NASA CONTRACTOR REPORT

NASA CR-149924

(Supersedes CR-144042)

(NASA-CR-149924) MASK ANALYSIS PROGRAM (MAP) REFERENCE MANUAL (M&S Computing, Inc.) 165 p HC $6.75 CSCL 09B

N76-28859

Unclas

G3/61 44611

MASK ANALYSIS PROGRAM (MAP) REFERENCE MANUAL

By C. L. Mitchell

M&S Computing, Inc.
Huntsville, Alabama 35805

June 23, 1976

Prepared for:

NASA - GEORGE C. MARSHALL SPACE FLIGHT CENTER
Marshall Space Flight Center, Alabama 35812

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO. NASA CR-149924*	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
4. TITLE AND SUBTITLE MASK ANALYSIS PROGRAM (MAP) REFERENCE MANUAL		5. REPORT DATE June 23, 1976 6. PERFORMING ORGANIZATION CODE EC45
7. AUTHOR(S) C. L. Mitchell		8. PERFORMING ORGANIZATION REPORT # 75-0039 , Revision 1
9. PERFORMING ORGANIZATION NAME AND ADDRESS M&S Computing, Inc. P. O. Box 5183 Huntsville, Alabama 35812		10. WORK UNIT NO. 11. CONTRACT OR GRANT NO. NAS8-25621
12. SPONSORING AGENCY NAME AND ADDRESS National Aeronautics and Space Administration George C. Marshall Space Flight Center Huntsville, Alabama 35812		13. TYPE OF REPORT & PERIOD COVERED Contractor Report 14. SPONSORING AGENCY CODE
15. SUPPLEMENTARY NOTES Electronics and Control Laboratory, Electronics Development Division Design Techniques Branch		

16. ABSTRACT

This document is intended to serve as a User's Manual and a Programmer's Manual for the Mask Analysis Program. The first portion of the document is devoted to the user. It contains all of the information required to execute MAP. The remainder of the document describes the details of MAP software logic. Although the information in this portion is not required to run the program, it is recommended that every user review it to gain an appreciation for the program functions.

*This report supersedes NASA CR-144042, dated October 21, 1975.

17. KEY WORDS	18. DISTRIBUTION STATEMENT Unclassified-Unlimited COR: John K. Gould EC01 J. Brooks Moore Director, E&C Lab

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	169	NTIS

MSFC - Form 3292 (Rev December 1972) For sale by National Technical Info[...]on Service, Springfield, Virginia 22151

PREFACE

This document is intended to serve as a User's Manual and a Programmer's Manual for the Mask Analysis Program. The first portion of the document is devoted to the user. It contains all of the information required to execute MAP. The remainder of the document describes the details of MAP software logic. Although the information in this portion is not required to run the program, it is recommended that every user review it to gain an appreciation for the program functions.

Prepared by:

C. L. Mitchell

Approved by:

K. H. Schonrock

TABLE OF CONTENTS

TABLE OF CONTENTS
(continued)

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In examining software designed specifically for artwork verification and other closely related graphic analysis functions, several major disadvantages become apparent:

- o Artwork verification and other analysis processes are generally limited to small designs due to techniques for which execution time and computer memory resources increase out of proportion to an increase in design density or complexity.

- o When artwork analysis techniques are too rigid or too specific, the software is useful for only limited technologies. The software will require constant updating, and eventually become obsolete.

- o If the software requires lengthy and cumbersome input data preparation, or if the input data is an output produced by another program, standalone usage may be prohibited. If the output format is a rigid interface to another program, its usefulness as a standalone program is questionable.

The Mask Analysis Program (MAP) was designed to avoid these pitfalls and to surpass the capability of the single function analysis programs. Two significant principles were applied in the design of MAP.

- o Data Base Simplicity - MAP operates on mask data which has been converted from its original form to orthogonal rectangles with identification numbers.

- o Maximum User Control - MAP processes are completely controlled by a unique command language. Among the mask analysis processes which the user may command are: qualitative and quantitative testing of mask areas, modification of masks, creation of new masks, and computation and sorting of mask data.

Among the major features of MAP are the following:

- Versatility - The user is not bound to a specific technology nor to specific mask tests. MAP analysis processes may be applied to a wide range of computer-aided design applications.

- Multipurpose Processing - MAP is not a single function processor restricted to artwork verification. It is also capable of performing the equally important tasks of device identification, nodal analysis, capacitance calculation, and logic equation generation.

- Variable User Interface - Input of masks to MAP requires no lengthy or cumbersome formatting or device identification. Input and output data is not restricted to a single format. A number of formats are available. In addition, MAP is structured such that implementation of new formats will require only the replacement of an isolated subroutine.

- Machine Independence - MAP processing is performed on an open-ended data structure basis; i.e., the number and size of masks analyzed are not restricted by available core memory. MAP was written in FORTRAN IV using only the widely compatible statement forms.

- High-Speed Processing - Because of the simplicity of rectangle processing, MAP is capable of greater processing speed than other accepted methods of mask processing.

2. PROGRAM STRUCTURE AND OPERATION

This section presents a general description of MAP processing flow and a general definition of processing terms which will be referred to in the remainder of the document. Sections 3, 4, and 5 present the details of specific processes for specific circumstances. Appendix A illustrates job setup examples for the actual execution of MAP.

2.1 Flow of Processing

Figure 2-1 illustrates the general flow of MAP processing. As shown, there are two major processes: rectangular refinement and command execution.

Rectangular refinement occurs at the beginning of execution and is the process of reading the graphic input data, converting the data to orthogonal rectangles if necessary, and storing the rectangular data on a file. This is a two pass process. On the first pass, smashing into rectangles occurs. This may consist of breaking orthogonal polygons into rectangles and/or breaking non-orthogonal polygons into thin horizontal rectangle slices which approximate the shape with a degree of accuracy chosen by the user. On the second pass, any necessary scaling or offsetting is performed.

Command execution is performed for the remainder of the run. Commands are processed one at a time. Data is read from the mask rectangle file, processing is performed, new data is stored on the rectangle file or an output file, and any user information is printed. Execution continues in this fashion until all of the commands have been processed.

2.2 Definition of Terms

2.2.1 Rectangle Definition and Identification

Figure 2-2 illustrates the MAP definition of an orthogonal rectangle. Six words are required to define a rectangle: two identification numbers, the lower left corner coordinates, and the upper right corner coordinates. Rectangles are processed and stored in this form.

The identification numbers have been provided to allow associations between rectangles to be established and identified by the user. These numbers are referred to in this document as the primary and secondary identifiers and the specific values of each will be discussed in accordance with each command in subsequent sections.

GENERAL FLOW OF PROCESSING

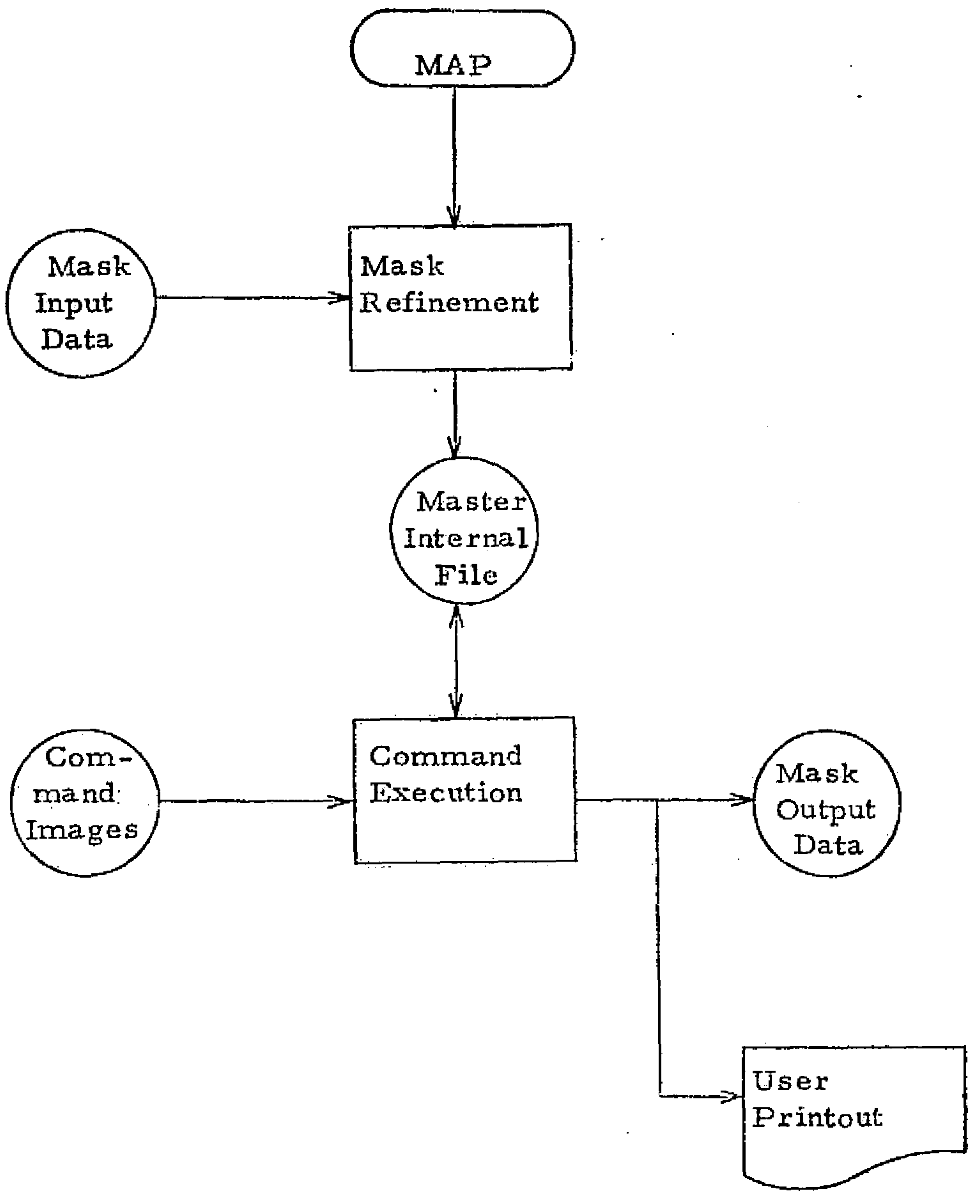


Figure 2-1

SIX-WORD RECTANGLE DEFINITION

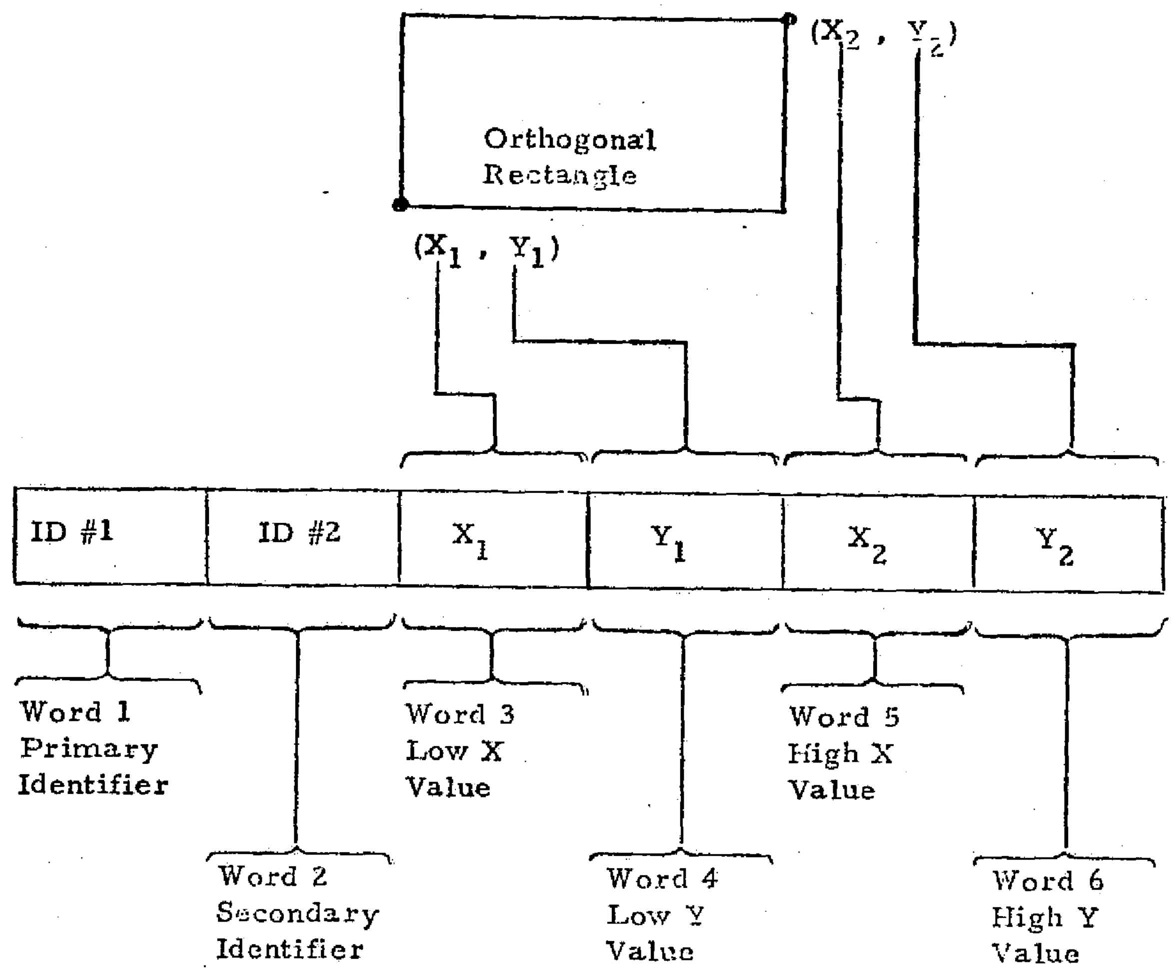


Figure 2-2

2.2.2 General Graphics Terms

Several terms have special connotations with regard to MAP. The term <u>mask</u> refers to the collection of graphic data which is on a single artwork level or which bears a certain qualitative significance. In other words, a mask need not conform to a strict technological mask definition. For example, a mask may be the resulting errors from an artwork test. Internal to MAP, a mask is a collection of rectangles which is stored in one or more records beginning at a mask position on the rectangle file, and which is accessed by a unique 4-character alphabetic name.

The term <u>mask areas</u> must be interpreted somewhat in context. However, in general, an area refers to one or more rectangles on a mask which have a collective significance which is often reflected in the rectangle identifiers.

The term <u>linkage</u> has a special MAP connotation. Rectangles on a mask are linked if they are physically connected. Rectangles on different masks are linked if they intersect or if they are each linked on their own masks to other rectangles which intersect.

3. GRAPHIC DATA INPUT

The graphic data associated with a set of masks may be described in many ways. One of the significant features of MAP is the wide variety of input data formats which it will accept. MAP will accept masks from mixed formats within the same run. Appendix B details each input format currently accepted. The structure of MAP is such that acceptance of additional formats requires only the addition of a small subroutine to interpret each format.

After a specific format is interpreted by MAP, the data is translated into an internal format allowing the most efficient processing capabilities. This internal format, as discussed in Section 2, is in the form of rectangle definitions.

Most users will probably find this 2-dimensional form quite suitable. However, the facility exists in MAP for handling 1-dimensional data. The user may specify that the input data for a particular run is to be treated as line segments. This requires the same storage space; line end points instead of rectangle diagonal end points are stored. One-dimensional data of this type is readily acceptable to many of the MAP processes. However, there are certain processes which are not meaningful with 1-dimensional data and the program will consider the data 2-dimensional. The user, therefore, should be cautious in selecting meaningful commands when processing linear data.

4. MAP COMMAND FORMATS

This section is devoted to the description of MAP commands. The MAP command language is the means by which the user initiates mask processing. This approach contributes to the versatility of the program and provides the maximum user control, since processing is performed exactly as commanded.

For a program using this command approach to be versatile, the command language must offer a wide range of capabilities. Table 4-1 is a summary of the commands comprising the high-level MAP language. The commands are listed in several categories:

- Preprocessing Commands are processed at the beginning of the execution to specify run options and assign names to the input masks.
- Input/Output Commands are provided to allow the user to conveniently transfer data.
- Operational Commands direct the performance of a long list of single and double mask operations which result in the creation of new masks.
- List Processing Commands direct several multiple mask processes which require complex list processing techniques.
- Dimensional Processing Commands are provided to initiate various dimensional calculations.
- Control Commands are provided to allow the user to change position in the command string to construct loops and branches in processing.

The user will find that the MAP language contains commands for very complex processes as well as very simple processes. Many of the operational processes, for example, are very basic processes. The user will find that several basic processes are often required to obtain a desired result, and there are often several methods of obtaining the same result. With practice, a user can quickly acquire the ability to set up efficient command strings. The user should not find it disturbing if a particular execution requires several hundred commands. MAP processing speed is very high, particularly for the very basic processes.

MAP COMMAND SUMMARY

COMMAND	DESCRIPTION
Preprocessing Commands:	
OPTN	Option specification
MASK	Input mask identification
Input/Output Commands:	
COMM	Comment printout
FILE	Special mask output
TEXT	Special text output
FREE	Mask storage release
Operational Commands:	
OPER	Mask operations
SPEC	Mask operation specifications
List Processing Commands:	
TRAC	Interconnection tracing
BOOL	Boolean equation generation
LIST	List cross-reference
Dimensional Processing Commands:	
AREA	Area computation
PERI	Perimeter computation
PARE	Area and perimeter computation
RANG	Range computation
Control Commands:	
SKIP	Unconditional routing
IFNL	Null condition routing

Table 4-1

This section presents a discussion of each of the command categories and a detailed description of each command format and usage considerations. The commands are input in a basically free format, are read as 80-character records, and are interpreted character-by-character.

The format descriptions which follow illustrate the general form, however, the following points should be noted when coding commands:

- Command names occupy the first four characters of the command record.
- Mask names must be alphabetic strings of one to four characters with no internal blanks and must not be the same as any of the MAP command names, operators, specifiers, or qualifiers.
- All numerical values must be whole numbers. Positive values should not be preceded by a "+" sign.
- All delimiters (commas, colons, slashes, equal signs, etc.) where specifically illustrated in the format description are required.
- Where blanks are indicated in the format description, any number of blanks may be entered and any number of blanks may be placed between the command name and the remainder of the command.
- Characters 73-80 of a command record are not interpreted as part of the command and, therefore, may be used for comments or identification.

4.1 Preprocessing Commands

The preprocessing commands are required for every MAP execution, first the OPTN command and then the required number of MASK commands. These two commands are read only through the command input unit, and should not be included with commands to be read from an alternate command input unit.

4.1.1 OPTN - Option Specification Command

Format:

OPTN p, a, d, f, o, s .

p = 0/1/2/3/4/5/ - degrees of printout.

a = 0/1/ - alternate input command unit, off/on.

d = 0/1/ - normal 2-dimensional data/1-dimensional data.

f = $\pm$ integer - input scale factor. Positive implies multiplication, negative implies division.

o = 0/1/ - automatic offset, on/off.

s = 0/integer - smash slice height after scaling when applicable.

Description:

The OPTN command is used to specify options which are to apply to an individual MAP run. It is the first command processed.

The user may select one of six degrees of printed output from the printer unit. Table 4-2 indicates the information provided for each degree. A value of 2 or 3 is recommended for most runs.

If an alternate command unit is specified, all commands following the last MASK command will be read from the alternate unit.

One- or two-dimensional data interpretation is in effect for the entire run.

The scale factor indicated is applied after and in addition to any scale factors indicated in the input data file.

Automatic offset adjusts all mask coordinates such that the lowest and leftmost point on any mask is point (0, 0). If automatic offset is not specified, offset will occur only if negative coordinate values are encountered in the data.

OPTIONAL INFORMATION OUTPUT

INFORMATION DESCRIPTION	PRINT OPTION
No printout	0
General information messages Error messages File and mask directory at beginning and end of run Command images as processed	1-5
File and mask directory after each process Listing of resultant mask data when PRNT specifier is used or when data is of a list form Timer output after each process	2-5
Listing of command file at the beginning of the run Listing of all resultant mask data	3-5
Basic debug output	4-5
Listing of intermediate processing mask results Extensive debug output	5

Table 4-2

The smash slice height should be specified as zero unless the input data contains non-orthogonal items. In that case, a smash slice height should be specified in accordance with the range in value of the data and the accuracy desired. Very thin slices will result in very accurate approximations but will increase the number of rectangles on the mask and reduce the processing speed. On the other hand, the slice height should not be too large. A rule of thumb is the maximum slice height should not exceed the shortest distance between any non-orthogonal line and any other line which is not directly connected. This will eliminate any discontinuities which could be created in smashing.

4.1.2 MASK - Input Mask Identification Command

Format:

MASK $u, t, s, m_1, \ldots, m_n$

u	=	integer ≥ 11 - mask input logical unit number. On machines which require a FORTRAN DEFINE FILE for a particular format file type, the value of u is restricted to the range of 11-15.
t	=	-1/0/1/2/3/4 - mask data format type. See Appendix B for complete description.
s	=	integer - number of levels to skip before accepting input data. For types -1, 3, and 4 it is with respect to the beginning of the file; for types 0, 1, and 2 it is with respect to the current file position.
$m_1, \ldots, m_n$	=	names to be assigned to the masks as they are input.

Description:

At least one MASK command is required for MAP execution. It must follow the OPTN command.

Multiple MASK commands are required when: data is to be input from more than one unit, the data consists of mixed formats, or the desired masks cannot be read in a continuous sequential manner from the input device.

The level skip capability is provided to allow inputting portions of mask files. After part of a file is skipped, successive mask levels are accepted until all the assigned names have been used.

4.2 Input/Output Commands

The commands in this category are all associated with input/output operations.

The COMM and FREE commands are primarily provided as a convenience to the user. The FILE and TEXT commands represent the major MAP facility for mask data output.

4.2.1 COMM - Comment Printout Command

Format:

COMM character string

character string = users comment of up to 68 characters to be printed on the listing.

Description:

The COMM command allows the user to document the MAP execution listing. Whenever a COMM command is encountered, the character string contained is printed on the run listing, centered on the page. These comments are also a benefit to the user by helping to maintain order and meaningfulness of the command file or card deck.

4.2.2 FILE - Special Mask Output Command

Format:

FILE u, t, m, l, f

u = 5/6/7/8/9 - output logical unit number.

t = 0/1/2/3- mask data format type. See Appendix B for complete description.

m = name of the mask to be output.

l = level number on output file. Required only for type 3 format.

f = 1/0 - new file to be started/no new file. Applicable only to type 3 format.

Description:

The FILE command initiates mask coordinate data output according to the format specified by the user. The coordinate data is output as rectangle descriptions. Where the format permits, the rectangle identifiers are included.

4.2.3 TEXT - Special Text Output Command

Format:

TEXT u, t, m, l, f

u = 5/6/7/8/9 - output logical unit number.

t = 0/1/3 - mask data format type. See Appendix B for complete description.

n = 1/2/3 - primary/secondary/both identifiers to be output.

m = name of the mask for which identifiers are output.

l = level number on output file. Required only for type 3 format.

f = 1/0 - new file to be started/no new file. Applicable only to type 3 format.

Description:

The TEXT command initiates output of rectangle identifiers. Included are the lower left coordinates of each associated rectangle. This type of information is very useful for displaying mask area identification when utilizing a graphics system with a CRT.

4.2.4 FREE - Mask Storage Release Command

Format:

FREE $m_1, \ldots, m_n$

$m_1, \ldots, m_n$ = names of the masks to be deleted from the master mask file.

Description:

The master mask file contains space for a specific number of masks. The number of masks which can be contained depends on the file assignment and the dimension of the directory arrays in the particular MAP version. When this number of masks is small due to machine or disk limitations, the user may need to delete masks which are no longer needed for processing, to provide file space for other masks. This is initiated through the FREE command. The program simply erases its directory information for the masks listed, releasing the file space for further use.

4.3 Operational Commands

Two operational commands, OPER and SPEC direct the performance of a number of simple mask operations. The OPER command is used to initiate an operation involving one or two masks, the result of which is the creation of a new mask. The SPEC command is used to define specifications to an operation. It precedes the OPER command of the operation for which the specifications apply.

Because the operations which are available and the specifications which can be applied are numerous, the general forms of the OPER and SPEC commands will be discussed first, each followed by a complete description of the specific forms.

4.3.1 OPER - Mask Operation Command

General Format:

OPER m_r = o$\cancel{b}$$m_1, m_2$$\cancel{b}$$q_1, q_2$

m_r = the name to be given to the mask resulting from the operation.

o = a 4-character operator identifying the operation to be performed.

m_1, m_2 = the names of the masks involved in the operation.

q_1, q_2 = any qualifiers or options associated with the operation where applicable.

General Description:

The OPER command simply initiates the creation of a new mask from a specified operation on one or two other masks. It is permissible for a mask being operated upon to also be named as the resultant mask, in which case, the resultant mask will assume the name and the other mask will be lost upon completion of the operation. Table 4-3 is a summary of the operational forms available with the OPER command.

SAME Format:

OPER m_r = SAMEb̶m_1

m_r = the name of the resultant mask.

m_1 = the name of the mask to which the resultant mask is equated.

SAME Description:

The SAME operation simply creates a mask which is a duplicate of another mask. Unless otherwise specified (by a SPEC command), the identifiers on the resultant mask will be the same as on the original mask.

NGTV Format:

OPER m_r = NGTVb̶m_1

m_r = the name of the resultant mask.

m_1 = the name of the mask of which the resultant mask is a negative.

NGTV Description:

The NGTV operation creates a mask which is the negative of another mask. The lower boundaries of the resultant mask are zero in both x and y directions. The upper boundaries are the maximum x and y values encoutered during preprocessing of the masks.

Unless otherwise specified (by a SPEC command), the primary and secondary identifiers for all of the rectangles on the resultant mask will be set to one and two respectively.

EDGE Format:

OPER m_r = ob̶m_1b̶q_1, q_2

m_r = the name of the resultant mask.

o = edge operator:

EDGE - extraction of all edges.
HEDG - horizontal edges only.
VEDG - vertical edges only.

SUMMARY OF OPERATIONS

OPERATOR FORM	MASKS INVOLVED	FUNCTION PERFORMED
SAME	1	Equation
NGTV	1	Negation
EDGE HEDG VEDG	1	Edge extraction
EXPN	1	Expansion
PLUS	2	Addition
INTR	2	Intersection
NINT	2	Non-intersection extraction
EXOR	2	Exclusive OR
LINK	1	Single mask linkage
LINK	2	Double mask linkage
NLNK	2	Non-linkage extraction
TWIX HTWX VTWX DTWX	1	Single mask spacing extraction
TWIX HTWX VTWX DTWX	2	Double mask spacing extraction
SPIN	1	z-axis rotation
FLIP	1	x- and/or y-axis rotation
PUSH	1	Offset
SCAL	1	Scaling
WNDW	1	Window restriction
PLAC	2	Cell placement

Table 4-3

m_1 = the name of the mask from which area edges are to be extracted.

q_1, q_2 = optional buried line removal qualifiers. One or two of the following:

SAM1 - same primary identifiers.
SAM2 - same secondary identifiers.
SAME - both identifiers the same.
DIF1 - different primary identifiers.
DIF2 - different secondary identifiers.
DIFF - both identifiers different.

EDGE Description:

Any form of the EDGE operation creates a new mask which contains the edges (or sides) of areas (rectangles or groups of rectangles) on another mask. Although the resultant edges are lines, they are stored and processed as widthless rectangles. The user has the option of extracting all edges, horizontal edges only, or vertical edges only by choosing the operator EDGE, HEDG, or VEDG respectively.

If no optional buried line removal qualifiers are specified, the resultant mask contains all of the horizontal and/or vertical edges of every rectangle. The buried line removal qualifiers allow the user to obtain the edges of mask areas (rectangle clusters). If several rectangles are adjacent or overlapping and only their periphery is desired, the internal edges may be eliminated. The line removal qualifiers represent the conditions upon which internal lines will be removed. The conditions are based on the relationships of the primary and secondary identifiers of any two adjacent or overlapping rectangles in question. Figure 4-1 illustrates several examples of EDGE operation results with various line removal qualifiers.

Unless otherwise specified (by a SPEC command), a resultant edge will retain the identifiers of the rectangle from which it was extracted.

EXPN Format:

OPER m_r = EXPNb̸m_1b̸x, y

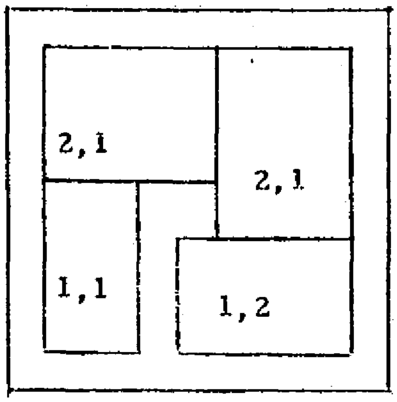
m_r = the name of the resultant mask.

m_1 = the name of the mask to be expanded.

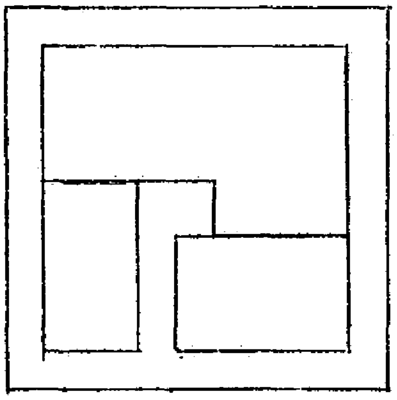
x, y = $\pm$ integer expansion values.

EXAMPLES OF EDGE OPERATION BURIED LINE REMOVAL

Mask A



Mask B - result of OPER B = EDGE A SAME



Mask C - result of OPER C = EDGE A DIF1

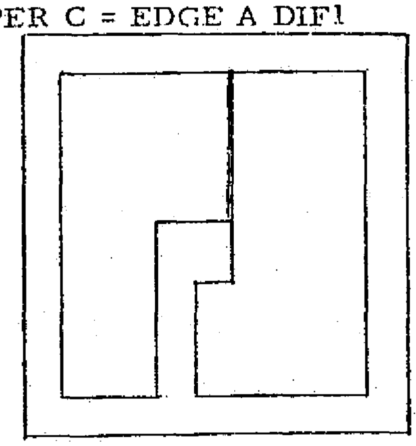


Figure 4-1

EXPN Description:

The EXPN operation creates a new mask which contains the rectangles of another mask expanded by given distances in the x and/or y directions. Negative expansion values will result in size reduction. The overall dimensional change is twice the expansion value; e.g., in the horizontal direction, the left side is shifted left a distance x, and the right side is shifted to the right also by a distance x.

Unless otherwise specified (by a SPEC command), the primary and secondary identifiers will be retained exactly as on the original mask.

Any portion of the resultant mask, which as a result of expansion, falls outside of the first quadrant (positive coordinates) or exceeds the maximum positive x or y limits, will be lost.

The reader will note that this operation is an additive process. For a multiplicative scaling, refer to the SCAL form of the OPER command.

PLUS Format:

OPER m_r = PLUSƀm_1, m_2

m_r = the name of the resultant mask.

m_1, m_2 = the names of the masks to be combined.

PLUS Description:

The PLUS operation simply creates a new mask which contains all of the rectangles of two other masks. Unless otherwise specified (by a SPEC command), the rectangles will retain their original primary and secondary identifiers.

INTR Format:

OPER m_r = INTRƀm_1, m_2ƀq_1, q_2

m_r = the name of the resultant mask.

m_1, m_2 = the names of the masks to be intersected.

q_1, q_2 = optional intersection qualifiers. One or two of the following:

SAM1 - same primary identifiers.
SAM2 - same secondary identifiers.
SAME - both identifiers the same.

DIF1 - different primary identifiers.
DIF2 - different secondary identifiers.
DIFF - both identifiers different.

INTR Description:

The INTR operation creates a new mask which contains rectangles representing the coincidence or intersection of two other masks.

If no qualifiers are specified the resultant mask contains all intersection areas of the two masks. The intersection qualifiers allow the user to specify what is a valid intersection between two rectangles based on some relationship between the rectangle identifiers, as listed above.

Unless otherwise specified (by a SPEC command), a rectangle on the resultant mask will have primary and secondary identifiers respectively reflecting the primary identifier of the rectangle on the first (m_1) mask and the primary identifier of the rectangle on the second (m_2) mask which intersected to form it.

NINT Format:

OPER m_r = NINTb̸m_1, m_2b̸q_1, q_2

m_r = the name of the resultant mask.

m_1 = the name of the mask containing the areas to be found exclusive.

m_2 = the name of the mask containing excluded areas.

q_1, q_2 = optional intersection qualifiers, see INTR Format.

NINT Description:

The NINT operation creates a new mask which contains all of the areas on the first mask (m_1) which do not intersect the second mask (m_2). The qualifiers are the same as for the INTR operation.

Unless otherwise specified (by a SPEC command), the resultant rectangles retain the original identifiers.

EXOR Format:

OPER m_r = EXORb̸m_1, m_2b̸q_1, q_2

m_r = the name of the resultant mask.

m_1, m_2 = the name of the mask on which rectangle linkages are to be found.

q_1, q_2 = optional intersection qualifiers. See INTR Format.

EXOR Description:

The EXOR operation creates a new mask which contains the exclusive OR of the areas on two other masks. The Boolean equivalent expression is $m_r = m_1 \cdot \overline{m}_2 + \overline{m}_1 \cdot m_2$. The qualifiers are the same as for the INTR operation.

Unless otherwise specified (by a SPEC command), the resultant rectangles will retain the identifiers from their respective original masks.

Single Mask LINK Format:

OPER m_r = LINKb̸m_1b̸q_1

m_r = the name of the resultant mask.

m_1 = the name of the mask on which rectangle linkages are to be found.

q_1 = optional inter-mask linkage qualifier:

POIN - minimum connection by a point.
LINE - minimum connection by a line.
AREA - minimum connection by an area.

Single Mask LINK Description:

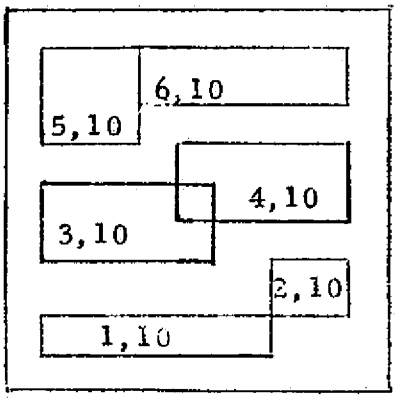
The LINK operation on a single mask creates a new mask containing the same rectangles as on the original mask but with their identifiers modified to reflect connections between rectangles. The resultant secondary identifiers are set equal to the original primary identifiers for each rectangle. The operation locates groups of physically connected rectangles. The rectangles within each group are assigned a new primary identifier which is equal to the lowest original primary identifier of the rectangles within the connected group.

The user may specify the minimum qualifications for a valid physical connection with the qualifiers listed above. If no qualifier is specified it defaults to POIN.

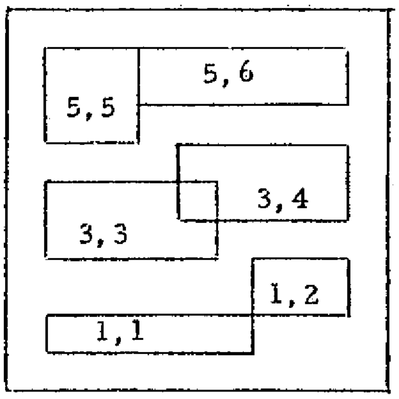
Figure 4-2 illustrates the resulting identification number assignments for each connection qualification for the single mask LINK operation.

EXAMPLE OF THE SINGLE MASK LINK OPERATION

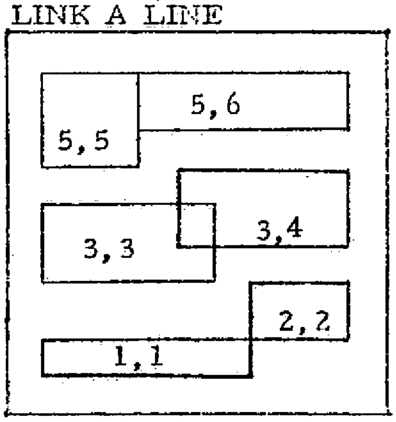
Mask A



Mask B - result of OPER B =
LINK A POIN



Mask C - result of OPER C =
LINK A LINE



Mask D - result of OPER D =
LINK A AREA

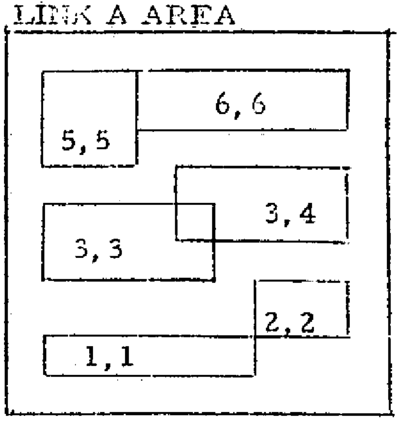


Figure 4-2

Double Mask LINK Format:

OPER m_r = LINKƀm_1,m_2ƀq_1,q_2

m_r = the name of the resultant mask.

m_1 = the name of the mask to be linked.

m_2 = the name of the mask to which linkage is performed.

q_1 = intra-mask linkage qualifier:

POIN	-	minimum connection by a point,
LINE	-	minimum connection by a line.
AREA	-	minimum connection by an area.
NONE	-	no intra-mask linkage.

q_2 = optional inter-mask linkage qualifier:

POIN	-	minimum connection by a point.
LINE	-	minimum connection by a line.
AREA	-	minimum connection by an area.

Double Mask LINK Description:

The double mask LINK operation first locates the connected groups of rectangles on the first mask (m_1) in the same manner as in a single mask LINK operation unless otherwise specified (q_1 = NONE). The qualifier q_1 is specified to indicate the conditions of a valid connection for this step.

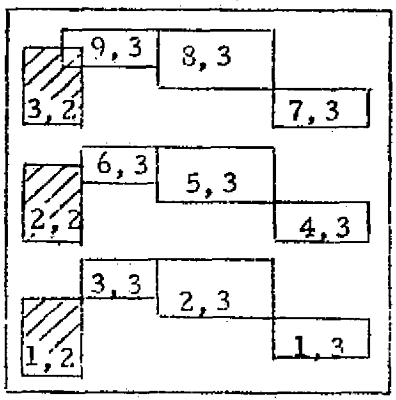
The next step is to locate only those groups of connected rectangles which are connected to any rectangle on the second mask (m_2). The qualifier, q_2, indicates what is a valid connection between masks.

The rectangles remaining after the final step constitute the resultant mask. The rectangle secondary identifiers (unless otherwise specified by a SPEC command) will be the original primary identifiers of the rectangles on mask m_1. The rectangle primary identifiers will be set to the primary identifier (unless otherwise specified by a SPEC command) of the first (lowest number) rectangle found on the second mask to which each is linked.

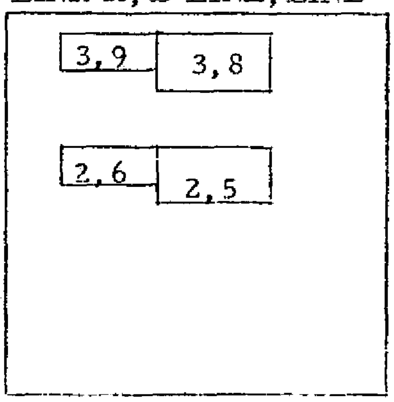
Figure 4-3 illustrates an example of the double mask LINK operation.

EXAMPLES OF THE DOUBLE MASK LINK AND NLNK OPERATIONS

Mask A and Mask B
(crosshatched)



Mask C - result of OPER C =
LINK A, B LINE, LINE



Mask D - result of OPER D =
NLNK A, B LINE, LINE

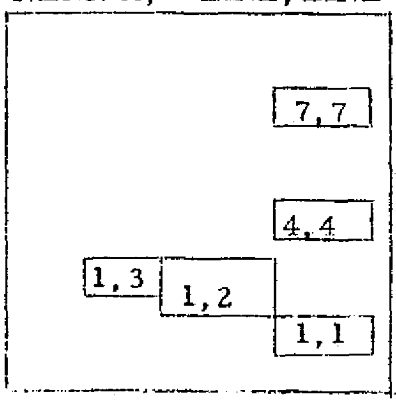


Figure 4-3

NLNK Format:

OPER m_r = NLNKƀm_1, m_2ƀq_1, q_2

m_r = the name of the resultant mask.

m_1 = the name of the mask to be found not linked.

m_2 = the name of the mask to which no linkage is to occur.

q_1 = intra-mask linkage qualifiers (see Double Mask LINK Format).

q_2 = optional inter-mask linkage qualifiers (see Double Mask LINK Format).

NLNK Description:

The NLNK operation is essentially the reverse of the double mask LINK operation; i.e., to find the rectangles on the first mask which are not linked to the second mask.

The first step is the same as for the double mask LINK operation. The second step is to find only the rectangles or groups of connected rectangles on the first mask (m_1) which are not connected to any rectangle on the second mask (m_2). The qualifiers indicate valid connections, and are identical in form and meaning to those for the double mask LINK.

Unless otherwise specified (by a SPEC command), the rectangle identifiers will be assigned the same as for a single mask LINK, i.e., as resulting from the first step of the operation.

Figure 4-3 illustrates an example of the NLNK operation.

Single Mask TWIX Format:

OPER m_r = oƀm_1ƀq_1

m_r = the name of the resultant mask.

o = spacing operator:

TWIX	-	horizontal and vertical spacing.
HTWX	-	horizontal spaces only.
VTWX	-	vertical spaces only.
DTWX	-	spaces diagonally off rectangle corners.

m_1 = the name of the mask upon which spaces are to be found.

q_1 = optional identifier qualifier:

SAME - same primary identifiers.
DIFF - different primary identifiers.

Single Mask TWIX Description:

Any one of the forms of the single mask TWIX operation is a process of locating spaces between rectangles on a mask. The user may extract horizontal and vertical, horizontal only, vertical only, or diagonal spaces by choosing one of the operator forms: TWIX, HTWX, VTWX, or DTWX, respectively.

The user may further qualify the extracted spaces by specifying a qualifier limiting valid spaces to only those between two rectangles with different primary identifiers. If no qualifier is given all spaces are extracted.

Unless otherwise specified (by a SPEC command), the space rectangles constituting the resultant mask will each have their primary identifier set to the primary identifier of the rectangle on the original mask which was to the left or below it, and each secondary identifier will be set to the primary identifier of the rectangle on the original mask which was to the right or above it.

Figure 4-4 illustrates examples of single mask TWIX operations.

Double Mask TWIX Format:

OPER m_r = o$\cancel{b}$$m_1$, $m_2$$\cancel{b}$$q_1$

m_r = the name of the resultant mask.

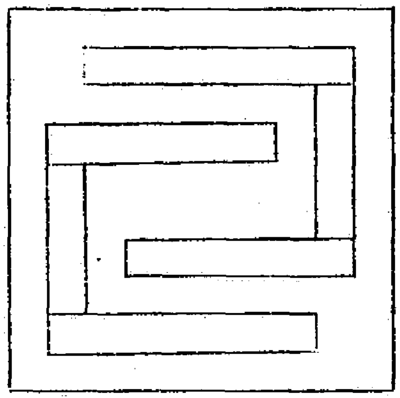
o = spacing operator:

TWIX - horizontal and vertical spaces.
HTWX - horizontal spaces only.
VTWX - vertical spaces only.
DTWX - spaces diagonally off rectangle corners.

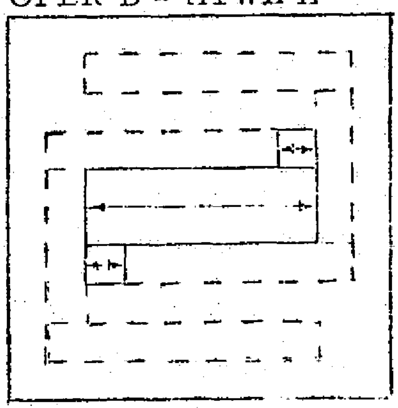
m_1, m_2 = the names of the masks between which spaces are to be found.

EXAMPLES OF SINGLE MASK TWIX OPERATION FORMS

Mask A



Mask A (broken lines)
and Mask B - result of
OPER B = HTWX A



Mask A (broken lines)
and Mask C - result of
OPER C = VTWX A

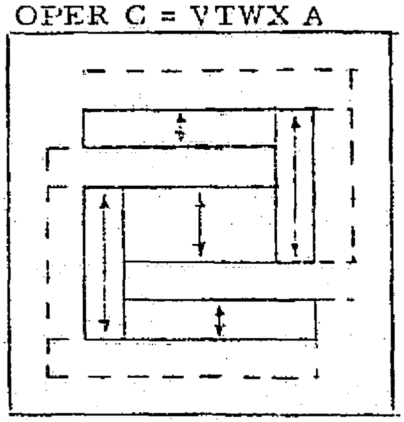


Figure 4-4

q_1 = optional identifier qualifier:

SAME - same primary identifiers.
DIFF - different primary identifiers.

Double Mask TWIX Description:

The forms of the double mask TWIX operation are the same as the single mask TWIX forms except that spaces extracted are between the rectangles of two masks; i.e., as would be seen if the two masks were overlayed. Only spaces between rectangles of different masks are extracted.

The identifier qualifier has the same meaning as for the single mask TWIX operation.

Unless otherwise specified (by a SPEC command), the space rectangles constituting the resultant mask will each have primary and secondary identifiers corresponding to the primary identifiers of the rectangle on the first mask (m_1) and the rectangle on the second mask (m_2), respectively, between which the space was located.

SPIN Format:

OPER m_r = SPINƀm_1ƀx_r, y_r, q

m_r = the name of the resultant mask.

m_1 = the name of the mask to be rotated.

x_r, y_r = integers > 0 - coordinates of the center of rotation.

q = 1/2/3 - the number of quadrants of rotation.

SPIN Description:

The result of the SPIN operation is the original mask rotated in the x-y plane about the point (x_r, y_r) q quadrants in the clockwise direction.

Unless otherwise specified (by a SPEC command) the resultant rectangle identifiers are the same as on the original mask.

Any portion of the mask, which, as a result of the SPIN operation, falls outside of the first quadrant (positive coordinates) or exceeds the maximum positive x or y limits, will be lost.

FLIP Format:

OPER m_r = FLIPbm_1bx_m, y_m

m_r = the name of the resultant mask.

m_1 = the name of the mask to be mirrored.

x_m = positive integer - x mirroring axis, or no x mirroring if zero.

y_m = positive integer - y mirroring axis, or no y mirroring if zero.

FLIP Description:

The result of the FLIP operation is the mirroring of a mask about a line $x = x_m$ and/or about a line $y = y_m$. Unless otherwise specified (by a SPEC command), the resultant rectangle identifiers are the same as on the original mask.

Any portion of the mask which, as a result of mirroring, falls outside of the first quadrant (positive coordinates) or exceeds the maximum positive x or y limits will be lost.

PUSH Format:

OPER m_r = PUSHbm_1bx_o, y_o

m_r = the name of the resultant mask.

m_1 = the name of the mask to be offset.

x_o = integer - x direction offset value.

y_o = integer - y direction offset value.

PUSH Description:

The result of the PUSH operation is the offsetting of the original mask by a value of x_o in the x direction and y_o in the y direction.

Unless otherwise specified (by a SPEC command), the resultant rectangle identifiers are the same as on the original mask.

Any portion of the mask which, as a result of the offset, falls outside of the first quadrant (positive coordinates) or exceeds the maximum positive x or y limits, will be lost.

SCAL Format:

OPER m_r = SCALbm_1bx_s, y_s

m_r = the name of the resultant mask.

m_1 = the name of the mask to be scaled.

x_s = integer scale factor in the x direction. Negative values indicate division. A zero is interpreted as no scaling.

y_s = integer scale factor in y direction. Negative values indicate division. A zero is interpreted as no scaling.

SCAL Description:

The result of the SCAL operation is the scaling of the original mask by given factors in the x and/or y directions. The reader will note that this is a multiplicative process. For an additive expansion or reduction see the EXPN form of the OPER command.

Unless otherwise specified (by a SPEC command), the resultant rectangle identifiers are the same as on the original mask.

Any portion of the mask which, as a result of the scaling process, falls outside of the first quadrant (positive coordinates) or exceeds the maximum positive x or y limits will be lost.

WNDW Format:

OPER m_r = WNDWbm_1bx_1, y_1, x_2, y_2

m_r = the name of the resultant mask.

m_1 = the name of the mask to be windowed.

x_1, y_1 = integers - the lower left corner coordinates of the rectangular window.

x_2, y_2 = integers - the upper right corner coordinates of the rectangular window.

WNDW Description:

The result of the WNDW operation is a mask which contains only the areas of the original mask which fall within the range of a rectangular window. The window is specified by its lower left corner point (x_1, y_1) and its upper right corner point (x_2, y_2).

Unless otherwise specified (by a SPEC command), the resultant rectangle identifiers are the same as on the original mask.

PLAC Format:

OPER m_r = PLAC ␢ m_1, m_2

m_r = the name of the resultant mask.

m_1 = the name of the mask containing the cells to be placed.

m_2 = the name of the mask containing placement location of the cells.

PLAC Description:

The PLAC operation creates a mask which is the result of placing the cells defined by the first mask (m_1) in the positions defined by the second mask (m_2).

Unless otherwise specified (by a SPEC command), the resultant placed cell rectangle identifiers will be the cell identifiers as on the first mask (m_1).

Any portion of the mask which, as a result of the placement process, falls outside of the first quadrant (positive coordinates) or exceeds the maximum positive x or y limits will be lost.

The mask containing the cell description (m_1) must be a valid cell mask. The mask containing the cell placement location (m_2) must be a valid placement mask. Refer to Appendix B for the data types and format descriptions of these special mask forms.

4.3.2 SPEC - Mask Operation Specification Command

General Format:

SPEC $s_1, s_2, \ldots, s_n$

$s_1, s_2, \ldots, s_n$ = specifications list.

General Description:

The SPEC command is used to define specifications to be applied to the operations defined by OPER commands. Table 4-4 is a summary of the available specification forms.

Any number or combination of specifications may be defined for a single operation. Multiple SPEC commands may be used where necessary. All SPEC commands must directly precede the OPER command for the operation to which they apply. Where conflicting or redundant specification types are contained in a SPEC command, the program accepts the last one encountered as valid. Once the operation has been performed, the program cancels the specifications.

In the following description of the individual specifier forms, the formats will each be illustrated as if they were used alone. Where several specifications are defined they are separated by commas as shown in the above general format.

Any rectangle identifier modifications as indicated by the specifiers E and R are performed in the very last step of any operation. The modifications are based on a previous identifier status. This identifier status is the default status and depends upon the operation form. The defaults are contained in the description of each OPER form in the preceding portion of this chapter.

PRNT Specifier Format:

SPEC PRNT

PRNT Specifier Description:

The PRNT specifier simply indicates the result of the operation is to be printed by the listing device. The six-word rectangle descriptions are printed one per line. The result is a six column list where the columns contain primary identifiers, secondary identifiers, low x-coordinates, low y-coordinates, high x-coordinates, high y-coordinates.

The user must be sure that the proper print option value has been specified by the OPTN command, as the PRNT specifier may be overridden.

TEMP Specifier Format:

SPEC TEMP

TEMP Specifier Description:

The TEMP specifier indicates that the result of the operation is to be discarded upon the completion of the operation. The TEMP specifier is obviously

SUMMARY OF OPERATIONAL SPECIFICATIONS

SPECIFIER FORM	APPLICATION TO OPERATIONAL RESULT
PRNT	Printed output
TEMP	Temporary storage
MIN	Minimum dimensional restrictions
MAX	Maximum dimensional restrictions
E	Identifier equation
R	Identifier replacement
S	Identifier starting number
I	Identifier incrementing value

Table 4-4

meaningless unless the output is printed. If it is used alone the operation will be wasted unless a PRNT specifier is defined or an appropriate print option was given in the OPTN command. This specifier is useful to avoid accumulating masks which only need to be printed.

MIN Specifier Format:

SPEC MINd = v

d = 1- character direction indicator:

A - all (x and y) directions.
X - x direction.
Y - y direction.
L - length.
W - width.
R - radial.

v = integer minimum dimension value.

MIN Specifier Description:

The MIN specifier allows the user to define the minimum acceptable dimensions of resultant mask areas. If some portion of an area does not meet the minimum dimension requirements, that portion will be deleted from the result. The minimum dimension restriction can be applied in any of the directions indicated above. Table 4-5 indicates which of the directions are valid for which operation forms.

MAX Specifier Format:

SPEC MAXd = v

d = 1-character direction indicator.

A - all (x and y) directions.
X - x direction.
Y - y direction.
L - length.
W - width.
R - radial.

v = integer maximum dimension value.

MIN AND MAX SPECIFIER'S APPLICATION TO OPERATIONS

SPECIFIER FORMS	APPLICATION TO OPER FORMS
MINA, MAXA	Applicable to any OPER form.
MINX, MAXX	Applicable to any OPER form.
MINY, MAXY	Applicable to any OPER form.
MINL, MAXL	Applicable to all EDGE forms to limit line lengths. Applicable to single and double mask TWIX, HTWX, and VTWX forms to limit the length of spacing runs.
MINW, MAXW	Applicable to single and double mask TWIX, HTWX, and VTWX forms to limit the spacing distance.
MINR, MAXR	Applicable to single and double mask DTWX forms limiting the diagonal spacing distance.

Table 4-5

MAX Specifier Description:

The MAX specifier allows the user to define the maximum acceptable dimensions of resultant mask areas. If some portion of an area exceeds the maximum dimension requirements, that portion will be deleted from the result. The maximum dimension restrictions can be applied in any of the directions listed. Table 4-5 indicates which of the directions are valid for which operation forms.

E Specifier Format:

SPEC $Ei_r@i_p$

i_r = the identifiers to be equated:

1 - primary.
2 - secondary.

i_p = the previous identifier to which the new identifier is being equated:

1 - primary.
2 - secondary.

E Specifier Description:

The E specifier is used to cause the resultant rectangle primary and/or secondary identifiers to be equated to the previous primary or secondary identifiers. A complete swap of identifiers is specified by E1@2 and E2@1.

R Specifier Format:

SPEC $Ri_r@i_p$

i_r = the identifiers to be replaced:

1 - primary.
2 - secondary.

i_p = the condition of the previous identifiers as the basis for incrementing the identifiers being replaced:

0 - unconditional replacement, identifier incremented for each rectangle.

1 - increment on a change in previous primary identifiers.

2 - increment on a change in previous secondary identifiers.

3 - increment on a change in either the primary or secondary identifiers.

R Specifier Description:

The R specifier is used to cause the resultant rectangle primary and/or secondary identifiers to be replaced based on some condition of the previous status of the identifiers.

When an R specifier has been given, the functional portion of the operation is performed normally. Upon completion, the resultant rectangles are arranged in an ordered list. The ordering is based first on the lowest primary identifier; when they are equal the next consideration is the lowest secondary identifier.

The replacement of identifiers is based on the position of the rectangles in the ordered list. An unconditional replacement ($i_p = 0$) is simply the assignment of new identifiers in sequential fashion through the list, incrementing the identifier for each rectangle. For the other replacement types ($i_p = 1$, 2, or 3) identifiers are replaced in the same sequential fashion through the list except the identifier value is only incremented upon conditions of change in previous identifiers. For $i_p = 1$ incrementing occurs when a new primary identifier is found, for $i_p = 2$ incrementing occurs when a new secondary identifier is found, and for $i_p = 3$ incrementing occurs when either a new primary or secondary has been found.

The S and I specifiers discussed later in this section are used in conjunction with the R specifier to define the starting value and the increment. If the S or I specifiers are not given, the starting value defaults to one and the increment defaults to one.

S Specifier Format:

SPEC Si_r@#bv, or

SPEC Si_r@i_1ƀm_1

i_r = the identifier being replaced for which the starting value is being defined:

1 - primary.
2 - secondary.

v = positive integer identifier starting value.

i_1 = the type of identifier to be looked up for use in computing starting value:

1 - primary.
2 - secondary.

m_1 = the name of the mask for which the highest i_1 identifier is looked up.

S Specifier Description:

The S specifier is used in conjunction with an R specifier to define the starting identifier to be used in the identifier replacement process.

There are two forms of the S specifier as shown above. The first form allows the user to specify a constant starting value. The second form allows the user to have a starting value calculated based upon the highest identifier found on another mask. The program looks up the highest type i_1 (primary or secondary) identifier assigned to any of the rectangles on the desired mask m_1. The program then adds one increment to this value to get the starting value for replacement.

When an R specifier is used without an S specifier the starting value defaults to one. When an S specifier is used without an R specifier, it is ignored.

Caution should be exercised in choosing a starting value. If an identifier exceeds the maximum integer value allowed for the particular computer, the run may produce erroneous results or may abort.

I Specifier Format:

SPEC Ii_r@#bv

i_r = the identifier being replaced for which the replacement incrementing value is being defined:

1 - primary.
2 - secondary.

v = integer increment ≥ 0.

I Specifier Description:

The I specifier is used in conjunction with an R specifier to define the value by which replacement numbers are changed for each time they are incremented. The increment value is simply given as a constant integer value.

When an R specifier is used without an I specifier the increment value defaults to one. When an I specifier is used without an R specifier, it is ignored.

Caution should be exercised in choosing large increment values. If an identifier exceeds the maximum integer value allowed for the particular computer, the run may produce erroneous results or may abort.

4.4 List Processing Commands

Three special list processing commands are provided for the MAP user. These commands direct complex multiple mask processes:

- o interconnection tracing (TRAC),
- o Boolean equation generation (BOOL), and
- o list cross-referencing (LIST).

These three commands have the same general format. The specific format for each is presented in the following sections. In all cases where the command definition becomes lengthy, it may be divided among several command records. In this case, the command string must be divided between items and an "*" must appear as the 72nd character on each record except the last.

4.4.1 TRAC - Interconnection Trace Command

Format:

TRAC $/i_1/i_2/\ldots/i_n/$

$/i_n/$ =nth item in the string, of the form $/m_n:m_a,\ldots m_i/$.

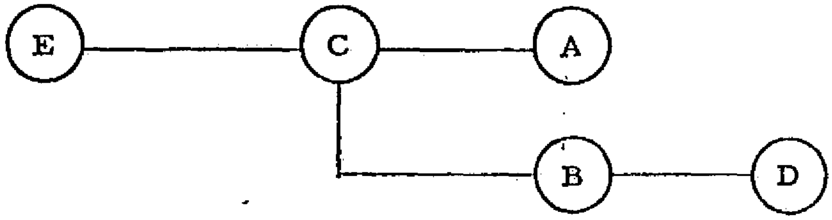
m_n =the name of the mask involved in the trace (reference mask).

m_a-m_i=the names of the masks which connect directly to the reference mask

Description:

The purpose of the TRAC process is to locate all interconnections (intersections) between a number of levels of masks. The result of this complex linkage process is the reassignment of rectangle identifiers to reflect all of the interconnections among the given masks.

The masks involved and their interconnection possibilities can be expressed as a connection tree. The tree is merely a graphic description of those masks which may be connected through intersection. The following illustration is an example of a connection tree of masks A through E.



This is interpreted: mask A connects to mask E if there is some area on mask A which intersects some area on mask C which intersects mask E, or is connected to another area on mask C which intersects mask E, and so on. If this condition exists then all of the areas on masks E, C, and A which contribute to this pathway, or node, will be given the same primary identification number.

The user represents the tree structure in the TRAC commands by indicating each mask name and the names of each mask to which it may immediately connect. For the above tree, the TRAC command might be represented as:

TRAC /E:C/C:A, B, E/A:C/B:C, D/D:B/

i.e., E connects to C, C connects to A, B, and E, and so on. The masks in the connection tree may be expressed in any order in the TRAC command, and the connection path identifier assignments will be based on that order.

The user should examine the primary identifiers of the rectangles of the masks involved before commanding execution of TRAC. Any rectangles with equal primary identifiers will be considered as the same node. If this is to be avoided, the primary identifiers on all the masks involved should be made unique. This may be done with a proper SPEC command preceding a SAME operation equating a mask to itself.

Appendix C illustrates a simple example of an electrical nodal trace using the TRAC command.

4.4.2 BOOL - Boolean Equation Generation Command

Format:

BOOL $/u/i_1/i_2/\ldots/i_n/$

u = 5/6/7/8/9 - logical unit to which Boolean data is to be written.

$/i_n/$ = nth item in the string, of the form $/m_n{:}m_a, m_b\ldots m_i/$ or of the form $/m_n{:}s/$ where:

m_n = the name of a mask for which Boolean equations are to be generated (reference masks).

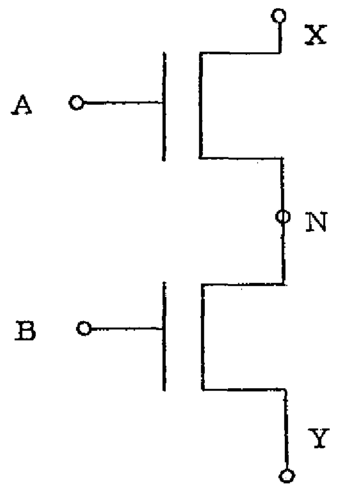
m_a-m_i = the names of the masks from which respectively each of i minterms may be derived for each item on the reference mask. If a mask name is preceded by the character "'" that mask's minterms will be inverted (NOTed).

s = 0/1 indicating a constant off or on state respectively.

Description:

The BOOL command process creates a list of Boolean equations which describes some interaction between masks, i.e., the status of some mask item as controlled by the status of other mask items. For example, the status of operation (ON/OFF) of a lightbulb may depend on the status of a switch and the status of voltage on the line; or the status of pipeline flow may depend on the status of a valve and status of fluid in the lines.

The equation generation process initiated by the BOOL command might best be described by a detailed example. A specific application for computer-aided-design is that of describing the ON/OFF logic state of each source (or drain) node in terms of the states of other nodes controlling it. The source (or drain) node state of a transistor is controlled by the state of its gate node and its drain (or source) node. The status of a node may be controlled by more than one transistor as shown below.



The status of node N is not only controlled by the states of nodes A and X, but also by the states of nodes B and Y. If the transistors are P-channel the Boolean expression for the status of node N is:

N = X AND NOT A OR Y AND NOT B,

if the transistors are N-channel the expression is:

N = X AND A OR Y AND B,

where each transistor contributes a minterm (e.g., X and A) to the expression. The equation generation process locates all the nodes controlling each node and generates the Boolean expression. The user need only specify the masks on which the minterms are located. Of course, there are nodes which have a constant On or OFF status, i.e., power and ground.

The equation generation process for CMOS requires the following masks to be identified:

-P guard band (PG)	-	where the primary sequence numbers are node identifications. Each node on this mask is grounded (0).
-N guard band (NG)	-	where the primary sequence numbers are node identifications. Each node on this mask is connected to power (1).
-P sources/drains (PSD) -P gates (PGM) -N sources/drains (NSD) -N gates (NGM)	-	where the primary sequence numbers are devices and secondary sequence numbers are nodes.

Thus, the BOOL command is expressed:

BOOL /u/PG:0/NG:1/PSD:PSD,'PGM/NSD:NSD,NGM/

where each item is the mask name of the node and mask name of its minterm components. The sign " ' " preceding the mask name PGM indicates that each minterm part located from that mask is to be a NOT state in the Boolean expression.

The Boolean expressions are printed out if an appropriate print option was specified in the OPTN command. The identifier data used to construct the expressions is written to the specified logical unit. Section 5.2 explains the format of this output.

4.4.3 LIST - List Cross-Reference Command

Format:

LIST /u/i_1/i_2/.../i_n/

u = 5/6/7/8/9 - logical unit to which the cross-reference list is to be written.

$/i_n/$ = nth item in the string, of the form $/m_n:m_a, m_b, \ldots, m_i/$ . .
where:

m_a-m_i - the names of up to three masks contribute to the cross-reference list.

Description:

The LIST process performs a cross-reference on the rectangle identifiers on several masks. Each item contains the name of the reference mask m_n and a list of associated masks m_a-m_i. The process creates a list where each entry (which corresponds to one unique primary/secondary identifier pair on mask m_n) contains the following:

- o word 1 - m_n rectangle primary identifier.
- o word 2 - m_n rectangle secondary identifier.
- o word 3 - m_a rectangle secondary identifier which has the same primary identifier (word 1).
- o word 4- word i+2 - same as word 3 for masks m_b-m_i.
- o last word - the item number n, (position in LIST command).

When there is more than one item specified in the command, the final list is created by combining and ordering all of the entries from each item's list.

The final cross-reference is written to the specified logical unit. Section 5.2 explains the details of this format. If an appropriate print option was specified in the OPTN command, the list will also be printed.

4.5 Dimensional Processing Commands

Several commands are provided for the user to direct dimensional computation process: areas, perimeters, and ranges. The result of any of these processes is a list of the numeric result of the particular computation.

The following pages describe each of the computations and the format of each command. In all cases, the resultant values are listed in scientific notation: a value times a power of ten. For simplicity only the value and exponent are listed.

4.5.1 AREA - Area Computation Command

Format:

AREA u,m,f

u = 5/6/7/8/9 - logical unit to which area data is to be written.

m = the name of the mask for which rectangle areas are to be computed.

f = ± integer factor ≠ 0 to be applied to the final result. Negative values imply division.

Description:

The result of the AREA computation is a list of the areas (length x width) of the rectangles on the mask, adjusted by the given factor. The list is printed in four columns (if an appropriate print option was specified in the option command): primary identifiers, secondary identifiers, area value, and exponent of ten multipliers. The list is written to the specified logical unit in the format described in Section 5.2.

4.5.2 PERI - Perimeter Computation Command

Format:

PERI u,m,f

u = 5/6/7/8/9 - logical unit to which perimeter data is to be written.

m = the name of the mask for which the rectangle perimeters are to be computed.

f = ± integer factor ≠ 0 to be applied to the final result. Negative values imply division.

Description:

The result of the PERI computation is a list of the perimeters of the rectangles on the mask, adjusted by the given factor. The list is printed in four columns (if an appropriate print option was specified in the option command): primary identifiers, secondary identifiers, perimeter values, exponent of ten multiplier. The list is written to the specified logical unit in the format described in Section 5.2.

4.5.3 PARE - Area and Perimeter Computation Command

Format:

PARE u, m, f_a, f_p

u = 5/6/7/8/9 - logical unit to which area and perimeter data is to be written.

m = the name of the mask for which the rectangle areas and perimeters are to be computed.

f_a = $\pm$ integer factor $\neq 0$ to be applied to the final area result. Negative values imply division.

f_p = $\pm$ integer factor $\neq 0$ to be applied to the final perimeter result. Negative values imply division.

Description:

The result of the PARE computation is simply a combination of the AREA and PERI computations. The list is printed in six columns (if an appropriate print option was specified in the option command): primary identifier, secondary identifier, area value, exponent of ten, perimeter value, exponent of ten. The list is written to the specified logical unit in the format described in Section 5.2.

4.5.4 RANG - Range Computation Command

Format:

RANG u, m, f_x, f_y

u = 5/6/7/8/9 - logical unit to which range data is to be written.

m = the name of the mask for which the rectangle x and y ranges are to be computed.

f_x = $\pm$ integer factor $\neq 0$ to be applied to the final x range result. Negative values imply division.

f_y = $\pm$ integer factor $\neq 0$ to be applied to the final y range result. Negative values imply division.

Description:

The result of the RANG computation is a list of the x ($x_2 - x_1$) and y ($y_2 - y_1$) ranges of the rectangle on the mask. The list is printed in six columns (if an appropriate print option was specified in the option command): primary identifier, secondary identifier, x range value, exponent of ten, y range value, exponent of ten. The list is written to the specified logical unit in the format described in Section 5.2.

4.6 Process Control Commands

Several commands have been provided to allow the user to control the manner in which the total command file is processed. The commands allow the user to have commands unconditionally skipped, or routing in the command file to occur based on a null mask condition.

With practice, a user will quickly learn to set up complex loops and branches in the command file where necessary. This capability also provides for reduction in run times where processes are only useful for non-null masks. The masks may be tested for a null condition and blocks of commands may be skipped if a null is found.

4.6.1 SKIP - Unconditional Routing Command

Format:

SKIP n

n = $\pm$ integer - number of commands to skip.

Description:

The SKIP command directs the program to move forward or backward in the command file. The command file is positioned to next read the nth command following (+) or preceding (-) the SKIP command.

4.6.2 IFNL - Null Condition Routing Command

Format:

IFNL m, n

m = name of the mask to be tested for null condition.

n = $\pm$ integer - number of commands to skip..

Description:

The IFNL commands direct the program to test for a null condition of the given mask, and if null, to move forward or backward in the command file. When this occurs the command file is positioned to next read the nth command following (+) or preceding (-) the IFNL command.

5. MAP OUTPUT

Several forms of data output are provided by MAP. The two major categories of output are informational output and mask data output. The following pages describe these output forms in detail.

5.1 Informational Printout

MAP informational output is sent entirely to the listing device. Informational output includes page headings, tables, command images, and messages documenting a particular MAP run. The degree of printout is optional as indicated in Table 4-2. The general form of the headings and tables is illustrated in the sample run listing given in Appendix C. Messages associated with abnormal processing conditions are listed in Table 5-1.

When any special debug executions are necessary, the user may specify a print option value of 4 or 5 on the OPTN command. Debug printout consists of single line strings of specific program variable values. This printout has been implemented for selected subroutines and generally occurs on entering or exiting the routine. The debug printout is divided into two categories: basic and extensive. The basic debug printout occurs at significant points in the processes. The extensive debug printout occurs in repeatedly-used routines and may produce a lengthy listing. A print option value of 4 yields only the basic form, a value of 5 yields both forms. Table 5-2 lists the subroutines where printout may occur and a description of the significance of the processing point where the line is printed. The reader may refer to the source listing for the names of the actual variables being printed.

5.2 Mask Data Output

The mask data output includes any type of data directly related to mask rectangle coordinates or identifiers.

The simplest form of this type is a printed listing of mask coordinates and identifiers. As previously mentioned, this is controlled by the print option of the OPTN command and the PRNT specifier of the SPEC command, for OPER commands.

Another output form for mask coordinates and identifiers is via the FILE and TEXT commands to secondary storage devices. The formats available are generally the same as for originally inputting mask data to MAP. Appendix B describes each format in detail. The masks involved in these commands will also be printed when the print option is greater than 2.

ABNORMAL PROCESSING CONDITION MESSAGES

MESSAGE	INTERPRETATION
# FATAL ERROR # CONDITION 1	The first command was not an OPTN command. The execution is terminated.
# FATAL ERROR # CONDITION 2	No MASK command was encountered immediately following the OPTN command. The execution is terminated.
# WARNING # CONDITION 3	A command has been encountered which does not contain a proper command type name as the first four characters. The command is ignored.
# WARNING # CONDITION 4	A command format error is present disabling complete interpretation of the command. An attempt is made to execute the command as it was interpreted.
# ERROR # CONDITION 5	A command format error is present preventing any meaningful interpretation of the command. The command is ignored.
# ERROR # CONDITION 6	No space is available on the mask file to write another intermediate or resultant mask. Execution of the command is terminated.
# WARNING # CONDITION 7	A mask needed for a process is not listed in the mask directory. The mask is assumed to be null. This condition may often occur for valid reasons since null masks are not listed in the mask directory. However, the user should check the spelling of the names given in the command whenever receiving the message.

Table 5-1

ABNORMAL PROCESSING CONDITION MESSAGES
(continued)

MESSAGE	INTERPRETATION
# WARNING # CONDITION 8	An intermediate result or a final resultant mask is null. This is a valid condition often encountered in MAP processing. When a resultant mask is null, this message is followed by a null mask message printed at the end of the command processing.
# ERROR # CONDITION 9	An invalid logical unit number has been specified in a command. Execution of the command is terminated.
# WARNING # CONDITION 10	An error has occurred in reading a command. An attempt is made to read the next command.
# WARNING # CONDITION 11	An error has occurred in writing mask data to an output device. An attempt is made to continue execution.
# WARNING # CONDITION 12	An error has occurred in reading mask data from an input device. An attempt is made to continue execution.

Table 5-1
(continued)

SUBROUTINE	DESCRIPTION
OP1	Processing path flags are printed preceding a single mask process step.
OP2	Processing path flags are printed preceding a mask multiplying process step.
OP3	Processing path flags are printed preceding a simple multi-pass processing step.
OP4	Processing path flags are printed preceding a complex multi-pass processing step. Pass flags are printed at the beginning of each pass.
SMASH	Mask parameters are printed preceding the first pass of input processing for an input mask.
I01	The input mode is printed preceding input of type -1 mask data. This is available only using a print option value of 5.
I01A	The input mode is printed preceding input of type 0 mask data. This is available only using a print option value of 5.
I01B	The input mode is printed preceding input of type 1 mask data. This is available only using a print option value of 5.
I01C	The input mode is printed preceding input of type 2 mask data. This is available only using a print option value of 5.
I01D	The input mode is printed preceding input of type 3 mask data. This is available only using a print option value of 5.
I01E	The input mode is printed preceding input of type 4 mask data. This is available only using a print option value of 5.

Table 5-2

DEBUG PRINTOUT
(Continued)

SUBROUTINE	DESCRIPTION
I04	Input and mask parameters are printed following the reading of a record from the mask file.
I05	Output and mask parameters are printed following the writing of a record to the mask file.
ORDER1	The entry data and ordering flags are printed following the entry of an item into an ordered list. This is available only using a print option value of 5.
ORDER2	Ordering flags are printed printed preceding the final process of sequencing an ordered list.
ORDER3	Ordering flags are printed preceding the final process of ordering large lists.
ORDER4	A pattern identifier is printed preceding the process of setting up an ordering priority pattern.
BOOK1	Mask parameters are printed following the location of a mask in the mask directory.
BOOK2	Mask parameters are printed following the deletion of an entry in the mask directory.
GEOM	Testing parameters are printed at the beginning of each step in a dimensional testing process.

Table 5-2
(Continued)

Another method of output for mask data is via the commands: BOOL, LIST, AREA, PERI, PARE, and RANG. These output forms are not strictly rectangle coordinates and identifiers, but are other forms of mask data which may be of use for more than casual informational purposes. In all cases, the results of these processes are printed when the print option is greater than 1. In addition, the resulting lists will be output to a secondary storage device. The records output to the device contain the same numeric data as is printed in type 0 output format. Refer to Appendix B for format details.

6. PROGRAM STRUCTURE

MAP FORTRAN code is in the form of a main program and a number of subprograms. The main program is organized into many small "inline routines." These inline routines are simple sections of code in the main program which are set apart by comments and specific statement label ranges. These inline routines represent specific processes; overall program flow, command processing, and operation forms.

Table 6-1 is a list of each of the inline routines and a brief functional description. Table 6-2 describes each of the actual FORTRAN subprograms. All of the subprograms are FORTRAN subroutines unless otherwise noted in the table. Further information regarding routine functions can be gained from the liberal comments in the MAP source listing. The details of the calling structure of these routines and subprograms are presented in Appendix D.

Although the core requirement for MAP is minimal, overlaying may be necessary on machines of very limited core resources. MAP is structured so that it may be readily overlayed. Figure 6-1 illustrates a recommended overlay structure. With this structure a maximum of about 65% of the program procedure can be resident at any time.

MAIN PROGRAM INLINE ROUTINE DESCRIPTIONS

NAME	DESCRIPTION
Program Flow Routines:	
INIT	Program initialization
ORTHO	Mask input and orthogonal refinement
COMMIE	Command interpretation and execution
FINIT	Normal program termination
Command and Processing Routines:	
COMM	User comment output
FILE	Coordinate storage
TEXT	Identifier storage
FREE	Mask storage release
OPER	Operation execution
SPEC	Operational specifications setup
TRAC	Nodal trace execution
BOOL	Boolean equation generation
LIST	Special list processing
AREA	Area computation
PERI	Perimeter computation
PARE	Area and perimeter computation
RANG	Range computation
SKIP	Unconditional command file repositioning
IFNL	Null mask condition command file repositioning

Table 6-1

MAIN PROGRAM INLINE ROUTINE DESCRIPTIONS
(continued)

NAME	DESCRIPTION
Operational Processing Routines:	
SAME	Equation
NGTV	Negation
EDGE	Edge extraction
EXPN	Expansion
PLUS	Addition
INTR	Intersection
NINT	Non-intersection extraction
EXOR	Exclusive OR
LINK1	Single mask linkage
LINK2	Double mask linkage
NLNK	Non-linkage extraction
TWIX1	Single mask spacing extraction
TWIX2	Double mask spacing extraction
SPIN	z-axis rotation
FLIP	x-axis and/or y-axis mirroring
PUSH	Offset
SCAL	Scaling
WNDW	Window extraction
PLAC	Cell placement

Table 6-1
(Continued)

SUBPROGRAM DESCRIPTIONS

NAME	DESCRIPTION
OP1	Single mask processing
OP2	Double mask processing with multiplexed input
OP3	Multiple pass double mask processing with repetitive secondary input
OP4	Multiple pass double mask processing with repetitive primary input
SMASH	Orthogonal smash processing
I01, I01A, I01B, I01C, I01D, I01E	User data input
I02	Command data input and output
I03	Command data decoding
I04	Binary mask data input
I05	Binary mask data output
I06A, I06B, I06C, I06D	User data output
I07	Normal user message printout
I08	Error message printout
ORDER1	Ordered list item entry
ORDER2	Ordered list sequencing
ORDER3	Large list ordering
ORDER4	Ordering priority pattern setup
ORDER5	Ordered identifier replacement

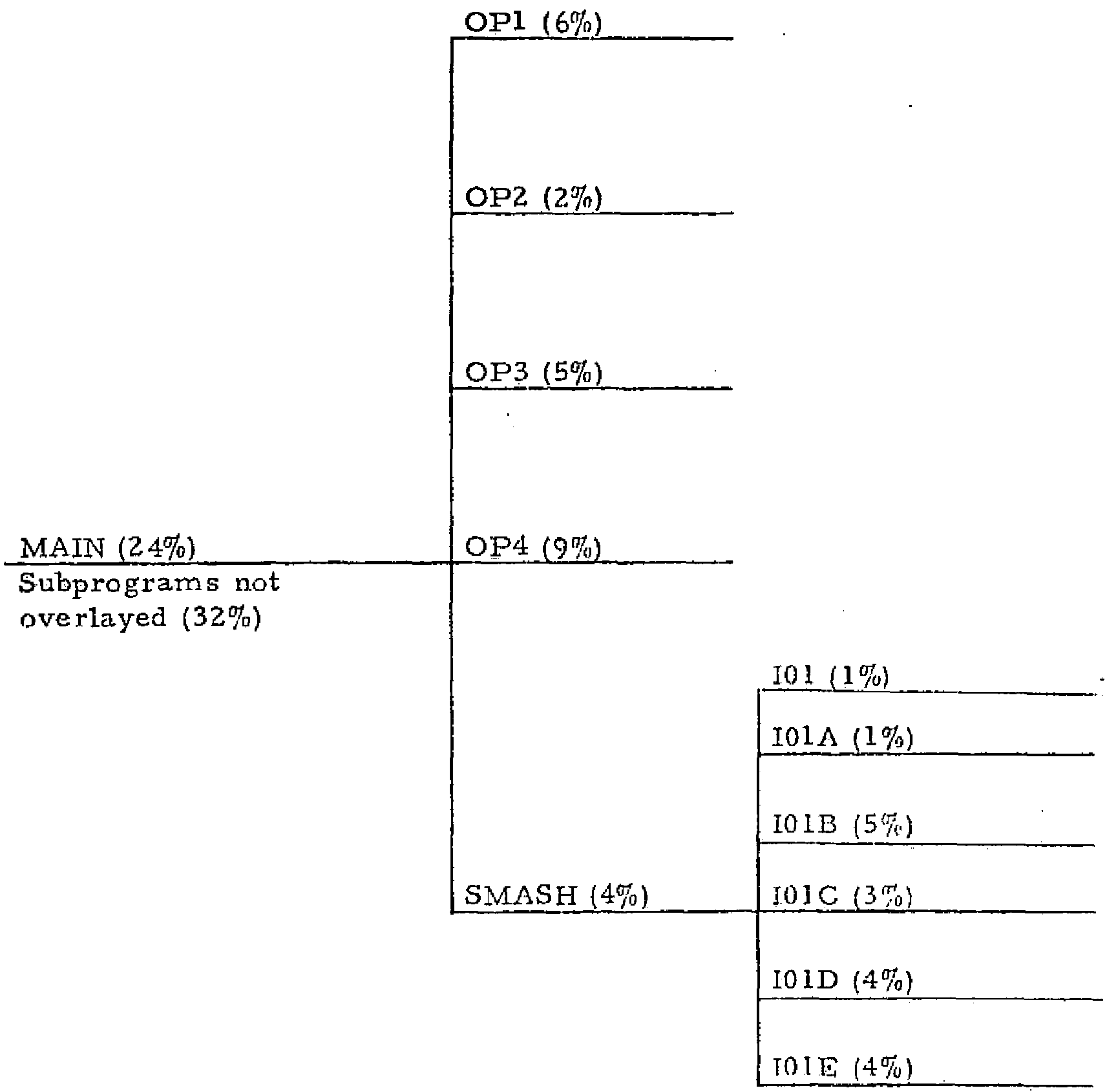
Table 6-2

SUBPROGRAM DESCRIPTIONS
(continued)

NAME	DESCRIPTION
BOOK1	Mask directory entry location
BOOK2	Mask directory entry deletion
GEOM	Minimum and maximum dimension testing
DEPEND	Isolation of machine dependent code
LOCATN	Mask file record address calculator (function)

Table 6-2
(continued)

RECOMMENDED OVERLAY STRUCTURE



Note: Percentages indicate approximate portions of the total procedure core requirement.

Figure 6-1

7. PROGRAM VARIABLES

At the outset of MAP development a great deal of consideration was given to the program variables. An attempt was made to choose meaningful scalar and array names. Appendix E presents a brief description of all of the program variables.

Most of the program variables are in blank common. The arrangement of common and the structure of the arrays was carefully designed to allow the program size to be modified by redimensioning the arrays.

The following discussion describes the arrangement of variables in common and methods of adjusting array dimensions. In addition, there is a brief description of the few variables whose values are set according to the characteristics of the computer.

7.1 Arrangement of Variables in Common

Table 7-1 illustrates the arrangement of variables in blank common. The variables have been positioned in categories.

The first category contains a group of scalar variables whose values are determined by program array sizes and machine characteristics, and must be initialized individually. These variables are all set to constant values in the subroutine DEPEND. They were placed in a group at the beginning so that the remainder of common could be initialized efficiently.

The intermediate portion of common contains general processing variables. The arrays in this portion will always remain of fixed length.

The last two categories of common variables contain all of the arrays which may be redimensioned to change the program size.

7.2 Adjustment of Array Dimensions

As previously mentioned, the arrays at the end of the common block may be redimensioned. The group of arrays shown as file directory arrays in Table 7-1 store vital information about program I/O. A portion of the arrays is devoted to a directory of the stored masks. Table 7-2 details the information stored in these arrays. All of the arrays have the same dimension, and the scalar DIREND is set (in subroutine DEPEND) to that dimension. This dimension limits the number of masks which can be stored at any time during a MAP run. It should be set at the maximum number of masks to be stored, plus three or four for intermediate scratch mask recordings, plus nine for the non-mask entries. In adjusting the size of the directory arrays, the programmer should keep in mind that available master mask file space must also be considered when determining

BLANK COMMON STRUCTURE

CATEGORY	VARIABLES
Version Variables	MACHIN, CHARAC, LARGE, WORDS, ITEMS, LEND, DIREND, SETEND, MU, FACT1, FACT2, FACT3, FACT4
List Pointers	BEG(8), END(8), LOC(9)
Ordered List Pointers	BEGO(3), ENDO(3), LOCO(3), ENTRY(6), START
Flags	OPTN1-OPTN6, MODE1-MODE8, MODE, PAT1-PAT5, SEQ1-SEQ10
Values	VAL1-VAL6, MAX1-MAX6, MIN1-MIN6, SPEC1-SPEC5
File Position Pointers	INP1, INP2, OUT1-OUT5, UNIT
List Segment Flags	FILE(8), STATUS(8), SEG1-SEG5, SEG
Alphanumeric and Command Image	B1, B4, DELIM(36), CARD(76), FIELD(34)
File Directory	NAME(DIREND), NUM1(DIREND), NUM2(DIREND), COUNT(DIREND), RECORD(DIREND)
Lists	SETUP(SETEND), LIST(I)

Table 7-1

DIRECTORY ARRAY'S USAGE

ARRAY	RANGE	USAGE DESCRIPTION
NAME(I)	I=1-4	Equivalent to scalars DESTIN, LAST, MASK, and TYPE.
	I=5-9	Name of last mask output to each logical unit I.
	I=10-DIREND	Name of mask stored in each file position I.
NUM1(I)	I=1-4	Equivalent to scalars CR, IN, LP, and AL.
	I=5-9	Running count of the number of masks output to each logical unit I.
	I=10-DIREND	Highest primary identifier on the mask stored in each file position I.
NUM2(I)	I=1-4	Equivalent to scalars LENGTH, PASS, SKIP, and STAT.
	I=5-9	Data type code of last mask output to each logical unit I.
	I=10-DIREND	Highest secondary identifier on the mask stored in each file position I.
COUNT(I)	I=1-4	Equivalent to scalars NEXT, TEST, ORD, and OR.
	I=5-9	Running count on number of records output to each logical unit I.
	I=10-DIREND	Starting record address of the mask stored in each file position I.
RECORD(I)	I=1-4	Used to store timer data.
	I=5-9	Number records written for the last mask output to each logical unit I.
	I=10-DIREND	Number of records for the mask stored in each file position I.

Table 7-2

the maximum number of masks to be stored during a run.

The arrays shown as lists in Table 7-1 are SETUP(I) and LIST(I). The SETUP array is used to stack processing codes for procedures requiring many steps. It is used to set up the course of action for the TRAC, BOOL, and LIST processes. SETUP(I) must be dimensioned to accommodate one entry for every mask name contained on any TRAC, BOOL, or LIST command record plus ten. The array is also used for other processes and must have a minimum dimension of 20 plus the maximum number of MIN and MAX specifiers used in any SPEC string. The scalar SETEND is set (in subroutine DEPEND) equal to the dimension of SETUP(I).

The array LIST(I) is the core area used to store mask data records during processing. The array may be divided up into "segments" as illustrated in Figure 7-1. The whole segments, first eight divisions, each contain exactly the number of words in a single mask record from the master file. These segments are used as input or output buffers as required by any particular process. They are also used to create ordered lists. The ordering process requires one third more space (i.e., eight words per rectangle instead of six words) during the creation of a list. When the list is sequenced it may then be output through a normal segment. Figure 7-1 illustrates the ordering configurations which may be used, each depending upon the number of segments required for input records.

As shown, the LIST array is divided into 8 2/3 segments plus 12 words for the small buffer at the end. To change the size of LIST(I), the number of rectangles per segment must be determined, multiplied by 6 (words per rectangle) and multiplied by 8 2/3. Twelve plus this value yields the dimension of LIST. There are several scalar values (set in subroutine DEPEND) which are associated with the dimension of LIST. ITEMS is set equal to the number of rectangles per segment (or record). WORDS is set to the number of words per segment (or record) and is simply 6 times ITEMS. Since the segment size must be equal to the size of a file record, the file must be accessed properly. The scalars FACT1-FACT4 are values used in a formula in the function LOCATN to compute a file address for any record.

A general rule for the most efficient dimensioning of these arrays is to first set the directory arrays and the SETUP array sizes as necessary. Then the LIST array should be dimensioned as high as core resources will allow. In general, doubling the size of the LIST array will reduce processing time by one fourth.

When any of these array dimensions are changed, the variable LEND must be reset (in subroutine DEPEND). LEND is a value used to zero part of blank common and is the number of words from BEG(1) through

LIST ARRAY USAGE

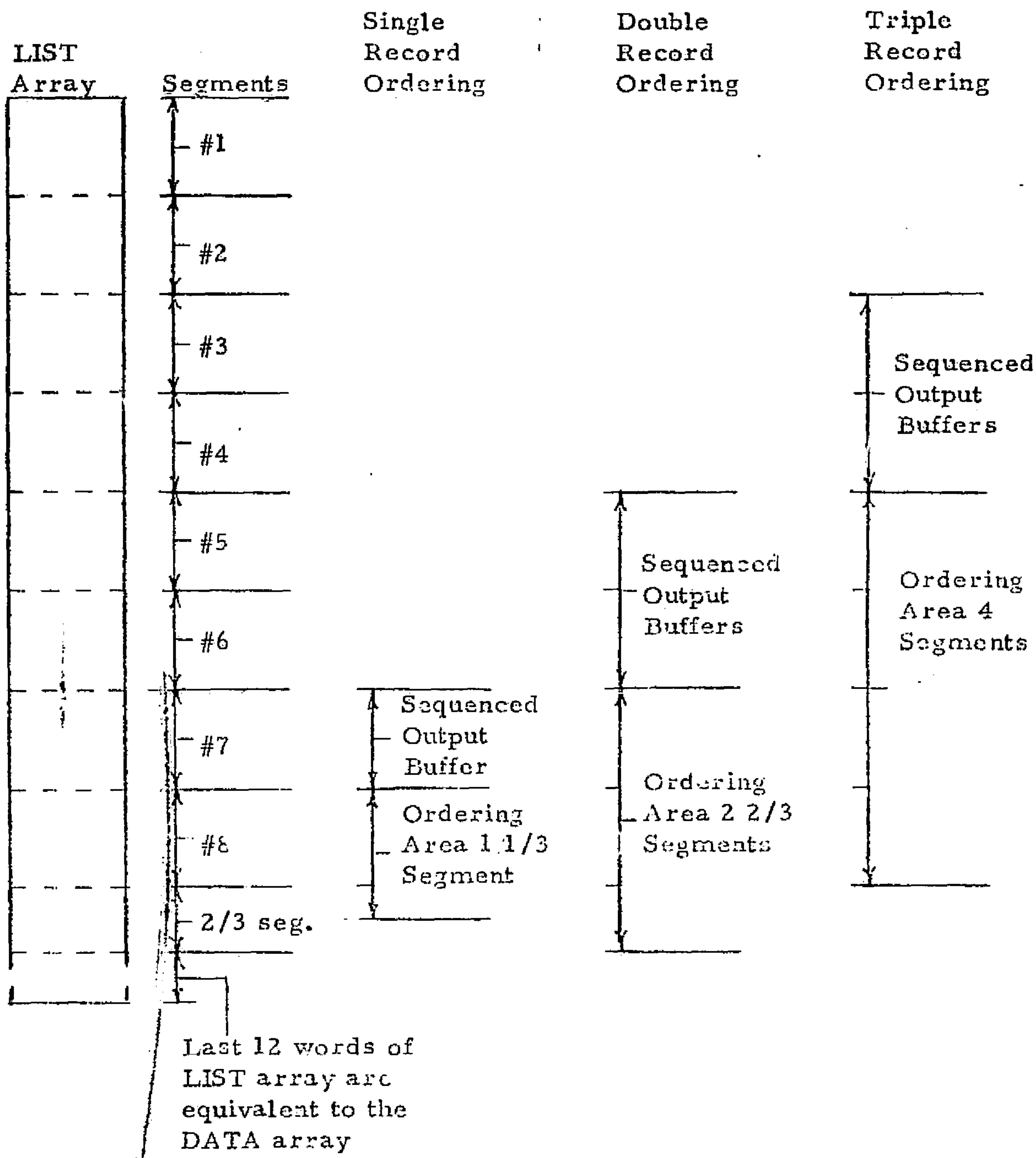


Figure 7-1

the last word in LIST. The following formula may be used to calculate the new value:

LEND = 272 + 5 x DIREND + SETEND + LIST array dimension,

for machines with single word REAL variables, or

LEND = 307 + 6 x DIREND + SETEND + LIST array dimension,

for machines with double word real variables.

7.3 Machine Dependent Variables

As previously mentioned, the first group of scalars in common have values that are determined by program array sizes and machine characteristics. The array size dependent values were discussed in the previous pages.

The machine dependent scalars are MACHIN, CHARAC, LARGE, and are set in subroutine DEPEND. MACHIN is set to the number of bits per word. CHARAC is set to the number of characters contained in a real variable. LARGE is set to the largest possible integer value.

The variables FACT1-FACT4, discussed in regard to the size of the LIST array, are also dependent upon the file addressing scheme of the computer. The variable LEND which is array size dependent also depends upon the word length of real variables as discussed in the previous sections.

APPENDIX A

JOB SETUP EXAMPLES

Table A-1 lists the logical unit assignments for executing MAP. Figures A-1, A-2, and A-3 illustrate the control commands for three typical MAP job setups.

LOGICAL UNIT ASSIGNMENTS

UNIT	DESCRIPTION
1	Command Input Unit (required). OPTN and MASK commands are always read from this unit.
2	Alternate Command Input Unit (optional). See alternate input command option, OPTN command.
3	Command Buffer Unit (required). This must be a sequential disk file.
4	Printed Outpu Unit (optional). See print option, OPTN command.
5-9	Mask Data Output (optional). See descriptions of FILE, TEXT, BOOL, LIST, AREA, PERI, PARE, and RANG commands.
10	Master Mask Unit (required). This must be a keyed or random access disk file depending on the computer.
11-n	Mask Data Input Units (at least one required). See MASK command description.

Table A-1

JOB SETUP EXAMPLE 1

```
!JOB
!LIMIT
!ASSIGN F:1,(DEVICE,CR)
!ASSIGN F:3,(FILE,TEMP),(OUTIN)
!ASSIGN F:4,(DEVICE,LP)
!ASSIGN F:10,(FILE,MASKS),(KEYED),(OUTIN)
!ASSIGN F:11,(FILE,DATA1),(IN)
!RUN (LMN,MAP)
!DATA
OPTN (alternate command input option off)
MASK (unit 11 specified)
(remainder of command deck)
!EOD
!FIN
```

This setup includes: all command input from card reader (F:1), printed output (F:4), and data input from a single file (F:11).

Figure A-1

JOB SETUP EXAMPLE 2

```
!JOB
!LIMIT
!ASSIGN F:1,(DEVICE,CR)
!ASSIGN F:2,(FILE,COMM),(IN)
!ASSIGN F:3,(FILE,TEMP),(OUTIN)
!ASSIGN F:4,(FILE,PRINT),(OUT),(SAVE)
!ASSIGN F:6,(FILE,RESULT),(OUT),(SAVE)
!ASSIGN F:10,(FILE,MASKS),(KEYED),(OUTIN)
!ASSIGN F:12,(FILE,DATA2),(IN)
!RUN (LMN,MAP)
!DATA
OPTN (alternate command input option on)
MASK (unit 12 specified)
!EOD
!FIN
```

This setup includes: OPTN and MASK command input from the card reader (F:1), remainder of commands input from a file (F:2), printer output stored on a file (F:4), special output stored on a file (F:6), and data input from a single file (F:12).

Figure A-2

JOB SETUP EXAMPLE 3

```
!JOB
!LIMIT
!ASSIGN F:1,(DEVICE,CR)
!ASSIGN F:3,(FILE,TEMP),(OUTIN)
!ASSIGN F:10,(FILE,MASKS),(KEYED),(OUTIN)
!ASSIGN F:11,(FILE,DATA1),(IN)
!ASSIGN F:12,(FILE,DATA2),(IN)
!ASSIGN F:13,(FILE,DATA3),(IN)
!RUN (LMN,MAP)
!DATA
OPTN  (alternate command input option off, zero print option)
MASK  (unit 11 specified)
MASK  (unit 12 specified)
MASK  (unit 13 specified)
(remainder of command deck)
!EOD
!FIN
```

This setup includes all commands input from card reader (F:1), no printer output, and data input from several files (F:11, F:12, F:13).

Figure A-3

APPENDIX B

USER FORMAT DESCRIPTIONS

The following discussion presents the specific details of the I/O formats for this particular version of MAP.

Data format types may be added or replaced in MAP as needed. All versions of MAP, however, must maintain the unique MAP format types -1 and 0.

Table B-1 lists the input and output MAP capabilities which apply to each format type.

Type -1 - MAP Mask File Format

Type -1 format is the format of the master mask file which is created during MAP executions. This is a random access binary file. The record length, the number of records allocated per mask, and the total length of the file are constants defined in the program. MAP will accept this format for input mask data. A file of this format may be externally constructed or saved from some previous MAP run.

Type 0 - MAP List Format

Type 0 format is the other unique MAP format. This format may be elected for FILE or TEXT output and is the standard non-printed output for BOOL, LIST, AREA, PERI, PARE, and RANG. Data is output as sets of records, one set per command. A set consists of:

- A starting record containing a single value written in I10 format. This value indicates the number of words per record for the remainder of the records in the set.
- Data records each containing the specified number of words (up to six) of data written in nI10 format.
- An end record of the same form as a data record except containing all -1 values. This serves as an end of set indicator.

MAP will accept type 0 format for input mask data.

MAP I/O FORMAT SUMMARY

I/O TYPE	INPUT CAPABILITIES	OUTPUT CAPABILITIES
Type -1: Mask File Format	Compatible	Automatically stored during execution.
Type 0: List Format	Compatible	Output via FILE, TEXT, BOOL, LIST, AREA, PERI, PARE, and RANG commands.
Type 1: PRF Format	All elements compatible except text	Output of blocks and text elements via FILE and TEXT commands.
Type 2: MANN Format	Compatible	Output via FILE command.
Type 3: AIDS Format	All elements compatible except text	Output of blocks and text elements via FILE and TEXT commands.
Type 4: Banning Cell Library Source Format	Compatible	Not applicable.

Table B-1

Type 1 - PRF Format

Type 1 format is the standard PRF format. It is accepted as a mask data input format and it may be used to output data via the FILE or TEXT command.

On input MAP will accept BLOCK, LINE, SHAPE, and COMPONENT data. TEXT data is ignored by MAP, and END COMPONENTS or END LEVEL is interpreted as the end of a mask.

On output, an END COMPONENTS record is written at the beginning of the file. The output for a FILE command is BLOCK data followed by an END LEVEL record. The TEXT command output is TEXT data where the rectangle identifiers are placed in the text string separated by a comma and given the lower left rectangle's coordinate as position points, all followed by an END LEVEL command.

Type 2 - MANN Format

Type 2 format is the standard MANN format. This format is acceptable for mask data input. MAP will accept rotated rectangle definitions and smash them into sets of orthogonal rectangles.

Type 2 format data can be output via the FILE command. The data output is strictly orthogonal rectangles.

Type 3 - AIDS Design File Format

Type 3 format is the standard Sigma 2 AIDS design file format. MAP will accept cell placement, shape, line, and block components as mask input data.

Output via the FILE command causes block elements to be written to a design file. Output via the TEXT command causes individual identifier digits to be stored in the design file as character elements. Type 3 format may be output to any design file for which the display parameter sector has been previously established. If a new file is to be started, the user must select the special new file option on the first FILE or TEXT command in the command set which is associated with the file. This causes a default display parameter sector to be written as the first record of the file.

Type 4 - Banning Cell Library Format

MAP will accept the Banning cell library source format as input. All cell shapes from a single level will be considered a single mask.

There is no provision in MAP for output in type 4 format.

APPENDIX C

MAP EXECUTION EXAMPLE

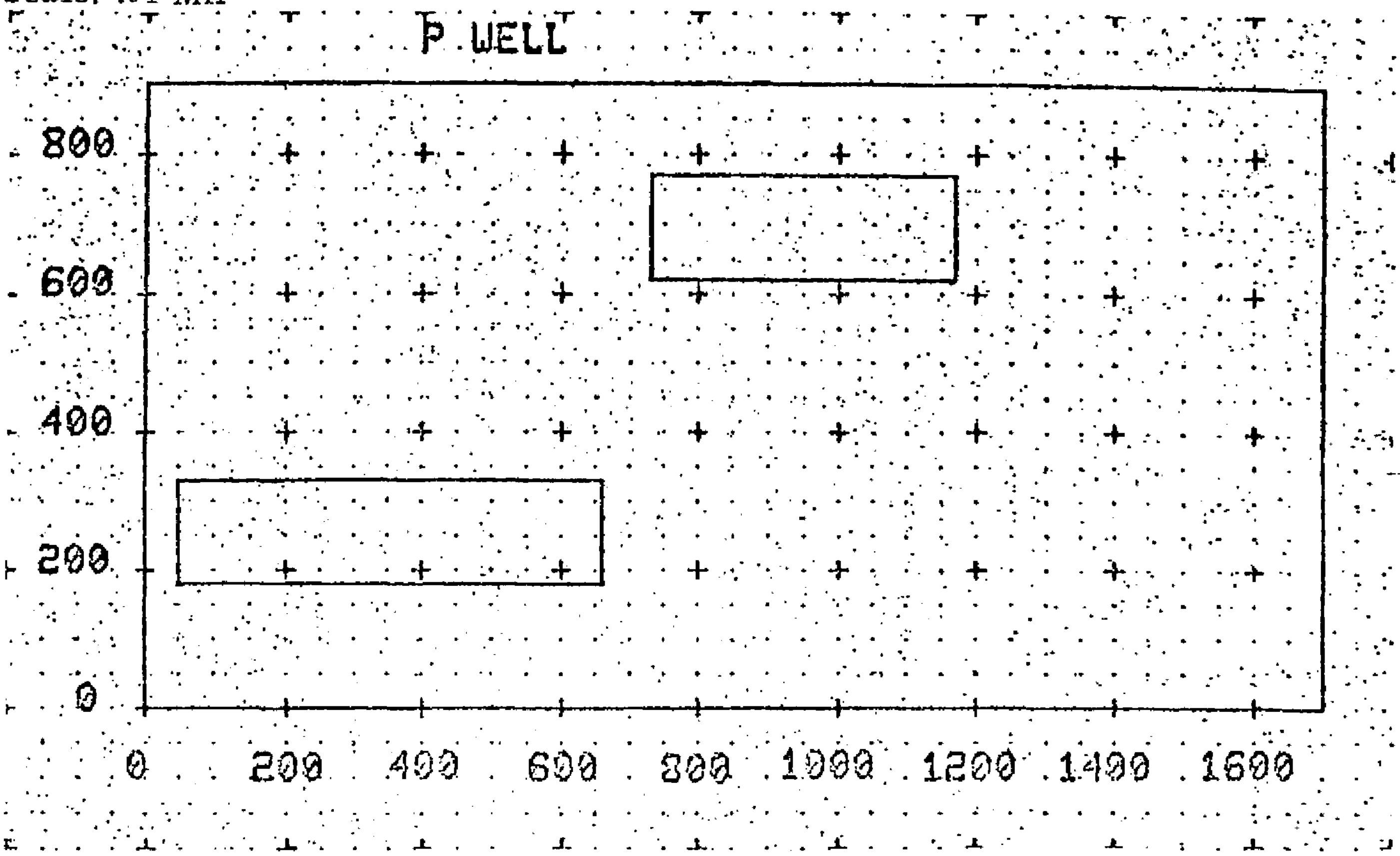
This appendix presents an example of a complete CMOS analysis execution. The command set used was specially developed for thorough analysis of CMOS masks. Among the functions performed are: nodal analysis, artwork verification, device identification, capacitance calculation, and equation generation.

The CMOS masks analyzed are very simple and should not be considered to be typical masks. Errors were designed into these masks to illustrate artwork verification capabilities of MAP. Figure C-1 illustrates the form of the original masks as displayed on a CRT. Figure C-2 illustrates the masks after the smashing operation performed by MAP following input of the original masks. Figure C-3 illustrates the N diffusion mask after the negation operation performed early in the run.

The remainder of this appendix contains a portion of the listing produced by the sample execution. The complete command set is presented in the first several pages. Due to the length of the remainder of the complete original listing, only representative portions of the command processing printout were selected for inclusion. The portions contained illustrate each of the analysis functions mentioned above.

ORIGINAL CMOS MASKS

Scale: .01 Mil



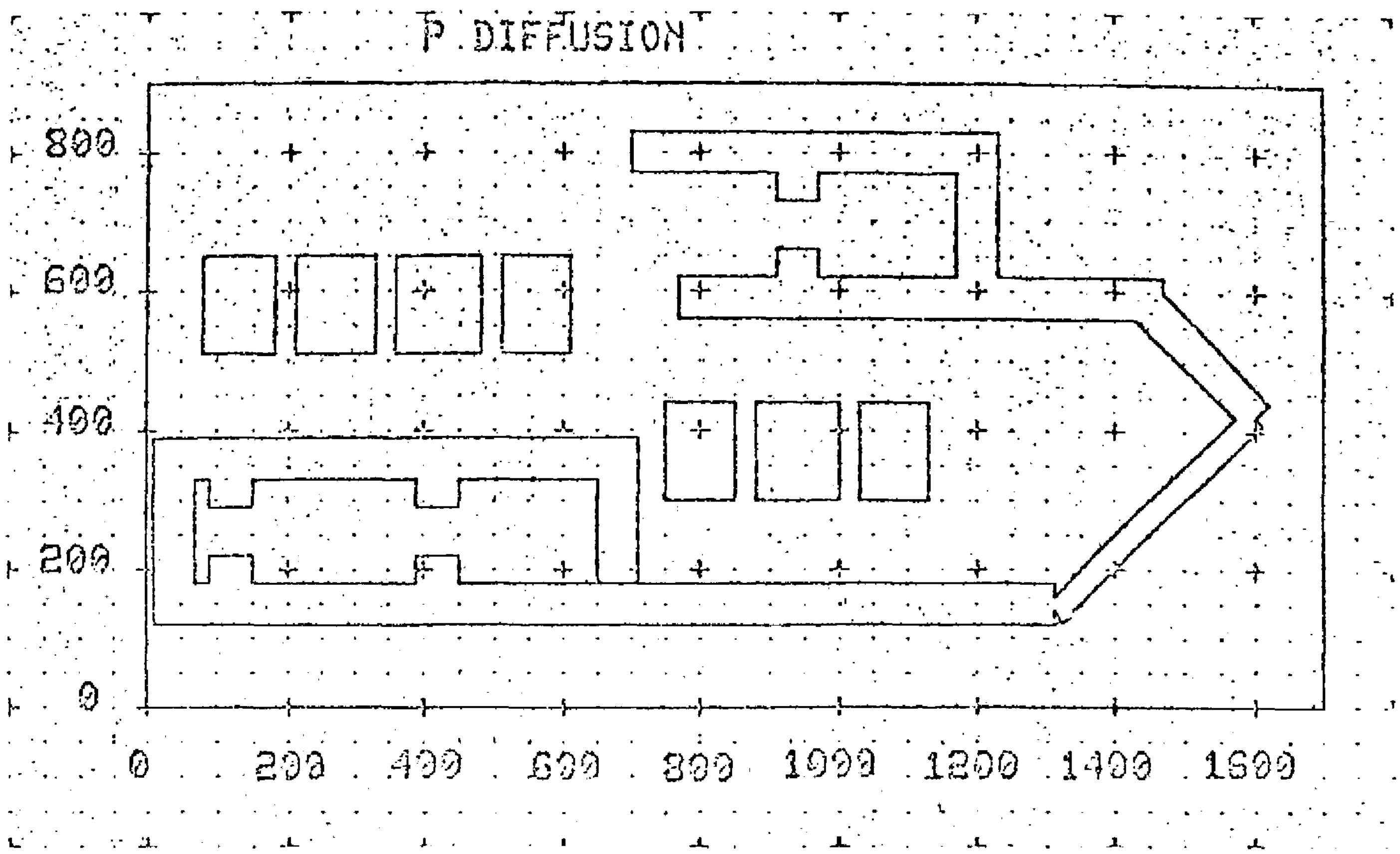
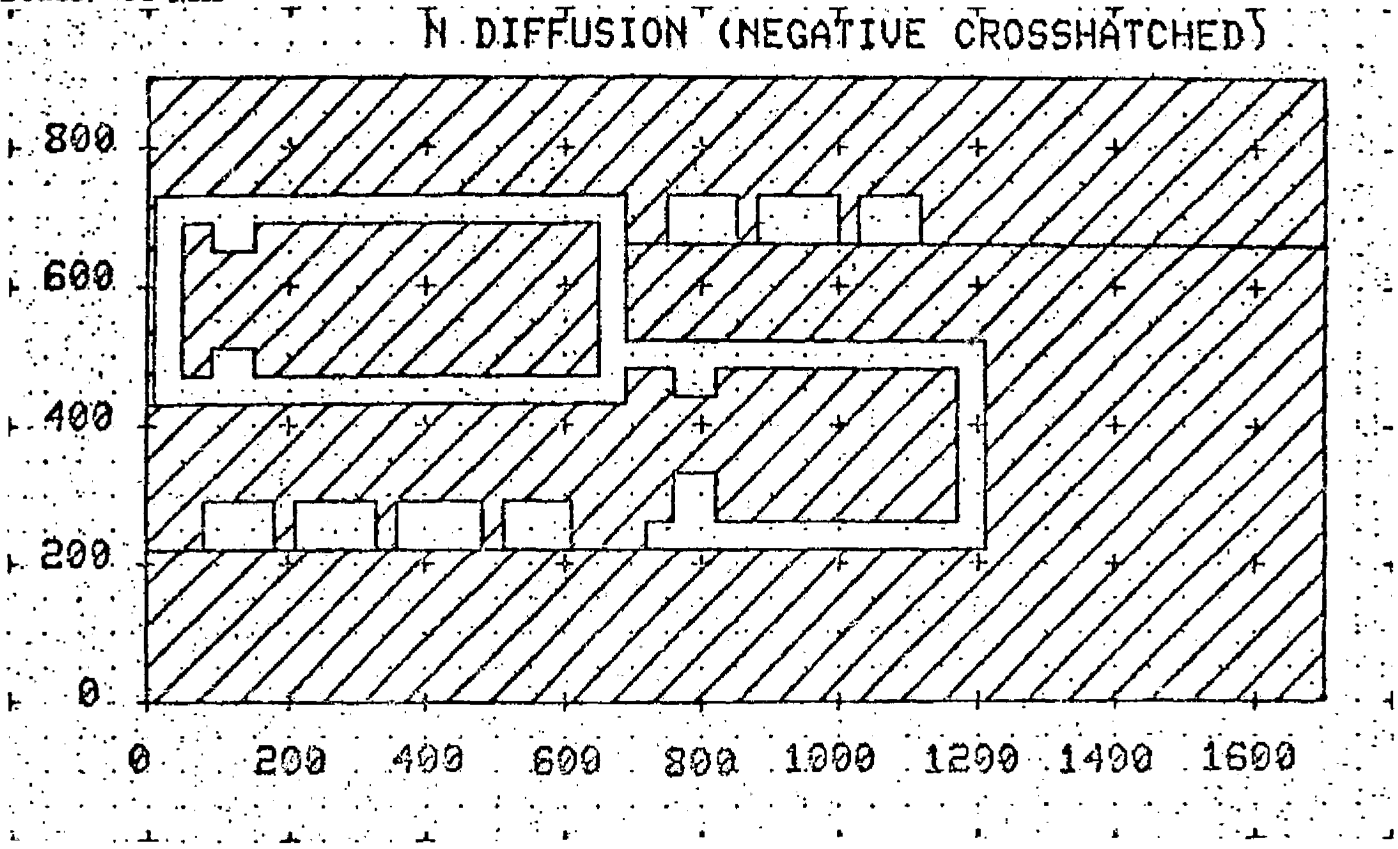


Figure C-1

Scale: .01 Mil



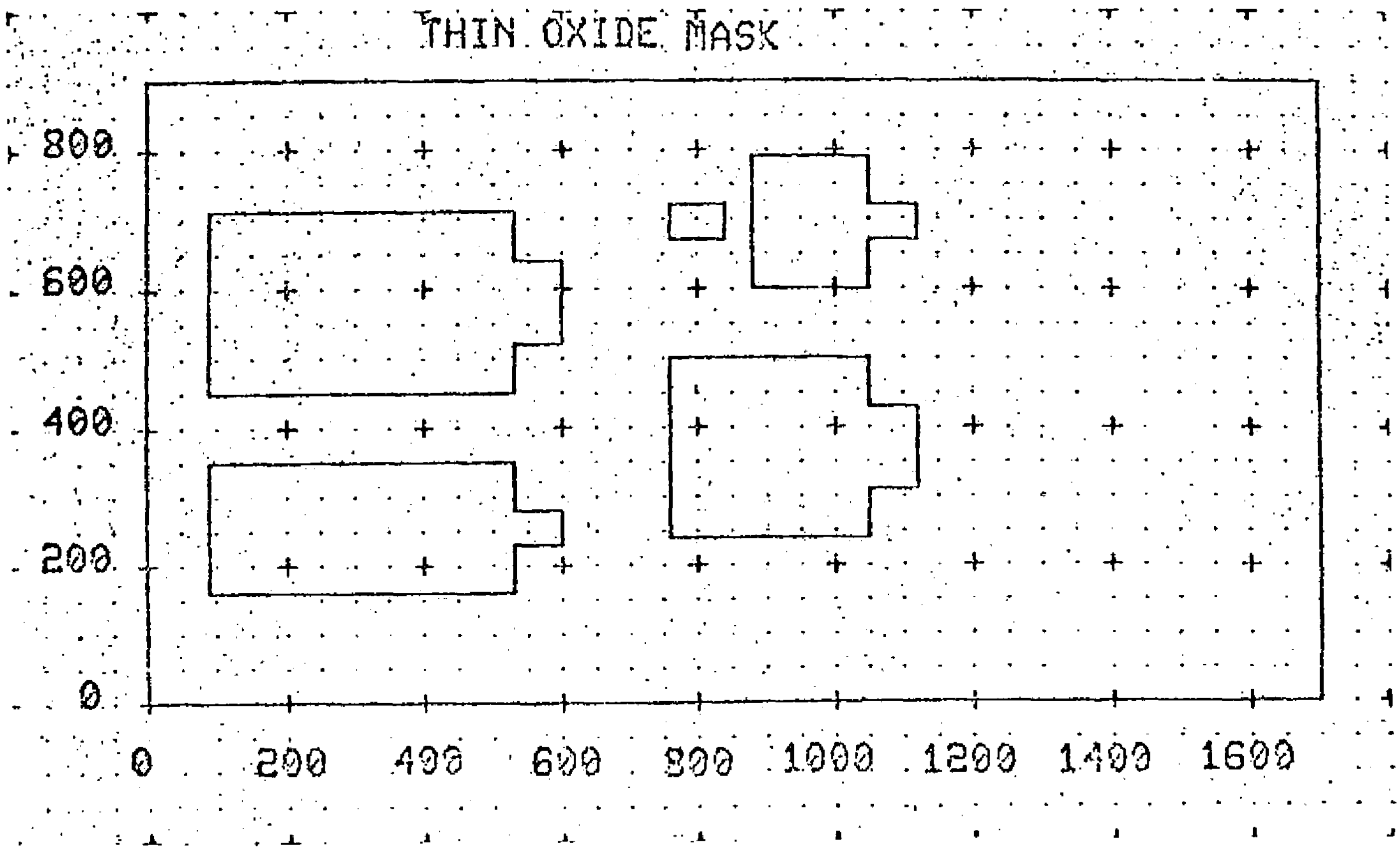
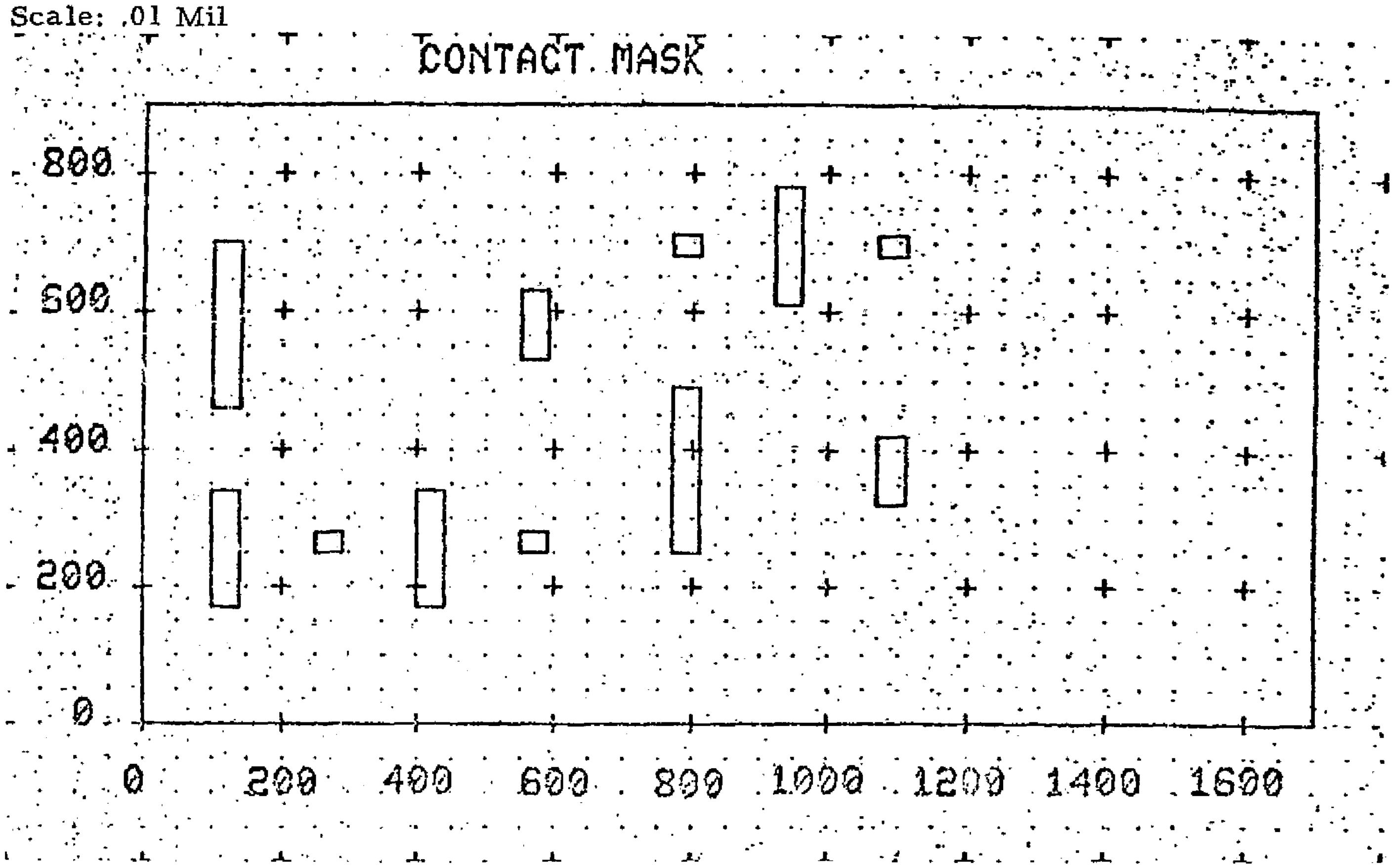


Figure C-1
(continued)

ORIGINAL CMOS MASKS
(continued)

Scale: .01 Mil



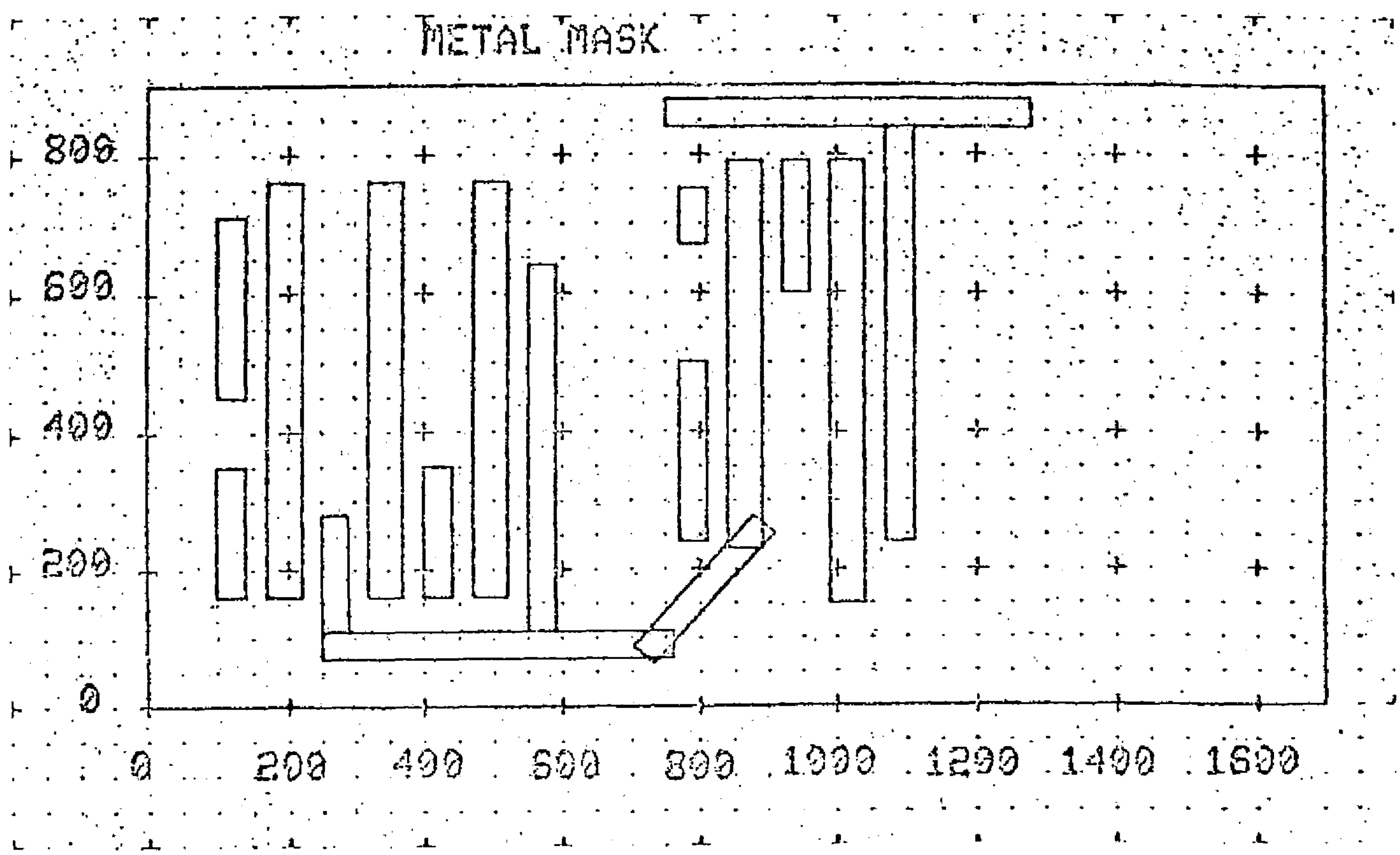
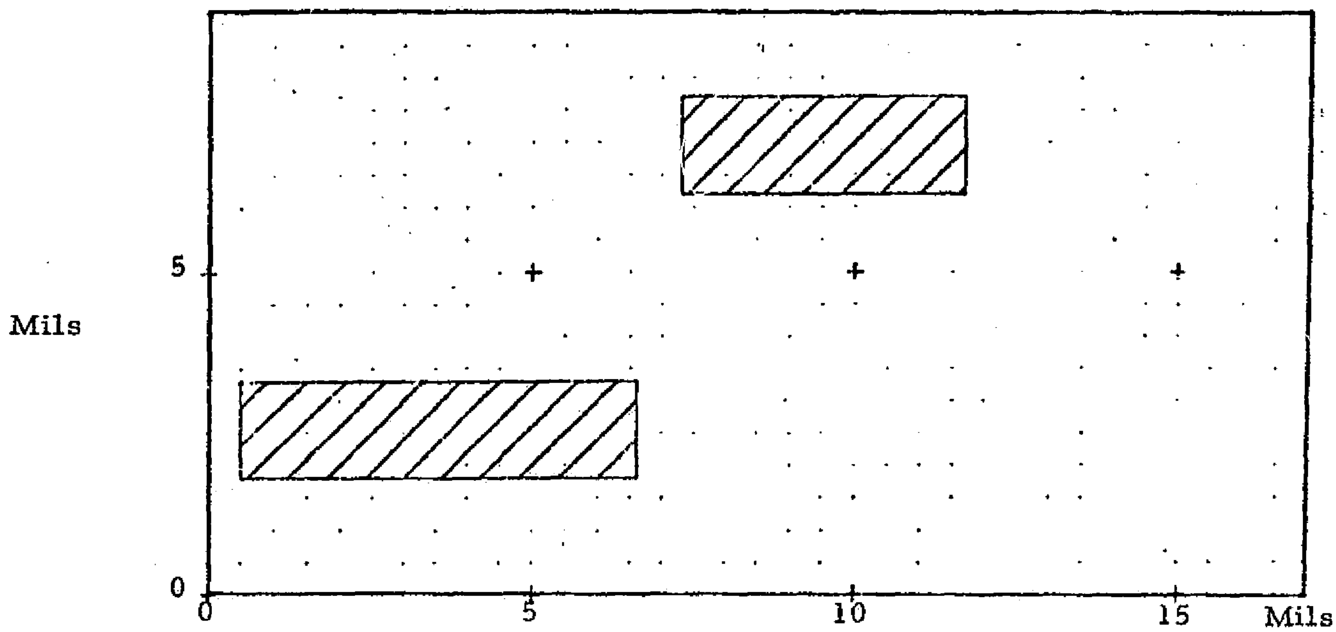


Figure C-1
(continued)

CMOS MASKS AFTER MAP SMASH

Mask W - Well Diffusion



Mask P - P Diffusion

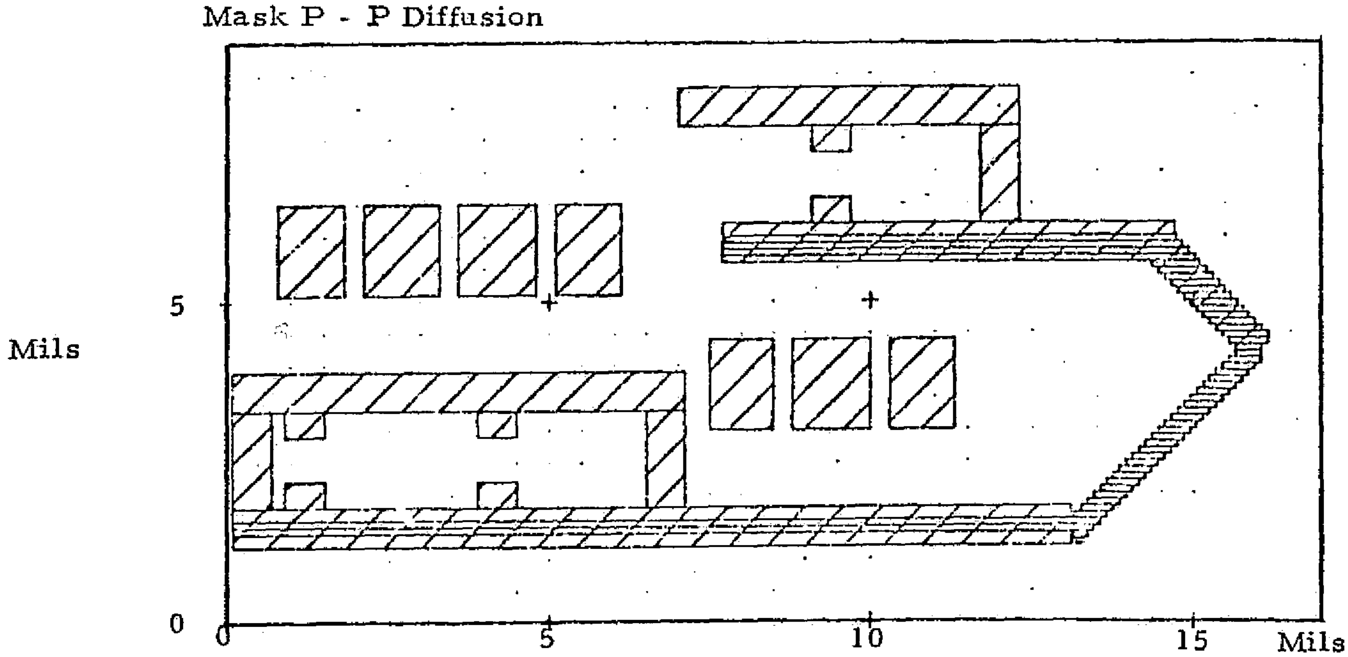
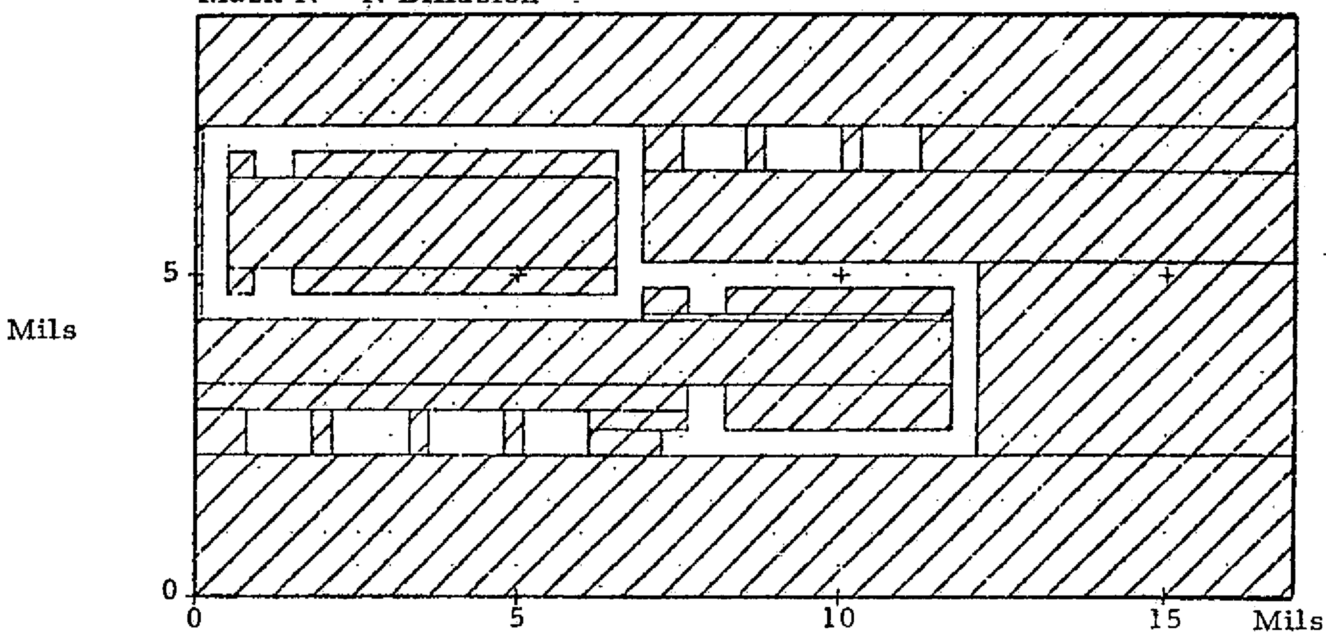


Figure C-2

CMOS MASKS AFTER MAP SMASH
(continued)

Mask N - N Diffusion



Mask T - Thin Oxide

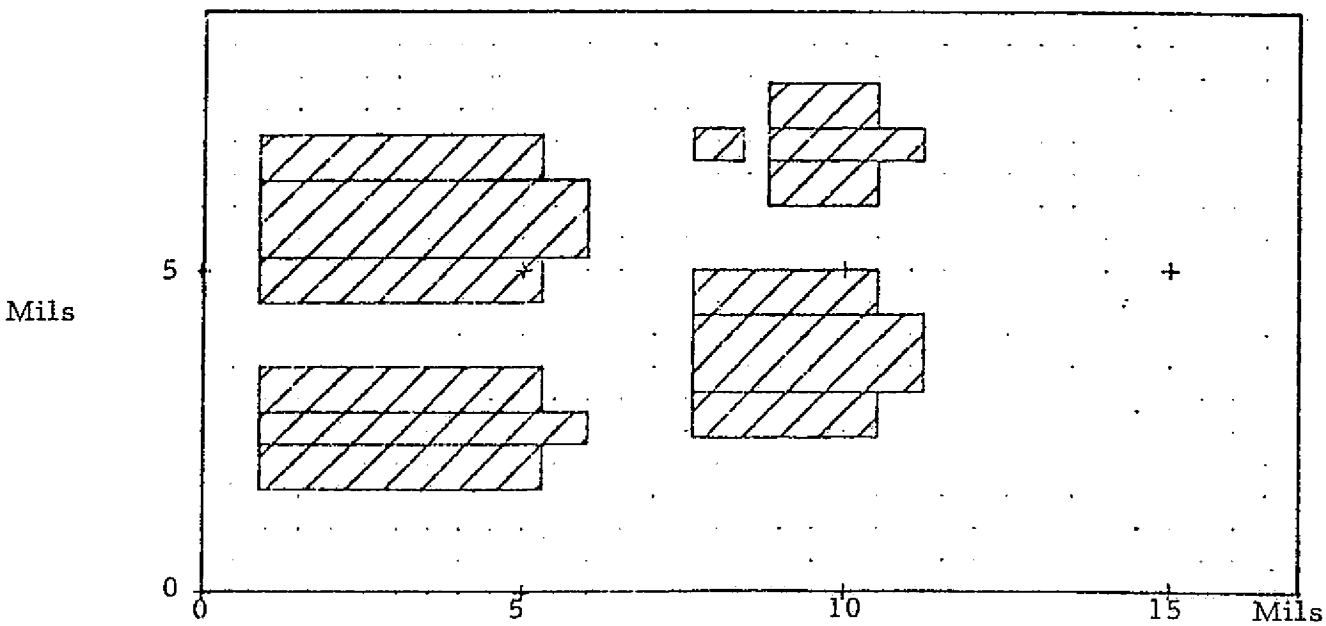
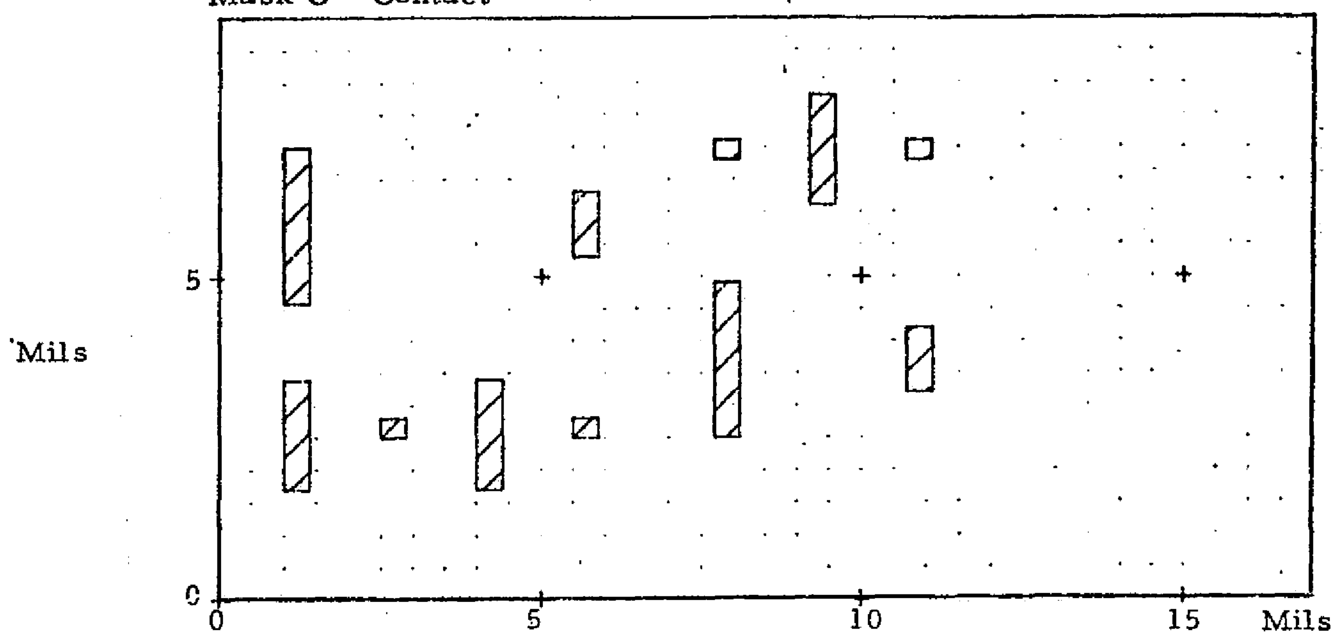


Figure C-2
(continued)

CMOS MASKS AFTER MAP SMASH
(continued)

Mask C - Contact



Mask M - Metallization

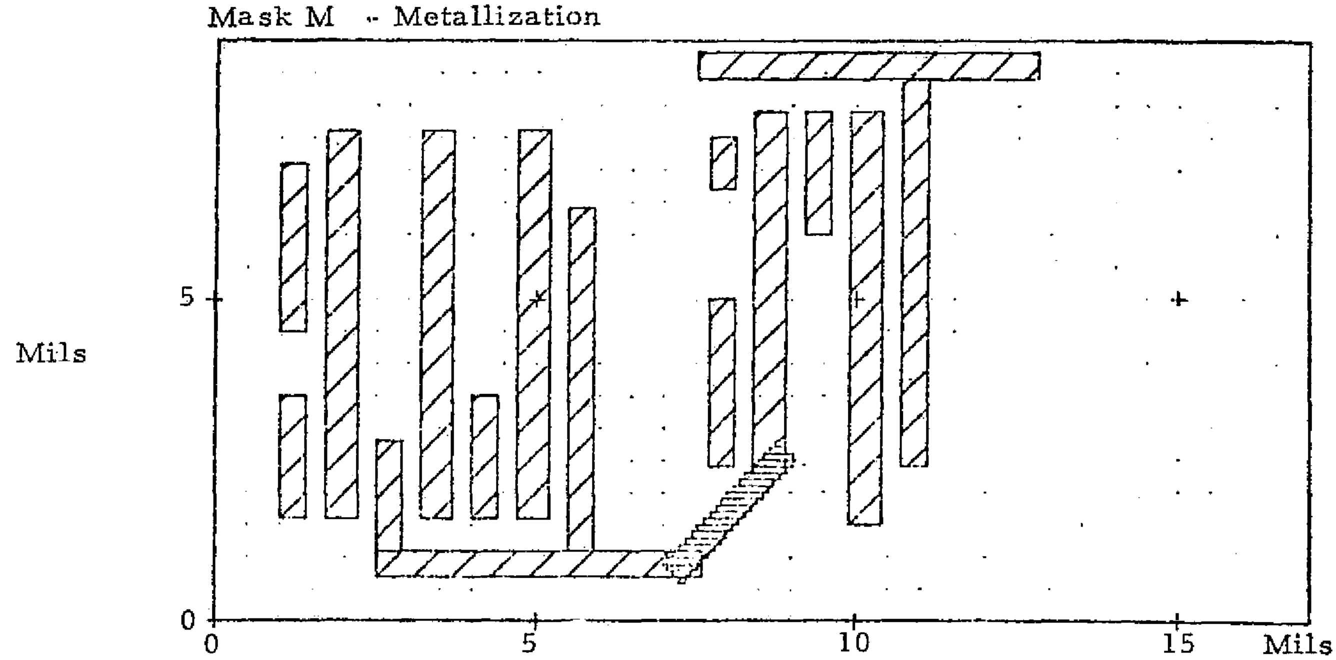


Figure C-2
(continued)

N MASK AFTER NEGATION

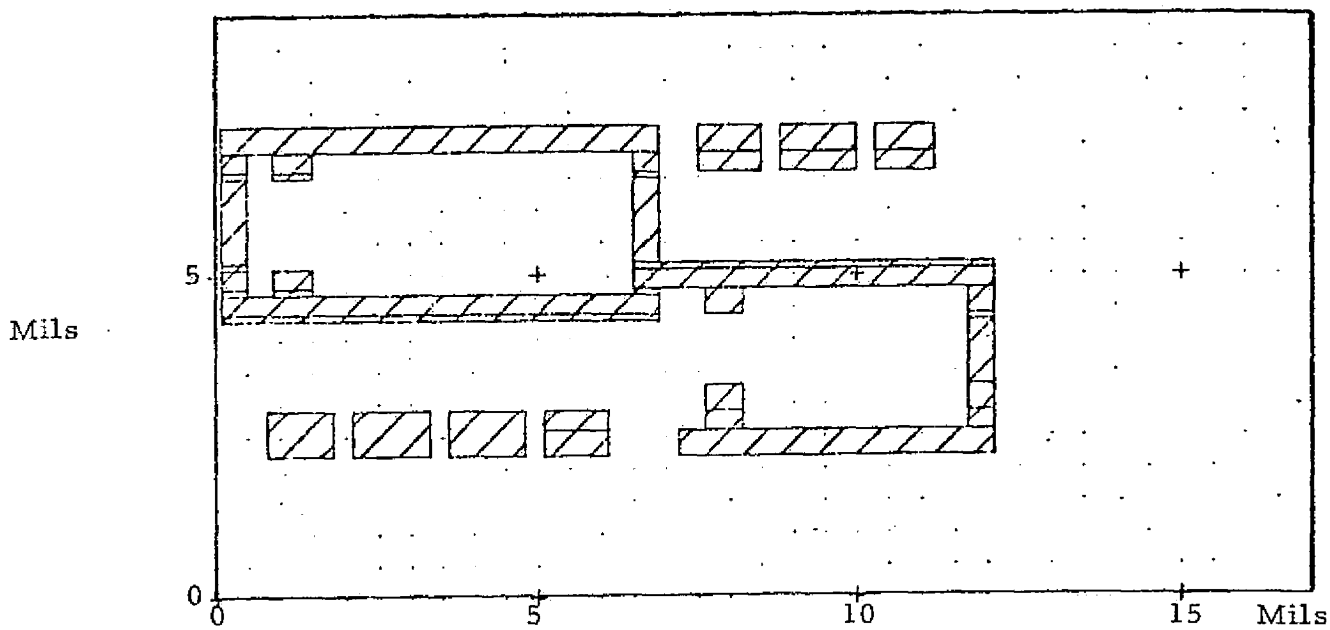


Figure C-3

* MAP *

```
OPTN  3,1,0,10,1,100                                                      <--------------------######
MASK  11,1,0,W,P,N,T,C,M                                                  <--------------------######
* DIRECTORY *
      FILES
      UNIT  SET  RECORDS  TOTAL  NAME  TYPE

      MASKS
      NAME  DIRECTORY  FILE  ADDRESS  RECORDS  IDENT # 1  IDENT # 2
      W        10        1         0      0        11          0
      P        11        2      3000      0        11          0
      N        12        3      6000      0        11          0
      T        13        4      9000      0        11          0
      C        14        5     12000      0        11          0
      M        15        6     15000      0        11          0
* TIME *                                                          CLOCK =    0
                                                                  DELTA =    0
------------------------------------------------------------------------------------------------

* COMMANDS *

COMM  UNIVERSAL CMOS MASK ANALYSIS PROCEDURE.  SCALE = 0.001 MIL    CC000001   <--------------------
IFNL  W,600                                                          CC000002   <--------------------
FILE  5,1,W                                                          CC000003   <--------------------
TEXT  5,1,3,W                                                        CC000004   <--------------------
IFNL  P,600                                                          CC000005   <--------------------
FILE  5,1,P                                                          CC000006   <--------------------
TEXT  5,1,3,P                                                        CC000007   <--------------------
IFNL  N,600                                                          CC000008   <--------------------
FILE  5,1,N                                                          CC000009   <--------------------
TEXT  5,1,3,N                                                        CC000010   <--------------------
IFNL  T,600                                                          CC000011   <--------------------
FILE  5,1,T                                                          CC000012   <--------------------
TEXT  5,1,3,T                                                        CC000013   <--------------------
IFNL  C,600                                                          CC000014   <--------------------
FILE  5,1,C                                                          CC000015   <--------------------
TEXT  5,1,3,C                                                        CC000016   <--------------------
IFNL  M,600                                                          CC000017   <--------------------
FILE  5,1,M                                                          CC000018   <--------------------
TEXT  5,1,3,M                                                        CC000019   <--------------------
COMM  NEGATION OF THE N MASK                                         CC000020   <--------------------
SPEC  R1@0,S1@1 M,R2@0,S2@# 3,I2# 0                                  CC000021   <--------------------
OPER  N = NGTV N                                                     CC000022   <--------------------
FILE  5,1,N                                                          CC000023   <--------------------
TEXT  5,1,3,N                                                        CC000024   <--------------------
COMM                                                                 CC000025   <--------------------
COMM      NODAL TRACE / RECTANGLE LINKAGE                            CC000026   <--------------------
COMM      ------------------------------                             CC000027   <--------------------
COMM                                                                 CC000028   <--------------------
COMM  NODAL TRACE                                                    CC000029   <--------------------
TRAC  /M:C/C:M,P,N/P:C/N:C/                                          CC000030   <--------------------
```

```
FILE  5,1,P                                                         CC000031    <-------------------
TEXT  5,1,3,P                                                       CC000032    <-------------------
FILE  5,1,N                                                         CC000033    <-------------------
TEXT  5,1,3,N                                                       CC000034    <-------------------
FILE  5,1,C                                                         CC000035    <-------------------
TEXT  5,1,3,C                                                       CC000036    <-------------------
FILE  5,1,M                                                         CC000037    <-------------------
TEXT  5,1,3,M                                                       CC000038    <-------------------
CØMM  LINKAGE ØF CØNNECTED WELL RECTANGLES AND RENUMBERING          CC000039    <-------------------
SPEC  R1Ø1,S1Ø1 N                                                   CC000040    <-------------------
ØPER  W = LINK W               LINE                                 CC000041    <-------------------
FILE  5,1,W                                                         CC000042    <-------------------
TEXT  5,1,3,W                                                       CC000043    <-------------------
CØMM  LINKAGE ØF CØNNECTED THIN ØXIDE RECTANGLES AND RENUMBERING    CC000044    <-------------------
SPEC  R1Ø1,S1Ø1 W                                                   CC000045    <-------------------
ØPER  T = LINK T               LINE                                 CC000046    <-------------------
FILE  5,1,T                                                         CC000047    <-------------------
TEXT  5,1,3,T                                                       CC000048    <-------------------
CØMM                                                                CC000049    <-------------------
CØMM       GENERAL ARTWØRK CHECKING - PHASE 1                       CC000050    <-------------------
CØMM       ----------------------------------                       CC000051    <-------------------
CØMM                                                                CC000052    <-------------------
CØMM  ERRØR:  MINIMUM N-P SPACING BETWEEN DIFFERENT NØDES = 0.4     CC000053    <-------------------
CØMM          MIL                                                   CC000054    <-------------------
SPEC  PRNT,MAXW=399                                                 CC000055    <-------------------
ØPER  NPSE = TWIX N,P          DIFF                                 CC000056    <-------------------
FILE  5,1,NPSE                                                      CC000057    <-------------------
TEXT  5,1,3,NPSE                                                    CC000058    <-------------------
FREE  NPSE                                                          CC000059    <-------------------
CØMM  ERRØR:  MINIMUM METAL LINE WIDTH = 0.4 MIL                    CC000060    <-------------------
SPEC  MAXX=399                                                      CC000061    <-------------------
ØPER  MLWX = SAME M                                                 CC000062    <-------------------
SPEC  MAXY=399                                                      CC000063    <-------------------
ØPER  MLWY = SAME M                                                 CC000064    <-------------------
SPEC  PRNT                                                          CC000065    <-------------------
ØPER  MLWE = PLUS MLWX,MLWY                                         CC000066    <-------------------
FREE  MLWX,MLWY                                                     CC000067    <-------------------
FILE  5,1,MLWE                                                      CC000068    <-------------------
TEXT  5,1,3,MLWE                                                    CC000069    <-------------------
FREE  MLWE                                                          CC000070    <-------------------
CØMM  ERRØR:  MINIMUM METAL-METAL SPACING = 0.3 MIL FØR LINES       CC000071    <-------------------
CØMM          < 10 MILS LØNG                                        CC000072    <-------------------
SPEC  PRNT,MAXW=299                                                 CC000073    <-------------------
ØPER  MMSE = TWIX M            DIFF                                 CC000074    <-------------------
FILE  5,1,MMSE                                                      CC000075    <-------------------
TEXT  5,1,3,MMSE                                                    CC000076    <-------------------
CØMM  ERRØR:  MINIMUM METAL-METAL SPACING = 0.4 MIL FØR LINES       CC000077    <-------------------
CØMM          > 10 MILS LØNG                                        CC000078    <-------------------
SPEC  PRNT,MINL=10001,MAXW=399                                      CC000079    <-------------------
ØPER  MMSE = TWIX M            DIFF                                 CC000080    <-------------------
FILE  5,1,MMSE                                                      CC000081    <-------------------
TEXT  5,1,3,MMSE                                                    CC000082    <-------------------
FREE  MMSE                                                          CC000083    <-------------------
CØMM  ERRØR: MINIMUM METAL/THIN ØXIDE SPACING = 0.2 MIL             CC000084    <-------------------
SPEC  PRNT,MAXW=199                                                 CC000085    <-------------------
```

```
ØPER  MTSE = TWIX M,T                                            CC000086   <--------------------
FILE  5,1,MTSE                                                   CC000087   <--------------------
TEXT  5,1,3,MTSE                                                 CC000088   <--------------------
FREE  MTSE                                                       CC000089   <--------------------
CØMM                                                             CC000090   <--------------------
CØMM  SEPARATIØN ØF P DIFFUSIØN INTØ ACTIVE ØR GUARD BAND AREAS  CC000091   <--------------------
ØPER  PA = NLNK P,W                LINE LINE                     CC000092   <--------------------
FILE  5,1,PA                                                     CC000093   <--------------------
TEXT  5,1,3,PA                                                   CC000094   <--------------------
ØPER  PG = NINT P,PA                                             CC000095   <--------------------
FILE  5,1,PG                                                     CC000096   <--------------------
TEXT  5,1,3,PG                                                   CC000097   <--------------------
CØMM  SEPARATIØN ØF N DIFFUSIØN INTØ ACTIVE ØR GUARD BAND AREAS  CC000098   <--------------------
ØPER  NA = INTR N,W                                              CC000099   <--------------------
FILE  5,1,NA                                                     CC000100   <--------------------
TEXT  5,1,3,NA                                                   CC000101   <--------------------
ØPER  NG = NINT N,NA                                             CC000102   <--------------------
IFNL  NG,10                                                      CC000103   <--------------------
FILE  5,1,NG                                                     CC000104   <--------------------
TEXT  5,1,3,NG                                                   CC000105   <--------------------
CØMM                                                             CC000106   <--------------------
CØMM  ERRØR:  MINIMUM N GUARD BAND WIDTH = 0.4 MIL               CC000107   <--------------------
SPEC  PRNT,MAXA=399                                              CC000108   <--------------------
ØPER  NGWE = SAME NG                                             CC000109   <--------------------
FILE  5,1,NGWE                                                   CC000110   <--------------------
TEXT  5,1,3,NGWE                                                 CC000111   <--------------------
FREE  NGWE                                                       CC000112   <--------------------
IFNL  PG,25                                                      CC000113   <--------------------
CØMM  ERRØR:  MINIMUM P GUARD BAND WIDTH ØUTSIDE ØF WELL = 0.6 MIL  CC000114   <--------------------
SPEC  PRNT,MAXA=599                                              CC000115   <--------------------
ØPER  PGWE = NINT PG,W                                           CC000116   <--------------------
FILE  5,1,PGWE                                                   CC000117   <--------------------
TEXT  5,1,3,PGWE                                                 CC000118   <--------------------
FREE  PGWE                                                       CC000119   <--------------------
CØMM  ERRØR:  WELL PERIMETER MUST MEET ØR ØVERLAP P GUARD BAND   CC000120   <--------------------
ØPER  WPER = EDGE W                SAM1                          CC000121   <--------------------
ØPER  WPER = EXPN WPER             5,5                           CC000122   <--------------------
SPEC  PRNT,MINA=10                                               CC000123   <--------------------
ØPER  WPER = NINT WPER,PG                                        CC000124   <--------------------
FILE  5,1,WPER                                                   CC000125   <--------------------
TEXT  5,1,3,WPER                                                 CC000126   <--------------------
FREE  WPER                                                       CC000127   <--------------------
IFNL  NG,10                                                      CC000128   <--------------------
CØMM  ERRØR:  N GUARD BAND MAY NØT INTERSECT P GUARD BAND, MAY   CC000129   <--------------------
CØMM          INDICATE AN ERRØR IN THE SEPARATIØN ØF N ACTIVE    CC000130   <--------------------
CØMM          AND N GUARD BAND AREAS DUE TØ IMPRØPER EMBEDDING   CC000131   <--------------------
CØMM          ØF N ACTIVE WITHIN THE WELL                        CC000132   <--------------------
SPEC  PRNT                                                       CC000133   <--------------------
ØPER  NWEE = INTR NG,PG                                          CC000134   <--------------------
FILE  5,1,NWEE                                                   CC000135   <--------------------
TEXT  5,1,3,NWEE                                                 CC000136   <--------------------
FREE  NWEE                                                       CC000137   <--------------------
IFNL  PA,118                                                     CC000138   <--------------------
CØMM  ERRØR:  N GUARD BAND MUST TØTALLY SURRØUND ALL P ACTIVE    CC000139   <--------------------
CØMM          AREAS                                              CC000140   <--------------------
```

```
ØPER  VNG = VTWX NG          SAME                          CC000141   <--------------------
ØPER  VNG = LINK VNG,PA      LINE,LINE                     CC000142   <--------------------
ØPER  HNG = HTWX NG          SAME                          CC000143   <--------------------
ØPER  HNG = LINK HNG,PA      LINE,LINE                     CC000144   <--------------------
SPEC  PRNT                                                 CC000145   <--------------------
ØPER  SURE = EXØR HNG,VNG                                  CC000146   <--------------------
FILE  5,1,SURE                                             CC000147   <--------------------
TEXT  5,1,3,SURE                                           CC000148   <--------------------
FREE  VNG,HNG,SURE                                         CC000149   <--------------------
CØMM                                                       CC000150   <--------------------
CØMM       P TRANSISTØR IDENTIFICATIØN / CHECKING          CC000151   <--------------------
CØMM       ------------------------------------            CC000152   <--------------------
CØMM                                                       CC000153   <--------------------
CØMM  LØCATIØN ØF PRØSPECTIVE P CHANNELS, ARBITRARILY ASSUMING    CC000154   <--------------------
CØMM  THAT ANY CHANNEL UP TØ 1 MIL LØNG WITH AT LEAST 0.1 MIL     CC000155   <--------------------
CØMM  METAL AND THIN ØXIDE WILL PERFØRM SØMEWHAT LIKE A TRANSISTØR CC000156   <--------------------
SPEC  MAXW=1000,R100,S10# 1,R200,S20# 1                    CC000157   <--------------------
ØPER  PPC = TWIX PA          DIFF                          CC000158   <--------------------
IFNL  PPC,97                                               CC000159   <--------------------
SPEC  MINA=100                                             CC000160   <--------------------
ØPER  PC = INTR PPC,T                                      CC000161   <--------------------
SPEC  MINA=100                                             CC000162   <--------------------
ØPER  PC = INTR PC,M                                       CC000163   <--------------------
ØPER  QPC = LINK PPC,PC      NØNE,AREA                     CC000164   <--------------------
SPEC  R101,S10# 1                                          CC000165   <--------------------
ØPER  QPC = LINK QPC         LINE                          CC000166   <--------------------
FILE  5,1,QPC                                              CC000167   <--------------------
TEXT  5,1,3,QPC                                            CC000168   <--------------------
FREE  PPC,PC                                               CC000169   <--------------------
CØMM                                                       CC000170   <--------------------
CØMM  ERRØR:  MINIMUM P CHANNEL LENGTH = 0.3 MIL, WIDTH = 0.7 MIL CC000171   <--------------------
SPEC  MAXW=299,MAXL=699                                    CC000172   <--------------------
ØPER  PCE = TWIX PA          DIFF                          CC000173   <--------------------
IFNL  PCE,5                                                CC000174   <--------------------
SPEC  PRNT                                                 CC000175   <--------------------
ØPER  PCE = LINK PCE,QPC     NØNE,AREA                     CC000176   <--------------------
FILE  5,1,PCE                                              CC000177   <--------------------
TEXT  5,1,3,PCE                                            CC000178   <--------------------
CØMM  ERRØR:  MINIMUM THIN ØXIDE WIDTH ØUTSIDE EDGES ØF P  CC000179   <--------------------
CØMM          CHANNELS = 0.2 MIL                           CC000180   <--------------------
ØPER  PCE = EXPN QPC         200,200                       CC000181   <--------------------
IFNL  PCE,5                                                CC000182   <--------------------
SPEC  PRNT                                                 CC000183   <--------------------
ØPER  PCE = NINT PCE,T                                     CC000184   <--------------------
FILE  5,1,PCE                                              CC000185   <--------------------
TEXT  5,1,3,PCE                                            CC000186   <--------------------
CØMM  ERRØR:  MINIMUM METAL WIDTH ØUTSIDE EDGES ØF P CHANNELS    CC000187   <--------------------
CØMM          = 0.1 MIL                                    CC000188   <--------------------
ØPER  PCE = EXPN QPC         100,100                       CC000189   <--------------------
IFNL  PCE,5                                                CC000190   <--------------------
SPEC  PRNT                                                 CC000191   <--------------------
ØPER  PCE = NINT PCE,M                                     CC000192   <--------------------
FILE  5,1,PCE                                              CC000193   <--------------------
TEXT  5,1,3,PCE                                            CC000194   <--------------------
FREE  PCE                                                  CC000195   <--------------------
```

```
CØMM                                                              CC000196   <-------------------
CØMM  LØCATIØN ØF P SØURCES AND DRAINS                            CC000197   <-------------------
SPEC  R120,S1Ø# 1                                                 CC000198   <-------------------
ØPER  PSD = SAME PA                                               CC000199   <-------------------
ØPER  PSD = LINK PSD,ØPC        LINE,LINE                         CC000200   <-------------------
ØPER  PSD = INTR PSD,PA                                           CC000201   <-------------------
FILE  5,1,PSD                                                     CC000202   <-------------------
TEXT  5,1,3,PSD                                                   CC000203   <-------------------
CØMM                                                              CC000204   <-------------------
CØMM  ERRØR:  P SØURCE AND DRAIN MINIMUM WIDTH = 0.1 MIL          CC000205   <-------------------
SPEC  PRNT,MAXA=99                                                CC000206   <-------------------
ØPER  PSDE = SAME PSD                                             CC000207   <-------------------
FILE  5,1,PSDE                                                    CC000208   <-------------------
TEXT  5,1,3,PSDE                                                  CC000209   <-------------------
CØMM  ERRØR:  MINIMUM SEPARATIØN ØF THIN ØXIDE TØ ØUTSIDE P       CC000210   <-------------------
CØMM          SØURCE AND DRAIN EDGES = 0.1 MIL                    CC000211   <-------------------
ØPER  PSDE = EXPN PSD           5,5                               CC000212   <-------------------
IFNL  PSDE,5                                                      CC000213   <-------------------
SPEC  PRNT,MAXA=104                                               CC000214   <-------------------
ØPER  PSDE = NINT PSDE,T                                          CC000215   <-------------------
FILE  5,1,PSDE                                                    CC000216   <-------------------
TEXT  5,1,3,PSDE                                                  CC000217   <-------------------
FREE  PSDE                                                        CC000218   <-------------------
CØMM                                                              CC000219   <-------------------
CØMM  LØCATIØN ØF P CHANNEL GATE METAL                            CC000220   <-------------------
SPEC  R120,S1Ø# 1                                                 CC000221   <-------------------
ØPER  PGM = SAME M                                                CC000222   <-------------------
ØPER  PGM = LINK PGM,ØPC        LINE,AREA                         CC000223   <-------------------
ØPER  PGM = INTR PGM,M                                            CC000224   <-------------------
FILE  5,1,PGM                                                     CC000225   <-------------------
TEXT  5,1,3,PGM                                                   CC000226   <-------------------
CØMM                                                              CC000227   <-------------------
CØMM  ERRØR:  MINIMUM P CHANNEL GATE METAL WIDTH = 0.5 MIL, A     CC000228   <-------------------
CØMM          REDUNDANT TEST SINCE CHANNEL WIDTH AND METAL WIDTH  CC000229   <-------------------
CØMM          ØUTSIDE CHANNEL HAVE BEEN CHECKED                   CC000230   <-------------------
ØPER  PGME = EXPN GPC           200,200                           CC000231   <-------------------
IFNL  PGME,5                                                      CC000232   <-------------------
SPEC  PRNT,MAXA=499                                               CC000233   <-------------------
ØPER  PGME = INTR PGM,PGME                                        CC000234   <-------------------
FILE  5,1,PGME                                                    CC000235   <-------------------
TEXT  5,1,3,PGME                                                  CC000236   <-------------------
CØMM  ERRØR:  MINIMUM P CHANNEL GATE METAL ØVERLAP ØN N GUARD     CC000237   <-------------------
CØMM          BAND = 0.2 MIL                                      CC000238   <-------------------
ØPER  PGME = EXPN NG            50,50                             CC000239   <-------------------
IFNL  PGME,5                                                      CC000240   <-------------------
SPEC  PRNT,MAXA=249                                               CC000241   <-------------------
ØPER  PGME = INTR PGM,PGME                                        CC000242   <-------------------
FILE  5,1,PGME                                                    CC000243   <-------------------
TEXT  5,1,3,PGME                                                  CC000244   <-------------------
CØMM  ERRØR:  P CHANNEL GATE METAL MUST CRØSS THE N GUARD BAND    CC000245   <-------------------
CØMM          INSIDE EDGE BEFØRE CRØSSING THE THICK ØXIDE STEP    CC000246   <-------------------
ØPER  PGME = LINK T,NG          LINE,LINE                         CC000247   <-------------------
ØPER  PGME = TWIX NG,PGME                                         CC000248   <-------------------
SPEC  PRNT                                                        CC000249   <-------------------
ØPER  PGME = INTR PGM,PGME                                        CC000250   <-------------------
```

```
FILE  5,1,PGME                                                        CC000251
TEXT  5,1,3,PGME                                                      CC000252
FREE  PGME                                                            CC000253
CØMM                                                                  CC000254
CØMM       N TRANSISTØR IDENTIFICATIØN / CHECKING                     CC000255
CØMM       ======================================                     CC000256
CØMM                                                                  CC000257
IFNL  NA,104                                                          CC000258
CØMM  LØCATIØN ØF PRØSPECIVE N CHANNELS, ARBITRARILY ASSUMING         CC000259
CØMM  THAT ANY CHANNEL UP TØ 1 MIL LØNG WITH AT LEAST 0.1 MIL         CC000260
CØMM  METAL AND THIN ØXIDE WILL PERFØRM SØMEWHAT LIKE A TRANSISTØR    CC000261
SPEC  MAXW=1000,R180,S18# 1,R280,S28# 1                               CC000262
ØPER  PNC = TWIX NA            DIFF                                   CC000263
IFNL  PNC,97                                                          CC000264
SPEC  MINA=100                                                        CC000265
ØPER  NC = INTR PNC,T                                                 CC000266
SPEC  MINA=100                                                        CC000267
ØPER  NC = INTR NC,M                                                  CC000268
ØPER  GNC = LINK PNC,NC        NØNE,AREA                              CC000269
SPEC  R121,S121 GPC                                                   CC000270
ØPER  GNC = LINK GNC           LINE                                   CC000271
FILE  5,1,GNC                                                         CC000272
TEXT  5,1,3,GNC                                                       CC000273
FREE  PNC,NC                                                          CC000274
CØMM                                                                  CC000275
CØMM  ERRØR:  MINIMUM N CHANNEL LENGTH = 0.3 MIL, WIDTH = 0.7 MIL     CC000276
SPEC  MAXW=299,MAXL=699                                               CC000277
ØPER  NCE = TWIX NA            DIFF                                   CC000278
IFNL  NCE,5                                                           CC000279
SPEC  PRNT                                                            CC000280
ØPER  NCE = LINK NCE,GNC       NØNE,AREA                              CC000281
FILE  5,1,NCE                                                         CC000282
TEXT  5,1,3,NCE                                                       CC000283
CØMM  ERRØR:  MINIMUM THIN ØXIDE WIDTH ØUTSIDE EDGES ØF N             CC000284
CØMM          CHANNELS = 0.2 MIL                                      CC000285
ØPER  NCE = EXPN GNC           200,200                                CC000286
IFNL  NCE,5                                                           CC000287
SPEC  PRNT                                                            CC000288
ØPER  NCE = NINT NCE,T                                                CC000289
FILE  5,1,NCE                                                         CC000290
TEXT  5,1,3,NCE                                                       CC000291
CØMM  ERRØR:  MINIMUM METAL WIDTH ØUTSIDE EDGES ØF N CHANNELS         CC000292
CØMM          = 0.1 MIL                                               CC000293
ØPER  NCE = EXPN GNC           100,100                                CC000294
IFNL  NCE,5                                                           CC000295
SPEC  PRNT                                                            CC000296
ØPER  NCE = NINT NCE,M                                                CC000297
FILE  5,1,NCE                                                         CC000298
TEXT  5,1,3,NCE                                                       CC000299
FREE  NCE                                                             CC000300
CØMM                                                                  CC000301
CØMM  LØCATIØN ØF N SØURCES AND DRAINS                                CC000302
SPEC  R180,S18# 1                                                     CC000303
ØPER  NSD = SAME NA                                                   CC000304
ØPER  NSD = LINK NSD,GNC       LINE,LINE                              CC000305
```

```
OPER  NSD = INTR NSD,NA                                           CC000306   <-------------------
FILE  5,1,NSD                                                     CC000307   <-------------------
TEXT  5,1,3,NSD                                                   CC000308   <-------------------
COMM                                                              CC000309   <-------------------
COMM  ERROR:  N SOURCE AND DRAIN MINIMUM WIDTH = 0.1 MIL          CC000310   <-------------------
SPEC  PRNT,MAXA=99                                                CC000311   <-------------------
OPER  NSDE = SAME NSD                                             CC000312   <-------------------
FILE  5,1,NSDE                                                    CC000313   <-------------------
TEXT  5,1,3,NSDE                                                  CC000314   <-------------------
COMM  ERROR:  MINIMUM SEPARATION OF THIN OXIDE TO OUTSIDE N       CC000315   <-------------------
COMM          SOURCE AND DRAIN EDGES = 0.1 MIL                    CC000316   <-------------------
OPER  NSDE = EXPN NSD           5,5                               CC000317   <-------------------
IFNL  NSDE,5                                                      CC000318   <-------------------
SPEC  PRNT,MAXA=104                                               CC000319   <-------------------
OPER  NSDE = NINT NSDE,T                                          CC000320   <-------------------
FILE  5,1,NSDE                                                    CC000321   <-------------------
TEXT  5,1,3,NSDE                                                  CC000322   <-------------------
FREE  NSDE                                                        CC000323   <-------------------
COMM                                                              CC000324   <-------------------
COMM  LOCATION OF N CHANNEL GATE METAL                            CC000325   <-------------------
SPEC  R100,S10# 1                                                 CC000326   <-------------------
OPER  NGM = SAME M                                                CC000327   <-------------------
OPER  NGM = LINK NGM,ONC        LINE,LINE                         CC000328   <-------------------
OPER  NGM = INTR NGM,M                                            CC000329   <-------------------
FILE  5,1,NGM                                                     CC000330   <-------------------
TEXT  5,1,3,NGM                                                   CC000331   <-------------------
COMM                                                              CC000332   <-------------------
COMM  ERROR:  MINIMUM N CHANNEL GATE METAL WIDTH = 0.5 MIL, A     CC000333   <-------------------
COMM          REDUNDANT TEST SINCE CHANNEL WIDTH AND METAL WIDTH  CC000334   <-------------------
COMM          OUTSIDE CHANNEL HAVE BEEN CHECKED                   CC000335   <-------------------
OPER  NGME = EXPN ONC           200,200                           CC000336   <-------------------
IFNL  NGME,5                                                      CC000337   <-------------------
SPEC  PRNT,MAXA=499                                               CC000338   <-------------------
OPER  NGME = INTR NGM,NGME                                        CC000339   <-------------------
FILE  5,1,NGME                                                    CC000340   <-------------------
TEXT  5,1,3,NGME                                                  CC000341   <-------------------
COMM  ERROR:  MINIMUM N CHANNEL GATE METAL OVERLAP ON P GUARD     CC000342   <-------------------
COMM          BAND = 0.2 MIL                                      CC000343   <-------------------
OPER  NGME = EXPN PG            50,50                             CC000344   <-------------------
IFNL  NGME,5                                                      CC000345   <-------------------
SPEC  PRNT,MAXA=249                                               CC000346   <-------------------
OPER  NGME = INTR NGM,NGME                                        CC000347   <-------------------
FILE  5,1,NGME                                                    CC000348   <-------------------
TEXT  5,1,3,NGME                                                  CC000349   <-------------------
COMM  ERROR:  N CHANNEL GATE METAL MUST CROSS THE P GUARD BAND    CC000350   <-------------------
COMM          INSIDE EDGE BEFORE CROSSING THE THICK OXIDE STEP    CC000351   <-------------------
OPER  NGME = LINK T,PG          LINE,LINE                         CC000352   <-------------------
OPER  NGME = TWIX NG,NGME                                         CC000353   <-------------------
SPEC  PRNT                                                        CC000354   <-------------------
OPER  NGME = INTR NGM,NGME                                        CC000355   <-------------------
FILE  5,1,NGME                                                    CC000356   <-------------------
TEXT  5,1,3,NGME                                                  CC000357   <-------------------
FREE  NGME                                                        CC000358   <-------------------
COMM                                                              CC000359   <-------------------
COMM       GENERAL ARTWORK CHECKING - PHASE 2                     CC000360   <-------------------
```

```
COMM     ----------------------------------------     CC000361   <-------------------
COMM                                                  CC000362   <-------------------
COMM  ERROR:  MINIMUM SPACING BETWEEN A NON SOURCE OR DRAIN      CC000363   <-------------------
COMM          DIFFUSION AND ANY OTHER P DIFFUSION WITH A         CC000364   <-------------------
COMM          DIFFERENT NODE NUMBER = 0.4 MIL                    CC000365   <-------------------
OPER  PNSD = NINT P,PSD                               CC000366   <-------------------
IFNL  PNSD,8                                          CC000367   <-------------------
FILE  5,1,PNSD                                        CC000368   <-------------------
TEXT  5,1,3,PNSD                                      CC000369   <-------------------
SPEC  PRNT,MAXW=399                                   CC000370   <-------------------
OPER  PPSE = TWIX PNSD,P        DIFF                  CC000371   <-------------------
FILE  5,1,PPSE                                        CC000372   <-------------------
TEXT  5,1,3,PPSE                                      CC000373   <-------------------
FREE  PPSE                                            CC000374   <-------------------
COMM                                                  CC000375   <-------------------
COMM  ERROR:  MINIMUM SPACING BETWEEN A NON SOURCE OR DRAIN N    CC000376   <-------------------
COMM          DIFFUSION AND ANY OTHER N DIFFUSION WITH A         CC000377   <-------------------
COMM          DIFFERENT NODE NUMBER = 0.4 MIL                    CC000378   <-------------------
OPER  NNSD = NINT N,NSD                               CC000379   <-------------------
IFNL  NNSD,8                                          CC000380   <-------------------
FILE  5,1,NNSD                                        CC000381   <-------------------
TEXT  5,1,3,NNSD                                      CC000382   <-------------------
SPEC  PRNT,MAXW=399                                   CC000383   <-------------------
OPER  NNSE = TWIX NNSD,N        DIFF                  CC000384   <-------------------
FILE  5,1,NNSE                                        CC000385   <-------------------
TEXT  5,1,3,NNSE                                      CC000386   <-------------------
FREE  NNSE                                            CC000387   <-------------------
COMM                                                  CC000388   <-------------------
COMM  ERROR:  MINIMUM CONTACT WIDTH = 0.3 MIL, LENGTH = 0.4 MIL, CC000389   <-------------------
COMM          I.E., ANY ITEM WITH A DIMENSION < 0.3 MIL OR ANY   CC000390   <-------------------
COMM          ITEM WITH BOTH DIMENSIONS IN THE RANGE BETWEEN 0.3 CC000391   <-------------------
COMM          AND 0.4 MIL ARE IN ERROR                           CC000392   <-------------------
SPEC  MINA=300                                        CC000393   <-------------------
OPER  CS = SAME C                                     CC000394   <-------------------
SPEC  PRNT                                            CC000395   <-------------------
OPER  CSE = NINT C,CS                                 CC000396   <-------------------
FILE  5,1,CSE                                         CC000397   <-------------------
TEXT  5,1,3,CSE                                       CC000398   <-------------------
SPEC  PRNT,MAXA=399                                   CC000399   <-------------------
OPER  CSE = SAME CS                                   CC000400   <-------------------
FILE  5,1,CSE                                         CC000401   <-------------------
TEXT  5,1,3,CSE                                       CC000402   <-------------------
FREE  CS,CSE                                          CC000403   <-------------------
COMM  ERROR:  MINIMUM SEPARATION OF STEPPED OPENING AND ANY      CC000404   <-------------------
COMM          CONTACT OPENING = 0.1 MIL                          CC000405   <-------------------
OPER  EXC = EXPN C              100,100               CC000406   <-------------------
SPEC  PRNT                                            CC000407   <-------------------
OPER  CTOE = NINT EXC,T                               CC000408   <-------------------
FILE  5,1,CTOE                                        CC000409   <-------------------
TEXT  5,1,3,CTOE                                      CC000410   <-------------------
FREE  EXC,CTOE                                        CC000411   <-------------------
COMM  ERROR:  MINIMUM SEPARATION OF CONTACT OPENING TO PERIPHERY CC000412   <-------------------
COMM          OF SOURCE OR DRAIN DIFFUSIONS = 0.2 MIL            CC000413   <-------------------
OPER  SD = PLUS PSD,NSD                               CC000414   <-------------------
SPEC  PRNT,MAXA=199                                   CC000415   <-------------------
```

```
OPER  SDCE = NINT SD,C                                   CC000416
FILE  5,1,SDCE                                           CC000417
TEXT  5,1,3,SDCE                                         CC000418
FREE  SDCE                                               CC000419
COMM  ERROR:  METAL MUST COMPLETELY COVER SOURCE AND DRAIN  CC000420
COMM          CONTACT OPENINGS                           CC000421
OPER  SDC = LINK C,SD             LINE,'FEA              CC000422
SPEC  PRNT                                               CC000423
OPER  MCOF = NINT SDC,M                                  CC000424
FILE  5,1,MCOE                                           CC000425
TEXT  5,1,3,MCOE                                         CC000426
COMM  ERROR:  MINIMUM METAL OVERLAP OVER EDGE OF OTHER CONTACT  CC000427
COMM          OPENINGS = 0.2 MIL                         CC000428
OPER  OC = NINT C,SDC                                    CC000429
IFNL  OC,6                                               CC000430
OPER  EXOC = EXPN OC             200,200                 CC000431
SPEC  PRNT                                               CC000432
OPER  MCOF = NINT EXOC,M                                 CC000433
FILE  5,1,MCOE                                           CC000434
TEXT  5,1,3,MCOE                                         CC000435
FREE  SDC,OC,EXOC,PNSD,NNSD,MCOF                         CC000436
COMM  ERROR:  MINIMUM PENETRATION OF THIN OXIDE WITHIN GUARD  CC000437
COMM          BAND FOR GATE METAL CROSSOVER = 0.2 MIL    CC000438
OPER  GM = PLUS PGM,NGM                                  CC000439
OPER  GB = PLUS PG,NG                                    CC000440
OPER  TOXF = INTR GM,GB                                  CC000441
IFNL  TOXF,5                                             CC000442
SPEC  PRNT,MAXA=199                                      CC000443
OPER  TOXF = INTR T,TOXE                                 CC000444
FILE  5,1,TOXE                                           CC000445
TEXT  5,1,3,TOXE                                         CC000446
FREE  GM,GB                                              CC000447
COMM  ERROR:  MINIMUM SEPARATION BETWEEN THIN OXIDE AND ANY  CC000448
COMM          OUTSIDE SOURCE OR DRAIN = 0.2 MIL          CC000449
SPEC  PRNT,MAXW=199                                      CC000450
OPER  TOXF = TWIX T,SD                                   CC000451
FILE  5,1,TOXE                                           CC000452
TEXT  5,1,3,TOXE                                         CC000453
FREE  SD                                                 CC000454
COMM  ERROR:  MINIMUM SEPARATION OF THIN OXIDE AND ANY OUTSIDE  CC000455
COMM          METAL = 0.2 MIL                            CC000456
SPEC  PRNT,MAXW=199                                      CC000457
OPER  TOXE = TWIX T,M                                    CC000458
FILE  5,1,TOXE                                           CC000459
TEXT  5,1,3,TOXE                                         CC000460
FREE  TOXE                                               CC000461
COMM                                                     CC000462
COMM       CAPACITANCE CALCULATION                       CC000463
COMM       -----------------------                       CC000464
COMM                                                     CC000465
COMM  EXPAND DIFFUSIONS TO COMPENSA  FOR UNDERCUT = 0.05 MILS  CC000466
OPER  EXP = EXPN P               50,50                   CC000467
OPER  EXN = EXPN N               50,50                   CC000468
COMM  CATEGORIZE THE METAL ACCORDING TO THICKNESS OF OXIDE  CC000469
COMM  BENEATH IT                                         CC000470
```

```
OPER  MTHN = INTR M,T                                             CC000471   <--------------------
OPER  MTHK = NINT M,MTHN                                          CC000472   <--------------------
COMM  SUBTRACT CONTACT WINDOW AREAS FROM THE THYN OXIDE METAL     CC000473   <--------------------
OPER  MTHN = NINT MTHN,C                                          CC000474   <--------------------
COMM                                                              CC000475   <--------------------
COMM  CALCULATE JUNCTION CAPACITAN-.S ASSUMING KA = 0.070 PF/SQ.  CC000476   <--------------------
COMM  MIL AND KP = 0.200 PF/MIL IN UNITS OF .001 PF               CC000477   <--------------------
PARE  6,EXP,-14285,-500                                           CC000478   <--------------------
COMM                                                              CC000479   <--------------------
COMM  CALCULATE CROSSOVER CAPACITANCES ASSUMING KO = 0.020 PF/SQ. CC000480   <--------------------
COMM  MIL THROUGH THICK OXIDE AND KO = 0.200 PF/SQ. MIL THROUGH   CC000481   <--------------------
COMM  THIN OXIDE IN .001 PF                                       CC000482   <--------------------
OPER  PTHK = INTR EXP,MTHK                                        CC000483   <--------------------
AREA  6,PTHK,-50000                                               CC000484   <--------------------
OPER  PTHN = INTR EXP,MTHN                                        CC000485   <--------------------
AREA  6,PTHN,-5000                                                CC000486   <--------------------
FREE  PTHK,PTHN                                                   CC000487   <--------------------
OPER  NTHK = INTR EXN,MTHK                                        CC000488   <--------------------
AREA  6,NTHK,-50000                                               CC000489   <--------------------
OPER  NTHN = INTR EXN,MTHN                                        CC000490   <--------------------
AREA  6,NTHN,-5000                                                CC000491   <--------------------
FREE  NTHK,NTHN                                                   CC000492   <--------------------
COMM                                                              CC000493   <--------------------
COMM  CALCULATE SUBSTRATE CAPACITANCE WITH THE SAME OXIDE         CC000494   <--------------------
COMM  COEFFICIENTS                                                CC000495   <--------------------
OPER  ALL = PLUS EXP,EXN                                          CC000496   <--------------------
OPER  STHK = NINT MTHK,ALL                                        CC000497   <--------------------
AREA  6,STHK,-50000                                               CC000498   <--------------------
OPER  STHN = NINT MTHN,ALL                                        CC000499   <--------------------
AREA  6,STHN,-5000                                                CC000500   <--------------------
FREE  STHK,STHN                                                   CC000501   <--------------------
COMM                                                              CC000502   <--------------------
COMM       EQUATION GENERATION                                    CC000503   <--------------------
COMM       -------------------                                    CC000504   <--------------------
COMM                                                              CC000505   <--------------------
COMM  GENERATE BOOLEAN TRANSISTOR EQUATIONS                       CC000506   <--------------------
SPE   .'11                                                        CC000507   <--------------------
OPE    G = SAME PG                                                CC000508   <--------------------
SPEC  F231                                                        CC000509   <--------------------
OPER  NG = SAME NG                                                CC000510   <--------------------
BOOL  /6/PG:0/NG:1/PSD:PSD,!PGM/NSD:NSD,NGM/                      CC000511   <--------------------
COMM                                                              CC000512   <--------------------
COMM       MISCELLANEOUS LIST PROCESSING                          CC000513   <--------------------
COMM       -----------------------------                          CC000514   <--------------------
COMM                                                              CC000515   <--------------------
COMM  STORE DEVICE/NODE LIST                                      CC000516   <--------------------
LIST  /6/PGM:PSD,PSD/NGM:NSD,NSD/                                 CC000517   <--------------------
COMM  PRINT LIST OF CHANNEL DIMENSIONS                            CC000518   <--------------------
RANG  6,OPC                                                       CC000519   <--------------------
RANG  6,ONC                                                       CC000520   <--------------------

   520 ACCEPTED

 * OFFSET *
     X          0
```

Y O

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
W	10	1	0	1	1	1
P	11	2	3000	2	1	2
N	12	3	6000	1	1	3
T	13	4	9000	1	1	4
C	14	5	12000	1	1	5
M	15	6	15000	1	1	6

* TIME *

CLOCK = 60
DELTA = 60

1 W

1	1	500	1800	6600	3300
2	1	7300	6200	11700	7700

1 P

3	2	100	1200	13100	1400
4	2	13175	1200	13250	1300
5	2	13125	1300	13350	1400
6	2	100	1400	13450	1500
7	2	100	1500	13550	1600
8	2	100	1600	13100	1800
9	2	13150	1600	13650	1700
10	2	13250	1700	13750	1800
11	2	100	1800	700	3300
12	2	900	1800	1500	2200
13	2	3900	1800	4500	2200
14	2	6500	1800	7100	3300
15	2	13350	1800	13850	1900
16	2	13450	1900	13950	2000
17	2	13550	2000	14050	2100
18	2	13650	2100	14150	2200
19	2	13750	2200	14250	2300
20	2	13850	2300	14350	2400
21	2	13950	2400	14450	2500
22	2	14050	2500	14550	2600
23	2	14150	2600	14650	2700
24	2	14250	2700	14750	2800
25	2	14350	2800	14850	2900
26	2	900	2900	1500	3300
27	2	3900	2900	4500	3300

C-20

28	2	14450	2900	14950	3000
29	2	7500	3000	8500	4400
30	2	8800	3000	10000	4400
31	2	10300	3000	11300	4400
32	2	14550	3000	15050	3100
33	2	14650	3100	15150	3200
34	2	14750	3200	15250	3300
35	2	100	3300	7100	3900
36	2	14850	3300	15350	3400
37	2	14950	3400	15450	3500
38	2	15050	3500	15550	3600
39	2	15150	3600	15650	3700
40	2	15250	3700	15750	3800
41	2	15350	3800	15850	3900
42	2	15450	3900	15950	4000
43	2	15550	4000	16050	4100
44	2	15650	4100	16050	4200
45	2	15650	4200	16050	4300
46	2	15550	4300	16150	4400
47	2	15450	4400	16153	4500
48	2	15350	4500	16059	4600
49	2	15250	4600	15965	4700
50	2	15150	4700	15871	4800
51	2	15050	4800	15777	4900
52	2	14950	4900	15683	5000

2 P

53	2	14850	5000	15589	5100
54	2	800	5100	1800	6500
55	2	2100	5100	3300	6500
56	2	3600	5100	4800	6500
57	2	5100	5100	6100	6500
58	2	14750	5100	15495	5200
59	2	14650	5200	15401	5300
60	2	14550	5300	15308	5400
61	2	14450	5400	15214	5500
62	2	14350	5500	15121	5600
63	2	7700	5600	15027	5700
64	2	7700	5700	14934	5800
65	2	7700	5800	14840	5900
66	2	7700	5900	14747	6000
67	2	7700	6000	14700	6200
68	2	9100	6200	9700	6600
69	2	11700	6200	12300	7700
70	2	9100	7300	9700	7700
71	2	7000	7700	12300	8300

1 N

72	3	0	0	17000	2200
73	3	0	2200	800	2900
74	3	1800	2200	2100	2900

75	3	3300	2200	3600	2700
76	3	4800	2200	5100	2900
77	3	6100	2200	7200	2600
78	3	12100	2200	17000	5200
79	3	6100	2600	7600	2900
80	3	6200	2600	11700	3300
81	3	0	2900	7600	3300
82	3	0	3300	11700	4300
83	3	0	4300	100	7300
84	3	6900	4300	11700	4400
85	3	6900	4400	7600	4800
86	3	8200	4400	11700	4800
87	3	500	4700	900	5100
88	3	1500	4700	6500	5100
89	3	500	5100	6500	6500
90	3	6900	5200	17000	6600
91	3	500	6500	900	6900
92	3	1500	6500	6500	6900
93	3	6900	6600	7500	7300
94	3	8500	6600	8800	7300
95	3	10000	6600	10300	7300
96	3	11200	6600	17000	7300
97	3	0	7300	17000	9000
					1 T
98	4	900	1600	5300	2300
99	4	900	2300	6000	2800
100	4	7600	2400	10500	3100
101	4	900	2800	5300	3500
102	4	7600	3100	11200	4300
103	4	7600	4300	10500	5000
104	4	900	4500	5300	5200
105	4	900	5200	6000	6400
106	4	8800	6000	10500	6700
107	4	900	6400	5300	7100
108	4	7600	6700	8600	7200
109	4	8300	6700	11200	7200
110	4	8800	7200	10500	7900
					1 C
111	5	1000	1700	1400	3400
112	5	4000	1700	4400	3400
113	5	2500	2500	2900	2800
114	5	5500	2500	5900	2800
115	5	7700	2500	8100	4900
116	5	10700	3200	11100	4200
117	5	1000	4600	1400	7000
118	5	5500	5300	5900	6300
119	5	9200	6100	9600	7800
120	5	7700	6500	8100	7100
121	5	10700	6800	11100	7100

1 M

122	6	7265	617	7348	700
123	6	2500	700	7600	1100
124	6	7168	700	7434	800
125	6	7066	800	7523	891
126	6	7021	891	7570	900
127	6	7072	900	7621	1000
128	6	7166	1000	7715	1100
129	6	2500	1100	2900	2800
130	6	5500	1100	5900	6400
131	6	7260	1100	7809	1200
132	6	7354	1200	7903	1300
133	6	7448	1300	7997	1400
134	6	7542	1400	8091	1500
135	6	7636	1500	8185	1600
136	6	9900	1500	10400	7900
137	6	1000	1600	1400	3500
138	6	1700	1600	2200	7600
139	6	3200	1600	3700	7600
140	6	4000	1600	4400	3500
141	6	4700	1600	5200	7600
142	6	7730	1600	8279	1700
143	6	7824	1700	8373	1800
144	6	7918	1800	8467	1900
145	6	8012	1900	8560	2000
146	6	8105	2000	8654	2100
147	6	8199	2100	8747	2200
148	6	8292	2200	8841	2300
149	6	8386	2300	8934	2400
150	6	8400	2300	8900	7900
151	6	7700	2400	8100	5000
152	6	8479	2400	9028	2500
153	6	10700	2400	11100	8400
154	6	8530	2500	9079	2509
155	6	8576	2509	9035	2600
156	6	8666	2600	8933	2700
157	6	8752	2700	8835	2783
158	6	1000	4500	1400	7100
159	6	9200	6000	9600	7900
160	6	7700	6700	8100	7500
161	6	7500	8400	12800	8800

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
P	10	1	0	2	71	2
N	11	2	3000	1	97	3
T	12	3	6000	1	110	4
C	13	4	9000	1	121	5
M	14	5	12000	1	161	6

W	16	7	18000	1	2	1

* TIME *

CLOCK = 63
DELTA = 3

Output for commands 1 through 24 not shown.

C-23

NODAL TRACE / RECTANGLE LINKAGE

CC000025
CC000026

NODAL TRACE

TRAC /M:C/C:M,P,N/P:C/N:C/ CC000030

CC000027
CC000028
CC000029
<-------------------- 30

WARNING
CONDITION 8
WARNING
CONDITION 8
* NULL MASK *
M

WARNING
CONDITION 8
WARNING
CONDITION 8
* NULL MASK *
C

WARNING
CONDITION 8
WARNING
CONDITION 8
WARNING
CONDITION 8
WARNING
CONDITION 8

C-21

1 M

1	122	7265	617	7348	700
1	123	2500	700	7600	1100
1	124	7168	700	7434	800
1	125	7066	800	7523	891
1	126	7021	891	7570	900
1	127	7072	900	7621	1000
1	128	7166	1000	7715	1100
1	129	2500	1100	2900	2800
1	130	5500	1100	5900	6400
1	131	7260	1100	7809	1200
1	132	7354	1200	7903	1300
1	133	7448	1300	7997	1400
1	134	7542	1400	8091	1500
1	135	7636	1500	8185	1600
2	136	9900	1500	10400	7900
3	137	1000	1600	1400	3500
4	138	1700	1600	2200	7600
5	139	3200	1600	3700	7600
3	140	4000	1600	4400	3500
6	141	4700	1600	5200	7600
1	142	7730	1600	8279	1700
1	143	7824	1700	8373	1800
1	144	7918	1800	8467	1900
1	145	8012	1900	8560	2000
1	146	8105	2000	8654	2100

1	147	8199	2100	8747	2200
1	148	8292	2200	8841	2300
1	149	8386	2300	8934	2400
1	150	8400	2300	8900	7900
7	151	7700	2400	8100	5000
1	152	8479	2400	9028	2500
8	153	10700	2400	11100	8400
1	154	8530	2500	9079	2509
1	155	8576	2509	9035	2600
1	156	8666	2600	8933	2700
1	157	8752	2700	8835	2783
7	158	1000	4500	1400	7100
3	159	9200	6000	9600	7900
9	160	7700	6700	8100	7500
8	161	7500	8400	12800	8800

1 C

3	111	1000	1700	1400	3400
3	112	4000	1700	4400	3400
1	113	2500	2500	2900	2800
1	114	5500	2500	5900	2800
7	115	7700	2500	8100	4900
8	116	10700	3200	11100	4200
7	117	1000	4600	1400	7000
1	118	5500	5300	5900	6300
3	119	9200	6100	9600	7800
9	120	7700	6800	8100	7100
8	121	10700	6800	11100	7100

1 P

3	3	100	1200	13100	1400
3	4	13175	1200	13250	1300
3	5	13125	1300	13350	1400
3	6	100	1400	13450	1500
3	7	100	1500	13550	1600
3	8	100	1600	13100	1800
3	9	13150	1600	13650	1700
3	10	13250	1700	13750	1800
3	11	100	1800	700	3300
3	12	900	1800	1500	2200
3	13	3900	1800	4500	2200
3	14	6500	1800	7100	3300
3	15	13350	1800	13850	1900
3	16	13450	1900	13950	2000
3	17	13550	2000	14050	2100
3	18	13650	2100	14150	2200
3	19	13750	2200	14250	2300
3	20	13850	2300	14350	2400
3	21	13950	2400	14450	2500
3	22	14050	2500	14550	2600
3	23	14150	2600	14650	2700

3	24	14250	2700	14750	2800
3	25	14350	2800	14850	2900
3	26	900	2900	1500	3300
3	27	3900	2900	4500	3300
3	28	14450	2900	14950	3000
7	29	7500	3000	8500	4400
10	30	8800	3000	10000	4400
8	31	10300	3000	11300	4400
3	32	14550	3000	15050	3100
3	33	14650	3100	15150	3200
3	34	14750	3200	15250	3300
3	35	100	3300	7100	3900
3	36	14850	3300	15350	3400
3	37	14950	3400	15450	3500
3	38	15050	3500	15550	3600
3	39	15150	3600	15650	3700
3	40	15250	3700	15750	3800
3	41	15350	3800	15850	3900
3	42	15450	3900	15950	4000
3	43	15550	4000	16050	4100
3	44	15650	4100	16050	4200
3	45	15650	4200	16050	4300
3	46	15550	4300	16150	4400
3	47	15450	4400	16153	4500
3	48	15350	4500	16059	4600
3	49	15250	4600	15965	4700
3	50	15150	4700	15871	4800
3	51	15050	4800	15777	4900
3	52	14950	4900	15683	5000

2 P

3	53	14850	5000	15589	5100
7	54	800	5100	1800	6500
11	55	2100	5100	3300	6500
12	56	3600	5100	4800	6500
1	57	5100	5100	6100	6500
3	58	14750	5100	15495	5200
3	59	14650	5200	15401	5300
3	60	14550	5300	15308	5400
3	61	14450	5400	15214	5500
3	62	14350	5500	15121	5600
3	63	7700	5600	15027	5700
3	64	7700	5700	14934	5800
3	65	7700	5800	14840	5900
3	66	7700	5900	14747	6000
3	67	7700	6000	14700	6200
3	68	9100	6200	9700	6600
3	69	11700	6200	12300	7700
3	70	9100	7300	9700	7700
3	71	7000	7700	12300	8300

1 N

C-26

3	162	800	2200	1800	2900
1	163	2100	2200	3300	2900
3	164	3600	2200	4800	2900
1	165	5100	2200	6100	2600
7	166	7200	2200	12100	2600
1	167	5100	2600	6100	2900
7	168	7600	2600	8200	2900
7	169	11700	2600	12100	2900
7	170	7600	2900	8200	3300
7	171	11700	2900	12100	3300
7	172	11700	3300	12100	4300
7	173	100	4300	6900	4400
7	174	11700	4300	12100	4400
7	175	100	4400	6900	4700
7	176	7600	4400	8200	4800
7	177	11700	4400	12100	4800
7	178	100	4700	500	4800
7	179	900	4700	1500	4800
7	180	6500	4700	6900	4800
7	181	100	4800	500	5100
7	182	900	4800	1500	5100
7	183	6500	4800	12100	5100
7	184	100	5100	500	5200
7	185	6500	5100	12100	5200
7	186	100	5200	500	6500
7	187	6500	5200	6900	6500
7	188	100	6500	500	6600
7	189	900	6500	1500	6600
7	190	6500	6500	6900	6600
7	191	100	6600	500	6900
7	192	900	6600	1500	6900
7	193	6500	6600	6900	6900
9	194	7500	6600	8500	6900
3	195	8800	6600	10000	6900
8	196	10300	6600	11200	6900
7	197	100	6900	6900	7300
9	198	7500	6900	8500	7300
3	199	8800	6900	10000	7300
8	200	10300	6900	11200	7300

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	14	44	470	N	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
M	11	2	3300	1	9	161
T	12	3	6300	1	110	4
C	13	4	9300	1	9	121
N	15	6	15000	1	9	200
W	16	7	18300	1	2	1
P	17	8	21600	2	12	71

* TIME *

CLOCK = 100

DELTA = 17

Output for commands 31 through 48 not shown.

```
                                                                  CC000049
                    GENERAL ARTWORK CHECKING - PHASE 1            CC000050
                    ----------------------------------            CC000051
                                                                  CC000052
                  ERROR!  MINIMUM N-P SPACING BETWEEN DIFFERENT NODES = 0.4   CC000053
                          MIL                                     CC000054
SPEC  PRNT,MAXW=399                                  CC000055   <------------------   55
OPER  NPSE = T,IX N,P          DIFF                  CC000056   <------------------   56
* NULL MASK *
    NPSE

# WARNING #
 CONDITION  8
# WARNING #
 CONDITION  8

                                                                              1 NPSE
                                   7      3    7100     2200     7200     2300
                                   7      3    7100     2300     7200     2400
                                   7      3    7100     2400     7200     2500
                                   7      3    7100     2500     7200     2600
* DIRECTORY *
    FILES
    UNIT   SET   RECORDS   TOTAL   NAME   TYPE
      5     26       18      878   T        1

    MASKS
    NAME   DIRECTORY  FILE  ADDRESS  RECORDS  IDENT # 1  IDENT # 2
    M          11       2     3300      1          9        161
    C          13       4     9300      1          9        121
    N          15       6    15300      1          9        200
    T          16       7    18300      1         16        110
    P          17       8    21600      2         12         71
    W          18       9    24300      1         11          2
    NPSE       20      11    30000      1          7          3

* TIME *                                                   CLOCK =  128
                                                           DELTA =   12
```

C-28

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

Output for commands 57 through 89 not shown.

```
                                                                                       CC000090
                             SEPARATION OF P DIFFUSION INTO ACTIVE OR GUARD BAND AREAS CC000091
   OPER  PA * NLNK P,H      LINE,LINE                          CC000092     <-------------------   92
   # WARNING #
    CONDITION  8
   # WARNING #
    CONDITION  3
   * NULL MASK *
       PA

                                                                                                 1 PA
```

1	1	5100	5100	6100	6500
7	7	7500	3000	8500	4400
7	7	800	5100	1800	6500
8	8	10300	3000	11300	4400
10	10	8800	3000	10000	4400
11	11	2100	5100	3300	6500
12	12	3600	5100	4800	6500

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	32	6	934	MTSE	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
M	11	2	3300	1	9	161
PA	12	3	6000	1	12	12
C	13	4	9300	1	9	121
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2

* TIME *

CLOCK = 163
DELTA = 2

Output for commands 93 through 94 not shown.

OPER PG = NINT P,PA   CC000095   <--------------------   95

1 PG

3	3	100	1200	13100	1400
3	4	13175	1200	13250	1300
3	5	13125	1300	13350	1400
3	6	100	1400	13450	1500
3	7	100	1500	13550	1600
3	8	100	1600	13100	1800
3	9	13150	1600	13650	1700
3	10	13250	1700	13750	1800
3	11	100	1800	700	3300
3	12	900	1800	1500	2200
3	13	3900	1800	4500	2200
3	14	6500	1800	7100	3300
3	15	13350	1800	13850	1900

3	16	13450	1900	13950	2000
3	17	13550	2000	14050	2100
3	18	13650	2100	14150	2200
3	19	13750	2200	14250	2300
3	20	13850	2300	14350	2400
3	21	13950	2400	14450	2500
3	22	14050	2500	14550	2600
3	23	14150	2600	14650	2700
3	24	14250	2700	14750	2800
3	25	14350	2800	14850	2900
3	26	900	2900	1500	3300
3	27	3900	2900	4500	3300
3	28	14450	2900	14950	3000
3	32	14550	3000	15050	3100
3	33	14650	2100	15150	3200
3	34	14750	3200	15250	3300
3	35	100	3300	7100	3900
3	36	14850	3300	15350	3400
3	37	14950	3400	15450	3500
3	38	15050	3500	15550	3600
3	39	15150	3600	15650	3700
3	40	15250	3700	15750	3800
3	41	15350	3800	15850	3900
3	42	15450	3900	15950	4000
3	43	15550	4000	16050	4100
3	44	15650	4100	16050	4200
3	45	15650	4200	16050	4300
3	46	15550	4300	16150	4400
3	47	15450	4400	16153	4500
3	48	15350	4500	16059	4600
3	49	15250	4600	15965	4700
3	50	15150	4700	15871	4800
3	51	15050	4800	15777	4900
3	52	14950	4900	15683	5000
3	53	14850	5000	15589	5100
3	58	14750	5100	15495	5200
3	59	14650	5200	15401	5300

2 PG

3	60	14550	5300	15308	5400
3	61	14450	5400	15214	5500
3	62	14350	5500	15121	5600
3	63	7700	5600	15027	5700
3	64	7700	5700	14934	5800
3	65	7700	5800	14840	5900
3	66	7700	5900	14747	6000
3	67	7700	6000	14700	6200
3	68	9100	6200	9700	6600
3	69	11700	6200	12300	7700
3	70	9100	7300	9700	7700
3	71	7000	7700	12300	8300

• DIRECTORY •
FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	34	12	958	PA	1

MASKS NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12000	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2

* TIME *

CLOCK = 166
DELTA = 2

Output for commands 96 through 98 not shown.

SEPARATION OF N DIFFUSION INTO ACTIVE OR GUARD BAND AREAS CC000098 .

OPER NA = INTR N,W CC000099 <-------------------- 99

1 NA

1	10	2100	2200	3300	2900
1	10	5100	2200	6100	2600
1	10	5100	2600	6100	2900
3	10	800	2200	1800	2900
3	10	3500	2200	4800	2900
3	11	8800	6600	10000	6900
3	11	8800	6900	10000	7300
8	11	10300	6600	11200	6900
8	11	10300	6900	11200	7300
9	11	7500	6600	8500	6900
9	11	7500	6900	8500	7300

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	36	67	1092	PG	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	0	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2

* TIME *

CLOCK = 172
DELTA = 1

Output for commands 99 through 101 not shown.

OPER NG = NINT N,NA   CC000102 <-------------------- 102

1 NG

7	166	7200	2200	12100	2600
7	168	7600	2600	8200	2900
7	169	11700	2600	12100	2900
7	170	7600	2900	8200	3300
7	171	11700	2900	12100	3300
7	172	11700	3300	12100	4300
7	173	100	4300	6900	4400

7	174	11700	4300	12100	4400
7	175	100	4400	6900	4700
7	176	7600	4400	8200	4800
7	177	11700	4400	12100	4800
7	178	100	4700	500	4800
7	179	900	4700	1500	4800
7	180	6500	4700	6900	4800
7	181	100	4800	500	5100
7	182	900	4800	1500	5100
7	183	6500	4800	12100	5100
7	184	100	5100	500	5200
7	185	6500	5100	12100	5200
7	186	100	5200	500	6500
7	187	6500	5200	6900	6500
7	188	100	6500	500	6600
7	189	900	6500	1500	6600
7	190	6500	6500	6500	6600
7	191	100	6600	500	6900
7	192	900	6600	1500	6900
7	193	6500	6600	6900	6900
7	197	100	6900	6900	7300

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	38	16	1124	NA	1

C-34

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3[illegible]00	1	9	161
PA	12	3	4300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27000	1	7	197

* TIME *

CLOCK = 174
DELTA = 1

Output for commands [illegible]3 through 149 not shown.

```
                                                                                 CC000150
                         P TRANSISTOR IDENTIFICATION / CHECKING                  CC000151
                         --------------------------------------                  CC000152
                                                                                 CC000153
                      LOCATION OF PROSPECTIVE P CHANNELS, ARBITRARILY ASSUMING   CC000154
                      THAT ANY CHANNEL UP TO 1 MIL LONG WITH AT LEAST 0.1 MIL    CC000155
                      METAL AND THIN OXIDE WILL PERFORM SOMEWHAT LIKE A TRANSISTOR CC000156
SPEC  MAXW=1000,R180,S12# 1,R220,S23# 1                         CC000157      <-------------------   157
OPER  PPC = TWIX PA          DIFF                               CC000158      <-------------------   158
* NULL MASK *
     PPC

# WARNING #
 CONDITION  8
# WARNING #
 CONDITION  8

                                                                                                   1 PPC

                                            1        1      8500      3000      8800      4400
                                            2        2     10000      3000     10300      4400
                                            3        3      1800      5100      2100      6500
                                            4        4      3300      5100      3600      6500
                                            5        5      4800      5100      5100      6500
* DIRECTORY *
     FILES
     UNIT   SET   RECORDS   TOTAL   NAME   TYPE
       5     46       41     1366   SUME     1

     MASKS
     NAME  DIRECTORY  FILE  ADDRESS  RECORDS  IDENT # 1  IDENT # 2
     NA        10       1      400       1          9         11
     M         11       2     3300       1          9        161
     PA        12       3     6300       1         12         12
     C         13       4     9300       1          9        121
     PG        14       5    12[illegible]       2          3         71
     N         15       6    15300       1          9        200
     T         16       7    18300       1         16        110
     P         17       8    21600       2         12         71
     W         18       9    24300       1         11          2
     NG        19      10    27300       1          7        197
     PPC       23      14    39000       1          5          5

* TIME *                                                        CLOCK =    214
                                                                DELTA =      2
```

```
IFNL  PPC,97                                                          CC000159    <--------------------    159
SPEC  MINA=100                                                        CC000160    <--------------------    160
OPER  PC = INTR PPC,T                                                 CC000161    <--------------------    161
* DIRECTORY *
      FILES
      UNIT  SET  RECORDS  TOTAL  NAME  TYPE
        5    46      41    1366  SURE    1

      MASKS
      NAME  DIRECTORY  FILE  ADDRESS  RECORDS  IDENT # 1  IDENT # 2
      NA       10        1      300      1          9         11
      M        11        2     3300      1          9        161
      PA       12        3     6300      1         12         12
      C        13        4     9300      1          9        121
      PG       14        5    12600      2          3         71
      N        15        6    15300      1          9        200
      T        16        7    18300      1         16        110
      P        17        8    21600      2         12         71
      W        18        9    24300      1         11          2
      NG       19       10    27300      1          7        197
      PPC      23       14    39300      1          5          5
      PC       24       15    42000      1          5         14
* TIME *                                                                   CLOCK =   216
                                                                           DELTA =     2
```

```
SPEC  MINA=100                                                        CC000162    <--------------------    162
OPER  PC = INTR PC,M                                                  CC000163    <--------------------    163
* DIRECTORY *
      FILES
      UNIT  SET  RECORDS  TOTAL  NAME  TYPE
        5    46      41    1366  SURE    1

      MASKS
      NAME  DIRECTORY  FILE  ADDRESS      RECORDS  IDENT # 1  IDENT # 2
      NA       10        1      300          1          9         11
      M        11        2     3300          1          9        161
      PA       12        3     6300          1         12         12
      C        13        4     9300          1          9        121
      PG       14        5    12600          2          3         71
      N        15        6    [illegible]    1          9        200
      T        16        7    [illegible]    1         16        110
      P        17        8    21600          2         12         71
      W        18        9    24300          1         11          2
      NG       19       10    27300          1          7        197
      PPC      23       14    39300          1          5          5
      PC       25       16    45000          1          5          6
* TIME *                                                                   CLOCK =   218
                                                                           DELTA =     2
```

```
OPER  OPC = LINK PPC,PC        NONE,AREA                          CC000164       <-------------------   164
# WARNING #
 CONDITION  8
# WARNING #
 CONDITION  8
* NULL MASK *
     OPC

                                                                                                 1 OPC
                                        1      1     8500    3000    8800    4400
                                        2      2    10000    3000   10300    4400
                                        3      3     1800    5100    2100    6500
                                        4      4     3300    5100    3600    6500
                                        5      5     4800    5100    5100    6500
* DIRECTORY *
     FILES
     UNIT  SET  RECORDS  TOTAL  NAME  TYPE
      5    46      41    1366   SURE    1

     MASKS
     NAME  DIRECTORY  FILE  ADDRESS  RECORDS  IDENT # 1  IDENT # 2
     NA       10        1     300       1         9        11
     M        11        2    3300       1         9       161
     PA       12        3    6300       1        12        12
     C        13        4    9300       1         9       121
     PG       14        5   12600       2         3        71
     N        15        6   15300       1         9       200
     T        16        7   18300       1        16       110
     P        17        8   21600       2        12        71
     W        18        9   24300       1        11         2
     NG       19       10   27300       1         7       197
     OPC      21       12   33000       1         5         5
     PPC      23       14   39300       1         5         5
     PC       25       16   45300       1         5         6
* TIME *                                                          CLOCK =   220
                                                                  DELTA =     2
-----------------------------------------------------------------------------------------------

SPEC  R101,S10# 1                                                 CC000165       <-------------------   165
OPER  OPC = LINK OPC           LINE                               CC000166       <-------------------   166
# WARNING #
 CONDITION  8
# WARNING #
 CONDITION  8
* NULL MASK *
     OPC
```

C-37

1 OPC

1	1	8500	3000	8800	4400
2	2	10000	3000	10300	4400
3	3	1800	5100	2100	6500
4	4	3300	5100	3600	6500
5	5	4800	5100	5100	6500

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	46	41	1366	SURE	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PPC	23	14	39300	1	5	5
OPC	24	15	42000	1	5	5
PC	25	16	45300	1	5	6

* TIME *

CLOCK = 220
DELTA = 0

Output for command 167 through 195 not shown.

```
                         LOCATION OF P SOURCES AND DRAINS                          CC000196
                                                                                    CC000197
SPEC  R180,S13# 1                                                CC000198    <-------------------  198
OPER  PSD = SAME PA                                              CC000199    <-------------------  199

                                                                                          1 PSD

                                                   1      1     5100     5100     6100     6500
                                                   2      7     7500     3000     8500     4400
                                                   3      7      800     5100     1800     6500
                                                   4      8    10300     3000    11300     4400
                                                   5     10     8800     3000    10000     4400
                                                   6     11     2100     5100     3300     6500
                                                   7     12     3600     5100     4800     6500
 * DIRECTORY *
     FILES
     UNIT   SET   RECORDS   TOTAL   NAME   TYPE
       5     48      10      1386   OPC      1

     MASKS
     NAME  DIRECTORY  FILE  ADDRESS  RECORDS  IDENT # 1  IDENT # 2
     NA       10        1      300      1          9         11
     M        11        2     3300      1          9        161
     PA       12        3     6300      1         12         12
     C        13        4     9300      1          9        121
     PS       14        5    12600      2          3         71
     N        15        6    15300      1          9        200
     T        16        7    18300      1         16        110
     P        17        8    21600      2         12         71
     W        18        9    24300      1         11          2
     NS       19       10    27300      1          7        197
     PSD      20       11    30400      1          7         12
     OPC      24       15    42300      1          5          5

 * TIME *                                                        CLOCK =   229
                                                                 DELTA =     1
-----------------------------------------------------------------------------------------------

OPER  PSD = LINK PSD/OPC        LINE/LINE                        CC000200    <-------------------  200
```

C-39

\# WARNING #
CONDITION 8
\# WARNING #
CONDITION 8
* NULL MASK *
PSD

1 PSD

1	2	7500	3000	8500	4400
1	5	8800	3000	10000	4400
2	4	10300	3000	11300	4400
2	5	8800	3000	10000	4400
3	3	800	5100	1800	6500
3	6	2100	5100	3300	6500
4	6	2100	5100	3300	6500
4	7	3600	5100	4800	6500
5	1	5100	5100	6100	6500
5	7	3600	5100	4800	6500

* DIRECTORY *
FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	48	10	1386	OPC	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	21	12	33000	1	5	7
OPC	24	15	42300	1	5	5

* TIME *
CLOCK = 231
DELTA = 2

OPER PSD = INTR PSD,PA CC000201 <-------------------- 201

1 PSD

1	7	7500	3000	8500	4400
1	10	8800	3000	10000	4400
2	8	10300	3000	11300	4400
2	10	8800	3000	10000	4400

3	7	800	5100	1800	6500
3	11	2100	5100	3300	6500
4	11	2100	5100	3300	6500
4	12	3600	5100	4800	6500
5	1	5100	5100	6100	6500
5	12	3600	5100	4300	6500

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	48	10	1386	GPC	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30000	1	5	12
GPC	24	15	42300	1	5	5

* TIME *

CLOCK = 231
DELTA = 0

Output for commands 202 through 218 not shown.

CC000219
LOCATION OF P CHANNEL GATE METAL CC000220

SPEC R120,S10# 1 CC000221 <-------------------- 221
OPER PGM = SAME M CC000222 <-------------------- 222

1 PGM

1	122	7265	617	7348	700
2	123	2500	700	7600	1100
3	124	7168	700	7434	800
4	125	7066	800	7523	891
5	126	7021	891	7570	900
6	127	7072	900	7621	1000
7	128	7166	1000	7715	1100
8	129	2500	1100	2900	2800
9	130	5500	1100	5900	6400
10	131	7260	1100	7809	1200
11	132	7354	1200	7903	1300
12	133	7448	1300	7997	1400
13	134	7542	1400	8091	1500
14	135	7636	1500	8185	1600
15	142	7730	1600	8279	1700
16	143	7824	1700	8373	1800
17	144	7918	1800	8467	1900
18	145	8012	1900	8560	2000
19	146	8105	2000	8654	2100
20	147	8199	2100	8747	2200
21	148	8292	2200	8841	2300
22	149	8386	2300	8934	2400
23	150	8400	2300	8900	7900
24	152	8479	2400	9028	2500
25	154	8530	2500	9079	2509
26	155	8576	2509	9035	2600
27	156	8666	2600	8933	2700
28	157	8752	2700	8835	2783
29	136	9900	1500	10400	7900
30	137	1000	1600	1400	3500
31	140	4000	1600	4400	3500
32	159	9200	6000	9600	7900
33	138	1700	1600	2200	7600
34	139	3200	1600	3700	7600
35	141	4700	1600	5200	7600
36	151	7700	2400	8100	5000
37	158	1000	4500	1400	7100
38	153	10700	2400	11100	8400
39	161	7500	8400	12800	8800
40	160	7700	6700	8100	7500

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	50	15	1416	PSU	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121

PO	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	16300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	21	12	33000	1	40	161
OPC	24	15	42300	1	5	5

* TIME *

CLOCK = 241
DELTA = 1

OPER PGM = LINK PGM,OPC LINE/AREA CC000223 <-------------------- 223
WARNING
CONDITION 6
WARNING
CONDITION 8
* NULL MASK *
PGM

1 PGM

1	1	7265	617	7348	700
1	2	2500	700	7600	1100
1	3	7168	700	7434	800
1	4	7066	800	7523	891
1	5	7021	891	7570	900
1	6	7072	900	7621	1000
1	7	7166	1000	7715	1100
1	8	2500	1100	2900	2800
1	9	5500	1100	5900	6400
1	10	7260	1100	7809	1200
1	11	7354	1200	7903	1300
1	12	7448	1300	7997	1400
1	13	7542	1400	8091	1500
1	14	7636	1500	8185	1600
1	15	7730	1600	8279	1700
1	16	7824	1700	8373	1800
1	17	7918	1800	8467	1900
1	18	8012	1900	8560	2000
1	19	8105	2000	8654	2100
1	20	8199	2100	8747	2200
1	21	8292	2200	8841	2300
1	22	8386	2300	8934	2400
1	23	8400	2300	8900	7900
1	24	8479	2400	9028	2500
1	25	8530	2500	9079	2509
1	26	8575	2509	9035	2600
1	27	8666	2600	8933	2700
1	28	8752	2700	8835	2783

2	29	9900	1500	10400	7900
3	33	1700	1600	2200	7600
4	34	3200	1600	3700	7600
5	35	4700	1600	5200	7600

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	50	15	1416	PSD	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	22	13	36000	1	5	35
OPC	24	15	42300	1	5	5

* TIME *

CLOCK = 243
DELTA = 2

OPER PGM = INTR PGM,M CC000224 <-------------------- 224

1 PGM

1	1	7265	617	7348	700
1	1	2500	700	7600	1100
1	1	7168	700	7434	800
1	1	7066	800	7523	891
1	1	7021	891	7570	900
1	1	7072	900	7600	1000
1	1	7072	900	7621	1000
1	1	7166	1000	7600	1100
1	1	7166	1000	7715	1100
1	1	2500	1100	2900	2800
1	1	5500	1100	5900	6400
1	1	7260	1100	7809	1200
1	1	7354	1200	7903	1300
1	1	7448	1300	7997	1400
1	1	7542	1400	8091	1500
1	1	7636	1500	8185	1600
1	1	7730	1600	8279	1700
1	1	7824	1700	8373	1800
1	1	7918	1800	8467	1900
1	1	8012	1900	8560	2000

1	1	8105	2000	8654	2100
1	1	8199	2100	8747	2200
1	1	8292	2200	8841	2300
1	1	8386	2300	8934	2400
1	1	8400	2300	8900	2400
1	1	8400	2300	8900	7700
1	1	8479	2400	8900	2500
1	1	8479	2400	9028	2500
1	1	8530	2500	8900	2509
1	1	8530	2500	9079	2509
1	1	8576	2509	8900	2600
1	1	8576	2509	9035	2600
1	1	8666	2600	8900	2700
1	1	8666	2600	8933	2700
1	1	8752	2700	8835	2783
2	2	9900	1500	10400	7900
3	4	1700	1600	2200	7600
4	5	3200	1600	3700	7600
5	6	4700	1600	5200	7600

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	50	15	1416	PSD	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	21	12	33000	1	5	6
CPC	24	15	42300	1	5	5

* TIME *

CLOCK = 244
DELTA = 1

Output for commands 225 through 253 not shown

```
                                                                                     CC000254
                             N TRANSISTOR IDENTIFICATION / CHECKING                  CC000255
                             --------------------------------------                  CC000256
                                                                                     CC000257
    IFNL  NA,104                                           CC000258        <------------------    258
                          LOCATION OF PROSPECTIVE N CHANNELS, ARBITRARILY ASSUMING   CC000259
                          THAT ANY CHANNEL UP TO 1 MIL LONG WITH AT LEAST 0.1 MIL    CC000260
                          METAL AND THIN OXIDE WILL PERFORM SOMEWHAT LIKE A TRANSISTOR CC000261
    SPEC  MAXW=1000,R1=0,S1=# 1,R2=0,S2=# 1                CC000262        <------------------    262
    OPER  PNC = TWIX NA          DIFF                      CC000263        <------------------    263
    * NULL MASK *
        PNC

    # WARNING #
     CONDITION  8
    # WARNING #
     CONDITION  8

                                                                                          1 PNC

                                            1       1      1800      2200      2100      2900
                                            2       2      3300      2200      3600      2600
                                            3       3      4800      2200      5100      2600
                                            4       4      3300      2600      3600      2900
                                            5       5      4800      2600      5100      2900
                                            6       6      8500      6600      8800      6900
                                            7       7     10000      6600     10300      6900
                                            8       8      8500      6900      8800      7[illegible]00
```

9 9 10000 6900 10300 7300

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	54	10	1524	PGME	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	[illegible]00	1	9	161
PA	12	3	[illegible]300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
GPC	24	15	42300	1	5	5
PNC	25	17	46000	1	9	9

* TIME *

CLOCK = 265
DELTA = 2

C-47

```
IFNL  PNC,97                                      CC000264     <--------------------  264
SPEC  MINA=100                                    CC000265     <--------------------  265
OPER  NC = INTR PNC,T                             CC000266     <--------------------  266
```

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	54	10	1524	PGME	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
GPC	24	15	42300	1	5	5
PNC	26	17	48300	1	9	9
NC	27	18	51000	1	9	16

* TIME *

CLOCK = 266

DELTA = 1

SPEC MINA=100 CC000267 <-------------------- 267

ØPER NC = INTR NC,M CC000268 <-------------------- 268

* DIRECTØRY *

FILES

UNIT	SET	RECØRDS	TØTAL	NAME	TYPE
5	54	10	1524	PGME	1

MASKS

NAME	DIRECTØRY	FILE	ADDRESS	RECØRDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
NC	23	14	390[illegible]	2	9	6
GPC	24	15	42[illegible]	1	5	5
PNC	26	17	48300	1	9	9

C-48

* TIME *

CLØCK = 270

DELTA = 4

ØPER ONC = LINK PNC,NC NØNE,AREA CC000269 <-------------------- 269

\# WARNING \#

CØNDITIØN 8

\# WARNING \#

CØNDITIØN 8

* NULL MASK *

ONC

1 ONC

1	1	1800	2200	2100	2900
2	2	3300	2200	3600	2600
2	4	3300	2600	3600	2900
3	3	4800	2200	5100	2600
3	5	4800	2600	5100	2900
4	2	3300	2200	3600	2600
4	4	3300	2600	3600	2900
5	3	4800	2200	5100	2600
5	5	4800	2600	5100	2900

6	6	8500	6600	8800	6900
6	8	8500	6900	8800	7300
7	7	10000	6600	10300	6900
7	9	10000	6900	10300	7300
8	6	8500	6600	8800	6900
8	8	8500	6900	8800	7300
9	7	10000	6600	10300	6900
9	9	10000	6900	10300	7300

* DIRECTØRY *

FILES

UNIT	SET	RECØRDS	TØTAL	NAME	TYPE
5	64	10	1524	PIME	1

MASKS

NAME	DIRECTØRY	FILE	ADDRESS	RECØRDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
[illegible]	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
NC	23	14	39600	2	9	6
OPC	24	15	42300	1	5	5
QNC	25	16	45000	1	9	9
PNC	26	17	48300	1	9	9

* TIME *

CLØCK = 272
DELTA = 2

SPEC R121,S121 OPC
ØPER QNC = LINK QNC   LINE   CC000270 <-------------------- 270
CC000271 <-------------------- 271
\# WARNING \#
CØNDITIØN 8
\# WARNING \#
CØNDITIØN 8
* NULL MASK *
QNC

1 QNC

6	1	1800	2200	2100	2900
7	2	3300	2200	3600	2600
7	2	3300	2600	3600	2900
7	4	3300	2200	3600	2600
7	4	3300	2600	3600	2900

8	3	4800	2200	5100	2600
8	3	4800	2600	5100	2900
8	5	4800	2200	5100	2600
8	5	4800	2600	5100	2900
9	6	8500	6600	8800	6900
9	6	8500	6900	8800	7300
9	8	8500	6600	8800	6900
9	8	8500	6900	8800	7300
10	7	10000	6600	10300	6900
10	7	10000	6900	10300	7300
10	9	10000	6600	10300	6900
10	9	10000	6900	10300	7300

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	54	10	1524	PGME	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	[illegible]	1	9	200
T	16	7	[illegible]	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
NC	23	14	39600	2	9	6
CPC	24	15	42300	1	5	5
PNC	26	17	48300	1	9	9
ONC	28	19	54000	1	10	9

* TIME *

CLOCK = 273
DELTA = 1

Output for commands 272 through 300 not shown.

LOCATION OF N SOURCES AND DRAINS

CC000301
CC000302

SPEC R190,S19# 1
OPER NSD * SAME NA

CC000303 <-------------------- 303
CC000304 <-------------------- 304

1 NSD

1	10	2100	2200	3300	2900
2	10	5100	2200	6100	2600
3	10	5100	2600	6100	2900
4	10	800	2200	1800	2900
5	10	3600	2200	4800	2900
6	11	8800	6600	10000	6900
7	11	8800	6900	10000	7300
8	11	10300	6600	11200	6900
9	11	10300	6900	11200	7300
10	11	7500	6600	8500	6900
11	11	7500	6900	8500	7300

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	58	13	1594	NCE	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
NSD	22	13	36000	1	11	11
CPC	24	15	42300	1	5	5
GNC	28	19	54300	1	10	9

* TIME *

CLOCK = 285
DELTA = 1

OPER NSD * LINK NSD,GNC LINE/LINE

CC000305 <-------------------- 305

WARNING
CONDITION 8
WARNING
CONDITION 8
* NULL MASK *
NSD

1 NSD

6	1	2100	2200	3300	2900
6	4	800	2200	1800	2900
7	1	2100	2200	3300	2900
7	5	3600	2200	4800	2900
8	2	5100	2200	6100	2600
8	3	5100	2600	6100	2900
8	5	3600	2200	4800	2900
9	6	8800	6600	10000	6900
9	7	8800	6900	10000	7300
9	10	7500	6600	8500	6900
9	11	7500	6900	8500	7300
10	6	8800	6600	10000	6900
10	7	8800	6900	10000	7300
10	8	10300	6600	11200	6900
10	9	10300	6900	11200	7300

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5 ,	58	13	1594	NCE	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
NSD	23	14	39000	1	10	11
GPC	24	15	42300	1	5	5
GNC	28	19	54300	1	10	9

C-52

* TIME *

CLOCK = 286
DELTA = 1

OPER NSD = INTR NSD,NA   CC000306   <-------------------- 306

1 NSD

6	1	2100	2200	3300	2900
6	3	800	2200	1800	2900
7	1	2100	2200	3300	2900
7	3	3600	2200	4800	2900
8	1	5100	2200	6100	2600

8	1	5100	2600	6100	2900
8	3	3600	2200	4800	2900
9	3	8800	6600	10000	6900
9	3	8800	6900	10000	7300
9	9	7500	6600	8500	6900
9	9	7500	6900	8500	7300
10	3	8800	6600	10000	6900
10	3	8800	6900	10000	7300
10	8	10300	6600	11200	6900
10	8	10300	6900	11200	7300

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	55	13	1594	NCE	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
Y	16	7	18300	1	15	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
NSD	22	13	36000	1	10	9
GPC	24	15	42300	1	5	5
GNC	28	19	54300	1	10	9

* TIME *

CLOCK = 287
DELTA = 1

Output for commands 307 through 323 not shown.

```
                                                                            CC000324
                    LØCATIØN ØF N CHANNEL GATE METAL                        CC000325
SPEC  R1ØO,S1Ø# 1                                     CC000326         <--------------------   326
ØPER  NGM = SAME M                                    CC000327         <--------------------   327

                                                                                               1 NGM
```

1	122	7265	617	7348	700
2	123	2500	700	7600	1100
3	124	7168	700	7434	800
4	125	7066	800	7523	891
5	126	7021	891	7570	900
6	127	7072	900	7621	1000
7	128	7166	1000	7715	1100
8	129	2500	1100	2900	2800
9	130	5500	1100	5900	6400

10	131	7260	1100	7809	1200
11	132	7354	1200	7903	1300
12	133	7448	1300	7997	1400
13	134	7542	1400	8091	1500
14	135	7636	1500	8185	1600
15	142	7730	1600	8279	1700
16	143	7824	1700	8373	1800
17	144	7918	1800	8467	1900
18	145	8012	1900	8560	2000
19	146	8105	2000	8654	2100
20	147	8199	2100	8747	2200
21	148	8292	2200	8841	2300
22	149	8386	2300	8934	2400
23	150	8400	2300	8900	7900
24	152	8479	2400	9028	2500
25	154	8530	2500	9079	2509
26	155	8576	2509	9035	2600
27	156	8666	2600	8933	2700
28	157	8752	2700	8835	2783
29	136	9900	1500	10400	7900
30	137	1000	1600	1400	3500
31	140	4000	1600	4400	3500
32	159	9200	6000	9600	7900
33	138	1700	1600	2200	7600
34	139	3200	1600	3700	7600
35	141	4700	1600	5200	7600
36	151	7700	2400	8100	5000
37	158	1000	4500	1400	7100
38	153	10700	2400	11100	8400
39	161	7500	8400	12800	8800
40	160	7700	6700	8100	7500

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	60	20	1634	NSU	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PSM	21	12	33300	1	5	6
NSD	22	13	36300	1	10	9
NGM	23	14	39000	1	40	161
OPC	24	15	42300	1	5	5
ONC	28	19	54300	1	10	9

* TIME *

CLOCK = 298
DELTA = 1

OPER NGM = LINK NGM,QNC LINE/LINE CC000328 <------------------- 328

\# WARNING \#
CONDITION 8
\# WARNING \#
CONDITION 8
* NULL MASK *
NGM

1 NGM

6	33	1700	1600	2200	7600
7	34	3200	1600	3700	7600
8	35	4700	1600	5200	7600
9	1	7265	617	7348	700
9	2	2500	700	7600	1100
9	3	7168	700	7434	800
9	4	7066	800	7523	891
9	5	7021	891	7570	900
9	6	7072	900	7621	1000
9	7	7166	1000	7715	1100
9	8	2500	1100	2900	2800
9	9	5500	1100	5900	6400
9	10	7260	1100	7809	1200
9	11	7354	1200	7903	1300
9	12	7448	1300	7997	1400
9	13	7542	1400	8091	1500
9	14	7636	1500	8185	1600
9	15	7730	1600	8279	1700
9	16	7824	1700	8373	1800
9	17	7918	1800	8467	1900
9	18	8012	1900	8560	2000
9	19	8105	2000	8654	2100
9	20	8199	2100	8747	2200
9	21	8292	2200	8841	2300
9	22	8386	2300	8934	2400
9	23	8400	2300	8900	7900
9	24	8479	2400	9028	2500
9	25	8530	2500	9079	2509
9	26	8576	2509	9035	2600
9	27	8666	2600	8933	2700
9	28	8752	2700	8835	2783
10	29	9900	1500	10400	7900

* DIRECTORY *
FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	60	20	1634	NSD	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
NSD	22	13	36300	1	10	9
GPC	24	15	42300	1	5	5
NGM	25	16	45000	1	10	35
CNC	28	19	54300	1	10	9

* TIME *

CLOCK = 300
DELTA = 2

BPER NGM = INTR NGM,M

CC000329 <-------------------- 329

1 NGM

6	4	1700	1600	2200	7600
7	5	3200	1600	3700	7600
8	6	4700	1600	5200	7600
9	1	7265	617	7348	700
9	1	2500	700	7600	1100
9	1	7168	700	7434	800
9	1	7066	800	7523	891
9	1	7021	891	7570	900
9	1	7072	900	7600	1000
9	1	7072	900	7621	1000
9	1	7166	1000	7600	1100
9	1	7166	1000	7715	1100
9	1	2500	1100	2900	2800
9	1	5500	1100	5900	6400
9	1	7260	1100	7809	1200
9	1	7354	1200	7903	1300
9	1	7448	1300	7997	1400
9	1	7542	1400	8091	1500
9	1	7636	1500	8185	1600
9	1	7730	1600	8279	1700
9	1	7824	1700	8373	1800
9	1	7918	1800	8467	1900
9	1	8012	1900	8560	2000
9	1	8105	2000	8654	2100
9	1	8199	2100	8747	2200
9	1	8292	2200	8841	2300
9	1	8386	2300	8934	2400

9	1	8400	2300	8900	2400
9	1	8400	2300	8900	7900
9	1	8479	2400	8900	2500
9	1	8479	2400	9028	2500
9	1	8530	2500	8900	2509
9	1	8530	2500	9079	2509
9	1	8576	2509	8900	2600
9	1	8576	2509	9035	2600
9	1	8666	2600	8900	2700
9	1	8666	2600	8933	2700
9	1	8752	2700	8835	2783
10	2	9900	1500	10400	7900

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	60	20	1634	NSD	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
NSD	22	13	36300	1	10	9
NGM	23	14	39000	1	10	6
OPC	24	15	42300	1	5	5
ONC	28	19	54300	1	10	9

C-58

* TIME *

CLOCK = 301
DELTA = 1

Output for commands 330 through 403 not shown.

ERROR: MINIMUM SEPARATION OF STEPPED OPENING AND ANY CC000404
CONTACT OPENING = 0.1 MIL CC000405

OPER EXC = EXPN C 100,100 CC000406 <-------------------- 406

1 EXC

3	111	900	1600	1500	3500
3	112	3900	1600	4500	3500

1	113	2400	2400	3000	2900
1	114	5400	2400	6000	2900
7	115	7600	2400	8200	5000
8	116	10600	3100	11200	4300
7	117	900	4500	1500	7100
1	118	5400	5200	6000	6400
3	119	9100	6000	9700	7900
9	120	7600	6700	8200	7200
8	121	10600	6700	11200	7200

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	68	33	1968	NNSD	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27[illegible]0	1	7	197
PSD	20	11	3[illegible]300	1	5	12
PGM	21	12	33300	1	5	6
NSD	22	13	36300	1	10	9
NGM	23	14	39300	1	10	6
CPC	24	15	42300	1	5	5
NNSD	25	16	45300	1	7	197
PNSD	26	17	48600	2	3	71
EXC	27	18	51000	1	9	121
CNC	28	19	54300	1	10	9

C-59

* TIME *

CLOCK = 368
DELTA = 0

SPEC PRNT CC000407 <-------------------- 407
OPER CTOE = NINT EXC,T CC000408 <-------------------- 408

1 CTOE

1	114	5400	2800	6000	2900

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	68	33	1968	NNSD	1

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2

NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
NSD	22	13	36300	1	10	9
NGM	23	14	39300	1	10	6
GPC	24	15	42300	1	5	5
NNSD	25	16	45300	1	7	197
PNSD	26	17	48600	2	3	71
EXC	27	18	51300	1	9	121
GNC	28	19	54300	1	10	9
CTBE	29	20	57000	1	1	114

* TIME *

CLOCK = 369
DELTA = 1

Output for commands 409 through 474 not shown.

CALCULATE JUNCTION CAPACITANCES ASSUMING KA = 0.070 PF/SQ. CC000475
MIL AND KP = 0.200 PF/MIL IN UNITS OF .001 PF CC000476
CC000477

PARE *,EXP,=14286,=500
CC000478 <------------------- 478
* AREA AND PERIMETER * (ID#1,ID#2,AREA,EXPONENT,PERIMETER,EXPONENT)

1 PARE

3	3	27509504	-5	53600160	-6
3	4	24499584	-7	15000056	-7
3	5	45499248	-7	21000064	-7
3	6	18829664	-5	54600160	-6
3	7	18969664	-5	55000176	-6
3	8	27509504	-5	53600160	-6
3	9	83998624	-7	32000112	-7
3	10	83998624	-7	32000112	-7

3	11	78398672	-6	92000336	-7
3	12	24499568	-6	48000160	-7
3	13	24499568	-6	48000160	-7
3	14	78398672	-6	92000336	-7
3	15	83998624	-7	32000112	-7
3	16	83998624	-7	32000112	-7
3	17	83998624	-7	32000112	-7
3	18	83998624	-7	32000112	-7
3	19	83998624	-7	32000112	-7
3	20	83998624	-7	32000112	-7
3	21	83998624	-7	32000112	-7
3	22	83998624	-7	32000112	-7
3	23	83998624	-7	32000112	-7
3	24	83998624	-7	32000112	-7
3	25	83998624	-7	32000112	-7
3	26	24499568	-6	48000160	-7
3	27	24499568	-6	48000160	-7
3	28	83998624	-7	32000112	-7
7	29	11549795	-5	10400033	-6
10	30	13649759	-5	11200036	-6
8	31	11549795	-5	10400033	-6
3	32	83998624	-7	32000112	-7
3	33	83998624	-7	32000112	-7
3	34	83998624	-7	32000112	-7
3	35	34789360	-5	31200096	-6
3	36	83998624	-7	32000112	-7
3	37	83998624	-7	32000112	-7
3	38	83998624	-7	32000112	-7
3	39	83998624	-7	32000112	-7
3	40	83998624	-7	32000112	-7
3	41	83998624	-7	32000112	-7
3	42	83998624	-7	32000112	-7
3	43	83998624	-7	32000112	-7
3	44	69998848	-7	28000096	-7
3	45	69998848	-7	28000096	-7
3	46	97998400	-7	36000128	-7
3	47	11241811	-6	40120128	-7
3	48	11325810	-6	40360144	-7
3	49	11409808	-6	40600144	-7
3	50	11493807	-6	40840144	-7
3	51	11577806	-6	41080144	-7
3	52	11661804	-6	41320144	-7

51 PARE

3	53	11745803	-6	41560144	-7
7	54	11549795	-5	10400033	-6
11	55	13649759	-5	11200036	-6
12	56	13649759	-5	11200036	-6
1	57	11549795	-5	10400033	-6
3	58	11829801	-6	41800144	-7
3	59	11913800	-6	42040144	-7
3	60	12011799	-6	42320144	-7
3	61	12095797	-6	42560144	-7

3	62	12193796	-6	42840144	-7
3	63	10397615	-5	30508096	-6
3	64	10267418	-5	30136080	-6
3	65	10[illegible]35821	-5	29760080	-6
3	66	10005622	-5	29388080	-6
3	67	14909737	-5	29600080	-6
3	68	24499568	-6	48000160	-7
3	69	78398672	-6	92000336	-7
3	70	24499568	-6	48000160	-7
3	71	26459504	-5	24400064	-6

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	74	6	2006	TOXE	1
6	1	71	71	PAHE	0

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	216[illegible]	2	12	71
W	18	9	24[illegible]0	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
NSD	22	13	36300	1	10	9
NGM	23	14	39300	1	10	6
QPC	24	15	42300	1	5	5
EXP	25	16	45600	2	12	71
EXN	26	17	48000	1	9	200
ONC	28	19	54300	1	10	9
MTHK	30	21	60000	1	9	161
MTHN	31	22	63000	2	9	16

* TIME *

CLOCK = 422
DELTA = 2

Output for commands 479 through 483 not shown.

```
AREA   6,PTHK,-50000                                   CC000484        <--------------------  484
* AREA *    (ID#1,ID#2,AREA,EXPONENT)
```

1 AREA

1	1	12000043	-7	0	0
3	1	24000080	-7	0	0
3	1	24000080	-7	0	0
3	1	54900224	-8	0	0
3	1	10980036	-7	0	0
3	1	10980036	-7	0	0
3	1	16000054	-7	0	0
3	1	16000054	-7	0	0
3	1	54900224	-8	0	0
3	1	54900224	-8	0	0
3	1	10980036	-7	0	0
3	1	12000043	-7	0	0
3	1	16000054	-7	0	0
3	1	54900224	-8	0	0
3	1	54900224	-8	0	0

3	1	10980036	-7	0	0
3	1	40000160	-8	0	0
3	1	24000080	-7	0	0
3	1	54900224	-8	0	0
3	1	54900224	-8	0	0
3	1	10980036	-7	0	0
3	1	10980036	-7	0	0
3	1	54900224	-8	0	0
3	1	56000192	-7	0	0
3	1	20000064	-7	0	0
3	1	20000064	-7	0	0
3	1	20000064	-7	0	0
3	1	15000056	-7	0	0
3	1	50000208	-8	0	0
3	1	40000160	-8	0	0
3	1	20000064	-7	0	0
3	1	20000064	-7	0	0
3	2	50000208	-8	0	0
3	2	15000056	-7	0	0
3	2	30000112	-7	0	0
3	2	20000064	-7	0	0
3	2	20000064	-7	0	0
3	2	20000064	-7	0	0
3	2	15000056	-7	0	0
3	2	50000208	-8	0	0
3	4	45000160	-7	0	0
3	5	45000160	-7	0	0
3	6	45000160	-7	0	0
3	8	16000054	-7	0	0
3	8	16000054	-7	0	0
3	8	16000054	-7	0	0
3	8	16000054	-7	0	0
3	8	24000080	-7	0	0
3	8	56000192	-7	0	0
8	8	12000043	-7	0	0

51 AREA

8	8	12000043	-7	0	0

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	74	6	2006	TEXE	1
6	2	53	124	AREA	0

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
PG	14	5	12600	2	3	71
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110

P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
NG	19	10	27300	1	7	197
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
NSD	22	13	36300	1	10	9
NGM	23	14	39000	1	10	6
QPC	24	15	42000	1	5	5
EXP	25	16	45600	2	12	71
EXN	26	17	48000	1	9	200
PTHK	27	18	51600	2	8	8
GNC	29	19	54300	1	10	9
MTHK	30	21	60300	1	9	161
MTHN	31	22	63000	2	9	16

* TIME *

CLOCK = 426
DELTA = 2

Output for commands 485 through 510 not shown.

BOOL /6/PG:0/NG:1/PSD:PSD,:PGM/NSD:NSD,NGM/ CC000511 <-------------------- 511

WARNING
CONDITION 8
WARNING
CONDITION 8
* EQUATIONS *

					1 BOOL
5	1	12	6	2	0
6	1	3	4	0	1
7	1	3	5	0	1
8	1	3	6	0	1
0	3	0	0	1	2
6	3	1	4	0	1
7	3	1	5	0	1
8	3	1	6	0	1
9	3	9	1	0	1
10	3	8	2	0	1
0	7	0	0	0	2
1	7	10	1	2	1
3	7	11	4	2	1
2	8	10	2	2	0
10	8	3	2	0	1

```
                                              9        9       3       1       0       0
                                              1       10       7       1       2       0
                                              2       10       8       2       2       1
                                              3       11       7       4       2       0
                                              4       11      12       5       2       1
                                              4       12      11       5       2       0
                                              5       12       1       6       2       1

   1 =                 12 AND    =6                    ITEM     5
        ØR              3 AND     4                    ITEM     6
        ØR              3 AND     5                    ITEM     7
        ØR              3 AND     6                    ITEM     8

   3 =        ØFF
        ØR              1 AND     4                    ITEM     6
        ØR              1 AND     5                    ITEM     7
        ØR              1 AND     6                    ITEM     8
        ØR              9 AND     1                    ITEM     9
        ØR              8 AND     2                    ITEM    10

   7 =        ØN
        ØR             10 AND    =1                    ITEM     1
        ØR             11 AND    =4                    ITEM     3

   8 =                 10 AND    =2                    ITEM     2
        ØR              3 AND     2                    ITEM    10

   9 =                  3 AND     1                    ITEM     9

  10 =                  7 AND    =1                    ITEM     1
        ØR              8 AND    =2                    ITEM     2

  11 =                  7 AND    =4                    ITEM     3
        ØR             12 AND    =5                    ITEM     4

  12 =                 11 AND    =5                    ITEM     4
        ØR              1 AND    =6                    ITEM     5
```

* DIRECTØRY *

FILES

UNIT	SET	RECØRDS	TØTAL	NAME	TYPE
5	74	6	2006	TØXE	1
6	8	24	405	BØØL	0

MASKS

NAME	DIRECTØRY	FILE	ADDRESS	RECØRDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
NG	14	5	12300	1	7	7
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
PSD	20	11	30300	1	5	12

PGM	21	12	33300	1	5	6
NSD	22	13	36300	1	10	9
NGM	23	14	39300	1	10	6
GPC	24	15	42300	1	5	5
EXP	25	16	45600	2	12	71
EXN	26	17	48300	1	9	200
ALL	27	18	51900	3	12	200
ONC	28	19	54300	1	10	9
PG	29	20	57600	2	3	3
MTHK	30	21	60300	1	9	161
MTHN	31	22	63600	2	9	16

* TIME *

CLOCK = 459
DELTA = 5

MISCELLANEOUS LIST PROCESSING

CC000512
CC000513
CC000514
CC000515
CC000516

STORE DEVICE/NODE LIST

LIST /6/PGM;PSD,PSD/NGM;NSD,NSD/ CC000517 <-------------------- 517

\# WARNING \#
CONDITION 8
\# WARNING \#
CONDITION 8

* LIST *

1 2 3 4 5

1 LIST

1	1	0	0	0	1
1	1	0	7	0	1
1	1	0	10	0	1
1	1	7	0	0	1
1	1	7	7	0	1
1	1	7	10	0	1
1	1	10	0	0	1
1	1	10	7	0	1
1	1	10	10	0	1
2	2	0	0	0	1
2	2	0	8	0	1
2	2	0	10	0	1
2	2	8	0	0	1
2	2	8	8	0	1
2	2	8	10	0	1
2	2	10	0	0	1
2	2	10	8	0	1
2	2	10	10	0	1
3	4	0	0	0	1
3	4	0	7	0	1
3	4	0	11	0	1
3	4	7	0	0	1

3	4	7	7	0	1
3	4	7	11	0	1
3	4	11	0	0	1
3	4	11	7	0	1
3	4	11	11	0	1
4	5	0	0	0	1
4	5	0	11	0	1
4	5	0	12	0	1
4	5	11	0	0	1
4	5	11	11	0	1
4	5	11	12	0	1
4	5	12	0	0	1
4	5	12	11	0	1
4	5	12	12	0	1
5	6	1	1	0	1
5	6	1	12	0	1
5	6	12	1	0	1
5	6	12	12	0	1
6	4	0	0	0	2
6	4	0	1	0	2
6	4	0	3	0	2
6	4	1	0	0	2
6	4	1	1	0	2
6	4	1	3	0	2
6	4	3	0	0	2
6	4	3	1	0	2
6	4	3	3	0	2
7	5	0	0	0	2

51 LIST

7	5	0	1	0	2
7	5	0	3	0	2
7	5	1	0	0	2
7	5	1	1	0	2
7	5	1	3	0	2
7	5	3	0	0	2
7	5	3	1	0	2
7	5	3	3	0	2
8	6	0	0	0	2
8	6	0	1	0	2
8	6	0	3	0	2
8	6	1	0	0	2
8	6	1	1	0	2
8	6	1	3	0	2
8	6	3	0	0	2
8	6	3	1	0	2
8	6	3	3	0	2
9	1	0	0	0	2
9	1	0	3	0	2
9	1	0	9	0	2
9	1	3	0	0	2
9	1	3	3	0	2
9	1	3	9	0	2

9	1	9	0	0	2
9	1	9	3	0	2
9	1	9	9	0	2
10	2	3	3	0	2
10	2	3	8	0	2
10	2	8	3	0	2
10	2	8	8	0	2

• DIRECTØRY •

FILES

UNIT	SET	RECØRDS	TØTAL	NAME	TYPE
5	74	6	2006	TF XE	1
6	9	82	487	L,ST	0

MASKS

NAME	DIRECTØRY	FILE	ADDRESS	RECØRDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
NG	14	5	12300	1	7	7
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
NSD	22	13	36300	1	10	9
NGM	23	14	39300	1	10	6
QPC	24	15	42300	1	5	5
EXP	25	16	45600	2	12	71
EXN	26	17	48300	1	9	200
ALL	27	18	51300	3	12	200
GNC	28	19	54300	1	10	9
PG	29	20	57600	2	3	3
MTHK	30	21	60300	1	9	161
MTHN	31	22	63600	2	9	16

• TIME •

CLØCK = 463
DELTA = 4

PRINT LIST ØF CHANNEL DIMENSIØNS

RANG 6,GPC
• RANGE • (ID#1,ID#2,X,EXPØNENT,Y,EXPØNENT)

CC000519 CC000518 <-------------------- 519

1 RANG

1	1	30000064	-5	14000027	-4
2	2	30000064	-5	14000027	-4
3	3	30000064	-5	14000027	-4
4	4	30000064	-5	14000027	-4
5	5	30000064	-5	14000027	-4

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	74	6	2006	TEXE	1
6	10	7	494	RANG	0

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
NG	14	5	12300	1	7	7
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
NSD	22	13	36300	1	10	9
NGM	23	14	39300	1	10	6
OPC	24	15	42300	1	5	5
EXP	25	16	45600	2	12	71
EXN	26	17	48300	1	9	200
ALL	27	18	51900	3	12	200
ONC	28	19	54300	1	10	9
PG	29	20	57600	2	3	3
MTHK	30	21	60300	1	9	161
MTHN	31	22	63600	2	9	16

C-70

* TIME *

CLOCK * 464
DELTA * 1

RANG 6,ONC

* RANGE * (ID#1,ID#2,X,EXPONENT,Y,EXPONENT)

CC000520 <--------------------- 520

1 RANG

6	1	30000064	-5	70000160	-5
7	2	30000064	-5	40000080	-5
7	2	30000064	-5	30000064	-5
7	4	30000064	-5	40000080	-5
7	4	30000064	-5	30000064	-5
8	3	30000064	-5	40000080	-5
8	3	30000064	-5	30000064	-5
8	5	30000064	-5	40000080	-5
8	5	30000064	-5	30000064	-5
9	6	30000064	-5	30000064	-5
9	6	30000064	-5	40000080	-5
9	8	30000064	-5	30000064	-5
9	8	30000064	-5	40000080	-5

10	7	30000064	-5	30000064	-5
10	7	30000064	-5	40000080	-5
10	9	30000064	-5	30000064	-5
10	9	30000064	-5	40000080	-5

* DIRECTORY *

FILES

UNIT	SET	RECORDS	TOTAL	NAME	TYPE
5	74	6	2006	TBXE	1
6	11	19	513	RANG	0

MASKS

NAME	DIRECTORY	FILE	ADDRESS	RECORDS	IDENT # 1	IDENT # 2
NA	10	1	300	1	9	11
M	11	2	3300	1	9	161
PA	12	3	6300	1	12	12
C	13	4	9300	1	9	121
NG	14	5	12300	1	7	7
N	15	6	15300	1	9	200
T	16	7	18300	1	16	110
P	17	8	21600	2	12	71
W	18	9	24300	1	11	2
PSD	20	11	30300	1	5	12
PGM	21	12	33300	1	5	6
NSD	22	13	36300	1	10	9
NGM	23	14	39300	1	10	6
GPC	24	15	42300	1	5	5
EXP	25	16	45600	2	12	71
EXN	26	17	48300	1	9	200
ALL	27	18	51900	3	12	200
GNC	28	19	54300	1	10	9
PG	29	20	57600	2	3	3
MTHK	30	21	60300	1	9	161
MTHN	31	22	63600	2	9	16

* TIME *

CLOCK = 465
DELTA = 1

****

* END *
STEP 0

APPENDIX D

ROUTINES CALLING STRUCTURE

This appendix presents the routines calling structure of MAP. Table D-1 lists each inline main program routine with the other inline routines and the subprograms which each may call. Table D-2 lists each subprogram with the other subprograms each may call.

INLINE ROUTINES CALLING STRUCTURE

CALLING ROUTINE	INLINE ROUTINES CALLED		SUBPROGRAMS CALLED	
INIT			I02 I03 I07	I08 DEPEND LOCATN
ORTHO			OP1 SMASH I07	BOOK1 BOOK2
COMMIE	COMM FILE TEXT FREE OPER SPEC TRAC BOOL	LIST AREA PERI PARE RANG SKIP IFNL	I02 I07 I08	BOOK2 LOCATN
FINIT			I07	
COMM			I07	
FILE			OP1 I03 I06A I06B	I06C I06D BOOK1
TEXT			OP1 I03 I06A I06B	I06C I06D BOOK1
FREE			I03	BOOK2
OPER	SAME NGTV EDGE EXPN PLUS INTR NINT EXOR LINK1 LINK2	NLNK TWIX1 TWIX2 SPIN FLIP PUSH SCAL WNDW PLAC	I03 I08	BOOK1 GEOM

Table D-1

INLINE ROUTINES CALLING STRUCTURE
(continued)

CALLING ROUTINE	INLINE ROUTINES CALLED	SUBPROGRAMS CALLED	
SPEC		I03 I08	BOOK1
TRAC	LINK1	OP1 OP2 OP3 OP4	I02 I03 BOOK1 BOOK2
BOOL		OP1 OP2 OP3 I02 I03	I06A I07 BOOK1 BOOK2
LIST		OP1 OP2 OP3 I02 I03	I06A I07 BOOK1 BOOK2
AREA		OP1 I03 I06A	I07 BOOK1
PERI		OP1 I03 I06A	I07 BOOK1
PARE		OP1 I03 I06A	I07 BOOK1
RANG		OP1 I03 I06A	I07 BOOK1

Table D-1
(continued)

INLINE ROUTINES CALLING STRUCTURE
(continued)

CALLING ROUTINE	INLINE ROUTINES CALLED		SUBPROGRAMS CALLED	
SKIP			I02	I03
IFNL	SKIP		I03	BOOK1
SAME			OP1	
NGTV	NINT		I05	BOOK1
EDGE			OP1 OP4	BOOK1
EXPN			OP1	
PLUS			OP1	OP2
INTR			OP3	
NINT			OP1	OP4
EXOR	NINT	PLUS	BOOK1	
LINK1			OP1 OP3 OP4	BOOK1 BOOK2
LINK2	LINK1		OP1 OP3	BOOK1
NLNK	LINK1		OP1 OP3	OP4 BOOK1
TWIX1			OP1 OP2 OP4	BOOK1 BOOK2 GEOM
TWIX2	TWIX1		OP2	BOOK1

Table D-1
(continued)

INLINE ROUTINES CALLING STRUCTURE
(continued)

CALLING ROUTINE	INLINE ROUTINES CALLED	SUBPROGRAMS CALLED
SPIN		OP1
FLIP		OP1
PUSH		OP1
SCAL		OP1
WNDW		OP1
PLAC		OP3

Table D-1
(continued)

CALLING SUBPROGRAM	SUBPROGRAMS CALLED		
OP1	I04 I05 ORDER1	ORDER2 ORDER3 ORDER4	LOCATN
OP2	I04 I05 ORDER1	ORDER2 ORDER3 ORDER4	LOCATN
OP3	I04 I05 ORDER1	ORDER2 ORDER3 ORDER4	LOCATN
OP4	I04 I05 ORDER1	ORDER2 ORDER3 ORDER4	BOOK1 BOOK2 LOCATN
SMASH	I01 I01A I01B	I01C I01D I01E	I05
I01	LOCATN	DEPEND	
I01A	I08	LOCATN	DEPEND
I01B	I08		
I01C	I08		
I01D			
I01E	I08		
I02	I08		
I03	I08		
I04	I08	DEPEND	LOCATN
I05	I06A I06B I06C	I06D I07 I08	LOCATN DEPEND

Table D-2

SUBPROGRAMS CALLING STRUCTURE
(Continued)

CALLING SUBPROGRAM	SUBPROGRAMS CALLED		
I06A	I07	I08	
I06B	I08		
I06C	I08		
I06D	I08	DEPEND	
CALLING SUBPROGRAM	SUBPROGRAMS CALLED		
I07	DEPEND		
I08			
ORDER1	ORDER2		
ORDER2	I05 I07	ORDER5 LOCATN	
ORDER3	I04 I05	ORDER5 BOOK1	
ORDER4			
ORDER5			
BOOK1	I08		
BOOK2	LOCATN		
GEOM	OP1 OP4	I05 BOOK1	BOOK2
DEPEND			
LOCATN			

Table D-2
(Continued)

APPENDIX E

DESCRIPTIONS OF PROGRAM VARIABLES

This appendix presents descriptions of MAP variables. Further knowledge of the variables may be gained from the liberal comments found in the MAP Source Listing.

Table E-1 presents each variable name and a brief definition. All variables are integer type unless otherwise noted. All are in blank common or equivalent to blank common variables unless identified as local. Equivalence is indicated by "variable = variable." Arrays are listed subscripted by their dimension. A subscript of "I" indicates the array dimension may be varied for different versions of the program.

VARIABLES DEFINITIONS

VARIABLE	DESCRIPTION
AL	Alternate command input logical unit number: AL = NUM1(4).
AP(24)	Real I01B local variable containing line widths for type 1 input data format.
BEG(8)	Index to LIST array for the beginning of each list segment.
BEGO(3)	Index to LIST array for the beginning of each ordered list.
B1	Single blank character.
B4	Real scalar containing four blank characters.
C(15)	Real MAIN local array containing the 4-character command names.
CARD(76)	Command image buffer storing columns 5-80.
CELL	Local I01E scalar containing a cell identifier from type 4 input data.
CHAR(11)	Local I06D array containing type 3 character design file entry words.
CHARAC	Number of characters which can be stored in word.
CON1	Real MAIN local variable containing the characters "OPTN".
CON2	Real MAIN local variable containing the characters "MASK".
COUNT(I)	Each entry is a count of records input or output for the mask occupying file location I.
CR	Normal command input logical unit number. CR = NUM1(1).

Table E-1

VARIABLES DEFINITIONS
(Continued)

VARIABLE	DESCRIPTION
D(7)	Local I03 array containing single characters considered delimiters on command records.
DATA(12)	Dummy array for temporary storage of output items: DATA(12) = last word of LIST.
DATA1 - DATA12	Scalar values for DATA array items: DATA1 = DATA(1).
DELIM(36)	Contains an index to D array for each delimiter preceding a valid field on a command image.
DESTIN	Four-character mask destination name.
DIG(11)	Local I06D array used for storing identifier digits for output as type 3 character definitions.
DIREND	Length of directory, i.e., upper bound for NAME, NUM1, NUM2, COUNT, and RECORD arrays.
DX1, DX2, DY1, DY2	Local SMASH scalars used for storing delta x and y values during smashing.
END(8)	Index to LIST array for the end of each list segment.
ENDO(3)	Index to LIST array for the end of each ordered list.
ENTRY(6)	Contains the priority of words considered when ordered lists are created.
FACT1 - FACT4	Factors used in the formula to calculate mask file record addresses.
FIELD (34)	Real array containing the valid fields located on a command image.
FILE(8)	Contains the file position of the masks associated with each LIST segment at any time.
FPX, FPY	Local I01E scalars containing polygon first point coordinates from type 4 input data.
I	Widely used local dummy scalar variable.

Table E-1
(Continued)

VARIABLES DEFINITIONS
(Continued)

VARIABLES	DESCRIPTION
IN	Graphic input data logical unit number: IN = NUM1(2).
INP1	File position of primary input mask.
INP2	File position of secondary input mask.
ITEMS	Number of six-word items per list segment.
J	Widely used local dummy scalar variable.
K	Widely used local dummy scalar variable.
L	Widely used local dummy scalar variable.
LARGE	Largest positive integer value.
LAST	Four-character name of the previously processed command: LAST = NAME(2).
LEND	Length of COMMON from BEG(1) through DATA12.
LENGTH	Record length in words for special list output.
LIST (I)	Main list for processing mask data.
LOC(9)	Index to LIST array for the current processing location for each list segment.
LOCO(3)	Index to LIST array for the current processing location for each ordered list.
LP	Printer formatted output logical unit number: LP = NUM1(3).
LPX, LPY	Local I01E scalars containing polygon last point coordinates from type 4 input data.
M	Widely used local dummy scalar variable.
MACHIN	Number of bits per word.

Table E-1
(Continued)

VARIABLES DEFINITIONS
(continued)

VARIABLES	DESCRIPTION
MASK	Four-character mask name.
MAXIM(6)	Maximum specified dimensions in the x, y, length, width, radial, or any directions: MAXIM(1) = MAX1.
MAX1 - MAX6	Scalar values for MAXIM array items.
MINIM(6)	Minimum specified dimensions in the x, y, length, width, radial, or all directions: MINIM(1) = MIN1.
MIN1 - MIN6	Scalar values for MINIM array items.
MODE	General processing mode.
MODE1	Processing mode for subroutines I01, I01A, I01B, I01C, I01D, and I01E.
MODE2	Processing mode for subroutine I02.
MODE3	Processing mode for subroutine I03.
MODE4	Processing mode for subroutine I04.
MODE5	Processing mode for subroutine I05.
MODE6	Processing mode for subroutines I06A, I06B, I06C, and I06D.
MODE7	Processing mode for subroutine I07.
MODE8	Processing mode for subroutine I08.
MU	Logical unit number for the mask storage file.
N	Widely used local dummy scalar variable.

Table E-1
(continued)

VARIABLES DEFINITIONS
(continued)

VARIABLES	DESCRIPTION
NAME(I)	Real array where each entry is the 4-character name for the mask occupying file location I.
NEXT	Indicator of additional steps in a string of processes.
NUM1(I)	Each entry contains the greatest primary identifier of the mask occupying file position I.
NUM2(I)	Each entry contains the greatest secondary identifier of the mask occupying file position I.
NXT	ORDER2, ORDER3, GEOM, and OP4 local storage of the scalar NEXT.
O(22)	Real MAIN local array containing the 4-character OPER command names.
OPTION(30)	Array encompassing a group of scalars: OPTION(1) = OPTN1, OPTION(30) = SEQ10.
OPTN1	Print option value.
OPTN2	Alternate input unit option.
OPTN3	Data mode option.
OPTN4	Scale factor option.
OPTN5	Offset override option.
OPTN6	Smash factor option.
OR	Value indicating list ordering priority pattern.
ORD	Ordered list segment length.
ORX, ORY	Local I01E scalars containing cell origin coordinates from type 4 input data.
OUT1	Mask output file position.

Table E-1
(Continued)

VARIABLES DEFINITIONS
(continued)

VARIABLES	DESCRIPTION
OUT2	Ordered mask output file position.
OUT3	Special list output logical unit.
OUT4	Printed output logical unit.
OUT5	Mask output file position.
P	Widely used local dummy scalar variable.
PARAM(8)	Real local I01B array storing parameters of type 1 input data prior to smashing.
PASS	Pass counter for multiple pass processes: PASS = NUM2(2).
PAT1	Path code for overall operations.
PAT2	Path code for a specific operation step.
PAT3 - PAT5	Miscellaneous path flags.
POINT(12)	Real local I01B array storing type 1 input data coordinates prior to smashing.
POINT(4)	Real local I01C array storing type 2 input data coordinates prior to smashing.
POINT(4)	Real local I01D array storing type 3 input data coordinates prior to smashing.
P1 - P4	Real local I01E scalars storing type 4 input data coordinates prior to smashing.
P1 - P6	Real local I01D scalars storing type 3 input data coordinates prior to smashing.
P5 - P8	Real local I01C array storing type 2 input data coordinates prior to smashing.
Q	Widely used local dummy scalar variable.
R	Widely used local dummy scalar variable.

Table E-1
(Continued)

VARIABLES	DESCRIPTION
REAL1 - REAL2	Real scalars local to OP1 and OP4 used to store floating point numbers resulting from mathematical operations.
RECORD(I)	Each entry contains a record count for the mask occupying file position I.
S(35)	Real MAIN local array containing the 4-character SPEC command forms.
S(36)	Local I06D array containing default information for the first record of a design file.
SCALE	Real local I01E scalar storing type 4 input data scale.
SEG	General list segment number.
SEG1	Primary input mask list segment number.
SEG2	Secondary input mask list segment number.
SEG3	Unordered mask output segment number.
SEG4	Ordered mask output segment number.
SEG5	Residue list segment number.
SEQ1	Preliminary identifier action code.
SEQ2	Final primary identifier action code.
SEQ3	Final secondary identifier action code.
SEQ4	Index to LIST array of identifiers to be considered for modification.
SEQ5	Primary identifier incrementing value.
SEQ6	Secondary identifier incrementing value.
SEQ7	Previous primary identifier replaced.
SEQ8	Previous secondary identifier replaced.

Table E-1
(Continued)

VARIABLES DEFINITIONS
(Continued)

VARIABLES	DESCRIPTION
SEQ9	Previous primary replacement identifier.
SEQ10	Previous secondary replacement identifier.
SETEND	Upper bound on SETUP array.
SETUP(I)	Array used to store information for multiple step processes.
SET1 - SET5	Scalar values for first five entries of SETUP arrays: SET1 - SETUP(1).
SKIP	Record skip flag for reading mask input data.
SPEC(5)	Array form of specification codes: SPEC(1) = SPEC1.
SPEC1	Printed output specification code.
SPEC2	Temporary storage specification code.
SPEC3	Identifier assignment specification code.
SPEC4	Minimum dimensional boundary specification code.
SPEC5	Maximum dimensional boundary specification code.
START	Address of first entry in an ordered list.
STAT	General I/O status flag.
STATE	Real I07 local array containing "ON" and "OFF" for Boolean printout.
STATUS(8)	Contains the I/O status of each list segment.
STRING(3)	Real I03 local array into which valid field command characters are encoded.
T(10)	Real MAIN local array containing the 4-character names of operation options.

Table E-1
(Continued)

VARIABLES	DESCRIPTION
TEMP(12)	I03 local array used to temporarily store command image characters.
TEST	Indicator for dimensional testing steps in a process string.
TIME	DEPEND local real scalar storing internal clock time.
TYPE	Real 4-character command type.
T1, T2	Local I01E scalars containing characters identifying valid type 4 format input records.
T1 - T6	Real I01B local scalars containing four characters identifying valid type 1 format input records.
UNIT	Mask file position indicator.
VALUE(6)	Miscellaneous values storage: VALUE(3) - VALUE(6) contain coordinates of the permissible window of masks for the run: VALUE(1) = VAL1.
VAL1 - VAL6	Scalar names for VALUE array items.
WORDS	Number of words in a list segment.
X	Local I01C scalar containing the character "X".
Z(11)	Local I03 array containing the characters "0" - "9" and "-".

Table E-1
(Continued)

☆U.S. GOVERNMENT PRINTING OFFICE 1976—641-255/19 REGION NO. 4