the following assumptions:

- computation of the transition zone assumes that master bars are approximately 700 feet apart.
- the furthest master/repeat set must be defined first
- VASI lights cannot be part of a dynamic coordinate system (refer to paragraph 3.7).

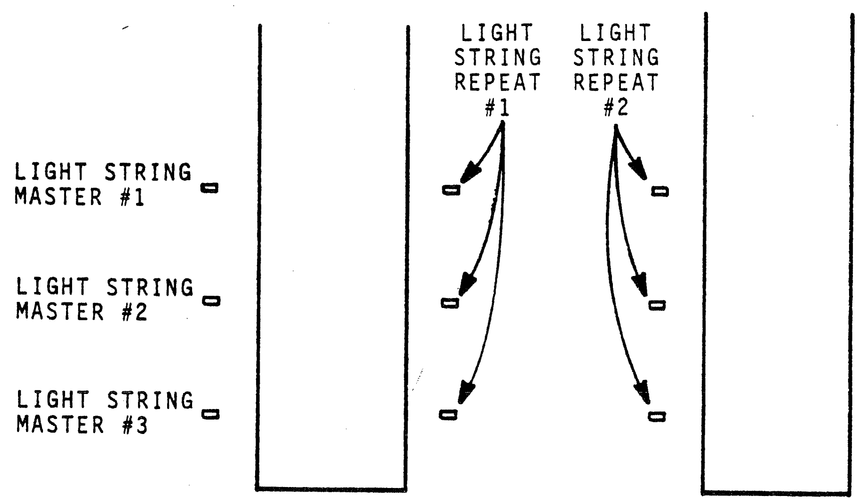


FIGURE 3-3
VASI LIGHT STRINGS

Strobe Light Strings (S)

Strobe light strings require the user to specify flash pattern characteristics in the form of off-time between sequential lights, period, and phase. The off-time is requested by the message OFT= to which the user should enter the integer number of frames between a particular light going off and the next sequential light being turned on. Flash period is requested by the message PER= and is the flash period in frames for the entire strobe string. Flash phase is requested by the message PHA= and is the phase delay in frames for this particular light string. After the flash pattern characteristics have been specified, the light characteristics discussed under normal light strings are entered.

Normal Light Strings (N)

Normal light string characteristics define the basic character of the lights being used in the model. The first parameter requested for input is the number of lights in the string. The message #LITS= is typed on the console to which the user should reply with a light count between 1 and 255. After entering the light count, the system will request the light string intensity by typing the message INT=. Valid light intensities range from Ø to 1.9967 where the maximum screen intensity is achieved with a model intensity of 1.0. Values greater than 1.0 will increase the fog penetration capability of the lights but the perceived intensity will not be greater.

All light strings have shape and directional characteristics that are specified in response to the message SAW= when typed on the console. The response to this prompt is an integer value which describes the shape, width, and directional characteristics. These responses are shown in table 3-1 and figure 3-4 shows the shape patterns.

If the user selects a curved string, the message ANG= will be typed to which the user should specify the rotation angle between successive lights in the string.

If the user selects a directional string, the message HDG= will be typed to which the user should respond with the directional heading, in degrees, for the directional light pattern.

After specifying the shape and width, the system will request the software switch number by prompting the user with the message SWT #=. The user should enter a value between Ø and 32 which defines the software switch to turn the light string on and off. If Ø is entered, the light string is assumed to always be turned on.

Associated with all light strings is a set of flags used to control special processing functions. The user will be asked if the current setting of these flags is to be changed by the message FLG> to which the user should reply Y or N. If the user responds with a Y, all of the flags described in table 3-2 will be requested. Whatever value is set in these flags will remain set if the user is inserting successive light strings.

After specifying all of the above light characteristics, the light string position must be defined as either a first and last point or a first point and spacing. If the user has selected the first/last point format, the prompts X=, Y=, and Z= will be typed for both the first and last points of the light string.

If the first/spacing point format is being used, the prompts X=, Y=, Z=, HDG=, ELEV= and DELTA= will be output. The user responses for X, Y, and Z define the location at which the string is to start, HDG defines the initial heading angle, ELEV defines the elevation angle, and DELTA defines the spacing between sequential lights. The system will compute and type out the last point coordinate of the light string.

When a curved string is being defined, the system will automatically select the first point/spacing format for that particular light string but does not change the format for other light strings.

NOVA-SP MODEL BUILDING CONTROL-TABLE LIGHT SWITCH NUMBERS

FUNCTION	RWY"0"	RWY"1"	RWY"2"	RWY"3"
NO SWITCH	0	8	16	24
VASI	1	9	17	25
REILS	2	10	18	26
CENTERLINE	3	11	19	27
TOUCHDOWN ZONE	4	12	20	28
HIRLS	5	13	21	29
APPROACH	6	14	22	30
STROBE	7	15	23	31

RESPONSE	DIRECTIONAL CHARACTER	SHAPE	WIDTH
Ø	omnidirectional	-	-
1	curved string	-	-
10	directional	1	28°
11	directional	1	53°
12	directional	1	90°
13	directional	1	152°
20	directional	2	28°
21	directional	2	53°
22	directional	2	90°
23	directional	2	152°
30	directional	3	28°
31	directional	3	53°
32	directional	3	90°
33	directional	3	152°

TABLE 3-1
LIGHT STRING DIRECTION, SHAPE, AND WIDTH CHARACTERISTICS

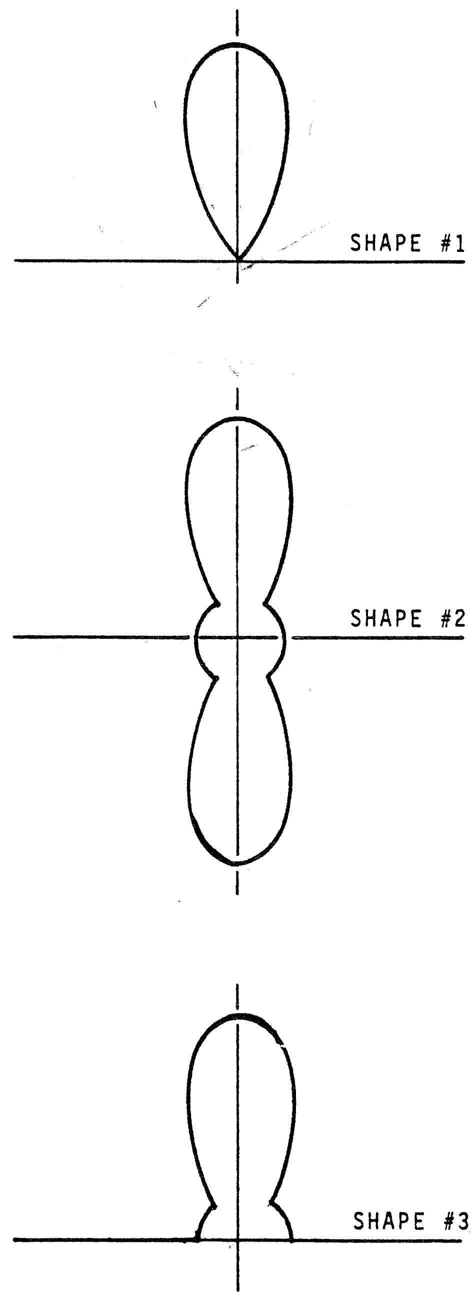


FIGURE 3-4
LIGHT SHAPES

PROMPT FOR FLAG	FLAG SETTING
FFB=	0 = apply fog function A to the light string 1 = apply fog function B to the light string Note: fog function B provides greater attenuation for mid range light strings such as those used in the airport area
*B=	0 = apply fog function to this light string 1 = do not apply fog function to this light string
*T=	0 = translate this string as required during hardware processing 1 = do not translate this string during hardware processing
*SP=	0 = do same point test 1 = do not do same point test Note that for strobe lights, the system will force *SP=1
PRI=	0 = light string does not take priority over all surfaces 1 = light string takes priority over all surfaces
IO=	0 = light string visible in all channels 1 = light string valid for instructor monitor only
CHAN=	0 = light string to be processed by all channels 1-8 = light string to be processed only by the specified channel

TABLE 3-2
LIGHT STRING FLAGS

3.5.1.7 Change Light String (C)

The change light string function provides the user the ability to modify the definition of an existing light string. The system will request the string number to be changed by typing the message STR #= on the console. After a legal string number has been entred, the system will retrieve the string definition from the light module and proceed to ask all appropriate questions as described under the insert string function. Any parameters that are to be changed are simply reentered when requested; prompts for parameters that are to remain unchanged are answered with a tab. As with the insert function, successive light strings after the first one can be modified by typing a tab when the string number is requested.

Under the current implementation of NSPBLD, the parameters HDG, ELEV, and DELTA must be reentered when specifying the string spacing. These values cannot be tabbed through to retain the current value.

If the user specifies a nonexistent light string, the system will output the error message NO ITEM and return to the prompt LIT>> to allow the user to select another function.

3.5.1.8 Kill Light Strings (K)

The kill light string function provides the user the facility to delete light strings from the light module. After selecting the kill function, the user will respond to the prompt STR #1= with the first string to be deleted followed by the last string to be deleted in response to the prompt STR #2=. If the second string name is less than the first string name, no strings are deleted. If part of the light strings that comprise a VASI set are deleted, the entire VASI set is deleted.

3.5.1.9 List Light Strings (L)

The list light string function provides the user the facility to list the light strings defined in the light module. The user will be asked to specify the first and last string to be listed in response to the prompts STR #1= and STR #2=. The following example shows the listing format for each type of light string.

```
#W30  FLASHING:  ONT=1          PER=15         PHA=0
   # LTS=4   INT=1.3125  SAW=12  HDG=344.5312  SWT #=7
   FFB=1 CHAN= 0   *B= 0 *T= 0 *SP=1 PRI=0 IO= 0
   START:  X=  7000.0000      Y=  -75.0000       Z=  1.0000
   END:    X=  7000.0000      Y=  -75.0000       Z=  1.0000
```

FLASHING (WHITE) LIGHT
PRINTOUT SAMPLE

```
#R2   ROTATING:  UP ANG= 19.9960   HDG=0.0000
   # LTS=2   INT=0.8984  SAW=23  HDG=0.0000    SWT #=0
   FFB=1 CHAN= 0   *B= 0 *T= 0 *SP=0 PRI=0 IO= 0
   START:  X=  7000.0000      Y=  75.0000        Z=  0.0000
   END:    X=  4900.6015      Y=  -2477.4609     Z=  0.0000
```

ROTATING (RED) LIGHT
PRINTOUT SAMPLE

```
#R15  VASI:       # MASTERS=2 # REPEATS=1 ELEV= 3.2500
   # LTS=3   INT=1.0000  SAW=11  HDG=180.0000  SWT #=5
   FFB=1 CHAN= 0   *B= 0 *T= 0 *SP=0 PRI=0 IO= 0
   START:  X=  200.0000       Y=  -160.0000      Z=  3.0000
   END:    X=  200.0000       Y=  -150.0000      Z=  3.0000

#R16  VASI:       # MASTERS=2 # REPEATS=1 ELEV= 3.2500
   # LTS=3   INT=1.0000  SAW=11  HDG=180.0000  SWT #=5
   FFB=1 CHAN= 0   *B= 0 *T= 0 *SP=0 PRI=0 IO= 0
   START:  X=  200.0000       Y=  160.0000       Z=  3.0000
   END:    X=  200.0000       Y=  150.0000       Z=  3.0000

#R17  VASI:       # MASTERS=2 # REPEATS=1 ELEV= 3.2500
   # LTS=3   INT=1.0000  SAW=11  HDG=180.0000  SWT #=5
   FFB=1 CHAN= 0   *B= 0 *T= 0 *SP=0 PRI=0 IO= 0
   START:  X=  -500.0000      Y=  160.0000       Z=  3.0000
   END:    X=  -500.0000      Y=  150.0000       Z=  3.0000

#R18  VASI:       # MASTERS=2 # REPEATS=1 ELEV= 3.2500
   # LTS=3   INT=1.0000  SAW=11  HDG=180.0000  SWT #=5
   FFB=1 CHAN= 0   *B= 0 *T= 0 *SP=0 PRI=0 IO= 0
   START:  X=  -500.0000      Y=  -160.0000      Z=  3.0000
   END:    X=  -500.0000      Y=  -150.0000      Z=  3.0000
```

VASI (RED) LIGHT
PRINTOUT SAMPLE

```
#W1    STROBE:      OFT=0              PER=15           PHA=0
   # LTS=11   INT=1.2500   SAW=32  HDG=180.0000  SWT #=7
   FFB=1 CHAN= 0    *B= 0 *T= 0 *SP=1 PRI=0 IO= 0
   START:   X=   -4000.0000       Y=   0.0000          Z=  15.0000
   END:     X=   -2000.0000       Y=   0.0000          Z=   5.0000
```

STROBE (WHITE) LIGHT
PRINTOUT SAMPLE

```
#R14  NORMAL:
   # LTS=20   INT=0.7500   SAW=11   HDG=180.0000   SWT #=3
   FFB=1 CHAN= 0    *B= 0 *T= 0 *SP=0 PRI=0 IO= 0
   START:   X=   6000.0000       Y=   0.0000          Z=   0.0000
   END:     X=   6950.0000       Y=   0.0000          Z=   0.0000
```

NORMAL (RED) LIGHT
PRINTOUT SAMPLE

```
#W32  NORMAL:
   # LTS=30   INT=1.0000   SAW=33   HDG=180.0000   SWT #=1
   FFB=1 CHAN= 0    *B= 0 *T= 0 *SP=0 PRI=0 IO= 0
   START:   X=   -4000.0000      Y=   7.0000          Z=  15.0000
   END:     X=   -1100.0000      Y=   7.0000          Z=   0.5000
```

NORMAL (WHITE) LIGHT
PRINTOUT SAMPLE

3.5.1.10 Select Position Mode (S)

The selection of first point/last point or first point/spacing for defining light string position is done using the select position mode function. In response to the prompt FL/SP> the user should specify F for first point/last point mode or S for first point/spacing mode.

3.5.1.11 Write Light Module (W)

After defining a light module, it is written on cassette tape or floppy disk by invoking the write function. After selecting the write function, the system will request the output file name by typing:

FILE NAME=

The user reply is either CS1: or CS2: (in foreground mode only) if the module is to be written onto cassette tape or FDØ: or FD1: followed by a file name from one to six characters in length if the module is to be written onto floppy disk. If output is to a cassette tape, the user is expected to have rewound the tape to load point if necessary.

If multiple output files are to be put onto cassette tape, they must be written sequentially without rewinding the tape. To add modules to the end of a cassette tape that already contains modules, the user must first read past the existing modules to position the tape before outputting a new module to the tape.

When the output is to be to floppy disk, the file name can be specified without the device name and the system will assume FDØ as the output device.

3.5.1.12 Blank Screen (B)

The blank screen function erases the screen of the Tektronix 4010 display. If the system does not include the Tektronix 4010 hardware, the command is ignored.

3.5.1.13 Display Light Strings (D)

For systems which contain the Tektronix 4010 display, the display function allows the user to specify those light strings in the active light module that are to be displayed. After the function has been selected, NSPBLD will request the strings to be displayed by typing the messages STR #1= and STR #2=. To these prompts, the user should respond with the first and last light strings to be displayed. If the system does not contain a Tektronix 4010 display, this command is ignored.

Note: View Position must be set first.

3.5.1.14 Set View Position (V)

The set view position function defines the viewing window for displaying light strings on the Tektronix 4010 display. To define the viewing window, the user will specify an X and Y ground position in response to the messages X= and Y= and a ground span to be covered by the display in response to the message SPAN=. If light strings have previously been displayed from the active module, the screen will be erased and the light strings displayed using the new viewing window. For systems without a Tektronix 4010 display, this command is ignored.

3.5.1.15 Find Light String (F)

For systems that have a Tektronix 4010 display, the find function allows the user to locate strings that are displayed on the screen. To use the find function, the user should position the crosshairs displayed on the screen over or near a light in question. Any printing character except an X should then be entered on the console in response to the message POS CROSS>. NSPBLD will then compare the crosshair position to all of the displayed light strings in the active light module and output, in listing format, any light strings near the crosshair position. This process may be repeated until the character X is entered in response to POS CROSS> and the system will then return to the LIT>> prompt.

3.5.1.16 Angle Rotation for Light Strings (A)

Selection of the angle rotation function allows the user to rotate all of the defined light strings in the active light module by a specified angle. The system will prompt the user with the message ANG= to which the user should enter the rotation angle in degrees.

3.5.1.17 Translation for Light Strings (T)

Selection of the translate function allows the user to translate all of the defined light strings in the active light module. The user should define the translation vector in response to the series of prompts X=, Y=, and Z=.

3.5.1.18 Print Statistics (N)

Statistical information concerning the light strings defined in the active light module will be output when the print statistics function is selected. Information that is printed out is shown in the following example.

```
LIT >> N

RED LITS =   332  RED STRS =  36
ORG LITS =     0  ORG STRS =   0
AMB LITS =    80  AMB STRS =   8
WHT LITS =  1478  WHT STRS =  60
GRN LITS =   118  GRN STRS =   6

TOT LITS =  2008  TOT STRS = 110

FLS = 4  ROT = 0  VASI = 2  SFL = 2

LIT >>
```

3.6 SURFACE MODULE

The surface module file contains all of the information that is required to define the surfaces in a model. A surface module contains edge and face definitions along with control and grouping information such as entry points, command lists, and priority trees. All surface modules in a model must contain the same number of faces.

This section describes the commands provided by NSPBLD to define and manipulate the data contained in a surface module.

3.6.1 EDITING FUNCTIONS

Editing functions provided for defining and manipulating edges and faces are selected by single character codes in response to the prompt SUR>> and are terminated by either a tab or carriage return. Functions that are available and the associated character code are as follows:

E = edit surface module
R = read surface module
Z = zero surface module
X = exit to module selection
P = define entry point
M = define mirror point vector
S = define center line management
I = insert edge, face, instruction, node
C = change edge, face, instruction, node
K = kill edge, face, instruction, node
L = list edge, face, instruction, node
W = write module
B = blank screen (Tektronix 4010 option)
D = display faces (Tektronix 4010 option)
V = set observer viewing position (Tektronix 4010 option)
F = find face (Tektronix 4010 option)
A = angular rotation
T = translate
N = print statistical information

Each of these functions and its relationship to generating edge and face definitions is discussed in the following paragraphs.

3.6.1.1 Edit Surface Module (E)

The edit module function provides the user the ability to declare the current module in memory to be active without destroying the data. This may be used after a program abort or a module write function to gain access to the surface module data. It is assumed by the system that the user is enabling access to valid surface module data.

3.6.1.2 Read Surface Module (R)

Previously generated surface modules are read into the modeling system with the read function. In response to the selection of the read function, the system will prompt the user by typing:

FILE NAME=

The user reply is either CS1: or CS2: (in foreground mode only) if the module is on cassette tape or FDØ: or FD1: followed by the file name from one to six characters in length if the module is on floppy disk. If input is from cassette tape the user is expected to have rewound the tape if necessary. If a file name is entered and the device name is omitted, FDØ, is assumed. If an active module currently exists, the system will ask for verification of the read command by typing VER> to which the user must respond with either "Y" to complete the read function or "N" to abort it.

3.6.1.3 Zero Surface Module (Z)

The zero function clears all edge and face definitions and resets all of the pointer structures in the surface module. This should be the first command executed when defining a new surface module. If an active module currently exists, the system will ask for verification of the zero command by typing VER> to which the user must respond with either "Y" to complete the zero function or "N" to abort it.

3.6.1.4 Exit (X)

When the user has completed surface module editing the exit function is used to allow selection of the editing mode for another type of module. If an active module still exists when the exit function is selected, the system will respond with the message:

ACT MOD
VER>

to which the user must reply with either a "Y" to exit or a "N" to abort the command.

3.6.1.5 Entry Point Definition (P)

Entry points are used to associate groups of edge or face definitions that are to be processed together by the hardware. Each surface module must have at least one defined entry point for edges and for faces and may have up to eight entry points. After selecting the entry point definition function, the user will receive the prompt:

ENTRY=

to which the user should respond by entering the name of the entry point to be defined, P1 through P8, terminated by either a tab or a carriage return. If a legal entry point name was entered, the system will request the first edge or face specification by typing ITEM #1= and waiting for input. After a legal edge or face name has been entered, a second edge or face specification will be requested by the message ITEM #2= typed on the console. The user should respond by entering the second edge or face name. When a second legal edge or face name has been entered, this will result in the entry point defining a group of edges or faces from the first through the second specification inclusively. Note that the system will require that both the first and second items be either edges or faces. If the user wishes to delete an entry point, define the entry point with the first edge or face name higher than the second edge or face name. Since the entry points are entered independently and define a contiguous group of edges or faces, care should be exercised not to overlap entry point definitions since this will result in incorrect edge or face groupings.

3.6.1.6 Define Mirror Point Vector (M)

Definition of a mirror point vector allows the user the ability to manage symmetrical faces such as end markers and touch down zones on runways without having to model them on both ends of the runway. The prompt MS= requests the user to input either Ø to indicate that an existing mirror point is to be deleted or one doesn't exist, or 1 to indicate a mirror point is to be defined or changed. If the mirror point is being defined or changed, the user will be prompted with X= and Y= to which the user should specify the X and Y coordinate values for the mirror point. The specified point represents the rotation origin about which the edges defined by entry point 1 will be rotated 180° by the real-time software (NSPFLY) when doing mirrored surface management. Figure 3-4 shows how some mirrored surfaces might be set up. The current implementation for mirrored surface management requires that all mirrored surfaces must be horizontal surfaces and cannot be part of a dynamic coordinate system. Edge definitions used in defining the mirrored surfaces must also be defined as shown in figure 3-5.

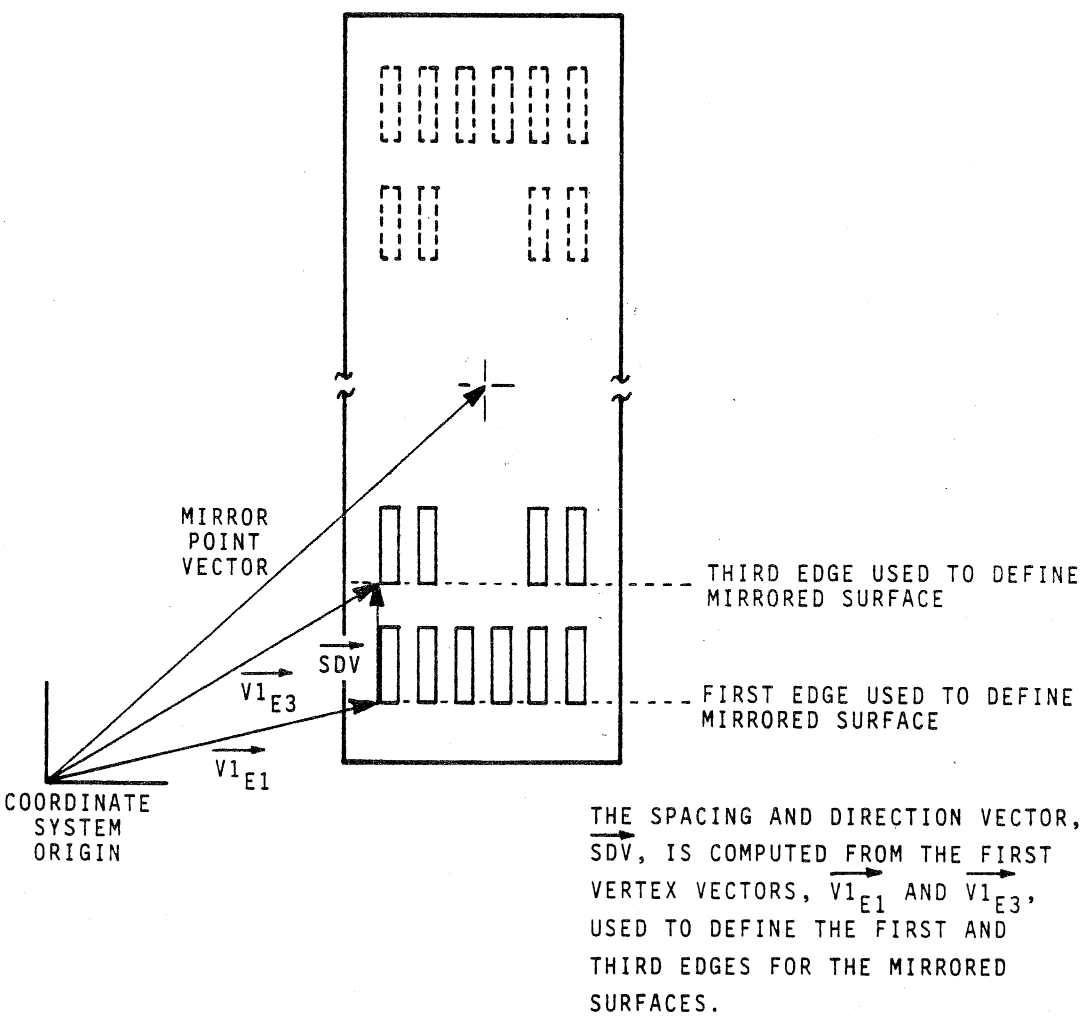
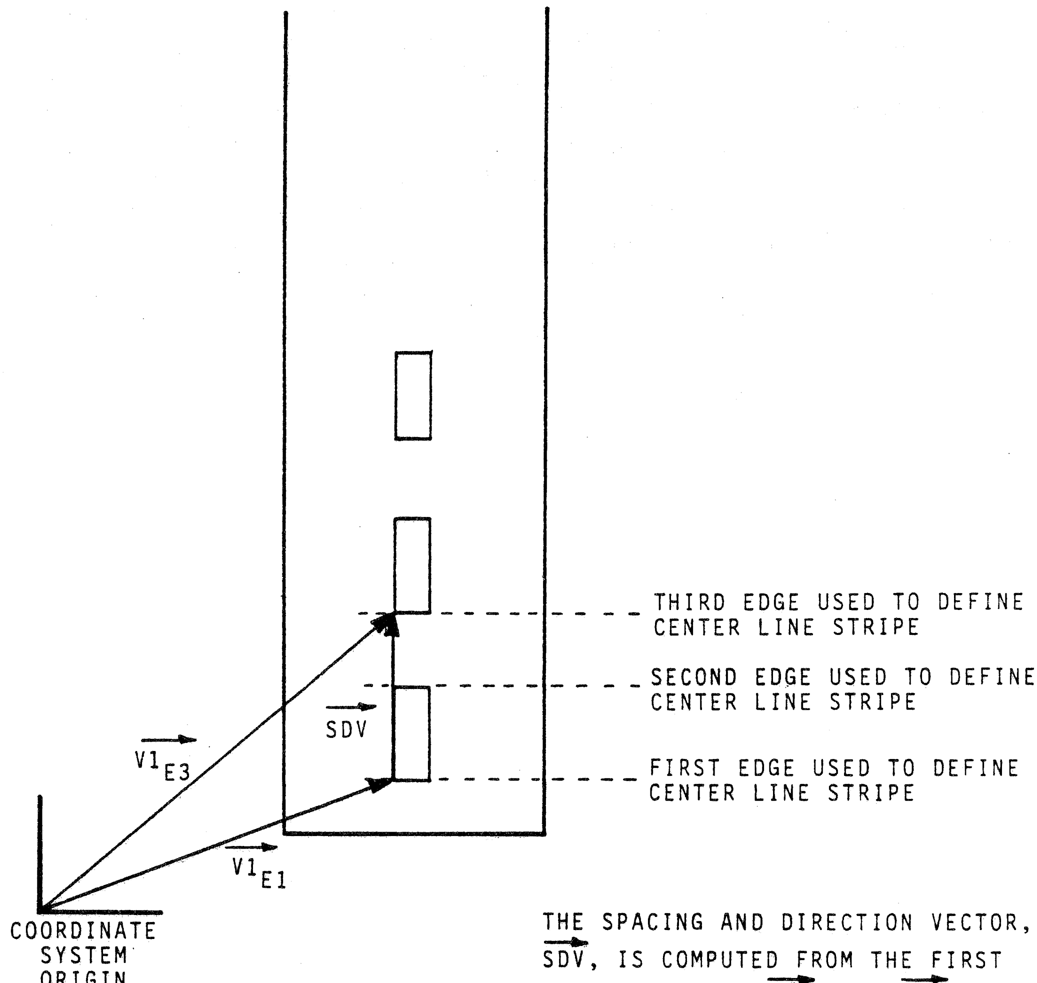


FIGURE 3-5
MIRROR SURFACE AND MIRROR POINT VECTOR

3.6.1.7 Define Center Line Management (S)

Definition of the center line management parameters allows the user to provide a center line strip for the entire length of the runway without modeling every segment of the strip. The user will be requested to specify the number of segments that were modeled by the prompt MODEL STRIPES= and the number of remaining stripes by the prompt REMAINING STRIPES=. The real-time system (NSPFLY) will use this information to manipulate those edges grouped by entry point 2 to do the center line management when those edges/faces are within visual range. When defining the edges to be used for center line face definitions, they must be defined as shown in figure 3-6 to provide the appropriate direction and spacing information for the real-time system. Edges and faces that are center line managed cannot be part of a dynamic coordinate system.



THE SPACING AND DIRECTION VECTOR, $\vec{SDV}$, IS COMPUTED FROM THE FIRST VERTEX VECTORS, $\vec{V1}_{E1}$ AND $\vec{V1}_{E3}$ USED TO DEFINE THE FIRST AND THIRD EDGES FOR THE STRIPE FACES.

FIGURE 3-6
EDGE DEFINITION FOR CENTER LINE MANAGEMENT

3.6.1.8 Insert Edge, Face, Instruction, and Node (I)

The insert function for surfaces allows the user to define edges, faces, command list instructions, and priority tree nodes for the active surface module. After selecting the insert function, the prompt ITEM #= will be output. The user selects the data type and number to be entered by responding with one of the following:

- E1 thru E64 to define edges
- F1 thru F64 to define faces
- I1 through I16 to define command list instructions
- N1 thru N5 to define priority tree nodes

After specifying what data type is to be entered, NSPBLD will request the input that is appropriate for that particular data type entry. If the user is entering several items of the same data type and they are in sequential order, the next item of that type will automatically be selected by typing a tab to the prompt after the first entry has been completed.

Insert Edge Definition

Edges are the basis for defining the faces for the system. An edge definition consists of some control flags and two vertices that define the edge location and direction. When inserting an edge definition the system will prompt the user with the message FLG> to which the user should respond with Y if the flag values are to be changed, and N if the current flag values are to be used. If the user chooses to set the flag values, the following flag prompts will be output.

*T=	0 = translate this edge as required during hardware processing
	1 = do not translate this edge during hardware processing.
CHAN=	0 = edge to be processed by all channels
	1-8 = edge to be processed only by the specified channel

Once the flag values have been set, they will remain at that value for any successive edge definitions as long as insert mode is not left.

After specifying the flags, the user must input the vertices that define the edge position and direction. NSPBLD will output the prompts X=, Y=, and Z= for the first vertex followed by the same prompts for the second vertex. The positive direction of the edge is from the first vertex to the second vertex (see figure 3-8).

Insert Face Definition

Faces are defined in the system by referencing four edge definitions in clockwise order. The sign of the edge name represents whether that edge is to be used in the positive or negative direction relative to the definition of the particular edge. In addition to referencing the defining edges, the face definition also contains information about intensity, illumination, and special processing.

After selecting a face for insertion, the system will prompt the user with the message INT= to define the face intensity. Valid responses are integer values from 0 to 63 which represent the 64 intensity levels for faces. Illumination functions for the face are defined by responding with a Y or N to the prompts LLL= for landing light lobe illumination, GND= for ground illumination, 3D= for 3-dimensional illumination, and OCC= if the surface is to be light occulting. Legal combinations of the illumination flags are shown by figure 3-7.

Flag values for face definitions may be changed by responding with a Y to the prompt FLG> when it is output. If the flag values are not to be changed, respond with an N. Flag prompts and the associated response for each flag is as follows:

*B=	0 =	apply sun shading to the face when defined by the command list
	1 =	do not apply sun shading to this face
*T=	0 =	translate this face as required during hardware processing
	1 =	do not translate this face during hardware processing
CHAN=	0 =	face to be processed by all channels
	1-8 =	face to be processed only by the specific channel

Once the flag values have been set, they will remain at that value for any successive face definitions as long as insert mode is not left.

After specifying the flags, the user must input the edge references that define the face. NSPBLD will output the prompts E1=, E2=, E3= and E4= to which the user should respond with the signed edge name for the four edges that define the face (see figures 3-8 and 3-9). Care should be taken when defining faces to make sure that they are planar. The modeling system does not check for planarity and errors in operation of the data base will occur if the faces are not planar. If the user wishes to define triangular faces rather than quadrilateral faces, simply repeat the first edge reference.

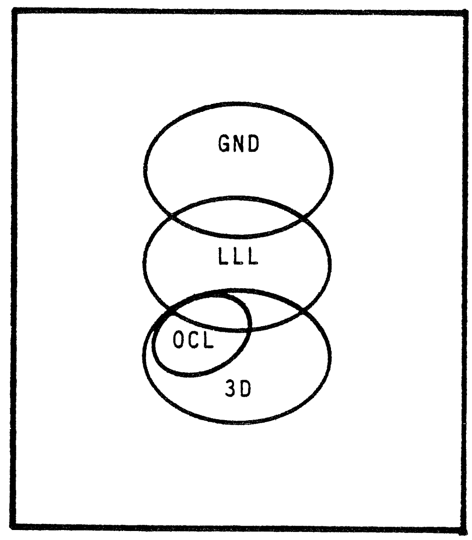


FIGURE 3-7
LEGAL COMBINATIONS OF ILLUMINATION FLAGS

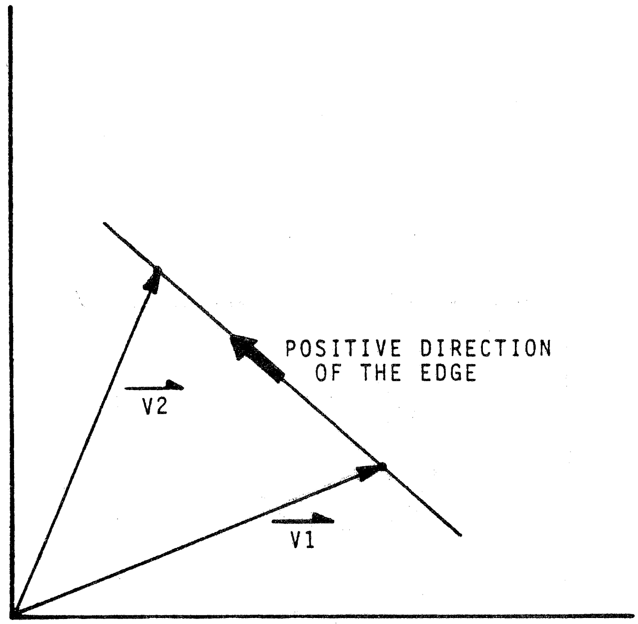


FIGURE 3-8
EDGE DEFINITION

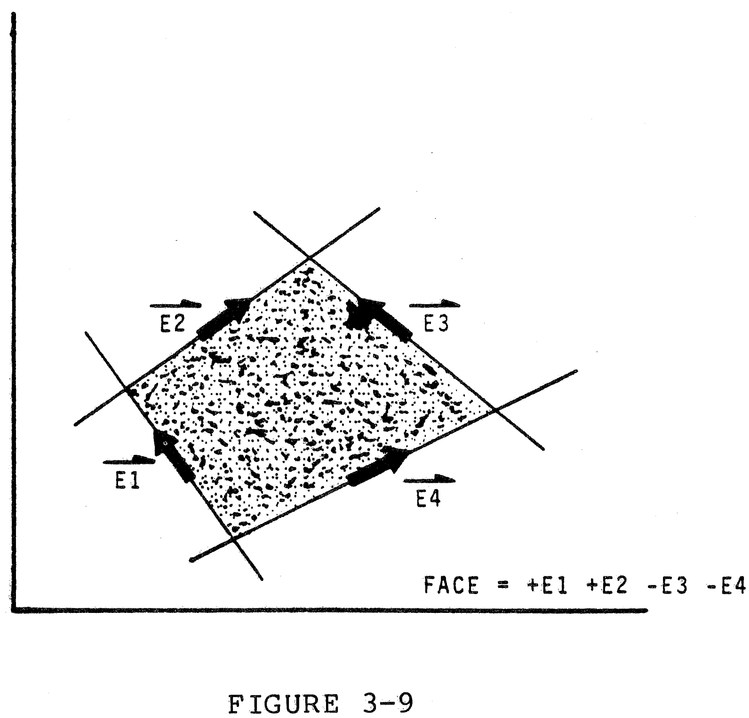


FIGURE 3-9
FACE DEFINITION

Insert Command List Instructions

Command list instructions provide control information to the real-time system about how the data contained in the model is to be processed. The command list in each surface module provides local control for processing the edge and face data in the particular surface module. Each surface module contains an edge and a face command list defined as follows. After specifying the instruction to be inserted by a response of I1 through I16 to the prompt ITEM #=, the system will request which command list, the one for edges or for faces, by the prompt EDG,FAC>. User response to this will be either E for the edge command list or F for the face command list. After selecting the command list, the command to be inserted will be requested by the prompt INST=. User responses to this prompt are the command strings defined in table 3-3. This table also provides a brief description of each command, any subsequent information requested for that command, and the appropriate responses to such subsequent requests.

The following examples are surface module command lists that the user may use in a model.

Example #1 - Normal processing of edges and faces

```
    Edge command list:  EDGE   ENTRY=P1
                        LINK

    Face command list:  FACE   ENTRY=P1
                        LINK
```

Example #2 - Surface module containing mirrored surfaces, center line managed surfaces, and normal surfaces

```
    Edge command list:  TMIR             mirrored surfaces
                        EDGE   ENTRY=P1    processing
                        MMIR
                        TVEC   MTX=T1    center line management
                        TADD   OPT=01      management processing
                        EDGE   ENTRY=P2
                        TVEC   MTX=T1    normal edge
                        EDGE   ENTRY=P3    processing
                        LINK

    Face command list:  FACE   ENTRY=P1
                        LINK
```

COMMAND	ADDITIONAL PROMPT	RESPONSE	FUNCTION
PVEC			Loads the P-vector into the matrix multiplier. The P-vector defines the raster parameters for the display.
SCHN			Concatenates the surface channel matrix for each channel to the P-vector. The channel matrix defines the orientation for each specific channel.
MTX	MTX=	T1 thru T7	Concatenates the transformation matrix specified to the current transformation matrix.
SUN			Loads the sun vector. Needed if using faces with *B=Ø.
MMIR			180° rotation matrix used for mirrored surface management.
TVEC	MTX=	T1 thru T7	Load the specified T-vector.
TADD	OPT=	01,02,03	Adds the specified delta vector to the current T-vector. Currently 01 is for center line management and the other options are undefined.
TMIR			Loads special T-vector for mirrored surfaces and concatenates a 180° rotation matrix.

TABLE 3-3 (PART A)
SURFACE MODULE COMMAND LIST INSTRUCTIONS

COMMAND	ADDITIONAL PROMPT	RESPONSE	FUNCTION
EDGE	ENTRY=	P1 thru P8	Define edges to be processed via specified entry point.
FACE	ENTRY=	P1 thru P8	Define faces to be processed via specified entry point.
LINK			Return to environment module command list.

TABLE 3-3 (PART B)
SURFACE MODULE COMMAND LIST INSTRUCTIONS

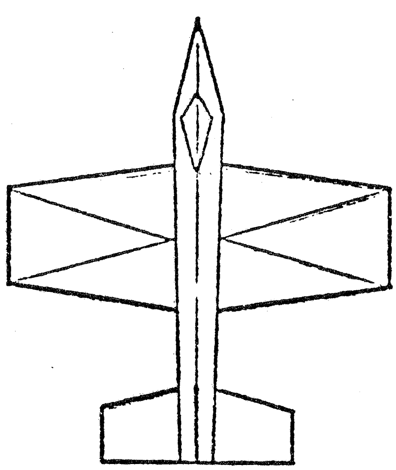
Insert Priority Tree Nodes

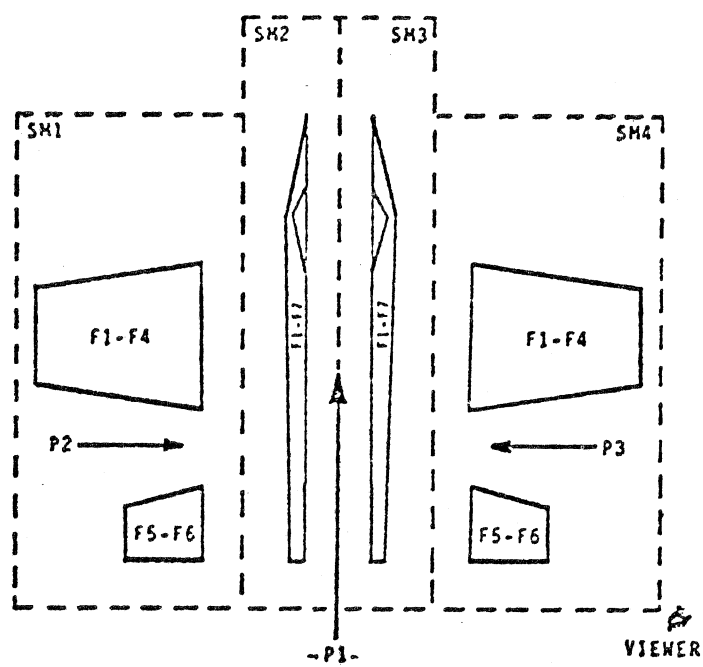
Tree Priority

There exists the option of listing clusters of faces as "leaves" on the end of branches of a binary tree. The order in which the tree is traversed dictates the relative priority of the faces. The traversal may be altered at any non-terminal node of a tree by a "plane test." This plane test checks the viewer position to determine false or true side of a predefined plane to give the branch of the node which highest priority is to follow. The true side of the plane is the side which the normal points. Such a node then has a "true son" and a "false son."

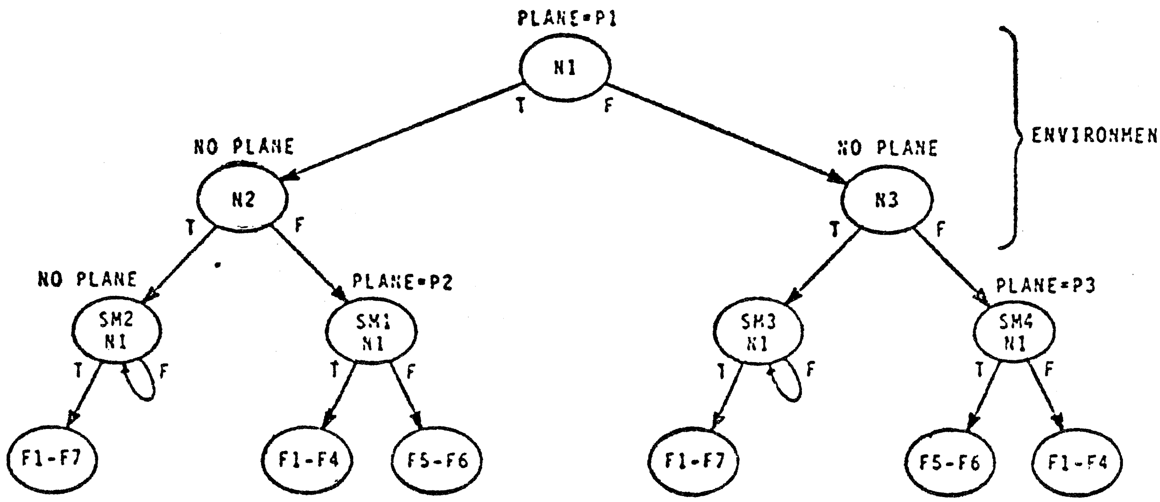
If there is no plane test, then the false son is given priority, meaning that all face clusters on the false son side of the node will have priority over those on the true son side.

The tree often spans environment and surface modules. The following example of an airplane demonstrates the use of these concepts:

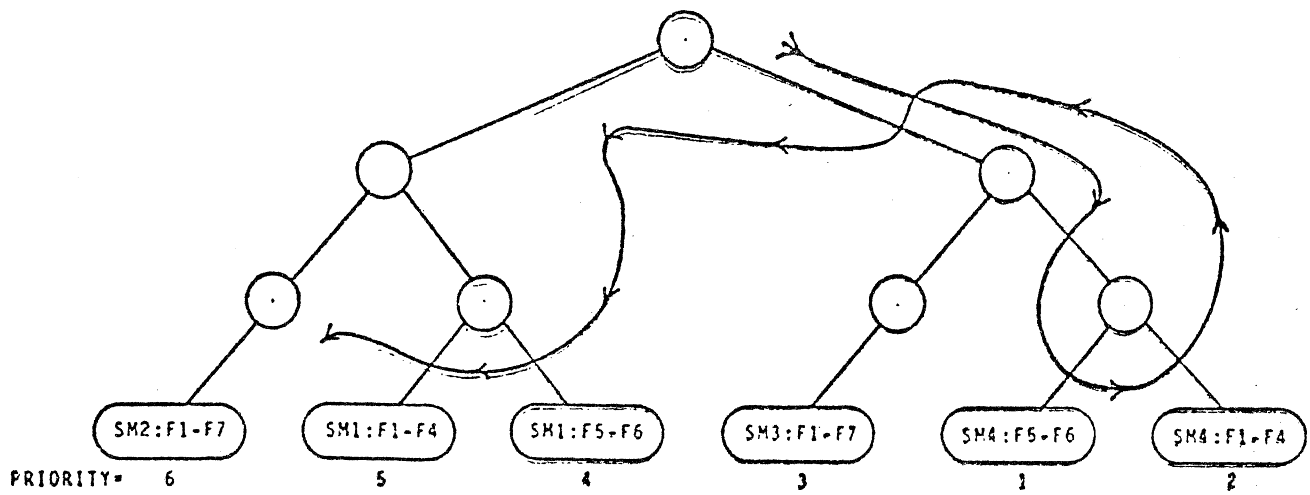




SMi refers to the surface module and Pi to the plane. The following tree would properly serve the model. Nodes N1-N3 are in the environment, while the remaining are in the surface modules.



If the viewer were located as shown above then the tree would be transversed in the following order with the face clusters receiving the given priorities.



A good exercise for the reader would be to change the viewpoint and traverse the tree accordingly.

Definition of priority tree nodes allows the user to modify the default priority order of the surfaces in a surface module. After specifying the priority tree node to be defined, the system will respond with the prompt <u>PLANE TEST></u> to which the user should reply with an N if no separating plane is to be defined for this node or a Y if a separating plane is to be defined. If the user is defining a separating plane and is doing a change on an existing node, the prompt <u>NEW PLANE></u> will be output at this point. The user should respond with a Y to define a new separating plane or an N if the existing one is to be used. When defining a separating plane, the system will request three vertices to define the plane. These vertices should be input in response to the prompts X=, Y=, and Z= which will be repeated three times, once for each vertex.

After all of the appropriate information about any separating plane has been input, the system will request the user to specify the true son and false son branches for this tree node. These inputs will be requested by the prompts TSON= and FSON=. User response to each of these prompts may be either another node number, N1 through N5, or a face number, F1 through F64. If the user specifies a face number, the prompt LAST FACE= will be output to request the last face number in the cluster being defined for this branch of the priority tree.

3.6.1.9 Change Edge, Face, Instruction, and Node (C)

The change function provides the user the ability to modify the definition of an existing edge, face, command list instruction, or priority tree node in the active surface module. After selecting the change function, the prompt ITEM #= will be output. At this point, the user should proceed as if an insert were being done on the specified item. For those parts of the definition that are not to be changed, simply answer the prompt with a tab. When a particular entry is to be changed, enter the new value.

If the user specifies a nonexistent item, the system will output the error message NO ITEM and return to the prompt SUR>> to allow the user to select another function.

3.6.1.10 Kill Edge, Face, Instruction, and Node (K)

The kill function provides the user the ability to delete items from the active surface module. After selecting the kill function, the system will request the user to specify the first and last item of the same type to be deleted in response to the prompts ITEM #1= and ITEM #2=. If the item number for the first item is greater than the second item, nothing is deleted and the command is ignored. Appropriate responses to the prompts are:

E1 thru E64 for edges
F1 thru F64 for faces
I1 thru I16 for command list instruction
N1 thru N5 for priority tree nodes

3.6.1.11 List Edge, Face, Instruction and Node (L)

The list function provides the user the ability to list edge definitions, face definitions, command list instructions and priority tree nodes in the active surface module. The user will be asked to specify the first and last item of the same type to be listed in response to the prompts ITEM #1= and ITEM #2=. Appropriate responses to the prompts are:

E1 thru E64 for edges
F1 thru F64 for faces
I1 thru I16 for command list instructions
N1 thru N5 for priority tree nodes

The following examples show the listing format for each data item.

```
ENTRY=P2

   #E2     *T= 0 CHAN= 0
      START:   X=  -20.0000          Y=  42.0000          Z=  -4.0000
      END:     X=  -11.0000          Y=  42.0000          Z=  -4.0000

   #E3     *T= 0 CHAN= 0
      START:   X=  -11.0000          Y=  42.0000          Z=  -4.0000
      END:     X=  4.0000            Y=  30.0000          Z=  -6.0000

   #E4     *T= 0 CHAN= 0
      START:   X=  -20.0000          Y=  42.0000          Z=  -4.0000
      END:     X=  -18.0000          Y=  30.0000          Z=  -6.0000
```

EDGE
PRINTOUT SAMPLE

```
ENTRY=P1

   #F1      INT=63    ACTUAL INT= 0.9843
      LLL=N GND=Y 3D= N OCC=N *B= 1 *T= 1 CHAN= 0
      EDGES:  -E1    -E1    -E1    -E1

   #F2      INT=15    ACTUAL INT= 0.2343
      LLL=N GND=N 3D= Y OCC=Y *B= 0 *T= 0 CHAN= 0
      EDGES:  E43    E42    -E41   E59

   #F3      INT=20    ACTUAL INT= 0.3125
      LLL=N GND=N 3D= Y OCC=Y *B= 0 *T= 0 CHAN= 0
      EDGES:  E60    -E59   E50    E50
```

FACE
PRINTOUT SAMPLE

```
#I1  FACE:       ENTRY=P1
#I2  LINK:       RETURN

#I1  EDGE:       ENTRY=P1
#I2  MTX :       MTX=T2
#I3  EDGE:       ENTRY=P2
#I4  LINK:       RETURN
```

COMMAND LIST INSTRUCTIONS
PRINTOUT SAMPLE

```
#N1    PLANE TEST >  Y
   0.9960  X+  0.0000  Y+  0.0000  Z = 51.9960
   FTR=  N1
   TSON= F37        LAST FACE=F40
   FSON= N2

#N2    PLANE TEST >  N
   FTR=  N1
   TSON= F32        LAST FACE=F36
   FSON= N3

#N3    PLANE TEST >  N
   FTR=  N2
   TSON= N4
   FSON= F1         LAST FACE=F3
```

PRIORITY TREE NODES
PRINTOUT SAMPLE

3.6.1.12 Write Surface Module (W)

After defining a surface module, it is written on cassette tape or floppy disk by invoking the write function. After selecting the write function, the system will request the output file name by typing:

FILE NAME=

The user reply is either CS1: or CS2: (in foreground mode only) if the module is to be written onto cassette tape or FDØ: or FD1: followed by a file name from one to six characters in length if the module is to be written onto floppy disk. If output is to cassette tape, the user is expected to have rewound the tape to load point if necessary. If multiple output files are to be put onto a cassette tape, they must be written sequentially without rewinding the tape. To add modules to the end of a cassette tape that already contains modules, the user must first read past the existing modules to position the tape before outputting a new module to the tape.

When the output is to be to floppy disk, the file name can be specified without the device name and the system will assume FDØ: as the output device.

3.6.1.13 Blank Screen (B)

The blank screen function erases the screen of the Tektronix 4010 display. If the system does not include the Tektronix 4010 hardware, the command is ignored.

Note: View position must be set first.

3.6.1.14 Display Faces (D)

For systems which contain the Tektronix 4010 display, the display function allows the user to specify those faces in the active surface module that are to be displayed. After the function has been selected, NSPBLD will request the faces to be displayed by typing the messages ITEM #1= and ITEM #2=. To these prompts, the user should respond with the first and last face to be displayed. If the system does not contain a Tektronix 4010 display, this command is ignored.

3.6.1.15 View Position (V)

The set view position function defines the viewing window for displaying faces on the Tektronix 4010 display. To define the viewing window, the user will specify an X and Y ground position in response to the messages X= and Y= and a ground span to be covered by the display in response to the message SPAN=. If faces have previously been displayed from the active module, the screen will be erased and the faces displayed using the new viewing window. For systems without a Tektronix 4010 display, this command is ignored.

3.6.1.16 Find Face (F)

For systems that have a Tektronix 4010 display, the find function allows the user to locate faces that are displayed on the screen. To use the find function, the user should position the crosshairs displayed on the screen over the face in question. Any printing character except an X should then be entered on the console in response to the message POS CROSS>. NSPBLD will then compare the crosshair position to all of the displayed faces in the active surface module and output, in listing format, any faces that surround the crosshair position. This process may be repeated until the character X is entered in response to POS CROSS> and the system will then return to the SUR>> prompt. For systems that do not have a Tektronix 4010 display, this command is ignored.

3.6.1.17 Angle Rotation for Edges and Faces (A)

Selection of the angle rotation function allows the user to rotate all of the defined edges and faces in the active surface modules by a specified angle. The system will prompt the user with the message ANG= to which the user should enter the rotation angle in degrees.

3.6.1.18 Translation for Edges and Face (T)

Selection of the translate function allows the user to translate all of the defined edges and faces in the active surface module. The user should define the translation vector in response to the series of prompts X=, Y=, and Z=.

3.6.1.19 Print Statistics (N)

Statistical information concerning the edges and faces defined in the active surface module will be output when the print statistics function is selected. Information that is printed out is shown in the following example.

```
SUR >> N

EDGES = 64  FACES = 42
SUR >>
```

3.7 ENVIRONMENT MODULE

The environment module file provides all of the information that is required to associate the light and surface modules that define a model. An environment module contains a module list which defines the required light and surface modules, a command list to control processing, a priority tree to control priority order, and several other lists associated with the global definitions of the model.

This section describes the commands provided by NSPBLD to define and manipulate the data contained in an environment module.

3.7.1 EDITING FUNCTIONS

Editing functions provided for defining and manipulating environment module elements are selected by single character codes in response to the prompt ENV>> and are terminated by either a tab or carriage return. Functions that are available and the associated character code are as follows:

E = edit environment module
R = read environment module
Z = zero environment module
X = exit to module selection
D = dynamic coordinate system map
S = sun vector
I = insert modules, runway offset, instruction, node
C = change modules, runway offset, instruction, node
K = kill modules, runway offset, instruction, node
L = list modules, runway offset, instruction node
W = write module

Each of these functions and its relationship to generating the environment module are discussed in the following paragraphs.

3.7.1.1 Edit Environment Module (E)

The edit module function provides the user the ability to declare the current module in memory to be active without destroying the data. This may be used after a program abort or a module write function to gain access to the environment module data. It is assumed by the system that the user is enabling access to valid environment module data.

3.7.1.2 Read Environment Module (R)

Previously generated environment modules are read into the modeling system using the read function. In response to the selection of the read function, the system will prompt the user by typing:

FILE NAME=

The user reply is either CS1: or CS2: (in foreground mode only) if the module is on cassette tape or FDØ: or FD1: followed by the file name from one to six characters in length if the module is on floppy disk. If input is from cassette tape, the user is expected to have rewound the tape if necessary. If a file name is entered and the device name is omitted, FDØ: is assumed. If an active module currently exists, the system will ask for verification of the read command by typing VER> to which the user must respond with either "Y" to complete the read command or "N" to abort it.

3.7.1.3 Zero Environment Module (Z)

The zero function clears all of the data in the environment module. This should be the first command executed when defining a new environment module. If an active module currently exists, the system will ask for verification of the zero command by typing VER> to which the user must respond with either "Y" to complete the zero command or "N" to abort it.

3.7.1.4 Exit (X)

When the user has completed environment module editing the exit function is used to allow selection of the editing mode for another type of module. If an active module still exists when the exit function is selected, the system will respond with the message:

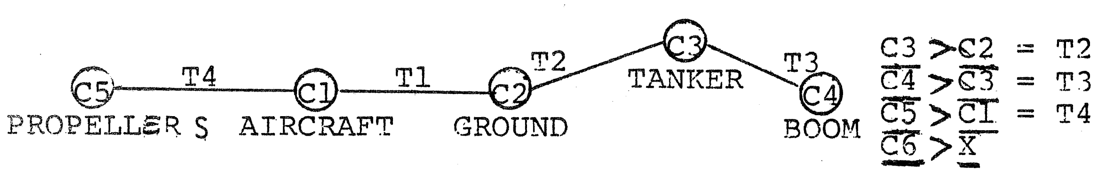
ACT MOD
VER>

to which the user must reply with either a "Y" to exit or a "N" to abort the command.

3.7.1.5 Define Dynamic Coordinate System Map (D)

The dynamic coordinate system map defines the relationship between different coordinate systems and the required transformations. In order to make an object move independently of the viewer aircraft it is necessary to define all lights and surfaces in the object relative to a different co-ordinate system. This co-ordinate system can be moved dynamically with respect to the first by defining a translation vector and rotational matrix in the command list.

Because there may be more than just the predefined co-ordinate systems C1 and C2 (aircraft and ground), it is imperative to map the relation between them. For example, consider the case of a refueling tanker with a movable boom docking with an aircraft with visible moving propellers. The following map must be entered:



The map says that the translation vector must be formed and the rotational matrices concatenated so that

1) the propellers move relative to the aircraft
2) the ground moves relative to the aircraft
3) the tanker moves relative to the ground
and 4) the boom moves relative to the tanker.

These are the relationships between the objects. Since C1 and C2 are predefined with the transformation T1, it does not need to be entered in the dialogue to the right. The character X indicates the end of the map. The underlined characters are those entered by the user. One more example of a map and a dialogue is as follows.

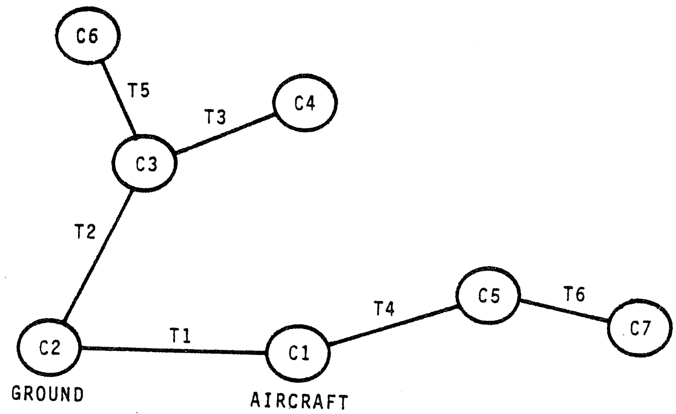


Figure defining coordinate system relationships and transformations

C3>C2 = T2
C4>C3 = T3
C5>C1 = T4
C6>C3 = T5
C7>C5 = T6
C8>X

Dialogue for entering example dynamic coordinate system map

Coordinate systems C1 and C2 are predefined to be the aircraft and ground respectively and associated with transformation T1. This relationship is assumed by the system and does not need to be entered.

After the user has entered the desired coordinate system map, return to the command selection prompt by typing the character X.

3.7.1.6 Define Sun Vector (S)

The define sun vector command provides the user the facility to specify the sun direction vector for sun shading. After selecting this function the user will be prompted to define the sun vector by the messages HDG= and ELEV=. The response to these prompts are the heading angle and elevation angle respectively.

3.7.1.7 Insert Module, Runway Offset, Instruction, and Node (I)

The insert function for environments allows the user to define module lists, runway offsets, command list instructions, and priority tree nodes for the active environment module. After selecting the insert function, the prompt ITEM #= will be output. The user selects the data type and number to be entered by responding with one of the following:

M1 thru M11 to define modules
R1 thru R4 to define runway offsets
I1 thru I64 to define command list instructions
N1 thru N6 to define priority tree nodes

After specifying what data type is to be entered, NSPBLD will request the input that is appropriate for that particular data type entry. If the user is entering several items of the same data type and they are in sequential order, the next item of that type will automatically be selected by typing a tab to the prompt after the first entry has been completed.

Insert Module List

The module list in the environment module provides the association between the environment module, surface modules, and light modules that make up the NSP model. After specifying the module to be defined, the prompt LIT,SUR> will request that the user specify what type of module is being defined. For light modules, the user should reply with an L, for surface modules, the user should reply with an S. After defining the module type, the prompt FILE NAME= requests the file name of the module being referenced. The user should input the appropriate one to six character file name.

For systems that do submodel switching, module list entries M1 through M4 are assumed to reference the four switchable submodels. Because of this implementation, users that do not use submodel switching are urged to define M1 through M4 to be the same surface module to prevent system crashes should a submodel switch be invoked inadvertantly or to maintain model compatability with systems that do (or will) use submodel switching.

Insert Runway Offset

The runway offset information provides location and heading for submodel swapping and center line management. After specifying the runway offset to be defined, the prompts X=, Y=, and HDG= will be output. The user should respond with the center location of the runway and the heading.

Insert Command List Instructions

Command list instructions provide control information to the real-time system about how the data contained in the model is to be processed. The command list in the environment module provides the control required to associate all of the light and surface modules that make up the model. After specifying the instruction to be inserted by a response of I1 through I64 to the prompt ITEM #=, the system will request the command to be inserted by the prompt INST=. User responses to this prompt are the command strings defined in table 3-4. This table also provides a brief description of each command, any subsequent information requested for that command, and the appropriate responses to such subsequent requests. The listing example for command lists in section 3.7.1.10 shows a typical command list for a four option system.

COMMAND	ADDITIONAL PROMPT	RESPONSE	FUNCTION
PVEC			Loads the P-vector into the matrix multiplier. The P-vector defines the raster parameters for the display.
SCHN			Concatenates the surface channel matrix for each channel to the P-vector. The channel matrix defines the orientation for each specific channel.
LCHN			Loads the light channel matrix.
MTX	MTX=	T1 thru T7	Concatenates the trans-mation matrix specified to the current transformation matrix.
SUN			Loads the sun vector. Needed if using faces with *B=Ø.
TVEC	MTX=	T1 thru T7	Load the specified T-vector.
STAR			Include the star data base lights.
DRAW			Loads the raster control information into the hard-ware. This command should follow the references to edge and face data but is not required if no faces are to be processed.

TABLE 3-4 (PART A)
ENVIRONMENT MODULE COMMAND LIST INSTRUCTIONS

COMMAND	ADDITIONAL PROMPT	RESPONSE	FUNCTION
STRT			Starts the display processor running. This command should follow the DRAW command but preceed light string processing.
LINK	MOD= EDG,FAC>	M1 thru M11 E,F	Go process the command list in the referenced surface module. Note that both the edge and face command lists in the surface module must be referenced in the environment module command list.
FOG	MOD= ENTRY=	M1 thru M11 P1 thru P8	Do fog processing on the light strings grouped by the specified entry point in the referenced light module.
LITE	MOD= ENTRY=	M1 thru M11 P1 thru P8	Process the light strings grouped by the specified entry point in the referenced light module.
END			Define the end of the command list.

TABLE 3-4 (PART B)
ENVIRONMENT MODULE COMMAND LIST INSTRUCTIONS

Insert Priority Tree Node

Definition of priority tree nodes allows the user to modify the default priority order for the surfaces that are contained in the model. The priority tree in the environment module references entire surface modules and/or subtrees in particular surface modules. After specifying the priority tree node to be defined, the system will respond with the prompt PLANE TEST> to which the user should reply with an N if no separating plane is to be defined for this node or a Y if a separating plane is to be defined. If the user is defining a separating plane and is doing a change on an existing node, the prompt NEW PLANE> will be output at this point. The user should respond with a Y to define a new separating plane or an N if the existing one is to be used. When defining a separating plane, the system will request three vertices to define the plane. These vertices should be input in response to the prompts X=, Y=, and Z= which will be repeated three times, once for each vertex.

If the user is defining priority tree node N1, which is where all priority tree processing begins, the system will now ask for the initial coordinate system specification by typing the prompt CS=. The user should define the initial coordinate system with a response of C1 through C8.

After all of the appropriate information about any separating plane or initial coordinate system has been input, the system will request the user to specify the true son and false son branches for this tree node. These inputs will be requested by the prompts TSON= and FSON=. User responses to each of these prompts may be either another node number, N1 through N6, or a surface module number, S1 through S4. If the user specifies a surface module number, the prompt NODE= will be output to request the node number in the surface module where priority processing is to continue in the referenced surface module.

For both the true and false son branches of the tree, the user will also be asked to specify the coordinate system that faces are defined in for that branch. The prompt CS= requests the coordinate system to which the user should reply with C1 through C8.

3.7.1.8 Change Module, Runway Offset, Instruction, and Node (C)

The change function provides the user the ability to modify the definition of an existing module, runway offset, command list instruction, or priority tree node in the active environment module. After selecting the change function, the prompt ITEM #= will be output. At this point, the user should proceed as if an insert were being done on the specified item. For those parts of the definition that are not to be changed, simply answer the prompt with a tab. When a particular entry is to be changed, enter the new value.

3.7.1.9 Kill Module, Runway Offset, Instruction, and Node (K)

The kill function provides the user the ability to delete items from the active environment module. After selecting the kill function, the system will request the user to specify the first and last item of the same type to be deleted in response to the prompts <u>ITEM #1=</u> and <u>ITEM #2=</u>. If the item number for the first item is greater than the second item, nothing is deleted and the command is ignored. Appropriate responses to the prompts are:

M1 thru M11 for module names
R1 thru R4 for runway offsets
I1 thur I64 for command list instructions
N1 thru N6 for priority tree nodes

3.7.1.10 List Module, Runway Offset, Instruction, and Node (L)

The list function provides the user the ability to list module names, runway offsets, command list instructions, and priority tree nodes in the active environment module. The user will be asked to specify the first and last item of the same type to be listed in response to the prompts <u>ITEM #1=</u> and <u>ITEM #2=</u>. Appropriate responses to the prompts are:

M1 thru M11 for module names
R1 thru R4 for runway offsets
I1 thru I64 for command list instructions
N1 thru N6 for priority tree nodes

The following examples show the listing format for each data item.

```
#M1   TYPE= S   FILE NAME=FDO:HKGS1R
#M2   TYPE= S   FILE NAME=FDO:HKGS1R
#M3   TYPE= S   FILE NAME=FDO:HKGS1R
#M4   TYPE= S   FILE NAME=FDO:HKGS4Z
#M5   TYPE= S   FILE NAME=FDO:HKGS5T
#M6   TYPE= S   FILE NAME=FDO:HKGS6Z
#M7   TYPE= S   FILE NAME=FDO:HKGS3Z
#M8   TYPE= L   FILE NAME=FDO:HKGL1A
#M9   TYPE= L   FILE NAME=FDO:HKGL2A
#M10  TYPE= L   FILE NAME=FDO:HKGL3A
#M11  TYPE= L   FILE NAME=FDO:HKGL4A
```

MODULE LIST
PRINTOUT SAMPLE

```
#R1   X=  0.0000       Y=  0.0000       HDG=0.0000
#R2   X=  2425.0000    Y=  6250.0000    HDG=355.4960
#R3   X=  7120.0000    Y=  -1125.0000   HDG=124.4960
#R4   X=  0.0000       Y=  0.0000       HDG=0.0000
```

RUNWAY OFFSET
PRINTOUT SAMPLE

```
#I1  TVEC:       MTX=T1
#I2  SUN :
#I3  LINK:       MOD=M1   TYPE= F
#I4  LINK:       MOD=M5   TYPE= F
#I5  LINK:       MOD=M6   TYPE= F
#I6  LINK:       MOD=M7   TYPE= F
#I7  PVEC:
#I8  SCHN:
#I9  MTX :       MTX=T1
#I10 LINK:       MOD=M1   TYPE= E
#I11 LINK:       MOD=M5   TYPE= E
#I12 LINK:       MOD=M6   TYPE= E
#I13 LINK:       MOD=M7   TYPE= E
#I14 DRAW:
#I15 STRT:
#I16 LCHN:
#I17 MTX :       MTX=T1
#I18 FOG :       MOD=M8   ENTRY=P1
#I19 LITE:       MOD=M8   ENTRY=P1
#I20 FOG :       MOD=M9   ENTRY=P1
#I21 LITE:       MOD=M9   ENTRY=P1
#I22 FOG :       MOD=M10  ENTRY=P1
#I23 LITE:       MOD=M10  ENTRY=P1
#I24 FOG :       MOD=M11  ENTRY=P1
#I25 LITE:       MOD=M11  ENTRY=P1
#I26 STAR:
#I27 END :
```

COMMAND LIST
PRINTOUT SAMPLE

```
#N1     PLANE TEST >  Y
   0.9960  X+  0.0000  Y+  0.0000  Z = 51.9960
   FTR=  N1
   TSON= F37        LAST FACE=F40
   FSON= N2

#N2     PLANE TEST >  N
   FTR=  N1
   TSON= F32        LAST FACE=F36
   FSON= N3

#N3     PLANE TEST >  N
   FTR=  N2
   TSON= N4
   FSON= F1         LAST FACE=F3

#N4     PLANE TEST >  Y
   0.0000  X+  -1.0000 Y+  0.0000  Z = -30.0000
   FTR=  N3
   TSON= F18        LAST FACE=F31
   FSON= F4         LAST FACE=F17
```

PRIORITY NODES
PRINTOUT SAMPLE

3.7.1.11 Write Environment Module (W)

After defining an environment module, it is written on cassette tape or floppy disk by invoking the write function. After selecting the write function, the system will request the output file name by typing:

FILE NAME=

The user reply is either CS1: or CS2: (in foreground mode only) if the module is to be written onto cassette tape or FDØ: or FD1: followed by a file name from one to six characters in length if the module is to be written onto floppy disk. If output is to cassette tape, the user is expected to have rewound the tape to load point if necessary. If multiple output files are to be put onto cassette tape, they must be written sequentially without rewinding the tape. To add modules to the end of a cassette tape that already contains modules, the user must first read past the existing modules to position the tape before outputting a new module to the tape.

When the output is to be to floppy disk, the file name can be specified without the device name and the system will assume FDØ: as the output device.

3.8 GROUP MODULE

The group module file provides all of the information that will be required by the real-time system, NSPFLY, to associate model numbers received from the host computer with the actual files on the disk that make up a particular model. In addition, information required for data base management is defined for each model. When the real-time system, NSPFLY, is started, it requests the name of the group module that defines what models are available for use. Using the information extracted from the group module, NSPFLY checks for the existence of all the specified files and sets up all of the appropriate internal tables required for operation.

This section describes the commands provided by NSPBLD to define and manipulate the data contained in a group module.

3.8.1 EDITING FUNCTIONS

Editing functions provided for defining and manipulating group module elements are selected by single character codes in response to the prompt GRP>> and are terminated by either a tab or carriage return. Functions that are available and the associated character code are as follows:

E = edit group module
R = read group module
Z = zero group module
X = exit group to module selection
I = insert group definition
C = change group definition
K = kill group definition
L = list group definition
W = write group module

Each of these functions and its relationship to generating the group module are discussed in the following paragraphs.

3.8.1.1 Edit Group Module (E)

The edit module function provides the user the ability to declare the current module in memory to be active without destroying the data. This may be used after a program abort or a module write function to gain access to the group module data. It is assumed by the system that the user is enabling access to valid group module data.

3.8.1.2 Read Group Module (R)

Previously generated group modules are read into the modeling system using the read function. In response to the selection of the read function, the system will prompt the user by typing:

FILE NAME=

The user reply is either CS1: or CS2: (in foreground mode only) if the module is on cassette tape or FDØ: or FD1: followed by the file name from one to six characters in length if the module is on floppy disk. If input is from cassette tape, the user is expected to have rewound the tape if necessary. If a file name is entered and the device name is omitted, FDØ: is assumed. If an active module currently exists, the system will ask for verification of the read command by typing VER> to which the user must respond with either "Y" to complete the read command or "N" to abort it.

3.8.1.3 Zero Group Module (Z)

The zero function clears all of the data in the group module. This should be the first command executed when defining a new environment module. If an active module currently exists, the system will ask for verification of the zero command by typing VER> to which the user must respond with either "Y" to complete the zero command or "N" to abort it.

3.8.1.4 Exit (X)

When the user has completed group module editing the exit function is used to allow selection of the editing modes for another type of module. If an active module still exists when the exit function is selected, the system will respond with the message:

ACT MOD
VER>

to which the user must reply with either a "Y" to exit or a "N" to abort the command.

3.8.1.5 Insert Group Definition (I)

After selecting the insert function, the prompt GRP #= will be output to request what group or model number is to be defined. The user should specify a group number with a response of GØ through G127. The system will next request the module number via the prompt MODULE #= to which the user should reply with M1 through M32. Module numbers simply associate an order with the environment modules to be specified as belonging to this group. This order is used in considering various environment choices during data base management. For systems that do not have data base management, M1 is used regardless of the number of environment modules defined under a particular group. After a module number has been selected, the prompt FILE NAME= requests the environment module name to be referenced. The reply must be a one to six character file name and is assumed to be on FDØ: during NSPFLY operation. When the environment module has been specified, the parameters associated with data base management will be requested. The prompts X=, Y=, DX=, and DY= will be output to which the user should reply with the center and size of the management square for this particular environment module.

3.8.1.6 Change Group Definition (C)

The change function provides the user the ability to modify the definition of an existing group in the active group module. After selecting the change function the prompt GRP #= will be output. At this point, the user should proceed as if an insert were being done. For no change, answer the prompt with a tab. When a particular entry is to be changed, enter the new value.

3.8.1.7 Kill Group Definition (K)

The kill function provides the user the facility to delete items from the active group module. After selecting the kill function, the system will request the group number by typing GRP #=. After entering the group number, the user will be asked to specify the first and last module in the group to be deleted in response to the prompts MOD #1= and MOD #2=. The entire group can be deleted by deleting all of the modules in the group.

3.8.1.8 List Group Definition (L)

The list function provides the user the facility to list group definitions in the active group module. The user will be asked to specify the first and last group definition to be listed in response to the prompts GRP #1= and GRP #2=. The user response should be the group numbers, GØ through G127, that are to be listed. The following example shows the listing format for group definitions.

```
#G0      MOD CNT=1
   # 1      FILE NAME=FD0:BRUENV
   ORIGIN:  X=   0.0000             Y=   0.0000
   DELTA:   X=   99968.0000         Y=   99968.0000

#G1      MOD CNT=1
   # 1      FILE NAME=FD0:FTNENV
   ORIGIN:  X=   0.0000             Y=   0.0000
   DELTA:   X=   99968.0000         Y=   99968.0000
```

3.8.1.9 Write Group Module (W)

After defining a group module, it is written on cassette tape or floppy disk by invoking the write function. After selecting the write function, the system will request the output file name by typing:

FILE NAME=

The user reply is either CS1: or CS2: (in foreground mode only) if the module is to be written onto cassette tape or FDØ: or FD1: followed by a file name from one to six characters in length if the module is to be written onto floppy disk. If output is to cassette tape, the user is expected to have rewound the tape to load point if necessary. If multiple output files are to be put onto a cassette tape, they must be written sequentially without rewinding the tape. To add modules to the end of a cassette tape that already contains modules, the user must first read past the existing modules to position the tape before outputting a new module to the tape.

When the output is to be to floppy disk, the file name can be specified with the device name and the system will assume FDØ: as the output device.

SECTION 4

MODEL MERGE PROGRAM NSPMER

4.1 INTRODUCTION

The model merge program, NSPMER, provides the user the facility to combine two light or surface modules into a single module. Merging of two modules results in the data in the second module specified being concatenated to the data in the first module. All of the control information such as entry points, command lists, and priority tree nodes are cleared and must be defined using NSPBLD after the merge is complete.

4.2 EXECUTION

Program loading and execution is accomplished in the following manner: where 4=Control, 5=Graphics, 6=Listings, F=Cursor Find

```
//ASSIGN,4,KEY.     OR //ASSIGN,4,TEK
//ASSIGN,6,KEY.     OR //ASSIGN,6,TEK
//ASSIGN,F,KEY.     OR //ASSIGN,F,TEK
//EXECUTE,FDØ,NSPMER.
```

Note: If TEK is selected, cursors are not available for use.

The program will be loaded and start execution. The program name and version number will be output as follows:

NSPMER VØ2-Ø1
NOVO-SP MODEL MERGE

This version number should be referenced when reporting any error in the modeling system.

The first prompt output to the user after the name and version number is MER>>. User response to this prompt should be either M to merge two modules or X to return to the monitor. After specifying the merge command, the prompts 1-ST FILE NAME= and 2-ND FILE NAME= will be output to request the two modules to be merged. The user reply to each prompt is either CS1: or CS2: (in foreground mode only) if the module is on cassette tape or FDØ: or FD1: followed by a file name from one to six characters in length if the module is on floppy disk. If input is from cassette tape, the user is expected to have positioned or rewound the tape if necessary. If a file name is entered and the device name is omitted, then FDØ: is assumed.

After specifying the input files to be merged, NSPMER will read the files and attempt to merge the information. If this is successful, the prompt OUTPUT FILE NAME= will be typed to request the name of the output file for the merged data. Specification of this file name is the same as for the input file. After the merged data has been output, the program returns to the prompt MER>>.

Entry pointers must be set in output file.

Error messages that are output by NSPMER are described in appendix B.

APPENDIX A

MODELING SYSTEM PROMPTS

A.1 INTRODUCTION

This appendix provides a summary of all prompts used for light modules, surface modules, environment modules, and group modules in NSPBLD and model merging in NSPMER. Each table is divided into columns for the prompt output by the system, the data type that is accepted for that prompt, the range of the response (minimum and maximum value), the units of the response, and any appropriate notes.

PROMPT	DATA TYPE	RANGE		UNITS	NOTES
		MIN	MAX		
LIT>>	ALPHABET				R, Z, X, E, P, I, C, K, L, S, W, B, D, V, F, A, T, N
FILE NAME=	FILE SPECIFICATIONS				CS0:, CS1:, FD0:X, FD1:X, Where X represents a one to six character file name.
VER>	YES/NO				
ENTRY=	NAME	P1	P8		
STR #= STR #1= STR #2=	NAME	R1 A1 O1 W1 G1	R110 A110 O110 W110 G110		
FLS, ROT, VASI, SFL, NML	ALPHABET				F, R, V, S, N, X
ONT=	INTEGER	0		FRAMES	
PER=	INTEGER	ON TIME VALUE	32,767	FRAMES	Flashing lights
			32,767		Strobe lights
PHA=	INTEGER	0		FRAMES	
OFT=	INTEGER	0	255	FRAMES	
OP ANG=	REAL	0	360	DEGREES	
HDG=	REAL	0	360	DEGREES	
#MASTERS=	INTEGER	2	3		
#REPEATS=	INTEGER	0	7		
ELEV=	REAL	0	360	DEGREES	
#LTS=	INTEGER	1	255		
INT=	REAL	0	1.9967		
SAW=	INTEGER	0 1 10 20 30	 13 23 33		Omnidirectional Curved Directional
SWT#=	INTEGER	0	32		
FLG>	YES/NO				
FFB= *B= *T= *SP= PRI= IO=	ZERO/ONE				
CHAN=	INTEGER	0	8		
X= Y= Z=	REAL	-2^{21}	$2^{21}-1$	FEET	
DELTA=	REAL	0	5NMI	FEET	
ANG=	REAL	0	360	DEGREES	
FL/SP>	ALPHABET				F, S
SPAN=	REAL	0	$2^{21}-1$	FEET	
POS CROSS>	ALPHABET				Any printing character

TABLE A-1
LIGHT MODULE ENTRY PARAMETERS

PROMPT	DATA TYPE	RANGE MIN	RANGE MAX	UNITS	NOTES
SUR>>	ALPHABET				R, Z, X, E, P, M, S, I, C, K, W, B, D, V, F, N, A, T
FILE NAME=	FILE SPECIFICATIONS				CS0;, CS1:, FD0:X, FD1:X, Where X represents a one to six character file name.
VER>	YES/NO				
ENTRY=	NAME	P1	P8		
ITEM #= ITEM #1= ITEM #2=	NAME	E1 F1 I1 N1	E64 F64 I16 N5		
MS=	ZERO/ONE				
X= Y= Z=	REAL	2^{-21}	$2^{21}-1$	FEET	
MODEL STRIPES=	INTEGER	0	15		
REMAINING STRIPES=	INTEGER	0	255		
FLG>	YES/NO				
*B= *T=	ZERO/ONE				
CHAN=	INTEGER	0	8		
INT=	INTEGER	0	63		
LLL= GND= 3D= OCC=	YES/NO				
E1= E2= E3= E4=	NAME	E1	E64		Edge names may be preceeded with a minus sign.
PLANE TEST>	YES/NO				
NEW PLANE>	YES/NO				
TSON= FSON=	NAME	N1 F1	N5 F64		
LAST FACE=	NAME	F1	F64		
EDG,FAC>	ALPHABET				E, F
INST=	TEXT				PVEC, SCHN, MTX, SUN, MMIR, TVEC, TAAD, TMIR, EDGE, FACE, LINK
MTX=	NAME	T1	T7		
OPT=	NAME	O1	O3		
SPAN=	REAL	0	$2^{21}-1$	FEET	
POS CROSS>	ALPHABET				Any printing character.

TABLE A-2
SURFACE MODULE ENTRY PARAMETERS

PROMPT	DATA TYPE	RANGE		UNITS	NOTES
		MIN	MAX		
ENV>>	ALPHABET				E, R, Z, X, D, S, I, C, K, L, W
FILE NAME=	FILE SPECIFICATIONS				CS0:, CS1:, FD0:X, FD1:X, Where X represents a one to six character file name.
VER>	YES/NO				
HDG= ELEV=	REAL	0	360	DEGREES	
ITEM #= ITEM #1= ITEM #2=	NAME	M1 R1 I1 N1	M11 R4 I64 N6		
LIT,SUR>	ALPHABET				L, S
X= Y=	REAL	-2^{21}	$2^{21}-1$	FEET	
INST=	TEXT				PVEC, SCHN, MTX, LCHN, SUN, TVEL, TADD, STAR, DRAW, STRT, END, LITE, LINK, FOG
MTX=	NAME	T1	T7		
OPT=	NAME	O1	O3		
MOD=	NAME	M1	M11		
ENTRY=	NAME	P1	P8		
EDG,FAC>	ALPHABET				E, F
PLANE TEST>	YES/NO				
CS=	NAME	C1	C8		
TSON= FSON=	NAME	N1 S1	N6 S4		
NODE=	NAME	N1	N5		

TABLE A-3
ENVIRONMENT MODULE ENTRY PARAMETERS

PROMPT	DATA TYPE	RANGE		UNITS	NOTES
		MIN	MAX		
GRP>>	ALPHABET				E, R, X, Z, W, I, K, L, C
FILE NAME=	FILE SPECIFICATIONS				CS0:, CS1:, FD0:X, FD1:X, Where X represents a one to six character file name.
VER>	YES/NO				
GRP #= GRP #1= GRP #2=	NAME	G0	G127		
MODULE #= MODULE #1= MODULE #2=	NAME	M1	M32		
X= Y= DX= DY=	REAL	-2^{21}	$2^{21}-1$	FEET	

TABLE A-4
GROUP MODULE ENTRY PARAMETERS

PROMPT	DATA TYPE	RANGE		UNITS	NOTES
		MIN	MAX		
MER>>	ALPHABET				M, X
1-ST FILE NAME= 2-ND FILE NAME= OUTPUT FILE NAME=					CS0:, CS1:, FD0:X, FD1:X, Where X represents a one to six character file name.
RETRY>	YES/NO				

TABLE A-5
MERGE ENTRY PARAMETERS

APPENDIX B

ERROR MESSAGES

B.1 INTRODUCTION

This appendix lists the error messages and interpretation for the NSPBLD and NSPMER programs.

General Error Messages - NSPBLD

SORRY	The user has requested a function that the modeling system software cannot provide
?	The response is not an appropriate answer. Reenter the requested data.
BAD NAME	The name specified by the user is not legal in the context it has been used. Enter a new name.
NO MOD	Editing functions are being issued but there is no active module. Issue a zero or read command.
BAD MOD TYPE	The module type read by the modeling system is not correct. Read the module under the appropriate set of editing functions.
NO ITEM	An item was specified that does not exist. Specify a new item name.
NO ROOM	No more room remains in the active module to store information of the type being specified.

JEOPARDY:	This message is followed by a list of face names whose definitions are not necessarily correct because one or more of the edges referenced have been deleted.

General Error Messages - NSPMER

TOO MANY FLASHING LIGHTS	The number of flashing light strings contained in the two modules being merged exceeds 6.
TOO MANY STROBES	The number of strobe light strings contained in the two modules being merged exceeds 6.
TOO MANY ROTATING LIGHTS	The number of rotating light strings contained in the two modules being merged exceeds 4.
TOO MANY VASIS	The number of VASI light strings contained in the two modules being merged exceeds 2.
TOO MANY STRINGS	The total number of light strings contained in the two modules being merged exceeds 110.
TOO MANY FACES	The number of faces contained in the two modules being merged exceeds 64.
TOO MANY EDGES	The number of edges contained in the two modules being merged exceeds 64.
TOO MANY SYMM. SURFACES	Symmetrical (mirrored) surfaces defined in both modules being merged.
TOO MANY RUNWAYS	Center line stripe management is defined in both modules being merged.

I/O Error Messages - NSPBLD and NSPMER

DUP FILE VER>	The file name specified on a write command already exists on the diskette. If the user responds with a Y, the file is overwritten; if the user responds with an N, the command is aborted.
CKS ERR	A checksum error was detected during an I/O operation.
I/O ERR u	An error was detected during an I/O operation. The error code values are shown below.

ERROR NUMBER	MEANING
1	file not opened
2	file not defined
3	end of file
4	no memory available
5	hardware error
6	disk full
7	illegal opcode
8	directory full
9	duplicate file name
10	bad logical unit
11	buffer outside user area
12	bad device name
13	bad type of extent
14	logical device tables full
15	no disk or door open
16	disk select error

APPENDIX C

MODELING SYSTEM FORMS

C.1 INTRODUCTION

This appendix provides samples of the modeling forms to be used during development of the various types of modules which comprise an NSP model.

SMK-95 SURFACE MODULE *(Edge)*							FILE NAME TRASIR						
EDGE	FLAG	T	CHAN	X	Y	Z	EDGE	FLAG	T	CHAN	X	Y	Z
E1	N			0	0	0							
				0	12000	0							
E64	Y	1	0	10000	-150000	0							
				10000	150000	0							

SAMPLE

PAGE _____ OF _____

SMK-95 SURFACE MODULE *(EDGE and FACE Instruction)*				FILE NAME TRASIR			
EDGE				FACE			
I-1	E	INST EDGE	ENTRY = PI	I-1	F	INST FACE	ENTRY = PI
I-2	E	INST LINK		I-2	F	INST LINK	
I-3	E	INST		I-3	F	INST	
I-4	E	INST		I-4	F	INST	
I-5	E	INST		I-5	F	INST	
I-6	E	INST		I-6	F	INST	
I-7	E	INST		I-7	F	INST	
I-8	E	INST		I-8	F	INST	
I-9	E	INST		I-9	F	INST	
I-10	E	INST		I-10	F	INST	
I-11	E	INST		I-11	F	INST	
I-12	E	INST		I-12	F	INST	
I-13	E	INST		I-13	F	INST	
I-14	E	INST		I-14	F	INST	
I-15	E	INST		I-15	F	INST	
I-16	E	INST		I-16	F	INST	

REMARKS

SAMPLE

PAGE ____ OF ____

SMK-95 SURFACE MODULE *(Command List)*		FILE NAME TRAS IR
MIRROR SURFACE		
MS= 1	X= 5500	Y= 0
STRIPES		
MODELED STRIPES= 6	STRIPES REMAING= 34	
ENTRY POINTERS		
ENTRY=P1 *(FACE)*	ITEM 1 F1	ITEM 2 F64
ENTRY=P1 *(EDGE)*	ITEM 1 E1	ITEM 2 E10
ENTRY=P2 *(EDGE)*	ITEM 1 E11	ITEM 2 E22
ENTRY=P3 *(EDGE)*	ITEM 1 E23	ITEM 2 E64

NODES

NODE	PLANE TEST	X	Y	Z
NODE N1: N1	PLANE TEST: N	X	Y	Z
		X	Y	Z
		X	Y	Z
	TSON: F1	LAST FACE: F5	FSON: N2	LAST FACE
NODE N2: N2	PLANE TEST: Y	X 0	Y 0	Z 0
		X 5000	Y 0	Z 500
		X 10 000	Y 0	Z 0
	TSON: F6	LAST FACE: F64	FSON: N2	LAST FACE
NODE N3	PLANE TEST	X	Y	Z
		X	Y	Z
		X	Y	Z
	TSON	LAST FACE	FSON	LAST FACE
NODE N4	PLANE TEST	X	Y	Z
		X	Y	Z
		X	Y	Z
	TSON	LAST FACE	FSON	LAST FACE
NODE N5	PLANE TEST	X	Y	Z
		X	Y	Z
		X	Y	Z
	TSON	LAST FACE	FSON	LAST FACE

SAMPLE

PAGE ____ OF ____

SMK-95 ENVIRONMENT MODULE *(Instruction)*		FILE NAME TRAEVN	
I 1	INST PVEC		
I 2	INST SCHN		
I 3	INST LCHN		
I 4	INST MTX	T1	
I 5	INST SUN		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		
I	INST		

SAMPLE

PAGE_____OF_____

SMK-95 ENVIRONMENT MODULE *(Priority Tree Node)*

FILE NAME: TRAENV

SAMPLE

NODE	PLANE TEST				
NODE N1: N1	PLANE TEST	X	Y	Z	
		X	Y	Z	
		X	Y	Z	
	CS= C2	TSON= S2	NODE= N1	CS= C2	
	FSON= S1		NODE= N2	CS= C2	
NODE N2	PLANE TEST	X	Y	Z	
		X	Y	Z	
		X	Y	Z	
	TSON=		NODE=	CS=	
	FSON=		NODE=	CS=	
NODE N3	PLANE TEST	X	Y	Z	
		X	Y	Z	
		X	Y	Z	
	TSON=		NODE=	CS=	
	FSON=		NODE=	CS=	
NODE N4	PLANE TEST	X	Y	Z	
		X	Y	Z	
		X	Y	Z	
	TSON=		NODE=	CS=	
	FSON=		NODE=	CS=	
NODE N5	PLANE TEST	X	Y	Z	
		X	Y	Z	
		X	Y	Z	
	TSON=		NODE=	CS=	
	FSON=		NODE=	CS=	
NODE N6	PLANE TEST	X	Y	Z	
		X	Y	Z	
		X	Y	Z	
	TSON=		NODE=	CS=	
	FSON=		NODE=	CS=	

PAGE_____ OF________

SMK-95 ENVIRONMENT MODULE		FILE NAME
MODULES		
MODULE NO	LIT, SUR	FILE NAME
M1	S	TRAS1R
M2	S	TRAS2R
M3	S	TRAS3Z
M4	S	TRAS4Z
M5	S	TRAS5T
M6	L	TRAL1A
M7	L	TRAL2A
M8	L	TRAL3A
M9	L	TRAL4A
M10		
M11		

RUNWAY			
RUNWAY NO	X	Y	HDG
R1	0	0	0
R2	0	-2000	0
R3			
R4			

DYNAMIC COORDINATE MAP		
C3	X	= T2
C4		= T3
C5		= T4
C6		= T5
C7		= T6
C8		

SAMPLE

SUN		
SUN	HDG = 210	ELEV = 350

PAGE ____ OF ______

SMK-95 LIGHT MODULE										FILE NAME TRALIA		
ENTRY = P1				STR #1 = R1				STR #2 = G110				
LIT/TYPE	STR #	# LTS 10	INT .7	SAW 0	ANG	SWT # 0				X 0	Y 0	Z 0
N	R1	ONT/OFT	PER	PHA	UP A	HDG	STR #2	# MAS	# RPTS	HEADING	ELEV	DELTA
		FLAG Y	FFB 1	CHAN 0	B 1	T 0	SP 0	PRI	IO	X 0	Y 90	Z 0
LIT/TYPE	STR #	# LTS 5	INT 1	SAW 0	ANG	SWT # 0				X 0	Y -5000	Z 0
F	A1	ONT/OFT 12	PER 15	PHA 30	UP A	HDG	STR #2	# MAS	# RPTS	HEADING	ELEV	DELTA
		FLAG N	FFB	CHAN	B	T	SP	PRI	IO	X 0	Y -5000	Z 0
LIT/TYPE	STR #	# LTS	INT	SAW	ANG 45	SWT #				X	Y	Z
A		ONT/OFT	PER	PHA	UP A	HDG	STR #2	# MAS	# RPTS	HEADING	ELEV	DELTA
		FLAG	FFB	CHAN	B	T	SP	PRI	IO	X	Y	Z
LIT/TYPE	STR #	# LTS	INT	SAW	ANG	SWT #				X 5000	Y -3000	Z 0
T		ONT/OFT	PER	PHA	UP A	HDG	STR #2	# MAS	# RPTS	HEADING	ELEV	DELTA
		FLAG	FFB	CHAN	B	T	SP	PRI	IO	X	Y	Z
LIT/TYPE	STR #	# LTS 8	INT	SAW	ANG	SWT #				X	Y	Z
C	R1	ONT/OFT	PER	PHA	UP A	HDG	STR #2	# MAS	# RPTS	HEADING	ELEV	DELTA
		FLAG	FFB	CHAN	B	T	SP	PRI	IO	X	Y	Z
LIT/TYPE	STR #	# LTS	INT	SAW	ANG	SWT #				X	Y	Z
		ONT/OFT	PER	PHA	UP A	HDG	STR #2	# MAS	# RPTS	HEADING	ELEV	DELTA
		FLAG	FFB	CHAN	B	T	SP	PRI	IO	X	Y	Z
LIT/TYPE	STR #	# LTS	INT	SAW	ANG	SWT #				X	Y	Z
		ONT/OFT	PER	PHA	UP A	HDG	STR #2	# MAS	# RPTS	HEADING	ELEV	DELTA
		FLAG	FFB	CHAN	B	T	SP	PRI	IO	X	Y	Z

SAMPLE

PAGE ____ OF ______