इंटरनेट मानक

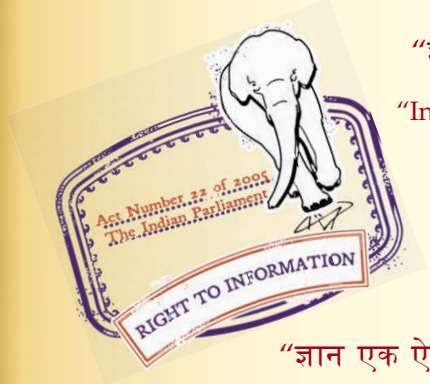
Disclosure to Promote the Right To Information

Whereas the Parliament of India has set out to provide a practical regime of right to information for citizens to secure access to information under the control of public authorities, in order to promote transparency and accountability in the working of every public authority, and whereas the attached publication of the Bureau of Indian Standards is of particular interest to the public, particularly disadvantaged communities and those engaged in the pursuit of education and knowledge, the attached public safety standard is made available to promote the timely dissemination of this information in an accurate manner to the public.

"जानने का अधिकार, जीने का अधिकार"
Mazdoor Kisan Shakti Sangathan
"The Right to Information, The Right to Live"

"पुराने को छोड नये के तरफ"
Jawaharlal Nehru
"Step Out From the Old to the New"

IS 15214 (2002): Generic Requirements for Implementation of Product Manufacturing Description Data and Transfer Methodology [LITD 5: Semiconductor and Other Electronic Components and Devices]



"ज्ञान से एक नये भारत का निर्माण"
Satyanarayan Gangaram Pitroda
"Invent a New India Using Knowledge"

"ज्ञान एक ऐसा खजाना है जो कभी चुराया नहीं जा सकता है"
Bhartṛhari—Nītiśatakam
"Knowledge is such a treasure which cannot be stolen"

BLANK PAGE

IS 15214 : 2002
IEC/PAS 62119 (1999)

भारतीय मानक

निर्मित उत्पाद के आंकड़ों के वर्णन और अन्तरण प्रणाली को कार्यान्वित करने की सामान्य आवश्यकताएँ

Indian Standard

GENERIC REQUIREMENTS FOR IMPLEMENTATION OF PRODUCT MANUFACTURING DESCRIPTION DATA AND TRANSFER METHODOLOGY

ICS 31.180

BUREAU OF INDIAN STANDARDS
MANAK BHAVAN, 9 BAHADUR SHAH ZAFAR MARG
NEW DELHI 110002

September 2002

Price Rs 800.00

NATIONAL FOREWORD

This Indian Standard which is identical with IEC/PAS 62119(1999) 'Generic requirements for implementation of product manufacturing description data and transfer methodology' issued by the International Electrotechnical Commission (IEC) was adopted by the Bureau of Indian Standards on the recommendation of Printed Circuits Sectional Committee and approval of the Electronics and Telecommunication Division Council.

The text of the IEC/PAS document has been approved as suitable for publication as Indian Standard without deviations. Certain conventions are, however, not identical to those used in Indian Standards. Attention is particularly drawn to the following:

a) Wherever the words 'International Standard' appear referring to this standard, they should be read as 'Indian Standard'.

b) Comma (,) has been used as a decimal marker while in Indian Standards, the current practice is to use a point (.) as the decimal marker.

CROSS REFERENCES

The technical committee responsible for the preparation of this standard has reviewed the provisions of the following International Standards and has decided that they are acceptable for use in conjunction with this standard:

IPC-T-50		Terms and definitions for interconnecting and packaging electronic circuits
IPC-2512	(ADMIN)	Sectional requirements for implementation of administrative methods for manufacturing data description
IPC-2513	(DRAWG)	Sectional requirements for implementation of drawing methods for manufacturing data description
IPC-2514	(BDFAB)	Sectional requirements for implementation of printed board fabrication data description
IPC-2515	(BDTST)	Sectional requirements for implementation of bare board product electrical testing data description
IPC-2516	(BDASM)	Sectional requirements for implementation of assembled board product manufacturing data description
IPC-2517	(ASEMT)	Sectional requirements for implementation of assembly in-circuit testing data description
IPC-2518	(PTLST)	Sectional requirements for implementation of part list product data description
IPC-2519	(MODEL)	Sectional requirements for information model data related to the printed board and printed board manufacturing description
IPC-2521		PWB fabrication data quality rating system
IPC-2522		Bareboard electrical test data quality rating system
IPC-2523		Printed board assembly data quality rating system
IPC-2524		In-circuit test data quality rating system
IPC-4101		Laminate/prepreg materials standard for printed boards

As in the Original Standard, this Page is Intentionally Left Blank

TABLE OF CONTENTS

Indian Standard

GENERIC REQUIREMENTS FOR IMPLEMENTATION OF PRODUCT MANUFACTURING DESCRIPTION DATA AND TRANSFER METHODOLOGY

1 SCOPE

This standard specifies data file formats used to describe printed board and printed board assembly products with details sufficient for tooling, manufacturing, assembly, inspection and testing requirements. These formats may be used for transmitting information between a printed board designer and a manufacturing or assembly facility. The files are also useful when the manufacturing cycle includes computer-aided processes and numerical control machines.

The information can be used for both manual and for digital interpretations. The data may be defined in either English or SI units.

1.1 Interpretation

"**Shall**", the emphatic form of the verb, is used throughout this standard whenever a requirement is intended to express a provision that is mandatory. Deviation from a **shall** requirement is not permitted, and compliance to test modules (CTMs), developed to check syntax and semantics, is required to ensure the correct relationships between entities in a file.

The words "should" and "may" are used whenever it is necessary to express non-mandatory provisions.

"Will" is used to express a declaration of purpose.

To assist the reader, the word **shall** is presented in bold characters.

1.2 GenCAM Focus

The GenCAM format requirements are provided in a series of standards focused on printed board manufacturing, assembly, inspection, and testing. This standard series consists of a generic standard (IPC-2511) which contains all the general requirements. There are seven sectionals that are focused on the details necessary to accumulate information in the single GenCAM file, that addresses the needs of the manufacturing disciplines producing a particular product. The sectional standards (IPC-2512 through 2518) paraphrase the important detailed requirements and provide suggested usage and examples for the topic covered by the sectional standard.

2 APPLICABLE DOCUMENTS

The following documents contain provisions which, through reference in this text, constitutes provisions of the IPC-2510 series of standards. All documents are subject to revision and parties who make agreements based on this generic standard are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below.

IPC-T-50		Terms and Definitions for Interconnecting and Packaging Electronic Circuits
IPC-2512	(ADMIN)	Sectional Requirements for Implementation of Administrative Methods for Manufacturing Data Description
IPC-2513	(DRAWG)	Sectional Requirements for Implementation of Drawing Methods for Manufacturing Data Description
IPC-2514	(BDFAB)	Sectional Requirements for Implementation of Printed Board Fabrication Data Description
IPC-2515	(BDTST)	Sectional Requirements for Implementation of Bare Board Product Electrical Testing Data Description

IPC-2516	(BDASM)	Sectional Requirements for Implementation of Assembled Board Product Manufacturing Data Description
IPC-2517	(ASEMT)	Sectional Requirements for Implementation of Assembly In-Circuit Testing Data Description
IPC-2518	(PTLST)	Sectional Requirements for Implementation of Part List Product Data Description
IPC-2519	(MODEL)	Sectional Requirements for Information Model Data Related to the Printed Board and Printed Board Manufacturing Descriptions
IPC-2521		PWB Fabrication Data Quality Rating System
IPC-2522		Bareboard Electrical Test Data Quality Rating System
IPC-2523		Printed Board Assembly Data Quality Rating System
IPC-2524		In-Circuit Test Data Quality Rating System
IPC-4101		Laminate/Prepreg Materials Standard for Printed Boards

3 REQUIREMENTS

The generic computer-aided manufacturing (GenCAM) format specifies data sections specifically for information interchange of data related to printed board manufacturing, assembly, inspection, and test.

GenCAM is comprised of twenty sections. These are shown in Tables 3-1 and 3-2.

Each section has a specific function or task respectively and is independent of each other. Accordingly, the information interchange for a specific purpose is possible only if the sections required for such a purpose have been prepared.

3.1 Category classification

Table 3-1 provides the name and purpose of each of the twenty sections.

Table 3-1 GenCAM Section Descriptions

Section Keyword	Purpose and Content
HEADER	Beginning of each file; includes name, company file type, number, revision, etc.
ADMINISTRATION	Describes the ordering information necessary for identifying responsibility, quantity of ordered parts and delivery schedule.
FIXTURES	Describes fixturing for bare board and assembled board testing.
PANELS	Panelization; includes description of manufacturing panels for printed boards and description of assembly arrays.
BOARDS	Description of board and coupons; includes outline of board/coupon, cutouts, tooling holes.
DRAWINGS	Describes engineering drawing and formatting requirements for complete printed board and printed board assembly descriptions.
PRIMITIVES	Describes simple and complex primitive physical shapes; includes lands, holes, standard patterns.
ARTWORKS	Functional and nonfunctional geometries developed by the user, i.e. user macros; includes shapes, logos, features not part of the circuit and other user-defined figures.
PATTERNS	Description for users to build libraries of reusable pads and padstacks.
MECHANICALS	Mechanical information; includes handles, nuts, bolts, heat sinks, brackets, holes.
LAYERS	Board manufacturing descriptions; includes conductive/nonconductive layer definition of silk screens, details of dielectric tolerances, separations and thickness.
PADSTACKS	CAD system data; includes pads and drilling information through the board.
PACKAGES	Describes a library of component packages; includes true package dimensions.
FAMILIES	Is used to describe logic families of components.
DEVICES	Component description; includes device part number.
COMPONENTS	Used to identify components; includes reference designators where appropriate.

Section Keyword	Purpose and Content
POWER	Power and ground; includes power injection parameters expected and permitted.
ROUTES	Conductor location information; includes the location of conductors on all layers.
TESTCONNECTS	Test point location; includes probe points, single name test point types and tester pin assignments.
CHANGES	Shows change data related to the existing design previously sent to the manufacturing site.

Section heading keywords **shall** be preceded by a dollar sign ($). Each section **shall** end with the header name preceded by the $END (i.e. $HEADER and $ENDHEADER). All keywords belonging to a section should be indented a minimum of two (2) spaces between the beginning and ending of a section in order to pass the readability test of the compliance test module. Once the indenture is defined (exact number of spaces), the practice will be consistent throughout the GenCAM file. The Horizontal Tab character (ASCII decimal 9) is not permitted for this requirement.

Nesting of sections **shall not** be permitted.

See Appendix A for a summary of section heading and ending keywords.

3.1.1 Categories and Content

Table 3-2 provides the section identifiers, and the relationship of which sections are necessary to support each of the significant seven standards related to the GenCAM format. The significant seven data sets and standards are administration, drawings, printed board fabrication, printed board test, assembly, assembled board inspection/test, and parts list. (IPC-2512 through IPC-2518)

Table 3-2 Section Name Relationships

Section Identifier	ADMIN	DRAWG	BDFAB	BDTST	BDASM	ASEMT	PTLST
HEADER	X	X	X	X	X	X	X
ADMINISTRATION	X	X	X	X	X	X	X
FIXTURES			X	X	X	X	
PANELS		X	X	X	X	X	
BOARDS		X	X	X	X	X	X
DRAWINGS	X	X	X	X	X	X	X
PRIMITIVES		X	X		X	X	
ARTWORKS		X	X		X		
PATTERNS		X	X	X	X	X	
MECHANICALS		X	X	X	X	X	X
LAYERS		X	X	X	X	X	
PADSTACKS				X	X	X	
PACKAGES		X		X	X	X	X
FAMILIES		X		X	X	X	X
DEVICES		X			X	X	X
COMPONENTS		X	X	X	X	X	X
POWER						X	
ROUTES		X	X	X	X	X	
TESTCONNECTS				X		X	
CHANGES	X*	X*	X*	X*	X*	X*	X*

* The CHANGES section is used independently to alter previously sent files. Included shall be a HEADER section (for revision status and identification) and an ADMINISTRATION section to show effectivity

Note: The letter "X" indicates general usage requirements. Each sectional standard provides additional granularity, and indicates mandatory and optional data ability requirements.

The twenty sections can be used in various combinations to provide information for manufacturing processes in printed board fabrication and printed board assembly characteristics. In each of the

significant seven data sets, there may be subsets to make up details for a specific process or equipment. An example of this might be a stencil file that is used to deposit solder paste on a printed board prior to assembly. Only specific information needed to produce the stencil would be necessary in order to establish the details for stencil manufacturing. The details of these characteristics are described within one of the significant seven standards in which that particular process applies.

It should be noted that in situations where a specific process is defined (i.e. stencil fabrication, glue dot positioning), some sections would be mandatory in order to provide the appropriate information, whereas some sections might be optional depending on the conditions of the particular manufacturing discipline.

3.2 Character Definition

The text used in the file for keywords and fixed field parameters **shall** be ASCII characters in English, in the range space (ASCII decimal 20) to ~ (ASCII decimal 126). The only control characters allowed are carriage return (ASCII decimal 13) and line feed (ASCII decimal 10). All other control characters, including TAB (ASCII decimal 9), and DEL (ASCII decimal 127) **shall not** be allowed. Each text line, including the very last line in the file, **shall** be terminated with at least one line feed character. Carriage returns are optional. Extra line feeds and carriage returns can be included when the file is generated but will be ignored when the file is read. A text end of file marker, such as control Z (ASCII decimal 26), should not be included in the file data.

Freeform text strings may include special native language character sets in any language that is documented in an ISO standard or has ISO registration (for examples of the ISO standards, see Table 3-3), provided that the character set code has been identified in the HEADER section.

Table 3-3 Example of Native Language Codes

Language	Character Set Designation	Standard	ISO Registration Number
Eastern European (Latin 2)	8-bit character set	ISO 8859-2	
English	ANSI ASCII	ANSI X3.4	006*
English	ISO 7-bit coded character set	ISO 646	002*
French			025
German			021
Italian			015
Katakana (Japanese)	JIS code	JIS X0201	013
Latin/Greek	8 bit character set	ISO 8859-7	
Norwegian – 1			060
Norwegian – 2			061
Portuguese			016
Spanish			017
Swedish			010
Western European (Latin 1)	8 bit character set	ISO 8859-1	

* ISO registration number 006 is equivalent to 002

3.3 Keyword Rules

All keywords **shall** be uppercase. There are section keywords and statement keywords. Section keywords are preceded by a $ (ASCII decimal 36); all statement keywords are followed by a colon (ASCII decimal 58) which separates them from the parameters identified as part of the keyword characteristics. There **shall** be one keyword per line. All section statements (except blank lines) **shall** start with a keyword and end with semicolon (ASCII decimal 59), but can span multiple text lines. See Appendix B for a listing of statement keywords.

Throughout this document, the word " keyword" **shall** refer to the capitalized reserved word that names the statement. The phrase "keyword statement" **shall** refer to the entire statement, including the keyword, all of the parameters, and ending with the semi-colon character.

3.4 Parameter Rules

Parameters are separated from each other by a comma character (ASCII decimal 44). The occurrence of optional but omitted parameters will be denoted by the appearance of a ",," (comma-comma) sequence except where the omitted parameter field(s) extend to the end of the line. In this case the terminating ";" (semicolon) is sufficient to denote the omitted parameter(s).

A fixed field parameter **shall** have the purpose defined in this specification (see Appendix C). All fixed field parameters **shall** be in upper case (capitals). A free field (string) can be any text that contains the characters as defined in the parameter description. All freeform alpha-numeric fields **shall** be contained in " (double-quote) character (ASCII decimal 34) pairs. When the string itself contains a double-quote character, that character **shall** be double-quoted.

Example:

"BOARD""Synthesizer""Assembly" could be the title of an assembly. This format indicates that the word "synthesizer" is surrounded by quotes.

All free fields, such as component names, can use upper or lower case and are always case sensitive. For example the component names "C1" and "c1" represent two different components.

3.4.1 Alpha-numeric Parameters

The following rules apply with the parameters identified as being a string of alphanumeric characters. Also identified are the rules for numbers and the GenCAM date format.

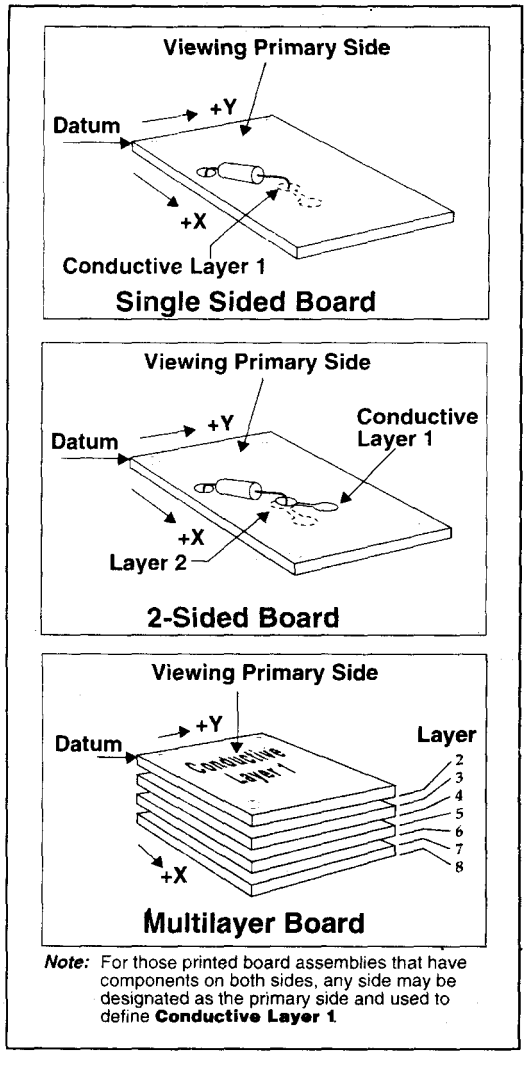
string	::=	SPACE (ASCII decimal 32) to ~ (ASCII decimal 126) inclusive This is the default character set and its use is assumed, unless another has been declared using the CHARACTERSET statement in the HEADER section of the GenCAM file.
number	::=	$3.4x10^{-38} \leq$ value $\leq 3.4x10^{38}$ where value can be positive, negative, integer or floating point, with 7 digit precision. Numbers are assumed to be positive but can be explicitly designated as positive by a preceding + (ASCII decimal 43) character. Negative numbers must be explicitly designated as negative by a preceding - (ASCII decimal 45) character. Exponentiation format **shall** be [±][0-9]*.[0-9]*[eE][0-9]*
p_number	::=	$0.0 \leq$ value $\leq 3.4x10^{38}$
p_integer	::=	1{0 1 2 3 4 5 6 7 8 9}n
date_format	::=	yyyy-mm-dd (numeric) - yyyy = year, mm = month, dd = day i.e., 1997-02-23

3.5 Coordinate Rules

The GenCAM file uses Cartesian coordinates, where the x coordinates become more positive going from left to right (west to east), and the y coordinates become more positive going from bottom to top (south to north), as viewed looking down at the primary side (TOP) of the board, coupon, or panel (see Figure 3-1).

The user has the responsibility to define the top of the printed board assembly. All sections of GenCAM will use the same TOP (primary side) selected. Once the point of origin of all coordinate data has been established, that location **shall** remain constant throughout the delivered GenCAM file.

When coordinates are identified by a parameter <xy_ref>, it is an indication that there are two separate fields for the X and Y coordinate. The first field represents the X, the second field represents the Y coordinate. The letters X and Y **shall not** appear in the GenCAM file, however the format is explicit in that xy pairs are denoted by the format parameter statement, and are enclosed by parentheses. Thus, three references to xy coordinates would result in three number sets, separated by commas i.e., (4937, 5178), (5937, 6492), (4326, 7893). Additional spaces to improve readability are optional.



Note: For those printed board assemblies that have components on both sides, any side may be designated as the primary side and used to define **Conductive Layer 1**.

Figure 3-1 Printed Board Viewing

3.6 Transformation Characteristics

The transformation parameter, <xform>, is used throughout this specification to define the location and orientation of physical features. Any primitive within the GenCAM file may be transformed by changing their characteristics. The transform includes placement, rotation, scaling, and mirror image changes. The transform is unique to a particular primitive, be it a standard or a user-defined primitive, the following rules apply:

<xform>	::=	<xy_ref>, [<rotation>], [<mirror>], [<scale_factor>]
<xy_ref>	::=	(<x_offset>, <y_offset>)
<rotation>	::=	<angular_measure>
<mirror>	::=	MIRROR - indicates that all x-values go to minus x-values.
<scale_factor>	::=	p_number - Scale multiplies all dimensions proportionally.
<x_offset>	::=	number
<y_offset>	::=	number

An additional transformation parameter, <location>, is also available in GenCAM. The <location> parameter is similar to the <xform> parameter except that the <scale_factor> parameter is not permitted. The intent of <location> is to prevent CAD or CAM systems from scaling those items where a change of scale is not appropriate. The rules for <location> are:

<location>	::=	<xy_ref>, [<rotation>], [<mirror>]
<xy_ref>	::=	(<x_offset>, <y_offset>)
<rotation>	::=	<angular_measure>
<mirror>	::=	MIRROR - indicates that all x-values go to minus x-values.
<x_offset>	::=	number
<y_offset>	::=	number

3.6.1 Scale factor description

Scale factor **shall** define a positive number, greater than or less than one. The scale factor is applied as a multiplier to both x and y dimensions evenly. It can be used to scale a coordinate, text, or other primitive dimensions. The scale factor does not apply to angular values or rotation direction values.

3.6.2 Rotation and mirror attributes

The rotation is a value in either degrees, or radians units. Positive rotation is in the counter-clockwise direction. Negative rotation is in the clockwise direction. Rotation defaults to 0.0, but can be applied to text, or any physical shape.

Mirror is the fixed field parameter MIRROR. MIRROR indicates that all "x" values go to a –x value.

The mirror and rotate attributes are shown in Figure 3-2. The example shows a unique artwork (14 pin DIP device) placed on the top and bottom of a board at 90 degree rotations. The following points should be noted:

1. The original shape (<rotation> = 0 and <mirror> = 0) is defined for the right edge of the board. It could have been defined for the bottom of the board. Whether the shape was defined for TOP or BOTTOM of the board the original shape will always have <rotation> = 0 and <mirror> = 0.
2. All shapes are viewed from the TOP of the board looking down and through the board.
3. Mirroring **shall** be applied before rotating.
4. MIRROR mirrors the component about the y-axis of the board. Mirroring is not done about the x or y axes of the component. The x coordinates of the shape become negative, the y coordinates remain the same.
5. Rotation is always counter clockwise as viewed from the right edge of the board, even if the component is on the BOTTOM.
6. Lands associated with the body go through the same shape mirroring and rotation.
7. Rotation and mirroring do not affect the component position on the board. The component is still placed using the shape origin.
8. The same rules apply to all GenCAM shapes, i.e. pads, patterns, artworks, boards, and panels.

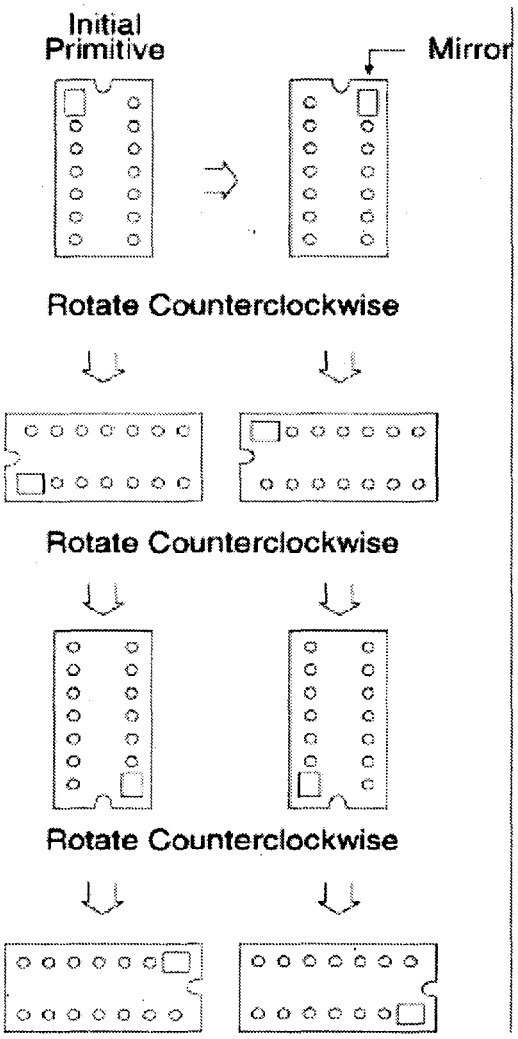


Figure 3-2 Mirror and Rotate Diagram

3.7 Data Organization and Identification Rules

GenCAM is intended to convey printed board and printed board assembly data in a single file. The information may be for a single electronic assembly or multiple electronic assemblies that may be manufactured on a single panel. Thus the rules for the boards, assemblies, panels and fixtures have specific requirements to permit segmentation of data within supporting data sections.

3.7.1 USEDIN

The keyword USEDIN is the identifier that permits grouping of data pertinent to one or more boards, assemblies, panels, or fixtures. The use of the USEDIN keyword statement is mandatory and **shall** be identified in every section as pertaining to those items contained in the HEADER section where the names are assigned.

ASSEMBLY: <usedin_name>, <assembly_name>, <assembly_number>, <assembly_revision>;
BOARD: <usedin_name>, <board_name>, <board_number>, <board_revision>;
PANEL: <usedin_name>, <panel_name>, <panel_number>, <panel_revision>;
FIXTURE:<usedin_name>, <fixture_name>, [<fixture_number>], [<fixture_revision>];

The <usedin_name> is unique only to the specific GenCAM file. A different name **shall** be assigned to each product defined in the HEADER section. Duplicating these names **shall not** be permitted. It is recommended that the selection of <usedin_name> have significance to those using the GenCAM file.

Support sections (i.e. PRIMITIVES, ARTWORKS, PATTERNS, ROUTES, COMPONENTS etc.) **shall** contain the indication to which <usedin_name> product the data belongs. If all of the data is pertaining to all products in the file, the USEDIN statement will have all usedin_names indicated in a string. Alternatively, the data may be segmented by multiple occurrences in a section. The USEDIN statement is modal, and **shall** apply to all data that follows until the occurrence of the next usedin statement, or the end of the section.

3.7.2 Explicit Definition

All data is explicitly defined, there is no implied change of data. For example, placing a component on the bottom of the board does not imply that its shape is mirrored. The shape **shall** be explicitly given a mirror parameter if the shape needs mirroring.

3.7.3 File Readability

When several keywords are used to define or add additional information to a preceding keyword, those keywords **shall** be indented under the parent to which they belong. The indenture **shall** be consistent throughout the file and be a minimum of two spaces. This same feature **shall** also be used to indent all keywords between the start and the end of a section, i.e., $BOARDS to $ENDBOARDS. The TAB (ASCII decimal 9) is not allowed when indenting blocks

Thus, the rule applies throughout the GenCAM file, and is established by the number of spaces that keywords are indented between the beginning and end of a section keyword. In addition to indenture, spaces are permitted between any parameters of a statement. Spaces are significant if they occur within a string (between quotes); spaces are insignificant if they are used primarily to improve readability of the GenCAM file.

The use of additional spaces between parameters may be random in that there may be 1, 2, 3, or more spaces at different increments depending on the desire of the tool or the user community to have particular parameters aligned for ease of readability.

4 GENERAL RULES

The following format key is provided to aid the reader in understanding the rules defined in this standard. Sections 4 through 6 provide details and examples in this format (BNF*) key. The text in Sections 4 through 6 supplements the BNF descriptions. In the case of differences between the text and the BNF, the BNF **shall** take precedence. A complete summary of the BNF descriptions of GenCAM is available on the web at http://www.gencam.org/docs/bnf/index.html.

< >	- denotes a single parameter field
[< >]	- denotes a single optional parameter field
A	- denotes exactly one of A.
0{< >}n	- denotes zero or more occurrences of a parameter field
{A | B | C}	- denotes exactly one of A or B or C
{A & B & C}	- denotes exactly one set of A and B and C (no order is implied)
1{A | B | C}2	- denotes no fewer than one, but no more than two of A, B or C.
1{A & B & C}2	- denotes at least one set, but no more than two sets of A, B and C.
1{A | B | C }n	- denotes some non-zero number (n) of A, B or C.
{A & 0{B}n & C}	- denotes exactly one set consisting of a single A and a single C and from zero or more occurrences of B

*Format popularized by John Backus and Steve Nauer

The convention within this document is that the parameter that applies to an item's name where it is defined, used the syntax "..._name" (e.g. <circle_name>). Where the defined item is later referenced (called), the syntax "..._ref" is used (e.g. <circle_ref>).

4.1 Description of Primitives

Many of the details within the various GenCAM sections require the ability to describe shapes. The PRIMITIVES section of a GenCAM file is specifically for defining and dimensioning these shapes. GenCAM supports the use of three distinct categories of standard shape primitive descriptions: Graphic Primitives, Simple Primitives and Complex Primitives. In addition, GenCAM supports the ability of the user to extend the collection of standard primitives.

GenCAM Graphic Primitives are single, line and arc segments.

LINE	CIRCARC	ELLIPARC

GenCAM Simple Primitives are common, closed shapes.

CIRCLE	RECTCENTER	RECTCORNER

GenCAM Complex Primitives are special shapes that are commonly used throughout the electronic design and manufacturing industry.

RECTCHAM	RECTROUND	OVAL
DIAMOND	HEXAGON	OCTAGON
DSHAPE	THERMAL	

The Standard Primitives collection can be extended by the user, through the use of special keywords. The keyword POLYGON can be used to define and name a group of GenCAM Graphic Primitives that form a single-image closed region. The keyword POLYLINE can be used to define and name a group of GenCAM Graphic Primitives that form a non-closed image.

4.2 GenCAM Primitive Enhancement Descriptors

The default condition of every primitive definition is as a solid, but zero-width line, with all closed regions being unfilled. Every primitive can be provided with *texture* to override the default condition through the use of the GenCAM primitive enhancement descriptors LINEDESC and PAINTDESC. These descriptors can be coupled with a primitive to identify its line width, type and end treatment, or the fill method of a closed region, respectively. The statements LINEDESC and PAINTDESC also reside in the PRIMITIVES section of the GenCAM file.

4.2.1 Line Enhancement Descriptors (LINEDESC)

The LINEDESC keyword **shall** be followed by a line description name. The LINEDESC keyword describes the characteristics of the line and includes the line width, the line type, the end characteristics, end modifications, and color. The following format for LINEDESC includes all the options and all possible parameters.

Format:

LINEDESC: <linedesc_name>, <line_width>, [<line_end>], [<color_ref>], [<line_type>],
[<line_space>], [<line_length>], [<line_mod>], [<mod_end>],
[<dim_A>], [<dim_B>], [<dim_C>];

Note: <line_type> defines the characteristics of the line that includes the line type designation and any dimensions associated with the line type. (see Figure 4-1) The default for <line_type> is SOLID. The default for <line_end> is ROUND with an <end_mod> of NONE. If no color is referenced, the default color is 0, 0, 0 (black).

4.2.1.1 Line width parameters <line_width>

Line width parameters **shall** be specified for primitive enhancers. Line width parameters specified for circles, rectangles and polygons determine the width of the perimeter line of these closed figures. When there is more than one dimension of <line_width> within the GenCAM file, the LINEDESC keyword is used multiple times and the differences are identified by the <linedesc_name> in order to be able to refer to that particular line description.

Example:

```
LINEDESC: "line1", 3.0;
```

A solid black line, 3.0 mm wide with round line ends

4.2.1.2 Line type parameters <line_type>

GenCAM supports the line types SOLID, DOTTED, DASHED, CENTER and (solid) ERASE. Line type parameters other than SOLID **shall** define the spacing characteristics for those lines shown in Figure 4-1. The words used for line type parameters are reserved, fixed field parameters and **shall** be indicated in upper case. The format used to describe line type parameters includes the fixed parameter followed by definitions of spaces and length of dots, dashes, center-lines, etc.

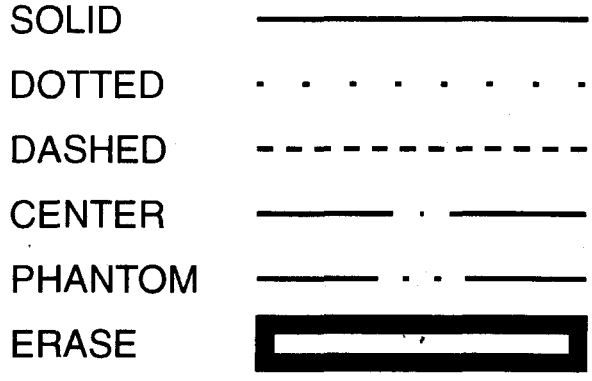


Figure 4-1 Line Type Parameter Descriptions

Example:

```
LINEDESC: "line2", 1.0, , , DOTTED, 2.5;
```

A dotted line 1.0mm diameter for the dot, 2.5mm for the space between dot centers.

```
LINEDESC: "line3",1.0, , , CENTER,1.5,2.8;
```

A dashed, dot, dash line, 1.0mm wide (also dot diameter), 1.5mm space between dot center and start or ending of dash, 2.8mm for length of dash.

4.2.1.3 Line end parameters <line_end>

The default for line end parameters is ROUND. When other parameters are used, they **shall** be denoted according to Figure 4-2.

<line_end> designations are:

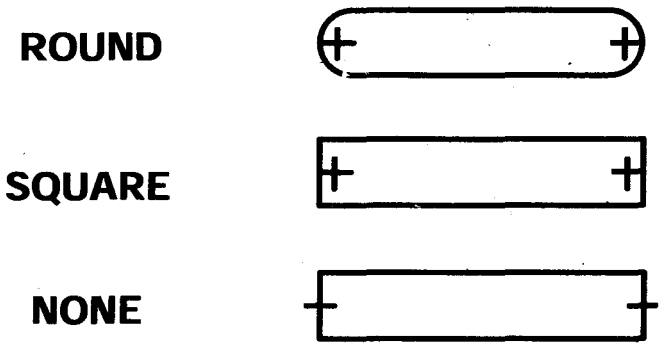


Figure 4-2 Line End Parameters

Note: The words ROUND, SQUARE, and NONE are fixed field parameters and are reserved words.

4.2.1.4 Line end modification parameters

Regarding the tear-drop, sub-land or other modifications to line end, only the processing attributes are defined in this format. (The figure generation for these modifications **shall** be performed in each CAD or CAM system). (see Figure 4-3)

<line_mod> designations:

taper (fillet)	TP,....
tear-drop	TD,....
Sub-lands	SL1,....
Sub-lands	SL2,....

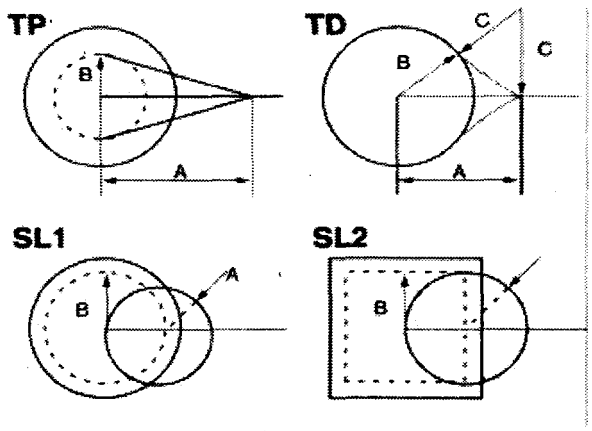


Figure 4-3 Line End Modification Parameters

The special line end modifications are defined by the fixed field parameter (TP, TD, SL1, SL2 or NONE); where NONE is the default parameter. The dimensions that describe line end modifications are A, B, and C as indicated in Figure 4-3. The words TP, TD, SL1, or SL2 are fixed field parameters.

The parameter <mod_end>, defines which end of the line is modified. There are three fixed field parameters that may be used to describe the requirement. The <mod_end> designations are:

BOTH: fixed field parameter for modifying both start and finish of line ends
START: fixed field parameter for describing the start point of a line
FINISH: fixed field parameter for describing the last point of a line

Note: The words BOTH, START, and FINISH are fixed field parameters.

Examples:

```
LINEDESC: "Line9", 1.0, SQUARE , , CENTER, 40.0, 8.0
                TP, BOTH, 2.5, 1.3;
```

A center line, black by default, 1.0 mm wide, where the length of the dash is 40.0 mm and the clearance between the center of the dot and the beginning and end of the dash is 0.8. Since the line ends are square, the dot is also square - 1.0 mm x 1.0 mm). Both ends of the line segment are tapered

```
LINEDESC: "line10", 1.0, ROUND, "Red", PHANTOM, 25.0, 8.0, TD, START,
2.0, 1.7, 1.6;
```

A red phantom line, 1.0 mm wide, where the dashes are 25.0 mm long, and the distance from beginning and end of the dash to the center of the dot, or the distance between dot centers is 8.0 mm. The starting end of the line is a teardrop, while the opposite end is round by default.

4.2.2 PAINT parameters (PAINTDESC)

The PAINTDESC keyword **shall** be followed by a paintdesc name and the parameters shown in Figure 4-4 and the appropriate parameters for HATCH and MESH.

PAINTDESC: <paintdesc_name>, <paint_type>, [<color_ref>], [<line_width>], [<pitch1>],[<angle1>], [<pitch2>], [<angle2>];

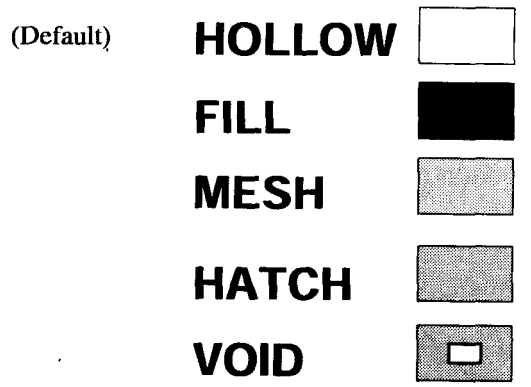


Figure 4-4 Paint Designation

The paint default is HOLLOW.
For MESH and HATCH, the default angle is 45.
The default color is 0, 0, 0 (black).

When combined with other primitives, the result of a FILL, HATCH, MESH, and HATCH with a VOID are shown in Figure 4-5. The implication in Figure 4-5 is that a LINEDESC description has also been invoked either separately or as part of a single record to provide the outline of those shapes that are painted with MESH, HATCH, or VOID. It should also be noted that if a VOID is instantiated in a FILL, the LINEDESC is not required.

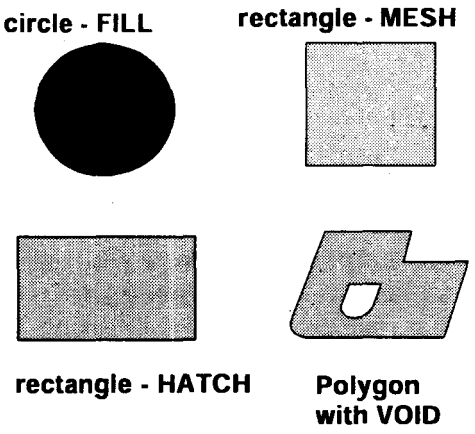


Figure 4-5 Paint Descriptions

ERASE and VOID remove the immediately previous, positive instance. VOID overlapping VOID only extends the VOID and does not remove a second positive instance.

4.2.3 Rules of Precedence

Both the LINEDESC and PAINTDESC statements have an optional <color_ref> parameter. The addition of an optional <color_ref> parameter, directly on the instance statement has the possibility of causing a conflict with the <color_ref> on the definition statement. Should this happen, the rule **shall** be that the <color_ref> specified of the more global enhancement statement (LINEDESC, PAINTDESC) **shall** override not only the default of the language, but also the <color_ref> on the instance statement. If no texture is provided, the definition statement takes precedence. Figure 4-6 shows the rules of precedence and indicates how usage textures override referenced textures included in the definition of an artwork or pattern.

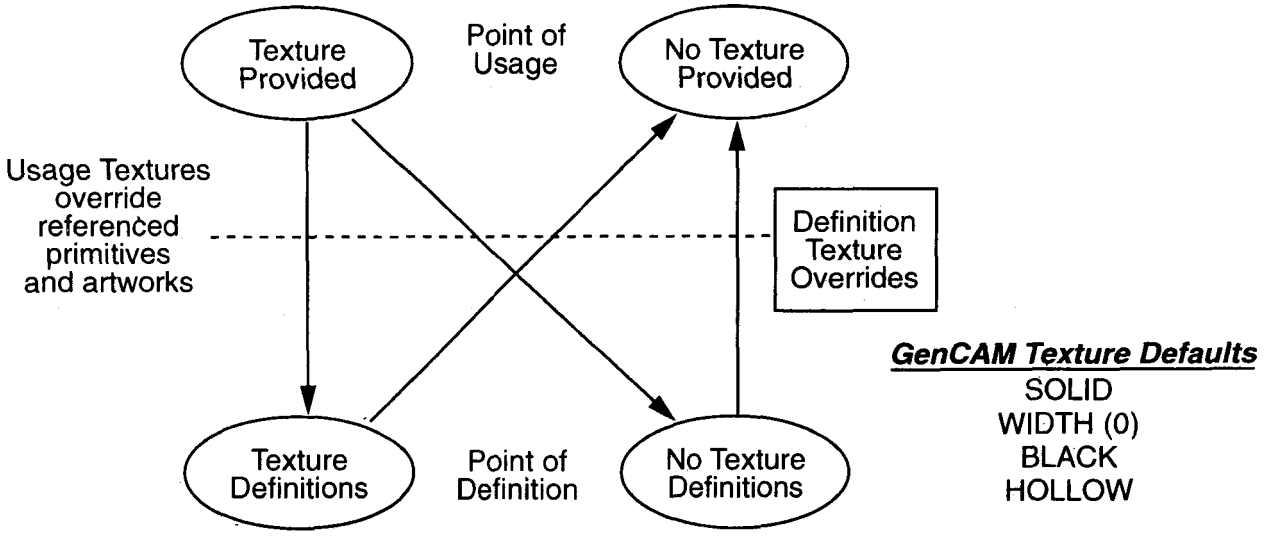


Figure 4-6 Rules of Precedence

4.2.4 Rules for Shape Descriptions

Shapes can be needed in many sections of the GenCAM file where they may be used to describe a conductive trace, a board outline, a hole, a drawing etc. Many instances of the same shape, in the same dimensions may be needed in different places in the GenCAM file. In recognition of this, GenCAM supports the reuse of distinct shapes through definition and reference rules.

GenCAM Standard Simple Primitive and GenCAM Standard Complex Primitive statements can appear in three forms. The first form, Named Form, is a definitional form, and can appear only in the PRIMITIVES section of the GenCAM file. The syntax of the Named Form does not support enhancement or transformation parameters. The other two forms are instance forms. The first of these, the Reference Form, creates a primitive instance by referencing a previously defined, Named Form primitive. The last form, the Un-named Form, creates an in-line instance definition. Unlike the Named Form, the syntax for these Reference and Un-named instance forms support enhancement through reference to a named LINEDESC and/or PAINTDESC. When used, both of these instance forms **shall** be owned by a higher order statement and **shall** be provided with a locating <xform>.

Named definitions of un-enhanced primitives and the primitive enhancements **shall** be located in the PRIMITIVES section where they are grouped and analyzed for completeness. Thus, a robust GenCAM file would have all enhancement descriptors (i.e. LINEDESC, PAINTDESC) and all named primitive descriptions (e.g. CIRCLE, OVAL), including user primitives (POLYLINE, POLYLINE), located in the PRIMITIVES section. These would then be called, provided with enhancement, and positioned at the correct location (xy coordinate) at the time they are required within the other sections.

4.3 Standard Graphic Primitives

The GenCAM Standard Graphic Primitives are line and arc segments, and are located by the x-y points of their definitions. Their statements shall have only an Un-Named form. They have no Named Form and no Reference Form. Within the PRIMITIVES Section of the GenCAM file, their syntax does not support enhancement. In any other applicable section their syntax does support reference to LINEDESC definitions. In all cases, they shall be owned by a higher order statement.

4.3.1 Straight line (LINE)

A straight line **shall** have all the characteristics and attributes necessary to fully describe the line based on its context. (See Figure 4-7)

Form:

PRIMITIVE Section:
LINE: <start_xy>, <end_xy>;

All Other Sections:
LINE: <start_xy>, <end_xy>, [<linedesc_ref>], [<color_ref>];

Note: A <color_ref> used here will be overridden by the color reference in the LINEDESC definition, if any.

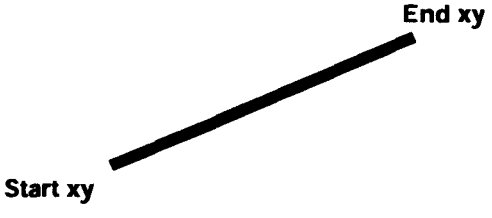


Figure 4-7 Line

Example:

PRIMITIVES Section:

```
LINE: (4973, 2967), (4735, 6524);
```

All Other Sections:

```
LINE: (4973, 2967), (4735, 6524), "line3", "cyan";
```

4.3.2 Circular arc (CIRCARC)

The keyword for a curved line may have one or two types of characteristics. Those characteristics identify a segment of a circle or an ellipse. The descriptions of the two types of curved lines are slightly different. The simpler is that of a circular arc (Figure 4-8).

Form:

PRIMITIVES Section:

CIRCARC: <start_xy>, <end_xy>, , [<direction>];

All Other Sections:

CIRCARC:<start_xy>, <end_xy>, , [<direction>], [<linedesc_ref>], [<color_ref>];

Note: <direction> is equal to clockwise (CLKW) or counterclockwise (CCLKW). The default is counterclockwise. CLKW and CCLKW are GenCAM reserved words.

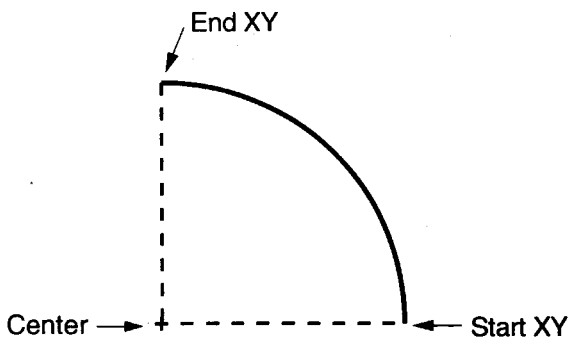


Figure 4-8 Circular Arc

The first parameter is the start coordinate of the arc, the second parameter is the end coordinate of the arc, and the third parameter is the center of the arc's circle.

Examples:

PRIMITIVES Section:

```
CIRCARC: (1000, 0), (0, 1000), (0, 0);
```

All Other Sections:

```
CIRCARC: (1000, 0), (0, 1000), (0, 0), "line3", "gray";
```

The CIRCARC **shall** be drawn from the start <start_xy> counterclockwise about the to the end <end_xy>, unless the direction parameter is defined as clockwise.

Note: Great care must be taken with the coordinates and direction of rotation when translating arcs from various CAD systems.

4.3.3 Ellipse arc (ELLIPARC)

The ELLIPARC is drawn from the start coordinate counterclockwise about the two focii to the end coordinate, unless the direction parameter is defined as clockwise (CLKW). ELLIPARCs are limited to greater than 0, and up to 360 degrees of rotation. Parabolic or hyperbolic curves are not supported. Rectangular coordinates are always used. (see Figure 4-9)

Elliptical arcs can be defined by the following characteristics once the description has been defined:

Form:

PRIMITIVES Section:
ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>, [<direction>];

All Other Sections:
ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
[<direction>], [<linedesc_ref>], [<color_ref>];

Note: <direction> is equal to clockwise (CLKW) or counterclockwise (CCLKW). The default is counterclockwise. CLKW and CCLKW are GenCAM reserved words.

The first parameter is the start coordinate of the arc, the second parameter is the end coordinate. The third and fourth parameters are the two focci of the ellipse.

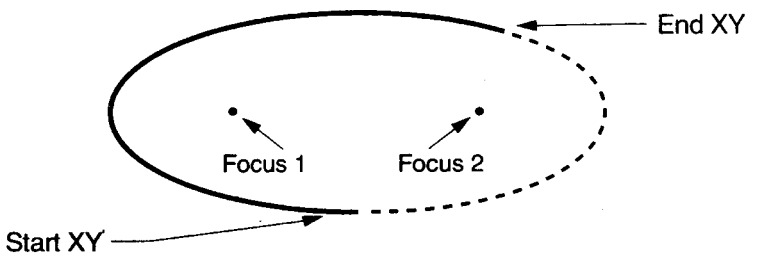


Figure 4-9 Ellipse arc

Examples:

PRIMITIVES Section:
```
ELLIPARC: (0, -200), (400, 200), (-400, 0), (400, 0), CLKW;
```

All Other Sections:
```
ELLIPARC: (0, -200), (400, 200), (-400, 0), (400, 0), CLKW,
          "line3", "Blue";
```

4.4 GenCAM Standard Simple Primitives

GenCAM Standard Simple Primitives are common, closed shapes. Unlike the GenCAM Standard Graphic Primitives, their definitions do not explicitly define their location. Instead, each has an understood point-of-origin to which a transformation **shall** be applied (See Section 3.6).

All three of the statement forms can be applied to GenCAM Standard Simple Primitives. Only the Named Form **shall** be valid within the PRIMITIVES section.

4.4.1 Circle (CIRCLE)

Named Form:

CIRCLE: <circle_name>, <diameter>;

Reference Form:

CIRCLE: <circle_ref>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;

Un-named Form:

CIRCLE: <diameter>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;

A circle is defined by its diameter. It is understood that the point of origin of a circle is its center. Line end parameters are not applicable, since a circle has no beginning and no end. Circles can be an outline, or can be filled, solid, mesh, or hatched. (See Figure 4-10)

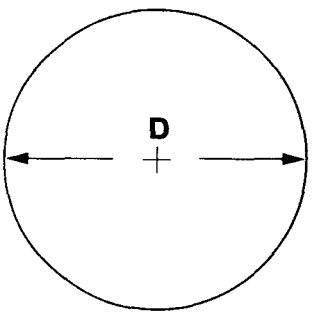


Figure 4-10 Circle

Examples:

```
CIRCLE: "circ2", 2.0;
CIRCLE: "hole2", "line2", ,"white", (16.3, 18.9);
CIRCLE: 2.0, "line2", , "white", (16.3, 18.9);
```

Note: In the first example, the CIRCLE "hole2" defaults to a line_width of 0.0 and/or PAINT of HOLLOW.

4.4.2 Rectangle corner (RECTCORNER)

The following rules apply for a rectangle whose descriptions are identified by the corner definition of the rectangle (See Figure 4-11).

Named Form:

RECTCORNER: <rectangle_name>, <p1>, <p2>;

Reference Form:

RECTCORNER: <rectangle_ref>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;

Un-named Form:

RECTCORNER: <p1>, <p2>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;

Note: The transform parameters apply to the point of origin of the RECTCORNER since it can be located by its defining points

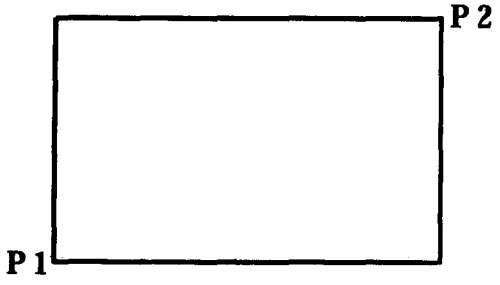


Figure 4-11 Rectangle (corner)

Note: The point of origin is (0, 0). This is usually the lower left corner of the rectangle. The rectangle is defined in the XY plane, but its instances can be rotated to any angle.
Note: Any rotation is of this shape is about the point 0,0; not about P1 or P2.

Examples:

```
RECTCORNER: "rect6", (1240, 3370), (4535,2355);
RECTCORNER: "rect6", "line2", , (0,0);
RECTCORNER: (0, 0), (2240, 3350), "line2", , , (0, 0), 30.0;
```

4.4.3 Rectangle Center (RECTCENTER)

All other rectangles are located by their center point; this is the point of origin. For the purpose of rotation, rectangles are always described in the horizontal plane. There are four corner types. These are those with square corners (identified as simple primitives), those with the chamfered corner, those with a rounded corner, and those where the entire corner is a radius equal to half of the height of the rectangle (identified as complex primitives).

The following rules apply for a simple primitive rectangle whose description is identified by the center of the rectangle, or its point of origin. (see Figure 4-12)

Named Form:

RECTCENTER: <rectangle_name>, <x_dimension>, <y_dimension>;

Reference Form:

RECTCENTER: <rectangle_ref>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;

Un-named Form:

RECTCENTER: <x_dimension>, <y_dimension>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

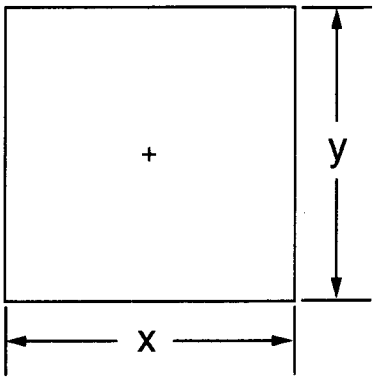


Figure 4-12 Rectangle (center)

Note: The center of the rectangle RECTCENTER is understood to coincide with 1/2X and 1/2Y. The rectangle is called by the center (in the direction originally described) followed by the xy coordinate.

Examples:

```
RECTCENTER: "rect3", 1.6, 2.8;
RECTCENTER: "3x4", , "FillViolet", , (4.3550, 3.6550);
RECTCENTER: 1.6, 2.8, "line12", "Hatch90", "Gray", (4.35, 3.65);
```

4.5 GenCAM Standard Complex Primitives

GenCAM Standard Complex Primitives are a family of closed shapes which are commonly used throughout the electronics design and manufacturing industry. Like the Standard Simple Primitives, their definitions do not explicitly define their location. Each has an understood point-of-origin to which a transformation can be applied. (See Section 3.6)

All three of the statement forms can be applied to GenCAM Standard Complex Primitives. Only the Named Form shall be valid within the PRIMITIVES section.

4.5.1 Rectangle, Chamfered Corners (RECTCHAM)

The following rules apply for a rectangle whose description includes a chamfer at each corner. (see Figure 4-13)

Named Form:

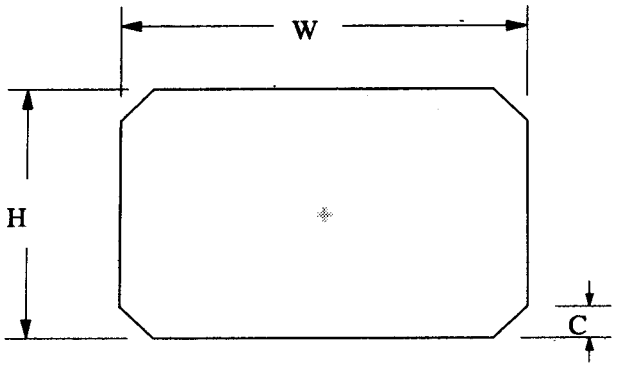
RECTCHAM: <rectcham_name>, <width>, <height>, <chamfer>;

Reference Form:

RECTCHAM: <rectcham_ref>,.[<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;

Un-named Form:

RECTCHAM: <width>, <height>, <chamfer>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;



W = Width
H = Height
C= Chamfer

Figure 4-13 Chamfered Rectangular Primitive

Note: The chamfer size is the length of one leg of the chamfer. The chamfer is 45°. The chamfered rectangle is defined in a horizontal position. The center of the rectangle is the point of origin.

4.5.2 Rectangle, Round Corners (RECTROUND)

The following rules apply for a rectangle whose description includes a radius at each corner. (see Figure 4-14)

Named Form:

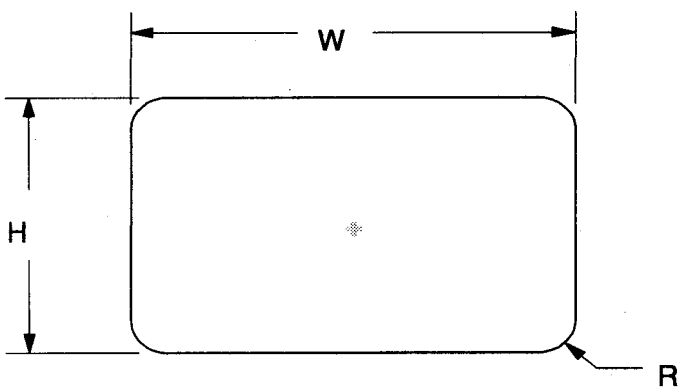
RECTROUND: <rectround_name>, <width>, <height>, <radius>;

Reference Form:

RECTROUND: <rectround_ref>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;

Un-named Form:

RECTROUND: <width>, <height>, <radius>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;



W = Width
H = Height
R = Radius

Figure 4-14 Rounded Rectangular Primitive

Note: The rounded rectangle is defined in the horizontal position. The center of the rounded rectangle is its point of origin.

4.5.3 Rectangle, Rounded Ends (OVAL)

The following rules apply for a rectangle whose description includes a complete radius at each end. (see Figure 4-15)

Named Form:

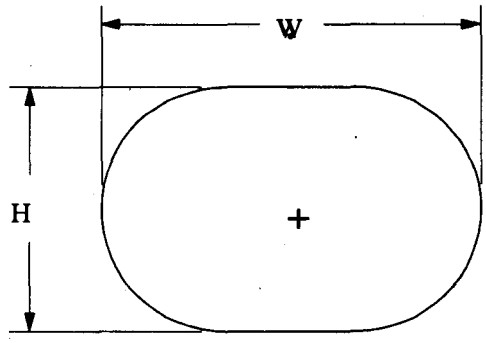
OVAL: <oval_name>, <width>, <height>;

Reference Form:

OVAL: <oval_ref>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;

Un-named Form:

OVAL: <width>, <height>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;



W = Width
H = Height

Figure 4-15 Oval Rectangular Primitive

Note: For the OVAL, the radius is always equal to ½ the height. The OVAL is defined in its horizontal position. The center of the OVAL is its point of origin

4.5.4 Rectangle, One-Sided (DSHAPE)

The D-shaped rectangles are similar to the other rectangle shapes except that the corner definition is only on one side. A single D-shaped primitive may be used to define the characteristics of the rectangle. Therefore, the shape attribute helps to delineate the type of corner that is use. (See Figure 4-16). The D-shaped designation is:

Named Form:

DSHAPE: <dshape_name>, <end_shape>, <orientation>, <width>, <height>, {<radius> | <chamfer>}1;

Reference Form:

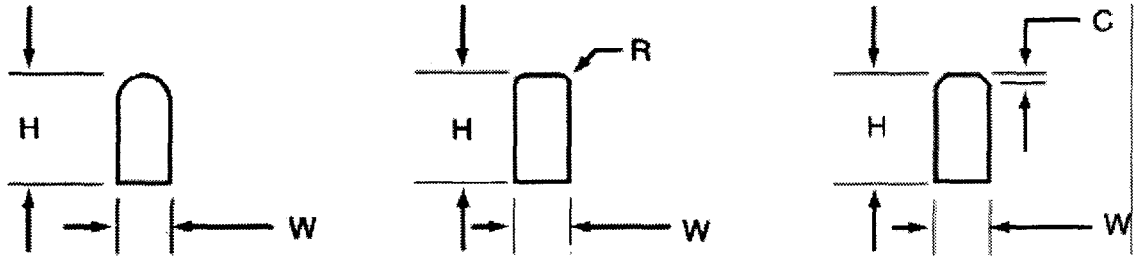
DSHAPE: <dshape_ref>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;

Un-named Form:

DSHAPE: <end_shape>, <orientation>, <width>, <height>, {<radius> | <chamfer>}1,
[<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;

The <end_shape> parameter is one of "ROUND", "FILLET", or "CHAMFER". North is where the fillet, round, or chamfer end is located. Thus, all primitive references are defined in their north direction. Calling for a south orientation would take the DSHAPE illustration shown in Figure 4-16 and rotate them 180° using the <xform> modifier for rotation. The radius/chamfer definition for a round D-shaped rectangle is ½ W. The rules for the <end_shape> fixed-field parameters are:

<end_shape> ::= {ROUND | FILLET | CHAMFER}1



W = Width
H = Height

Figure 4-16 D-Shaped Rectangle Primitives – Round, Fillet, Chamfer

Note: The point of origin of D-shaped rectangles is defined in the north direction. The center of the D-shaped rectangle is the point of origin.

4.5.5 Rectangle, Rotated (DIAMOND)

A diamond is essentially a square rectangle with its corners pointed north, south, east, and west. The characteristics of the diamond shown in Figure 4-17 indicate that the sides are always equal. However, since both a height and width are provided for the diamond, the "H" and "W" dimensions could be different, therefore providing a diamond that is elongated either horizontally or vertically. The rule for diamonds is that the north and south point pair and the east and west point pair must always lie in the same plane. The definition for the complex primitive diamond is:

Named Form:

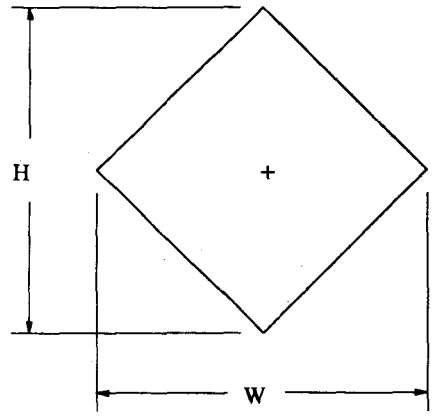
DIAMOND: <diamond_name>, <width>, <height>;

Reference Form:

DIAMOND: <diamond_ref>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;

Un-named Form:

DIAMOND: <width>, <height>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;



W = Width
H = Height

Figure 4-17 Diamond Primitive

Note: The center of the diamond is defined with its major axis pointing north. The center of the diamond is the point of origin.

4.5.6 HEXAGON

The hexagon complex primitive is defined with the points in the horizontal plane. The <xform> parameter can change the orientation when a hexagon is referenced or described in the un-named form. The rules for hexagons are that they are a six-sided feature, with the length of each of the sides being equal. Therefore, the only dimension that is necessary is to provide the distance across the points of the hexagon. The following rules apply and are illustrated in Figure 4-18.

Named Form:

HEXAGON: <hexagon_name>, <point_to_point>;

Reference Form:

HEXAGON: <hexagon_ref>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;

Un-Named Form:

HEXAGON: <point_to_point>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;

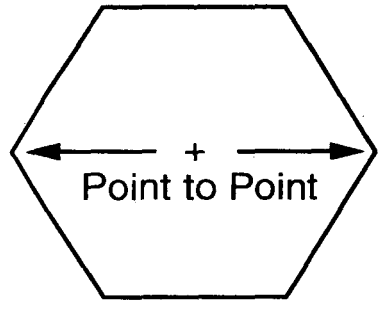


Figure 4-18 Hexagon Primitive

Note: The direction of points defines the way the hexagon is oriented. The center of the hexagon is the point of origin for this complex primitive.

4.5.7 OCTAGON

The octagon is an eight-sided primitive where each of the eight sides is equal. The octagon is defined with the flat surfaces in the horizontal and vertical plane. The only dimension needed to describe the octagon is the distance across the points which occur at 22½° off the horizontal or vertical planes. The following rules apply and are illustrated in Figure 4-19.

Named Form:

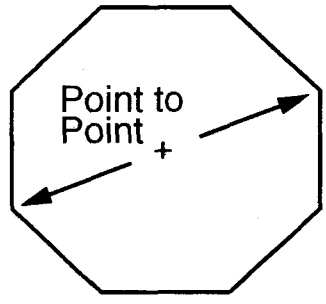
OCTAGON: <octagon_name>, <point_to_point>;

Reference Form:

OCTAGON: <octagon_ref>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;

Un-named Form:

OCTAGON: <point_to_point>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;



Point_to_Point = Dimension

Figure 4-19 Octagon Primitive

Note: The point of origin is the center of the octagon.

4.5.8 THERMALS

Thermals are those shapes and features used in the padstacks of GenCAM and are defined in various characteristics. Thermals may be square, round or octagonal and have varying numbers of spokes. Therefore, the rules for thermals are intended to cover all the variations that can exist for thermal as shown in Figures 4-20, 4-21, and 4-22.

It should be understood that there are no LINEDESC or PAINTDESC requirements for thermals. The default condition of thermals is that they *remove* material of a plane, assuming a PAINTDESC of VOID. Thus, a ROUND thermal primitive with four spokes would have four separate areas of material removed in a plane.

The rules for thermals are:

Named Form:

THERMAL: <thermal_name>, <therm_shape>, <outer_diameter>, <inner_diameter>,
[<spoke_count>], [<spoke_width>], [<spoke_start_angle>],
[<spoke_end_shape>];

Reference Form:

THERMAL: <thermal_ref>, <xform>;

Un-named Form:

THERMAL: <therm_shape>, <outer_diameter>, <inner_diameter>, [<spoke_count>], [<spoke_width>], [<spoke_start_angle>], [<spoke_end_shape>], <xform>;

The <therm_shape>, fixed-field parameters are:
<therm_shape> ::= {ROUND | SQUARE | OCTAGON}1

One of these is required for each thermal defined. In addition, the <spoke_end_shape> also has three fixed-field parameters. They are:
<spoke_end_shape> ::= {ROUND | SQUARE | PARALLEL}1

The default condition for the optional parameters of thermals are:
<spoke_count> ::= 0
<spoke_width> ::= the <inner_diameter> subtracted from the <outer_diameter>
<spoke_start_angle> ::= 45 Degrees
<spoke_end_shape> ::= SQUARE

Note: If the <spoke_count> is not defined (zero), the other three optional parameters do not apply, and would represent a donut thermal shape (see 4.5.8.1).

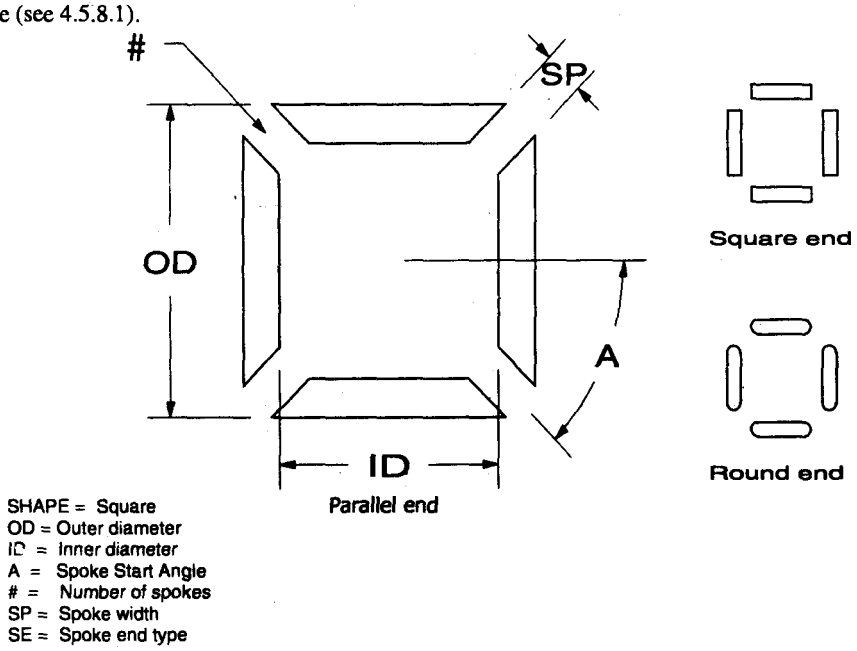
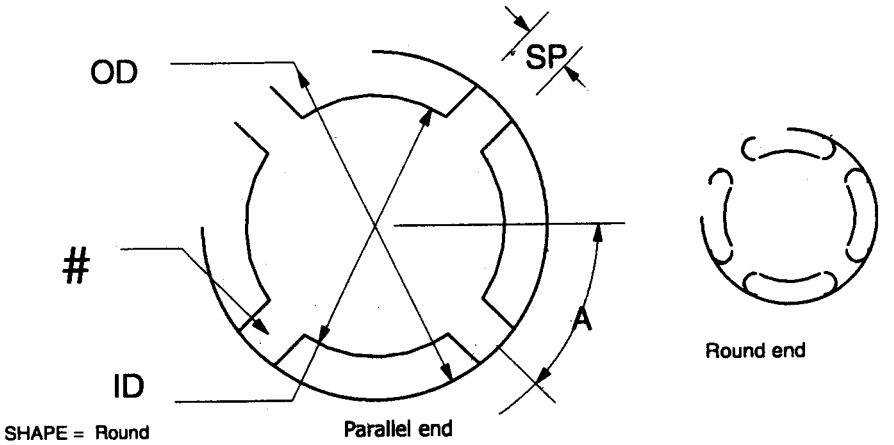
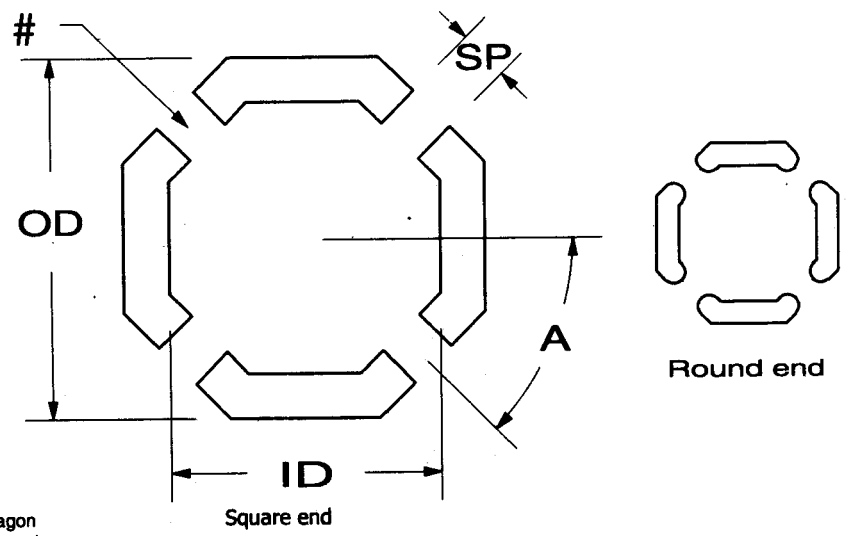


Figure 4-20 Square Thermal Primitive



SHAPE = Round
OD = Outer diameter
ID = Inner diameter
A = Spoke Start Angle
# = Number of spokes
SP = Spoke width
SE = Spoke end type

Figure 4-21 Round Thermal Primitive



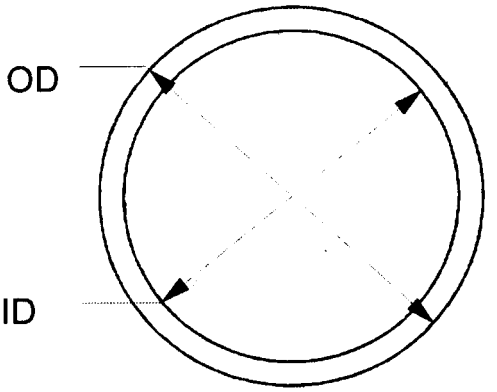
SHAPE = Octagon
OD = Outer diameter
ID = Inner diameter
A = Spoke Start Angle
# = Number of spokes
SP = Spoke width
SE = Spoke end type

Figure 4-22 Octagonal Thermal Primitive

Note: The examples shown in Figures 4-20 through 4-22 illustrate standard thermal primitive definitions. Many thermal primitive configurations can be generated using different spoke numbers and end types. The center of the overall thermals is the point of origin of the primitive.

4.5.8.1 Donuts

Donuts are thermals that have no spokes. They are used many times for non-functional lands on an interlayer plane, where the land is not connected to the plane. GenCAM defines donuts using the THERMAL statement. The donut-shape may be either round, square, or octagon shaped and is illustrated in Figures 4-23, 4-24, and 4-25.



OD= Outside Diameter
ID= Inside Diameter

Figure 4-23 Donut Round

A typical example of a donut used as a thermal would be:

Named form:

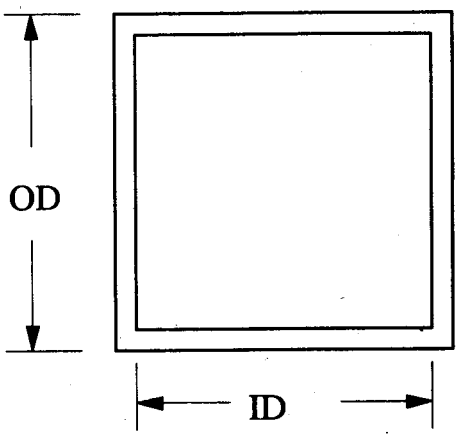
```
THERMAL: "land12", ROUND, 12.5, 12.0;
```

Reference Form:

```
THERMAL: "land12", (1200, 2400);
```

Un-named Form:

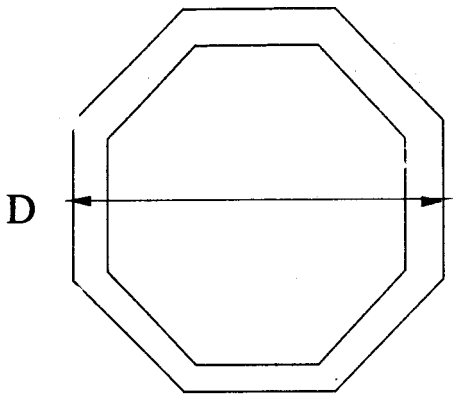
```
THERMAL: ROUND, 12.5, 12.0, 0, , , , (1200, 2400);
```



OD= Outside Diameter
ID= Inside Diameter

Figure 4-24 Donut Square

Note: It should be noted that the characteristics for shape of the donut is either ROUND, SQUARE, or OCTAGON. The center of the ROUND, SQUARE, or OCTAGON donut is the point of origin.



D = Outside Diameter

Figure 4-25 Donut Octagon

4.6 GenCAM User-Defined Primitives

User primitives are those that are defined as single images using the POLYGON and POLYLINE statements. The POLYGON, POLYLINE statements are followed by such Reference Form and/or Un-named Form instance definitions as are necessary to complete the description. To show the ownership relation, the primitive statements are indented for readability.

4.6.1 POLYLINE

Polylines are any shape that is defined by a collection of continuous GenCAM Graphic Primitives. All of the enhancements for line segments are applicable to this defined primitive.

Named form:

```
POLYLINE: <polyline_name>;
        :>              1{
                           LINE: <start_xy>, <end_xy>;
                           | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>];
                           | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                                       [<direction>];
        <:              }n
```

Reference Form:

```
POLYLINE: <polyline_ref>, [<linedesc_ref>], [<color_ref>], <xform>;
```

Un-named Form:

```
POLYLINE: [<linedesc_ref>], [<color_ref>];
        :>              1{
                        LINE: <start_xy>, <end_xy>, [<linedesc_ref>], [<color_ref>];
                        | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>],
                              [<linedesc_ref>], [<color_ref>];
                        | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                              [<direction>], [<linedesc_ref>], [<color_ref>];
        <:              }n
```

The Reference Form and the Un-named Form are referred to throughout this document as <polyline_builder>. The rules for <polyline_builder> are:

```
<polyline_builder>  ::= 1{
                          POLYLINE: <polyline_ref>, <linedesc_ref> , [<color_ref>], <xform>;
                        | POLYLINE: [<linedesc_ref>], [<color_ref>];
                    :>          1{
                                LINE: <start_xy>, <end_xy>, [<linedesc_ref>], [<color_ref>];
                                | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>],
                                        [<linedesc_ref>], [<color_ref>];
                                | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                                        [<direction>], [<linedesc_ref>], [<color_ref>];
                    <:          }n
                      }n
```

The Named Form **shall** only be valid in the PRIMITIVES section. The Reference and Un-named forms **shall** only appear in all other sections, and only when owned by a higher order statement. (i.e. ARTWORK, PATTERN, OUTLINE, etc., as shown in Figure 4-26) This is referred to in this document as a <polyline_builder>.

Parent textures have precedence over child texture parameters. For example, any <color_ref> parameter on a LINE, CIRCARC or ELLIPARC statement within the POLYLINE definition will be over-ridden by any <color_ref> parameter on the POLYLINE statement. This is also true of the <linedesc_ref> parameters. For example, if the LINEDESC that is referenced in the POLYLINE statement describes a SOLID line, and a LINE statement in the POLYLINE definition references a LINEDESC that describes a DOTTED line, then that line segment of the POLYLINE will be SOLID.

Examples:

```
POLYLINE: "linesegments";
  LINE: (15.0,12.0), (22.0,12.0);
  LINE: (22.0,12.0), (22.0,2.0);
  LINE: (22.0,2.0), (30.0,2.0);
```

```
POLYLINE: "linesegment", li12, ,
          (0.0, 0,0);
  LINE: (15.0,12.0), (22.0,12.0);
  LINE: (22.0,12.0), (22.0,2.0);
  LINE: (22.0,2.0), (30.0,2.0);
LINEDESC: "li12", 3, , gray, SOLID;
```

an alternate GenCAM description would be:

```
ARTWORK: "lineseg1";
  LINE: (15.0, 12.0), (22.0, 12.0), 3, gray, S
  LINE: (22.0, 12.0), (22.0, 2.0), 3, gray, SO
  LINE: (22.0,2.0), (30.0,2.0) 3, gray, SOLID;
```

```
ARTWORK: "lineseg1";
POLYLINE: "linesegment", "line12", , (0.0, 0.0);
LINEDESC: "line12";
```

Note: This format is under the rules of <polyline_builder> ar provides the texture under the named artwork "linesegment".

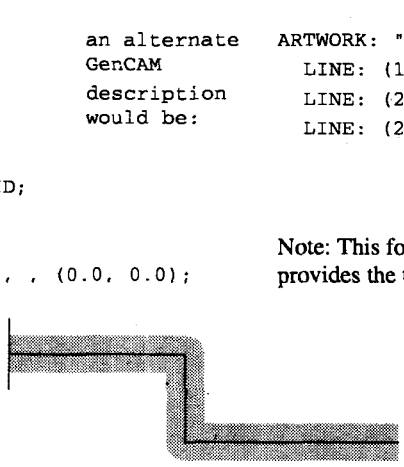


Figure 4-26 Three segment line description

4.6.2 POLYGON

A Polygon is any closed shape that is defined by continuous line segments that do not cross. All of the parameters for line segments and paint are applicable to this defined primitive. The shape of the polygon is user-defined. The rules for polygons are such that the final coordinate of the polygon **shall** be the same as the starting coordinate to signify that the shape is a closed shape. There is no automatic closing performed. Polygons are always positioned by their point of origin. Polygons may also be rotated about

their point of origin in order to provide the user the opportunity to define a polygon and position it by the coordinate and then rotate it.

Polygons may define the absence or presence of material. Figure 4-27 shows the characteristics for two polygons that have different paintdesc characteristics.

The following rules apply:

Named form:

```
POLYGON: <polygon_name>;
            :>          2{
                        LINE: <start_xy>, <end_xy>;
                        | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>];
                        | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                                    [<direction>];
            <:          }n
```

Reference Form:

```
POLYGON: <polygon_ref>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>], <xform>;
```

Un-Named form:

```
POLYGON: [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>];
            :>          2{
                        LINE: <start_xy>, <end_xy>, [<linedesc_ref>], [<color_ref>];
                        | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>],
                            [<linedesc_ref>], [<color_ref>];
                        | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                            [<direction>], [<linedesc_ref>], [<color_ref>];
            <:          }n
```

The Reference Form and the Un-named Form are referred to throughout this document as <polygon_builder>. The rules for <polygon_builder> are:

```
<polygon_builder> ::= 1{
                        POLYGON: <polygon_ref>, [<linedesc_ref>], [<paintdesc_ref>],
                                [<color_ref>], <xform>;
                        | POLYGON: [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>];
                        :>          2{
                                    LINE: <start_xy>, <end_xy>, [<linedesc_ref>], [<color_ref>];
                                    | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>],
                                        [<linedesc_ref>], [<color_ref>];
                                    | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                                        [<direction>], [<linedesc_ref>], [<color_ref>];
                        <:          }n
                      }n
```

Examples:

```
POLYGON: "line6","HATCH6", "Black";
  LINE: (0, 0), (5, 0);                                  [p1 to p2]
  LINE: (5, 0), (5, -1);                                 [p2 to p3]
  LINE: (5, -1), (8, -1);                                [p3 to p4]
  LINE: (8, -1), (8, -6);                                [p4 to p5]
  LINE: (8, -6), (1.5, -6);                              [p5 to p6]
  CIRCARC: (1.5, -6), (0, -4.5), (1.5, -4.5), CLKW;      [p6 to p8]
  LINE:(-4.5, 0), (0, 0);                                [p8 to p9]

POLYGON:"line6","VOID";
  LINE: (2, -3), (4, -3);                                [p10 to p11]
  LINE: (4, -3), (4, -5);                                [p11 to p12]
  CIRCARC: (4, -5), (2, -5), (3, -5), CLKW;              [p12 to p14]
  LINE: (2, -5), (2, -3);                                [p14 to p15]
```

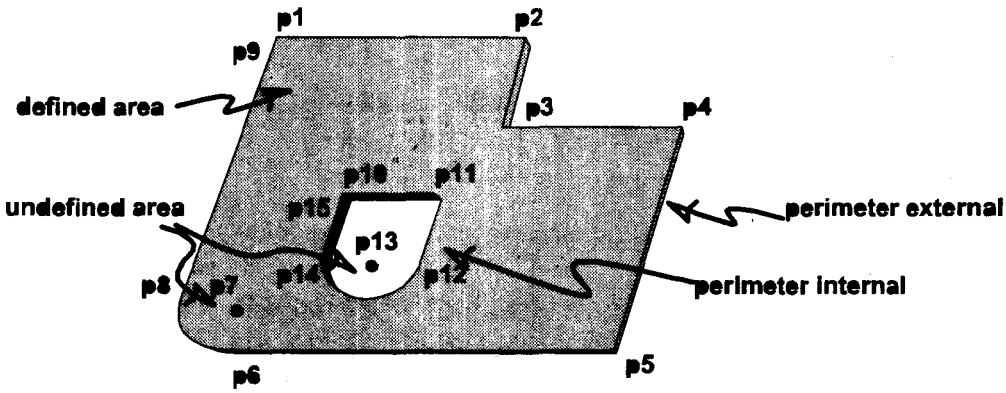


Figure 4-27 Polygon - User Primitive

Note: Polygon definitions may be invoked by identifying their characteristics including the location, coordinate, and a rotation angle. The primitive name is used as the reference.

4.7 Description of Artworks (ARTWORKS)

GenCAM supports the concept of shapes that are more complex than those that can be defined as primitives, through the ARTWORKS section of the GenCAM file. The GenCAM ARTWORKS section is specifically for defining and naming enhanced shapes for later reference. ARTWORKS can be defined as single images or multiple images, using standard or user-defined primitives. The resulting form can be connected or disconnected, but all of the defining primitives **shall** reference the same point of origin.

The ARTWORKS section supports special purpose categories of shapes and collections of shapes, through the statements TARGET, LOGO, UFONT, FEATURE, as well as the generic statement ARTWORK.

Artworks are constructed using either references to primitives as defined in the PRIMITIVES section of the GenCAM file, or using in-line primitive definitions. Throughout this document, these are referred to as <shape_builder> constructions.

<shape_builder> ::= 1{
LINE: <start_xy>, <end_xy>, [<linedesc_ref>], [<color_ref>];

| CIRCARC: <start_xy>, <end_xy>, , [<direction>], [<linedesc_ref>],
[<color_ref>];

| ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>, [<direction>]
[<linedesc_ref>], [<color_ref>];

| CIRCLE:<diameter>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| CIRCLE:<circle_ref>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| RECTCENTER: <x_dimension>, <y_dimension>, [<linedesc_ref>],
[<paintdesc_ref>], [<color_ref>], <xform>;

| RECTCENTER: <rectangle_ref>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| RECTCORNER: <p1>, <p2>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| RECTCORNER: <rectangle_ref>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| RECTCHAM: <width>, <height>, <chamfer>, [<linedesc_ref>],
[<paintdesc_ref>], [<color_ref>], <xform>;

| RECTCHAM: <rectcham_ref>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| RECTROUND: <width>, <height>, <radius>, [<linedesc_ref>],
[<paintdesc_ref>], [<color_ref>], <xform>;

| RECTROUND: <rectround_ref>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| OVAL: <width>, <height>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>],<xform>:

| OVAL: <oval_ref>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| DSHAPE: <end_shape>, <orientation>, <width>, <height>,{<radius> |
<chamfer>}1, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| DSHAPE: <dshape_ref>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| DIAMOND: <width>, <height>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

```
| DIAMOND: <diamond_ref>, [<linedesc_ref>], [<paintdesc_ref>],
                [<color_ref>], <xform>;

| HEXAGON: <point_to_point>, [<linedesc_ref>],
                [<paintdesc_ref>], [<color_ref>], <xform>;

| HEXAGON: <hexagon_ref>, [<linedesc_ref>], [<paintdesc_ref>],
                [<color_ref>], <xform>;

| OCTAGON: <point_to_point>, [<linedesc_ref>], [<paintdesc_ref>],
                [<color_ref>], <xform>;

| OCTAGON: <octagon_ref>, [<linedesc_ref>], [<paintdesc_ref>],
                [<color_ref>], <xform>;

| POLYGON: <polygon_ref>, [<linedesc_ref>], [<paintdesc_ref>],
                [<color_ref>], <xform>;

|POLYGON: [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>];
:>          2{
            LINE: <start_xy>, <end_xy>, [<linedesc_ref>], [<color_ref>];
            | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>],
                [<linedesc_ref>], [<color_ref>];
            | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                [<direction>], [<linedesc_ref>], [<color_ref>];
<:          }n

| POLYLINE: <polyline_ref>, [<linedesc_ref>], [<color_ref>], <xform>;

| POLYLINE: [<linedesc_ref>], [<color_ref>];
:>          1{
            LINE: <start_xy>, <end_xy>, [<linedesc_ref>], [<color_ref>];
            | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>],
                [<linedesc_ref>], [<color_ref>];
            | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                [<direction>], [<linedesc_ref>], [<color_ref>];
<:          }n
        }n
```

4.7.1 TARGETS

Targets are used in printed board descriptions as different patterns to assist in registration of layers to each other, alignment of boards/panels to tooling positions, and to mark significant information on the part, drawing, board, panel or assembly. The following targets have been designated as reserved user-primitives. The formats are as follows:

Named Forms:

```
TARGET: <target_name>, <function>, {<padstack_ref> | <artwork_ref>}1, [<color_ref>];
```

TARGET: <target_name>, <function>, [<color_ref>];
:> {
<shape_builder>
<: }

Reference Form:

TARGET: <target_ref>, [<layer_ref>], [<color_ref>], <xform>;

A <layer_ref> parameter is required if the definition references or defines a multilayered artwork. The <layer_ref> parameter is not valid if the TARGET definition references a pad stack.

Three fixed-field parameters have been reserved for the <function> parameter of the TARGET. The reserved words are:

<function> ::= {REGISTRATION | ALIGNMENT | MARKER}1

4.7.1.1 ALIGNMENT (Fiducials)

Fiducials are solid, typically round images defined by a diameter. (see Figure 4-28) GenCAM uses the TARGET statement with the reserved word ALIGNMENT for these uses.

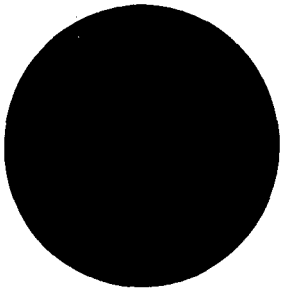


Figure 4-28 Fiducial Artwork

Note: The center of the fiducial is the point of origin.

Example:

```
TARGET: "fiducial1", ALIGNMENT, "artwork3", "Black";
```

4.7.1.2 ALIGNMENT (Butterfly)

A butterfly target is useful as an alignment feature in phototool imaging. (see Figure 4-29) GenCAM uses the TARGET statement and the reserved word ALIGNMENT for this. The butterfly target **shall** be defined as:

```
TARGET: "butterfly3", ALIGNMENT, "artwork16", "Black";
```

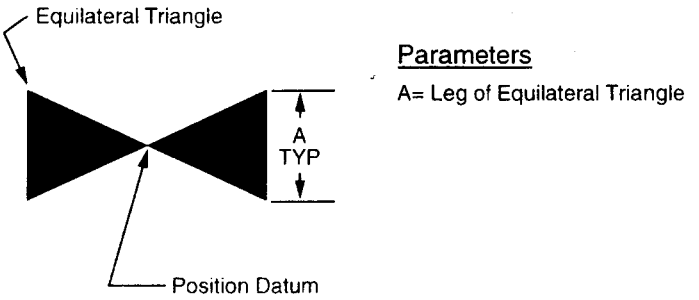


Figure 4-29 Butterfly Artwork

Note: The point where the two equilateral triangles touch is the point of origin. The butterfly artwork is defined in the horizontal position.

4.7.1.3 REGISTRATION (Moire)

Moire patterns are used to register images related to phototools. These are therefore used as targets that allow the registration of one layer to another. Figure 4-30 shows a typical Moire. GenCAM uses the TARGET statement and the reserved word REGISTRATION for this use.

Example:

```
TARGET: "moire1", REGISTRATION, "artwork3";
```

Note: When the "artwork3" is referred to, it's coding would be as follows:

Example:

```
ARTWORK: "artwork3";
  CIRCLE: 4.5, "line10";
  CIRCLE: 4.0, "line10";
  CIRCLE: 3.5, "line10";
  CIRCLE: 3.0, "line10";
  CIRCLE: 2.5, "line10";
  CIRCLE: 2.0, "line10";
```

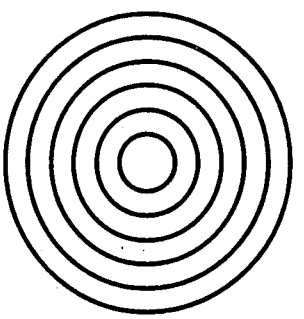


Figure 4-30 Moire ARTWORK

Note: The center of the MOIRE pattern is the point of origin.

4.7.1.4 MARKER (Bullseye)

The bullseye may also be used as a marker in many applications. (see Figure 4-31) GenCAM uses the TARGET statement and the reserved word MARKER for this use.

Example:

```
TARGET: "bullseye", MARKER, "artwork1";
```

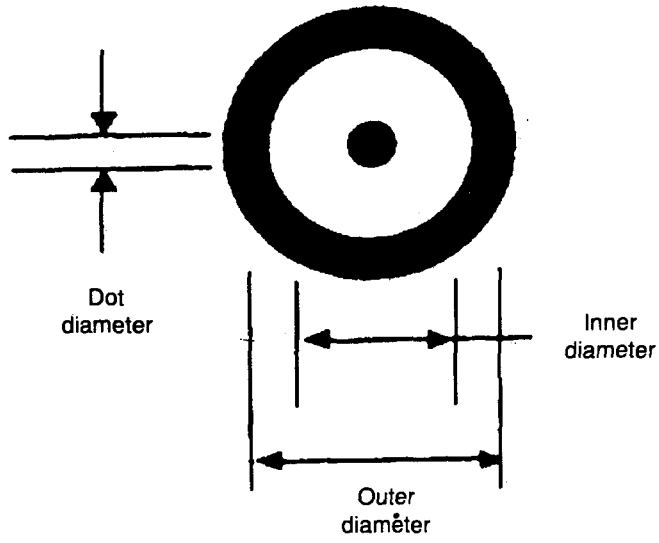


Figure 4-31 Bullseye ARTWORK

Note: The center of the bullseye is the point of origin. It is described as an ARTWORK with a center circle filled, and a DONUT around it.

4.7.2 FEATURE

A variety of features exist that are used in printed boards, printed board drawings, and printed board assembly. Some of these relate to the characteristic for defining polarity of electronic components. The following symbols are appropriate for GenCAM:

Note: Features are also considered as ARTWORKS. They are, therefore, also defined in the ARTWORKS Section.

Named Form:

FEATURE: <feature_name>, [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>];
:> {
<shape_builder>
<: }

Reference Form:

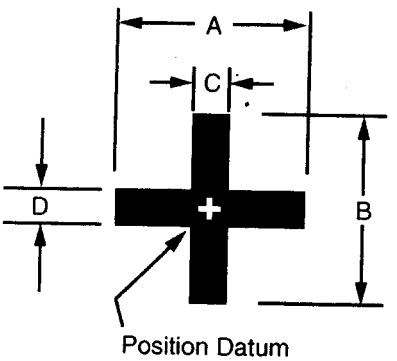
FEATURE: <feature_ref>, [<layer_ref>], [<color_ref>], <xform>;

4.7.2.1 FEATURE - Plus-Sign

The Plus-Sign is a set of lines that cross one another as defined in Figure 4-32. The following characteristics exist and are defined for the cross:

```
FEATURE: "plus_sign1", , FILL, "Black";
  POLYGON: "Cross1";
    LINE: (-1, -4), (-1, -1);
    LINE: (-1, -1), (-1, -1);
    LINE: (-4, -1), (-4, 1);
```

```
LINE: (-4, 1), (-1, 1);
LINE: (-1, 1), (-1, 4);
LINE: (-1, 4), (1, 4);
LINE: (1, 4), (1, 1);
LINE: (1, 1), (4, 1);
LINE: (4, 1), (4, -1);
LINE: (4, -1), (1, -1);
LINE: (1, -1), (1, -4);
LINE: (1, -4), (-1, 4);
```



Parameters

A = Horizontal Member length
B = Vertical Member Length
C = Horizontal Member Width
D = Vertical member Width

Figure 4-32 Crosshair Artwork Feature

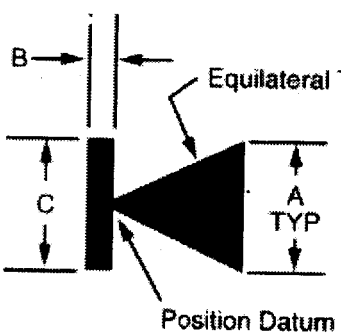
Note: The center of the crosshair is the point of origin.

The Crosshair could also be defined in ARTWORKS as two rectangles and their center. Therefore, the ARTWORK name identifies "rectangle1" having the dimensions "A" and "D", and "rectangle2" having the dimensions "B" and "C".

4.7.2.2 FEATURE - Diode

The diode feature is a triangle with the apex of the triangle touching a vertical line as shown in Figure 4-33. The following characteristics exist and are defined for the diode.

```
FEATURE: "diode7", , FILL, "Black";
  POLYGON: "PolyTriangle1";
   LINE: (0, 0), (4, 2);
   LINE: (4, 2), (4, -2);
   LINE: (4, -2), (0, 0);
  RECTCENTER: "Rectangle1x4", , , , (0, -0.5);
```



Parameters

A= Leg of Equilateral Triangle
B= Width of Bar
C=Length of Bar

Figure 4-33 Diode Artwork Feature

Figure 4-34 shows the shape constructions that are supported by GenCAM, and their relationships to each other.

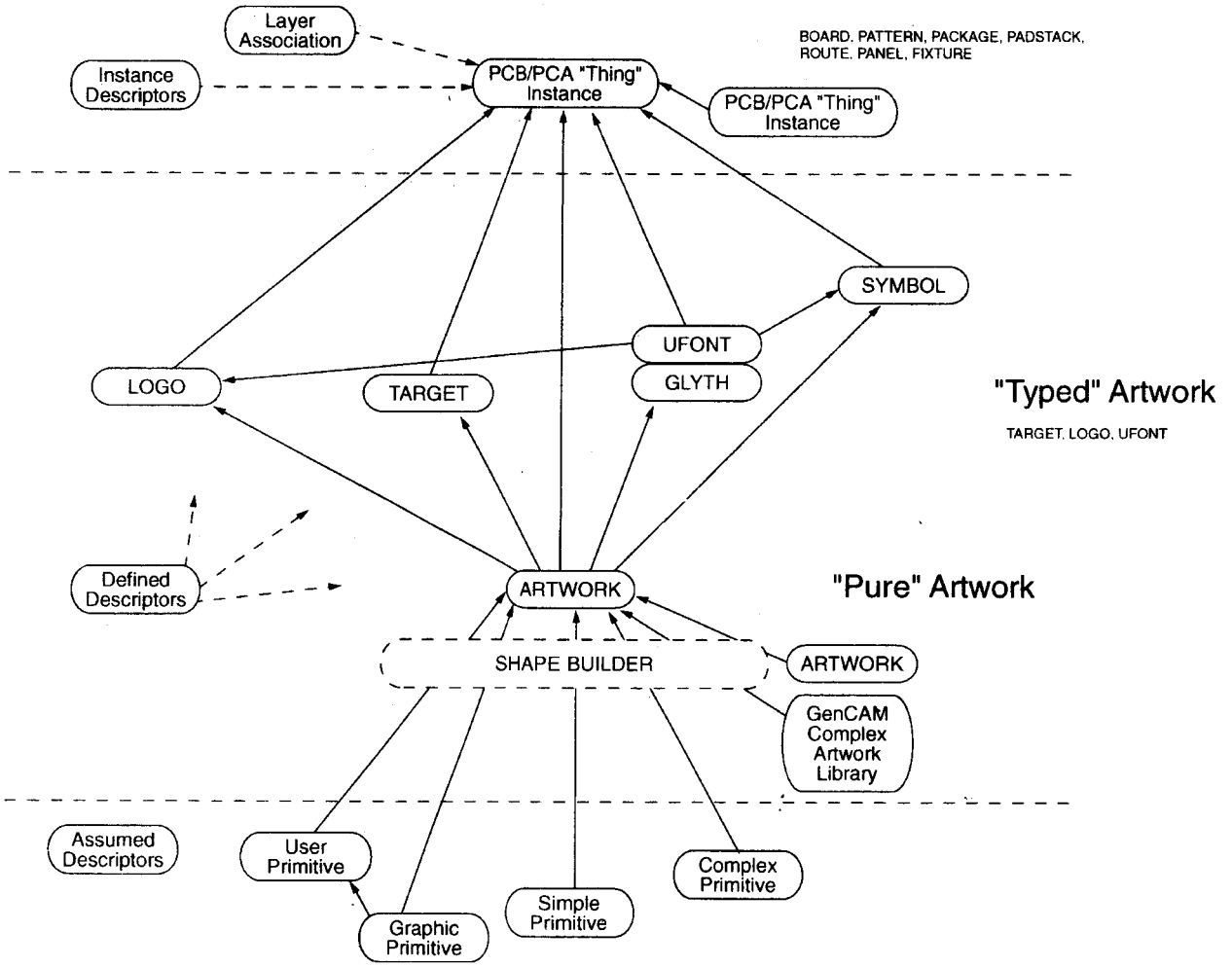


Figure 4-34 Shape Construction

4.7.3 Text parameters (TEXT)

All text is contained in a text box. The text box is defined by the lower x and y coordinate (<p1>) and the upper x and y (<p2>) coordinate. In accordance with the following format:

<text> ::= TEXT: <text_string>, <p1>, <p2>, [<font_ref>], [<color_ref>], <xform>;

The <font_ref> may reference any acceptable font known in the printing industry. *True-Type* ™ fonts are available in both UNIX and PC based CAD and CAM systems, but it should be noted that they are not necessarily identical from system to system. Another acceptable font is *Open-Type* ™. If not specified, the default font for GenCAM is *Helvetica.*

It is important that all text is enclosed in a textbox. This includes upper and lower case letters, as well as all line widths, line descriptions, and line ends. (see Figure 4-35)

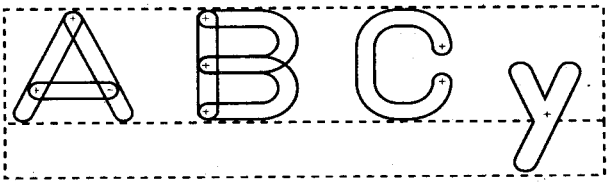


Figure 4-35 Text box to round end character relationships

4.7.3.1 Text character height

Text character height **shall** be controlled by the text box and is identified as in Figure 4-36. The height is positive; a negative height is accomplished through mirroring. Character height is expressed in incremental units of the dimensional characteristics of the file, through the limits (xy coordinates) of the text box. Both upper and lower case letters **shall** be inside the text box. Included in this requirement are the extensions of such descending letters as lower case "g", "q", "y", "j" and "p".

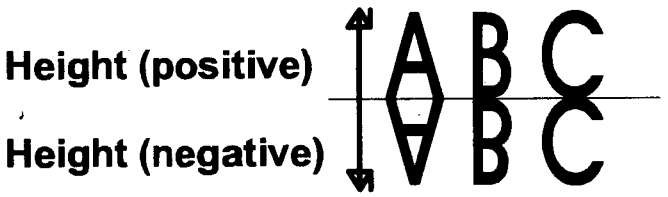


Figure 4-36 Text Character Height

4.7.3.2 Text rotation angle

Figure 4-37 shows the rotation of the text.

```
TEXT: "ABC", (0, 0), (8, 4), , "Blue", (12.0, 6.3);
```

The text box is rotated 30° about the lower xy coordinate.

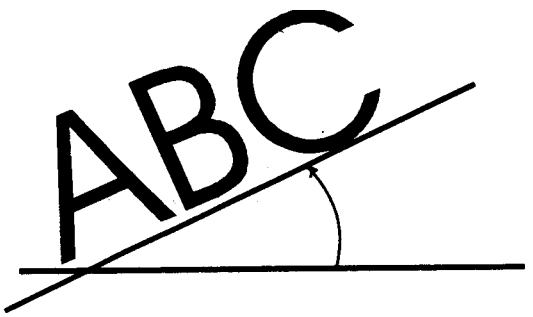


Figure 4-37 Rotation Angle

4.7.4 User-defined character sets (UFONT)

User defined text requires that each character is decided by the user. User defined character sets are defined in the ARTWORKS section of the GenCAM file using the UFONT statement keyword. The UFONT name identifies the collection of letters or symbols established by the user.

The same rules for user fonts apply that can be interpreted to fit within a text box. The major difference is that since UFONT is one character at a time, the entire font is described using a character code for each symbol established as a GLYPH. The GLYPH character code, therefore, identifies the symbol or letter type.

4.7.4.1 Character definition (GLYPH)

When a user defines his own font, font style, or symbols, they are identified one at a time. Standard graphic primitives are used to describe the information and are indented to show relationship with various GLYPH characteristics. Format for UFONT is as follows:

```
<user_font>     ::=     UFONT: <font_name>;
                :>          1{
                                GLYPH: <char_code>, <p1>, <p2>;
                :>                {
                                        <shape_builder>
                <:                }
                <:              }n

<p1>            ::=     <xy_ref> - p1 and p2 define the bounding box of the text string
<p2>            ::=     <xy_ref>
<char_code>     ::=     p_integer
```

The characteristics of all symbols continue with an indenture showing the type of lines, circular arcs, and ellipses used to make these various characteristics.

4.7.5 LOGOS

Logos are user-defined primitives. They may be scaled or used or incorporated into the characteristics for a printed board or printed board assembly. In many instances, the logos are used to identify the company or the equipment which uses a particular characteristic. Some examples are shown in Figure 4-38:

Figure 4-38 Examples of Logos

Because of various different primitives, or previously defined artworks, in order to accomplish the characteristics for a logo description, they are usually provided a user macro name. Logos are then made up of various different primitives in order to accomplish the characteristics for a logo description. The rules for logos are that they must have a start and an end. The information contained within the logo are line segments, circles, other primitives in order to describe the details of the logo. The following rules apply:

Named Form:

```
LOGO: <logo_name>, [<color_ref>];
                :>          1{
                                <shape_builder>
                            | TEXT: <text_string>, <p1>, <p2>, [<font_ref>], [<color_ref>],
                                <xform>;
                <:            }n
```

Reference Form:

```
LOGO: <logo_ref>, [<layer_ref>], [<color_ref>], <xform>;
```

Note: The point of origin for all primitives used to describe the logo must be the same. The point of origin is used to position the logo and/or transform the image.

4.7.6 PATTERNS

User defined patterns are those that consist of multiple images, using multiple standard primitives or artworks. The resulting form can be disconnected, but all of the defined primitives or artworks **shall** reference the same point of origin. The format is as follows:

Named Form:

```
PATTERN: <pattern_name>, [<color_ref>];
        :>          {
                     1{PADSTACK: <padstack_ref>, <pin_physical_ref>, <xform>;}n
                     & 0{ARTWORK: <artwork_ref>, [<layer_ref>], [<color_ref>],
                            <xform>;}n
                     & 0{TARGET: <target_ref>, [<layer_ref>], [<color_ref>],
                            <xform>;}n
                     & 0{FEATURE: <feature_ref>, [<layer_ref>], [<color_ref>],
                            <xform>;}n
        <:           }

PATTERN: <pattern_name>, [<color_ref>];
        :>  {
                  <shape_builder>
        <:  }
```

Reference Form:

```
PATTERN: <pattern_ref>, [<layer_ref>], [<color_ref>], <xform>;
```

A <layer_ref> parameter is required if the definition references or defines a multilayer artwork. The <layer_ref> parameter is not valid if the named PATTERN references a pad stack.

PATTERNS are used instead of ARTWORKS to define those images that are associated with pins of the component and the device placed on the board.

4.7.6.1 Land Patterns

Land patterns are special user-defined patterns made up of land shapes mostly used for surface mounting. The PATTERN may be generic in that they reference an ARTWORK, however in most instances patterns used for component mounting should use a PADSTACK definition. When the PADSTACK is referenced, a pin is also assigned. The <pin_physical_ref> **shall** match the assignment made in the PACKAGE section of the GenCAM file. Some examples of user defined land patterns is shown in Figure 4-39 through 4-41.

```
$PRIMITIVES
  RECTCENTER: "land1",2.0,8.0;
  POLYGON: "poly2";
    LINE: (-0.4, 1.0), (0.4, 1.0);
    LINE: (0.4, 1.0), (0.4, 2.0);
    LINE: (0.4, 2.0), (0.2, 0.2);
    LINE: (0.2, 0.2), (0.2, -0.2);
    LINE: (0.2, -0.2), (0.4, -0.2);
    LINE: (0.4, -0.2), (0.4, -1.0);
    LINE: (0.4, -1.0), (-0.4, -1.0);
    LINE: (-0.4, -1.0), (-0.4, 1.0);
  PAINTDESC: "solid", FILL;
$ENRPRIMITIVES
```

```
$PADSTACKS
  PADSTACK: "Pad1";
   PAD: "BOTH","land1", , solid, , , (0,0)
  PADSTACK: "Notch";
   PAD: "BOTH", "poly2", , solid, , ,(0,0);
$ENDPADSTACKS

$PATTERNS
  PATTERN: "MelfStd";
    PADSTACK: "Pad1", "cathode", (-3.2, 0.0);
    PADSTACK: "Pad1", "anode", (3.2, 0.0);
  PATTERN: "MelfNotched";
    PADSTACK: "Notch", "cathode", (-3.2, 0.0);
    PADSTACK: "Notch", "anode", (3.2, 0.0),
              180.0;
$ENDPATTERNS
```

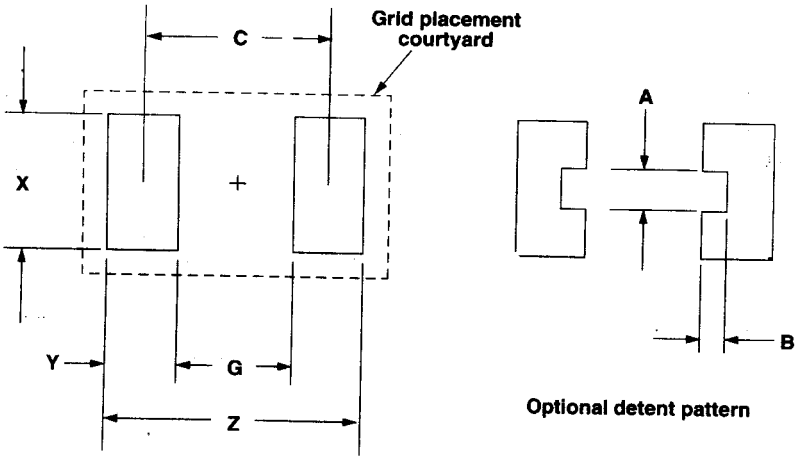


Figure 4-39 MELF Pattern

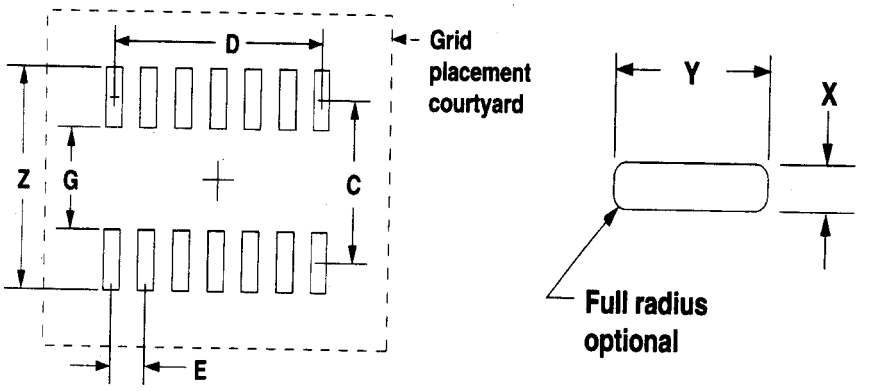


Figure 4-40 SOIC Pattern

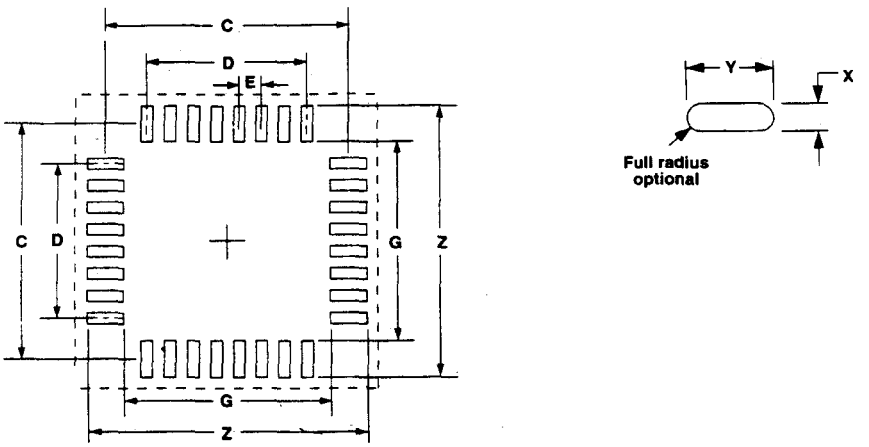


Figure 4-41 Quad flat pack

4.7.6.2 SYMBOL (Schematic or Logic)

Schematic or logic symbols are special user-defined patterns. They are used to describe the information for the characteristics of an electronic symbol intended to have a relationship to the package or the device containing the electronic elements. These symbols are most useful in attempting to take the same methodology used to draw the symbol in DRAWINGS, and correlate it to the manner in which the logic elements are contained within a package. The rules for SYMBOL consist of the following:

Named Form:

SYMBOL: <symbol_name>, [<color_ref>];
:> {
<shape_builder>
& 0 {ARTWORK: <artwork_ref>, [<color_ref>], <xform>;}n
& 0 {TEXT: <text_string>, <p1>, <p2>, [<font_ref>], [<color_ref>],
<xform>;}n
& 0{ SYMPIN: <sympin_name>, <location>;}n
<: }

Reference Form:

SYMBOL: <symbol_ref>, <comp_refdes>, <circuit_ref>, [<color_ref>], <xform>;

Figure 4-42 shows an example of using the SYMBOL definition in the PATTERNS section and correlating that to the DRAWING section.

```
$PATTERNS
  SYMBOL: "NAND";
    CIRCLE: 2.0, , ,(1.3, 2.0);
    LINE: (2.3, 1.7), (2.6, 1.7);
    LINE: (3.5, 2.3), (3.5, 3.6);
    LINE: (4.6, 2.3), (4.2, 2.3);
    CIRCARC: (3.2, 2.2), (2.1,
             3.4),(0.0,0.0);
    SYMPIN: "I1", (3.2, 3.6);
    SYMPIN: "I2", (4.3, 2.4);
    SYMPIN: "Out", (2.4, 1.6);
  SYMBOL: "SWITCH";
    CIRCLE: 2.5, , , ,(4.6, 5.0);
    CIRCLE: 3.7, , , ,(3.8, 4.8);
    LINE: (3.4, 2.4), (3.2, 4.5);
    LINE: (2.4, 4.6), (3.2, 3.2);
    SYMPIN: "p1", (2.4, 4.5);
    SYMPIN: "p2", (3.4, 4.5);
  SYMBOL: "GROUND";
    LINE: (6.7, 3.4), (6.7, 4.6);
    LINE: (6.8, 3.4), (6.8, 4.6);
    SYMPIN: "G", (6.4, 3.3);
  SYMBOL: "CONN";
    LINE: (4.3, 2.2), (4.3, 2.8);
    LINE: (4.0, 2.2), (4.0, 2.8);
    SYMPIN: "C", (4.3,2.0);
  SYMBOL: "AFRAME";
    RECTCENTER: 1.6, 2.8, , , , (3.5,2.4);
    RECTCENTER: 5.2, 3.6, , , , (2.1,2.5);
$ENDPATTERNS
```

```
$DRAWINGS
  SCHEMATIC: "SHEET1",V1;
    FRAME: "AFRAME",(0,0);
      TEXT: "SAMPLE1", (0,0),(2.1, 3.4),
            , ,(3.4, 2.5);
    SYMBOL: "NAND", "U1", "1", ,(3.4, 4.5);
      TEXT: "U1-1", (0,0), (1.4, 3.4), ,
                  (4.5, 3.6);
      SYMPIN: "I1", "-RESET";
      SYMPIN: "I2", "-Q";
      SYMPIN: "OUT", "+Q";
    SYMBOL: "NAND","U1", "2", ,(2.3, 4.4);
      TEXT: "U1-2", (0,0), (1.4, 3.4), ,
            (3.4, 4.2);
      SYMPIN: "I1", "+Q";
      SYMPIN: "I2", "-SET";
      SYMPIN: "OUT", "-Q";
    SYMBOL: "SWITCH", "S1", "1", , (1.0, 2.5);
      TEXT: "U1-2", (0,0), (1.4, 3.4), ,
            (4.6, 3.4);
      SYMPIN: "p1", "-RESET";
      SYMPIN: "p2", "GND";
    SYMBOL: "SWITCH", "S2", "1", ,(3.5, 6.5);
      TEXT: "S2", (0,0), (1.4, 3.4), ,
            (2.4,3.5);
      SYMPIN: "p1", "-SET";
      SYMPIN: "p2", "GND";
    SYMBOL: "GROUND", , , , (3.4, 2.3);
      SYMPIN: "G", "GND";
    SYMBOL: "GROUND", , , , (4.2, 3.5);
      SYMPIN: "G", "GND";
    SYMBOL: "CONN", , , ,(5.2,4.6);
      TEXT: "+Q", (0,0), (1.4,3.4), ,
            (2.7, 3.8);
      SYMPIN: "C", "+Q";
    SYMBOL: "CONN", , , ,(1.1, 2.3);
      TEXT: "-Q", ,(0,0), (1.4, 3.4),
            (0.4, 2.5);
      SYMPIN: "C", "-Q";
    NET: "-RESET";
      LINE: (4.5, 2.3), (3.4, 2.3);
    NET: "-SET";
      LINE: (3.2, 4.5), (3.2, 5.6);
      CIRCLE: 3.25, , , ,(3.5, 4.6);
    NET: "-Q";
      LINE: (4.5, 3.4), (4.5, 1.6);
      CIRCLE: 3.25, , , , (1.3, 2.8);
    NET: "+Q";
      LINE: (1.2, 3.6), (1.2, 4.7);
    NET: "GROUND";
      LINE: (1.4, 2.7), (2.3, 3.6);
$ENDDRAWINGS
```

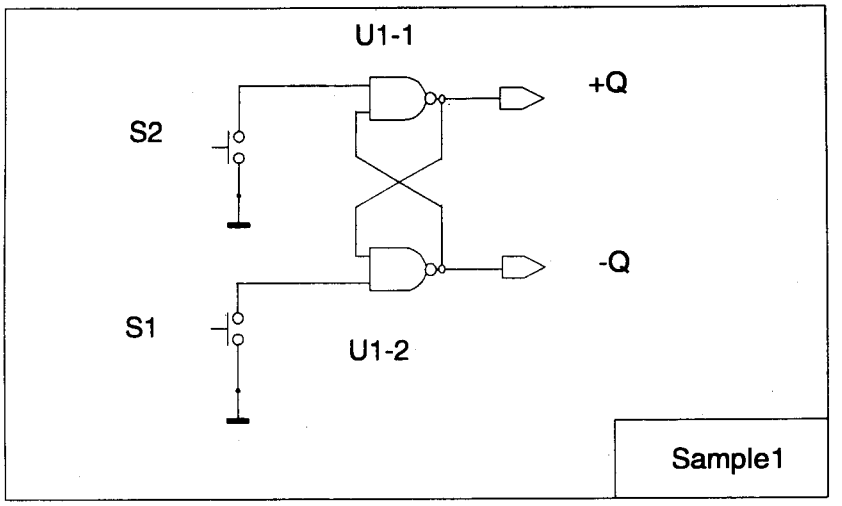


Figure 4-42 Logic Drawing Example

4.7.7 Artwork Pattern Parameter Summary

There are many parameters contained within GenCAM descriptions. The use of primitive information and artwork information to create patterns is of use in many of the GenCAM sections. Figure 4-43 is an illustration that provides the relationship of four levels of data within GenCAM. These levels are:

- Graphic shape definition - use of primitives both standard and user
- PCB and PCA symbol definition - symbols used on printed circuit boards and printed circuit board assemblies as well as their panelization characteristics
- PCB and PCA design definition - physical location of layers, conductors, parts, test parameters for printed boards and printed board assemblies
- Assembly fixture definition - panelization for both the bare board and the assembly and how these might be contained within a fixture.

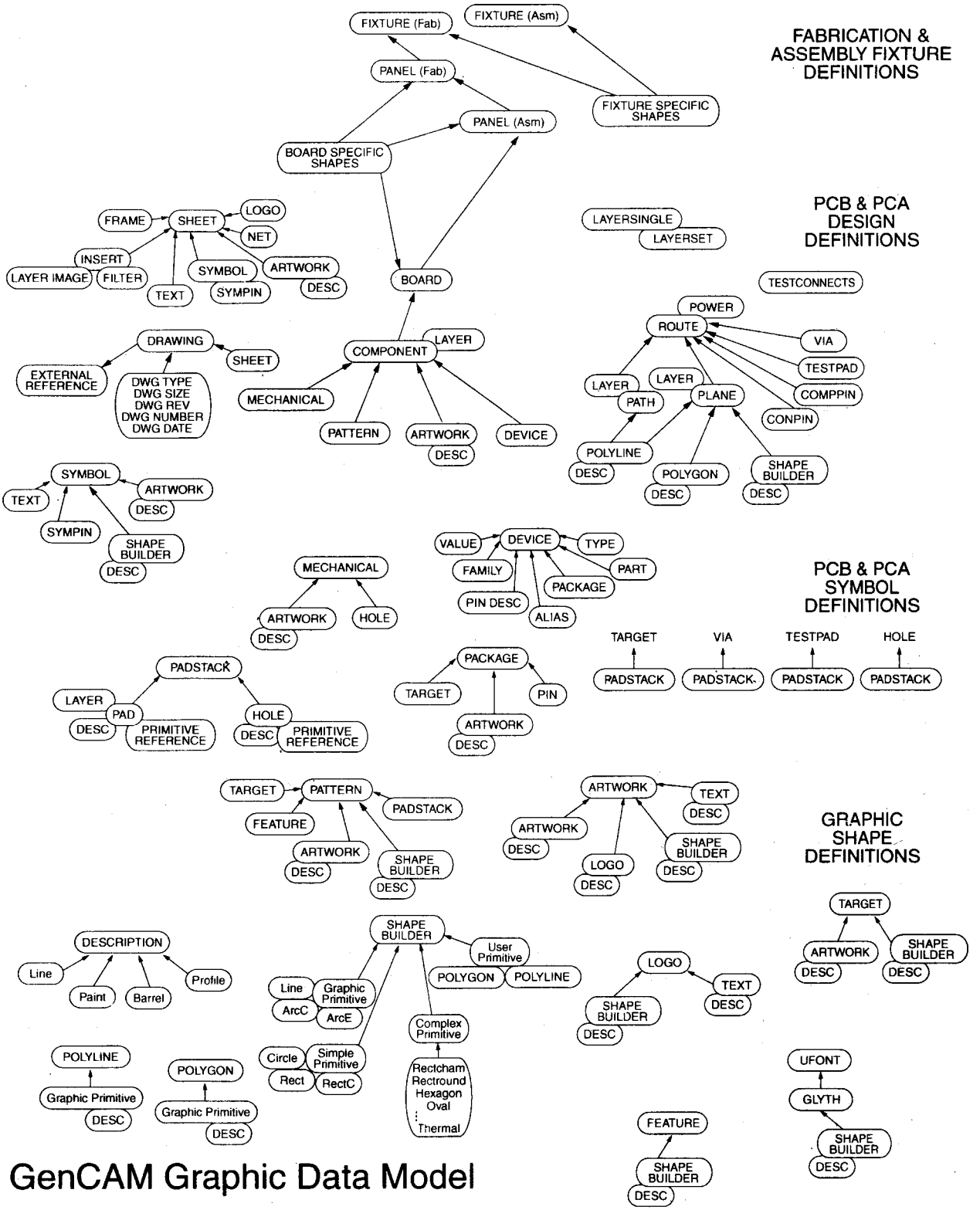


Figure 4-43 GenCAM Graphic Data Model

4.8 SECTIONS

The following generic sections define the characteristics of the GenCAM structure. Each file has its own description and may be made up of several keywords and parameters. All delivered files must have a HEADER as shown in paragraph 4.8.2. The HEADER describes the details that are necessary for managing the GenCAM files that follow.

4.8.1 General Requirements

There are several general requirements that relate to all the sections of a GenCAM file. Some of the requirements are format; others are specific modifications or additional information that may be provided for a particular GenCAM section.

The requirements for file readability are stated in 3.7.3, which indicates that keywords **shall** be indented under any "parent" keyword to which they belong. To aid in the interpretation of the requirements of section 4.8, the symbol ":>", has been applied to the format description to indicate that the following keyword statement, or parameter, **shall** be indented under the preceding keyword. The symbol "<:" is used to indicate that the indentation or parent/child relationship has been completed.

4.8.1.1 The ATTRIBUTE Statement

The ATTRIBUTE statement can be used after any statement within the GenCAM file. The appearance of an ATTRIBUTE statement provides additional information or reference to external data. The ATTRIBUTE is associated with the most immediate prior statement, which is not an ATTRIBUTE, or a MODIFIED statement. The format for attributes consists of three parameters.

```
<attribute_def>      ::=   ATTRIBUTE: <attribute_group>, <attribute_name>, <attribute_value>;
  <attribute_group>  ::=   string
  <attribute_name>   ::=   string
  <attribute_value>  ::=   string
```

All three parameters for an ATTRIBUTE statement are required. The first parameter, <attribute_group>, **shall** be one of the registered GenCAM group names, which represent public available specifications or standards, i.e., IPC-4101. The second parameter, <attribute_name>, **shall** serve to identify the attribute within that group. The third parameter, <attribute_value>, **shall** serve to provide the value to be associated with the named attribute. The current list of registered attributes is available from the web at http://www.gencam.org/registered/attributes. The following example associates an ATTRIBUTE with a LINE statement:

```
LINE: (3.4, 4.0), (3.6, 4.0);
  ATTRIBUTE: "external", "URL", "http://www.acme.com/standard/lines";
```

Only registered attributes are allowed in a GenCAM file. Conforming GenCAM applications must support registered attributes. Attributes should be registered with IPC in order to guarantee unique group names.

Attributes are supported by the language, but are only recorded as strings and passed on as strings without keyword parameter recognition. An attribute statement is assumed to reference the previous, statement.

4.8.1.2 The MODIFIED Statement

Any GenCAM statement may be followed by the MODIFIED statement. The modified keyword is used to associate information relevant to a particular change in a GenCAM file with information about the reason for the change. The MODIFIED keyword pertains to the prior statement.

<modified_def> ::= MODIFIED: <person_ref>, <mod_date>, <mod_description>;
<person_ref> ::= string
<mod_date> ::= date_format
<mod_description> ::= string

All three parameters for a MODIFIED statement are required. The first parameter, <person_ref>, **shall** reference a PERSON defined in the ADMINISTRATION section of the GenCAM file. The second parameter, <mod_date>, is a modification date and uses the standard GenCAM date format. The third parameter, <mod_description>, is the modification text and is a freeform string that provides information as to exactly what has been changed.

Example:

UNITS: THOU10;
MODIFIED: "Bob Neal", 1998-03-24, "all dimensions adjusted to THOU10";

4.8.1.3 The AUDIT Statement

Any GenCAM statement may be followed by an AUDIT statement. The AUDIT statement is used to associate a conformance grade with the statement it describes. This statement may be used to evaluate quality, completeness, design rule adherence, or dam principles.

<audit_def> ::= AUDIT: <audit_type>, <person_ref>, <point_loss>, [<explanation>];
<audit_type> ::= {BDFAB | BDTST | BDASM | ASEMT}1
<person_ref> ::= string
<point_loss> ::= p_number – must be a value between 1 and 10
<explanation> ::= string

All three parameters of an AUDIT statement are required. The first parameter, <audit_type>, **shall** refer to the type of audit being performed. One of the fixed field parameters BDFAB, BDTST, BDASM, or ASEMT **shall** be used. The second parameter, <person_ref>, **shall** reference a PERSON defined in the ADMINISTRATION section of the GenCAM file. The third parameter, <point_loss>, defines the points lost due to an error in using the GenCAM format. The fourth parameter, <explanation> is optional. It is a freeform string than can provide the reason for the point loss. Each category begins with a score of 10 points. Violations subtract points from this score. A score of zero (0) **shall** indicate that the job is completely unacceptable and cannot be started. The point scores and their contribution are defined in IPC-2521 (BDFAB), IPC-2522 (BDTST), IPC-2523 (BDASM), and IPC-2524 (ASEMT).

Example:

AUDIT: BDFAB, "DWB", 3, "No tolerance provided for hole diameters";

4.8.2 HEADER

The HEADER section **shall** be required at the beginning of every GenCAM file. The HEADER **shall** include the GENCAM keyword. Other keywords are used to describe the characteristics of the file, and identifies the products contained there-in.

HEADER ::= $HEADER
:> {
GENCAM: <gencam_rev>;
& 0{ CHARACTERSET: <iso_ref>;}1
& GENERATEDBY: <software_package>, <software_rev>;
& 1{ASSEMBLY: <usedin_name>, <assembly_name>,
<assembly_number>, <assembly_revision>;}n
& 1{BOARD: <usedin_name>, <board_name>, <board_number>,

```
                        <board_revision>;}n
                    & 0{PANEL: <usedin_name>, <panel_name>, <panel_number>,
                        <panel_revision>;}n
                    & 0{FIXTURE: <usedin_name>, <fixture_name>, [<fixture_number>],
                        [<fixture_revision>];}n
                    & UNITS: <dimension>, [<grid_value>];
                    & ANGLEUNITS: <angular_unit>;
                    & 1{COLOR: <color_name>, <R>, <G>, <B>;}n
                    & HISTORY: <increment>;
                <:  }
                    $ENDHEADER

<gencam_rev>        ::=  string  - represents GenCAM revision i.e., 1.0, 1.1, 2.0
<iso_ref>           ::=  string  - ISO Native Language Registration (see Table 3.3)
<software_package>  ::=  string
<software_rev>      ::=  string
<usedin_name>       ::=  string  - unique within the GenCAM file
<assembly_name>     ::=  string
<assembly_number>   ::=  string
<assembly_revision> ::=  string
<board_name>        ::=  string
<board_number>      ::=  string
<board_revision>    ::=  string
<panel_name>        ::=  string
<panel_number>      ::=  string
<panel_revision>    ::=  string
<fixture_name>      ::=  string
<fixture_number>    ::=  string
<fixture_revision>  ::=  string

<dimension>         ::=  {
                         MM          - Millimeter(s)
                        | UM         - Micrometer(s)
                        | MM100      - hundredths of millimeter
                        | INCH       - Inch
                        | THOU       - Inch/1,000
                        | THOU10     - Inch/10,000
                        | USERCM     - Units to the Centimeter
                        | USERMM     - Units to the Millimeter
                        | USERUM     - Units to the Micrometer
                        | USERINCH   - Units to the Inch
                        }1

<grid_value>        ::=  p_integer   -A dimensional value to determine the grid system
                                     used by the CAD system
<angular_unit>      ::=  {RADIANS | DEGREES}1
<increment>         ::=  p_integer   -Initially zero, and incremented every time changes
                                     are incorporated into the file.
<color_name>        ::=  string
<R>                 ::=  p_integer   - (0 - 255) - 0 represents the absence of intensity. i.e,
                                                   0, 0, 0 = "black"; 255, 255, 255 = "white"
<G>                 ::=  p_integer   - (0 - 255)
<B>                 ::=  p_integer   - (0 - 255)
```

4.8.2.1 GENCAM

The GENCAM statement is used to identify the version of GenCAM to which the file conforms. The first parameter, <gencam_rev>, is a string which is to be in a "1.2.3" notation. The "1" is the major revision number, the "2", is a minor revision number, etc.

4.8.2.2 CHARACTERSET

The CHARACTERSET statement defines the character codes that are valid for freeform string parameters throughout the GenCAM file. The first parameter, <iso_ref>, is a valid ISO registration identifier as defined in Table 3.3. If not specified, the default is ISO-002, 7-bit ANSI ASCII character codes. All characters not in freeform strings must be ISO-002 characters.

4.8.2.3 GENERATEDBY

The GENERATEDBY statement identifies how the GenCAM file was created. The first parameter, <software_package>, is a registered name for the software package that generated the GenCAM file. A list of registered package names is available at http://www.gencam.org/registered/software_packages.
The second parameter, <software_rev>, is a revision string indicating the revision of the software package that generated the GenCAM file

4.8.2.4 ASSEMBLY

Each ASSEMBLY statement identifies an assembly that is described in the GenCAM file. The first parameter, <usedin_name>, is a freeform string that is unique to the GenCAM file in which the assembly is used. The second parameter, <assembly_name>, identifies the assembly by name. The third parameter, <assembly_number>, is used to provide the part number of the assembly. The fourth parameter, <assembly_revision>, is used to describe the assembly revision.

4.8.2.5 BOARD

Each BOARD statement identifies a board that is described in the GenCAM file. The first parameter, <usedin_name>, is a freeform string that is unique to the GenCAM file in which the board is defined. The second parameter, <board_name>, identifies the board by name. The third parameter, <board_number>, is used to identify the part number of the board. The fourth parameter, <board_revision>, is used to describe the board revision.

4.8.2.6 PANEL

Each PANEL statement identifies a panel that is described in the GenCAM file. The first parameter, <usedin_name>, is a freeform string that is unique to the GenCAM file in which the panel is used. The second parameter, <panel_name>, identifies the panel or subpanel name. The third parameter, <panel_number>, is used to identify the part number of the panel. The fourth parameter, <panel_revision>, is used to identify the panel revision.

4.8.2.7 FIXTURE

Each FIXTURE statement identifies a fixture that is described in the GenCAM file. The first parameter, <usedin_name>, is a freeform string that is unique to the GenCAM file in which the fixture is defined. The second parameter, <fixture_name>, identifies the fixture by name. The third parameter, <fixture_number>, is used to identify the part number of the fixture. The fourth parameter, <fixture_revision>, is used to describe the fixture revision.

4.8.2.8 UNITS

The UNITS statement defines the unit of measure for length used throughout the GenCAM file. The first parameter, <dimension>, is a fixed field parameter which defines the unit of measure for the file. If the unit of measure is a USER defined unit of type USERCM, USERMM, USERUM, or USERINCH then a

second optional parameter, <grid_value> is required. The <grid_value> is a divisor applied to the base USER unit. (see 6.3). For example, if the USERMM dimension is used and if each grid unit is to be 0.025 millimeters then the <grid_value> would be 40 (1millimeter/40units = .0.025millimeters). For this example <grid_value> the length of LINE: (0,0),(10,0); would be 0.25 millimeters.

4.8.2.9 ANGLEUNITS

The ANGLEUNITS statement defines the angular unit of measure used throughout the GenCAM file. The first parameter, <angular_unit>, is a fixed field parameter which is either DEGREES or RADIANS.

4.8.2.10 COLOR

The COLOR statement is used to name and define colors that are referenced throughout the GenCAM file. Each COLOR statement defines the hue and intensity of one reference color. The definition is given in the emission spectrum, by providing three values in the range 0 to 255 decimal for each of red, green and blue, respectively. The first parameter, <color_name>, is a string that defines the name of the color. The second parameter, <R>, defines the red color intensity, the third parameter, <G>, defines the green color intensity, and the fourth parameter, <B>, defines the blue intensity.

4.8.2.11 HISTORY

The HISTORY statement holds a simple sequential serial change number for the GenCAM file.. The only parameter, <increment>, must be set to zero when the GenCAM file is first produced. This parameter **shall** be incremented every time the GenCAM file is modified.

HEADER Section Example

```
$HEADER
  GENCAM: "1.3";
  GENERATEDBY: "SIML", "P22";
  ASSEMBLY: "C100", "Modem C100 mrboard", "11149-14811",
          "Rev 566g 20";
    ATTRIBUTE: "alpha", "m_part", "BIS 9600";
    ATTRIBUTE: "alpha", "m_desc", "Issue 2";
    MODIFIED: "Harry", 1990-09-20, "changed part number";
  BOARD: "ModemBd", "11354-66540", "R1.3a", "Ver7";
  UNITS: THOU10, 50;
  ANGLEUNITS: DEGREES;
  COLOR: "Black", 0, 0, 0;
  COLOR: "White", 255, 255, 255;
  COLOR: "Shaded", 127, 127, 127;
  HISTORY: 1;
$ENDHEADER
```

4.8.3 ADMINISTRATION

The ADMINISTRATION section is used to identify and describe the customer requirement to the manufacturer, and vice versa. The following BNF defines the valid statements allowed in the ADMINISTRATION section. The BNF also defines whether a statement is mandatory and when, where, and how often it can be used in the section.

<u>ADMINISTRATION</u> ::= $ADMINISTRATION
:> {
0{ASSEMBLY: <usedin_ref>, <line_item>, [<quantity>];}n
& 0{BOARD: <usedin_ref>, <line_item>, [<quantity>];}n
& 0{PANEL: <usedin_ref>, <line_item>, [<quantity>];}n

```
                    & 0{FIXTURE:<usedin_ref>, <line_item>, [<quantity>];}n
                    & TRANSACTION: <trans_type>, <trans_number>, <trans_date>;
                    & 1{
                      PERSON: <person_identifier>, <name>, <enterprise>,
                        <street_addr>, <city>, <state_prov>, <country>, <postal_code>,
                        [<phone>], [<fax>], [<email>], [<url>], [<title>];
                     }n
                    & SENT: <person_ref>;
                    & RECEIVED: <person_ref>;
                    & 0{DESIGNER: <person_ref>;}1
                    & 0{ENGINEER: <person_ref>;}1
                    & 0{BUYER: <person_ref>;}1
                    & 0{CUSTOMERSERVICE: <person_ref>;}1
                    & 0{ACCEPT: <person_ref>;}1
                    & 0{BILLTO: <person_ref>;}1
                    & 1{SCHEDULE: <line_item_ref>, <delivery_date>, <count>;}n
                    & 0{COMMENT: <comment>;}n
               <:   }
                    $ENDADMINISTRATION
```

<usedin_ref>	::=	string -reference to assembly, board, or panel assembly(s), board(s), or panel(s) contained in the file and denoted in the HEADER Section
<line_item>	::=	string - May be a positive integer or an alpha-numeric.
<quantity>	::=	p_integer
<trans_type>	::=	{PO | RFQ | RFP | RFA | CO}1 - type of transaction; either purchase order, request for quote, request for proposal, request for audit, or change order.
<trans_number>	::=	string
<trans_date>	::=	date_format
<person_identifier>	::=	string - unique identifier for an individual who may have multiple responsibilities i.e. send, receive, design, or buy.
<person_ref>	::=	string -a reference to a <person_identifier>
<name>	::=	string - the name of an individual (first, middle, last)
<enterprise>	::=	string - company name | SELF
<street_addr>	::=	string
<city>	::=	string
<state_prov>	::=	string
<country>	::=	string
<postal_code>	::=	string
<phone>	::=	string
<fax>	::=	string -fax machine phone number
<email>	::=	string
<url>	::=	string
<title>	::=	string
<line_item_ref>	::=	string - May be a positive integer or an alpha-numeric.
<delivery_date>	::=	date_format
<count>	::=	p_number
<comment>	::=	string

4.8.3.1 ASSEMBLY

Each ASSEMBLY statement references an assembly as described in the HEADER section, providing order information. The first parameter, <usedin_ref>, references a unique GenCAM <usedin_name>

which must have been declared in the HEADER section. The second parameter, <line_item>, defines the item in a multi-part order, and is an internal identifier that is unique within the GenCAM file. The third parameter, <quantity>, defines the total quantity of the item being ordered using this GenCAM file.

4.8.3.2 BOARD

Each BOARD statement references a board as described in the HEADER section. The first parameter, <usedin_ref>, references a unique name within the GenCAM file, <usedin_name>, which **shall** have been declared by the BOARD statement in the HEADER section. The second parameter, <line_item>, defines the item in a multi-part order, and is an internal identifier that is unique within the GenCAM file. The third parameter, <quantity>, defines the total quantity of the item being ordered using this GenCAM file.

4.8.3.3 PANEL

Each PANEL statement references a panel as described in the HEADER section. The first parameter <usedin_ref>, references a unique <usedin_name>, which **shall** have been declared in the HEADER section of the GenCAM file. The second parameter, <line_item>, defines the item in a multi-part order, and is an internal identifier that is unique within the GenCAM file. The third parameter,<quantity>, defines the total quantity of the item being ordered using this GenCAM file.

4.8.3.4 FIXTURE

Each FIXTURE statement references a fixture as described in the HEADER section. The first parameter, <usedin_ref>, references a unique name, <usedin_name>, within the GenCAM file which **shall** have been declared in the HEADER section. The second parameter, <line_item>, defines the item in a multi-part order, and is an internal identifier that is unique within the GenCAM file. The third parameter,<quantity>, defines the total quantity of the item being ordered using this GenCAM file.

4.8.3.5 TRANSACTION

The TRANSACTION statement identifies the type of transaction the GenCAM file represents. The first parameter, <trans_type>, is a fixed field parameter which identifies the type of transaction, either a purchase order (PO), request for quote (RFQ), request for proposal or information (RFP), request for audit (RFA), or a change order (CO). The second parameter, <trans_number>, is a unique identifier for the transaction. This identifier is defined outside the GenCAM file and is a freeform string. The third parameter, <trans_date>, specifies the date on which the transaction was initiated.

4.8.3.6 PERSON

The PERSON statement defines a unique identifier, <person_identifier>, and associates information about an individual with that unique identifier. The <person_identifier> is used within the GenCAM file whenever a statement parameter requires a reference to an individual. The first parameter, <person_identifier>, is a person identifier unique to the GenCAM file. The second parameter, <name>, is the person's full name. The third parameter, <enterprise>, is the person's company or enterprise; if no enterprise exists, the term SELF should be used. The fourth parameter, <street_address>, is the street address of the enterprise. The fifth parameter, <city>, is the city. The sixth parameter,<state_prov>, is the state or province. The seventh parameter, <country>, is the country. The eighth parameter, <postal_code>, is the postal code. The ninth parameter, <phone>, is the phone number. The tenth parameter, <fax>, is the fax machine number. The eleventh parameter, <email>, is the email address. The twelfth parameter, <url>, is the Web address of the individual. The thirteenth parameter, <title>, is the title of the person used within the enterprise.

4.8.3.7 SENT

The SENT statement identifies the person sending out the GenCAM file. The first parameter, <person_ref>, maps to one of the <person_identifiers> defined in a PERSON statement in the ADMINISTRATION section.

4.8.3.8 RECEIVED

The RECEIVED statement identifies the person receiving the GenCAM file. The first parameter, <person_ref>, maps to one of the <person_identifiers> defined in a PERSON statement in the ADMINISTRATION section.

4.8.3.9 DESIGNER

The DESIGNER statement identifies the designer of the assembly or assemblies described in the GenCAM file. The first parameter, <person_ref>, maps to one of the <person_identifiers> defined in a PERSON statement in the ADMINISTRATION section.

4.8.3.10 ENGINEER

The ENGINEER statement identifies the engineer who is responsible for the assembly or assemblies described in the GenCAM file. The first parameter, <person_ref>, maps to one of the <person_identifiers> defined in a PERSON statement in the ADMINISTRATION section.

4.8.3.11 BUYER

The BUYER statement identifies the person who is responsible for payment. The first parameter, <person_ref>, maps to one of the <person_identifiers> defined in a PERSON statement in the ADMINISTRATION section.

4.8.3.12 CUSTOMERSERVICE

The CUSTOMERSERVICE statement identifies the customer service representative who is responsible for the account. The first parameter, <person_ref>, maps to one of the <person_identifiers> defined in a PERSON statement in the ADMINISTRATION section.

4.8.3.13 ACCEPT

The ACCEPT statement identifies the person in the receiving department who takes possession of the shipment in the name of the enterprise. The first parameter, <person_ref>, maps to one of the <person_identifiers> defined in a PERSON statement in the ADMINISTRATION section.

4.8.3.14 BILLTO

The BILLTO statement identifies the person in the billing or purchasing department to whom the billing should be addressed. The first parameter, <person_ref>, maps to one of the <person_identifiers> defined in a PERSON statement in the ADMINISTRATION section.

4.8.3.15 SCHEDULE

The SCHEDULE statement defines the schedule for the delivery of partial orders. The first parameter, <line_item_ref>, is a reference to a <line_item> defined within the GenCAM file. The second parameter, <delivery_date>, is the date by which the item is to be received. The third parameter, <count>, is the quantity of the items that are to be received on the date specified. The total <count> parameters for any line item should match the total quantity identified for the TRANSACTION of the GenCAM file.

Note: Documents associated with the $ADMINISTRATION section **shall** be contained in $DRAWINGS. That section will define the document types, size, number, revision, date, and all pertinent product descriptions.

4.8.3.16 COMMENT

The COMMENT statement can be used to add information that is considered relevant to the ADMINISTRATION section. The only parameter, <comment>, is a freeform string.

ADMINISTRATION Section Example

```
$ADMINISTRATION
  ASSEMBLY: "Modem", "C100", 1000;
  TRANSACTION: PO, "C10437", "1997-11-13";
  SENT: "Bill";
  PERSON: "bill", "Bill Johnsrud", "Arch Networks", "1525 Central
          Ave.", "Canata", "Ontario", "Canada", "K2K2E6",
          "613.519.3434", "613.519.5960", "bjohnsrud@arch.com";
  RECEIVED: "Hank";
  PERSON: "Hank", "Henry Wright", "FabMax Inc.", "3400 E.
          Industrial Way, #5","Santa Clara", "California",
          "USA", "95130", "408.525.3881", "408.525-3860";
  DESIGNER: "Chris";
  PERSON: "Chris", "Christopher Wible", "Arch Networks", "1525
          Central Avenue", "Kanata", "Ontario","Canada",
          "K2K2E6", "613.519.3544", "613.519.5960",
          "cwible@arch.com", "MTS";
  BUYER: "Miller";
  PERSON: "Miller", "Anne Miller", "Arch Networks", "1525 Central
          Avenue","Kanata", "Ontario", "Canada", "K2K2E6",
          "613.519.8723", "613.519.5961";
  BILLTO: "Miller";
  SCHEDULE: 1, "1997-12-15", 150;
  SCHEDULE: 2, "1998-01-31", 250;
  SCHEDULE: 3, "1998-03-01", 600;
  COMMENT: "Ship all to North American Contract Manufacturer.
           Import Not Guaranteed.";
$ENDADMINISTRATION
```

4.8.4 FIXTURES

The FIXTURES section provides information that is pertinent to one or more fixtures. It also serves to associate board(s) and/or panel(s) with a fixture, providing placement parameters for each. The FIXTURES section is optional, and has the following form:

```
FIXTURES    ::=  $FIXTURES
            :>   1{
                   FIXTURE: <usedin_ref>, <fixture_app>, [<fixture_function>];
            :>     {
                     1{
                        OUTLINE: <layer_ref>;
            :>              1{<polygon_builder>}n
            <:          }n
                      & 0{
                        HOLE: <hole_name>, <hole_type>, <primitive_ref>,
                             <barreldesc_ref>, [<profiledesc_ref>], [<layer_ref>], <xform>;
                        }n
                      & 0{
                        CUTOUT: <cutout_name>, [<layer_ref>];
            :>            {
                                 <shape_builder>
            <:            }
                        }n
                      & 0{
```

```
                    WELL: <well_name>, <nominal_angle>, <surface>, <cutmode>,
                    <rem_value>, [<barreldesc_ref>], [<profiledesc_ref>];
              :>        {
                              <shape_builder>
              <:        }
                    }n
                & 0{
                    SLOT: <slot_name>, <slot_width>, [<layer_ref>];
              :>        <polyline_builder>
              <:    }n
                & 0{
                    KEEPOUT: <keepout_name>, <keepout_type>, [<layer_ref>];
              :>        <polygon_builder>
              <:    }n
                & 1{
                    PLACEMENT: <usedin_ref>, <location>;
                    }n
              <:    }
              <: }n
                $ENDFIXTURES
```

<usedin_ref>	::=	string - for FIXTURE statement references a fixture_name - for PLACEMENT statement references a BOARD or PANEL
<fixture_app>	::=	{BDFAB | BDTST | BDASM | ASEMT}1
<fixture_function>	::=	string
<hole_name>	::=	string
<hole_type>	::=	{ELECTRICAL | MECHANICAL | TOOLING}1
<primitive_ref>	::=	{<circle_ref> | <rectctr_ref> | <rectcnr_ref> | <rectcham_ref> | <rectround_ref> |<oval_ref> | <dshape_ref> |<diamond_ref> | <hexagon_ref> | <octagon_ref>}1
<circle_ref>	::=	string
<rectctr_ref>	::=	string
<rectcnr_ref>	::=	string
<rectcham_ref>	::=	string
<rectround_ref>	::=	string
<oval_ref>	::=	string
<dshape_ref>	::=	string
<diamond_ref>	::=	string
<hexagon_ref>	::=	string
<octagon_ref>	::=	string
<barreldesc_ref>	::=	string
<layer_ref>	::=	{<layersingle_ref> | <layerset_ref>}1
<layersingle_ref>	::=	string
<layerset_ref>	::=	string
<xform>	::=	<xy_ref>, [<rotation>], [<mirror>], [<scale_factor>]
<cutout_name>	::=	string
<well_name>	::=	string
<nominal_angle>	::=	angle_units
<surface>	::=	{TOP | BOTTOM | BOTH}1
<cutmode>	::=	{REMOVE | REMAIN}1
<rem_value>	::=	p_number
<profiledesc_ref>	::=	string
<slot_name>	::=	string
<slot_width>	::=	p_number

<keepout_name> ::= string
<keepout_type> ::= {COMPONENT | VIA | ROUTE | TESTPIN | TESTPROBE | BOARD}1
<location> ::= <xy_ref>, [<rotation>], [<mirror>]

4.8.4.1 FIXTURE

The FIXTURE statement is used to identify the fixture assembly. The first parameter, <usedin_ref>, is required. It is a freeform string, and **shall** identify the fixture by its <usedin_name> as defined in the HEADER section of the GenCAM file. The second parameter, <fixture_app>, is also required, and **shall** identify the fixture application as one of BDFAB, BDTST, BDASM or ASEMT. The third parameter, <fixture_function>, is optional. It is a freeform string that can be used to describe the fixture function. (i.e. "Stencil", "Gluedot")

4.8.4.2 OUTLINE

The OUTLINE statement is used to define the outer periphery of the fixture, using one or more polygon shapes. The first parameter, <layer_ref>, is required and is used to reference a single layer or layer set associated with the fixture. An OUTLINE statement **shall** be followed by either a POLYGON statement that references and locates a closed shape, or by sufficient GenCAM simple primitive statements to describe and locate the closed shape(s). The point of origin of the major outline **shall** be the location to which all the other fixture characteristics, i.e., holes, cutouts, slots, etc. are referenced.

4.8.4.3 HOLE

The HOLE statement is optional, and can be used to name and define a drilled, milled or punched feature of the fixture. The first parameter, <hole_name>, is required. It is a freeform string that serves to uniquely name the hole instance. Each hole defined for the fixture **shall** have a unique name within the fixture description. The second parameter, <hole_type>, is required and **shall** identify the hole use as one of ELECTRICAL, MECHANICAL or TOOLING. The third parameter, <primitive_ref>, is required and **shall** reference a closed, standard or user defined shape, as defined in the PRIMITIVES section of the GenCAM file. The fourth parameter, <barreldesc_ref>, is required and **shall** reference a barrel description as defined in the PRIMITIVES section of the GenCAM file. The fifth parameter, <profiledesc_ref>, is optional. It can serve to define tolerances on the hole size and plating. The sixth parameter, <layer_ref>, is optional. It can serve to reference a single layer or a layer set through which the hole is drilled, as defined in the LAYERS section of the GenCAM file. If a <layer_ref> parameter is not included then all layers are assumed. The last parameter, <xform>, defines the xy location of the hole. The x-y transform is required and **shall** define the relative position of the point of origin of the hole primitive with respect to the fixture origin. The rotation parameter is optional, and can define the angle between the position of the primitive as defined, and its orientation on the fixture, measured counter-clockwise. Any mirroring **shall** be done before the shape is rotated. The mirror parameter is optional, and can be used to indicate that the shape is mirrored.

4.8.4.4 CUTOUT

The CUTOUT statement is used to name and define an internal area of the fixture, which has had one or more layers cut away. The first parameter, <cutout_name>, is required. It is a freeform string that serves to uniquely name the cutout instance. The second parameter, <layer_ref>, is optional, and is used to reference the layer set associated with the cutout. When omitted, a cutout is assumed to include all layers. A CUTOUT statement **shall** be followed by one or more in-line or referenced primitive definitions identified as being part of the set of <shape_builder> descriptions. If simple primitives are used they **shall** describe a closed shape.

4.8.4.5 WELL

The WELL statement is optional, and can be used to identify material removal from the panel. The first parameter, <well_name>, is required. It is a freeform string that **shall** serve to uniquely identify the well instance. The second parameter, <nominal_angle>, is required and **shall** define the nominal angle between the sidewalls of the well. An angle of zero **shall** be understood to define vertical, parallel walls. The third parameter, <surface>, is also required, and **shall** be one of the fixed fields TOP, BOTTOM or BOTH, and **shall** define the side from which the well is milled. The fourth parameter, <cutmode> is also required, and **shall** be one of the fixed fields REMOVE or REMAIN. The fifth parameter, <rem_value> supplements the <cutmode> parameter, and provides the measure of the material removed or the measure of the material that remains. The six parameter, <barreldesc_ref>, is optional, and can reference a barrel description as defined in the PRIMITIVES section of the GenCAM file if the WELL is plated, coated or filled; the default is HOLLOW. The seventh parameter, <profiledesc_ref>, is optional. It can serve to define tolerances on any plating of the well, and when present, **shall** reference a finish profile description as defined in the PRIMITIVES section of the GenCAM file. A WELL statement **shall** be followed by one or more in-line or referenced primitive definitions identified as being part of the set of <shape_builder> descriptions. If simple primitives are used they **shall** describe a closed shape.

4.8.4.6 SLOT

The SLOT statement is optional, and can be used to identify the absence of material in the fixture, which takes the form of a slot. The first parameter, <slot_name>, is required. It is a freeform string that **shall** serve to uniquely identify the slot instance. The second parameter, <slot_width>, is also required, and **shall** define width of the slot. The default condition for a slot is vertical walls, completely through the fixture. The third parameter, <layer_ref>, is optional and is used when a fixture has layer definition to indicate the layer of the fixture that should contain the slot.

The SLOT description **shall** immediately be followed by either a POLYLINE statement that references and locates the slot shape, or by one or more GenCAM Graphic Primitive statements which are used to describe the slot, from beginning to end, as a form of polyline. It is assumed that slots have rounded corners at their ends. The definition of the SLOT is to the center of the radius at the xy location ends of the polyline.

4.8.4.7 KEEPOUT

The KEEPOUT statement is optional, and can be used to name and define an enclosed area of the fixture, inside of which is not accessible for certain manufacturing or test uses. The first parameter, <keepout_name>, is required. It is a freeform string that serves to uniquely name the keepout instance. The second parameter, <keepout_type>, is also required. The keepouts type **shall** be one of COMPONENT, ROUTE, TESTPROBE, TESTPIN or BOARD. The third parameter, <layer_ref>, is optional, and **shall** define the surface to which the keepout region is applied, either a single layer, TOP or BOTTOM, or layerset defined in the LAYERS section. When omitted, both TOP and BOTTOM are assumed. A KEEPOUT statement **shall** be followed by one or more POLYGON statements that reference and locate the keepout shape(s), or by sufficient GenCAM simple primitive statements to describe and locate the closed shape(s).

4.8.4.8 PLACEMENT

The PLACEMENT statement is used to identify each board or panel that is to be placed on the fixture, and to define its location on the fixture. The first parameter, <usedin_ref>, is required. It is a freeform string, and **shall** reference a BOARD or PANEL by its <usedin_name> as defined in the HEADER section of the GenCAM file. The second parameter, <location>, provides the relative x-y position and optional rotation and mirror for the particular board or panel indicated, with respect to fixture's point-of-origin

A FIXTURES Section Example

```
$FIXTURES
  FIXTURE: "ScanDual", BDTST,"Test probe translation #1";
    ATTRIBUTE: "HP", "KIT", "44200L";
    ATTRIBUTE: "HP", "ACTUATION", "PNEUMATIC";
    ATTRIBUTE: "HP", "CONNECT", "SEMIAUTOMATIC";
    ATTRIBUTE: "HP", "TOPPROBES", "ON";
    HOLE: "Thole1", TOOLING, "Round1875", "B1875", , ,
          (41550, 59550);
    HOLE: "Thole2", TOOLING, "Round1875", "B1875", , ,
          (3550, 3550);
    PLACEMENT: "Scan_Xmt", (30198, -77677), 0.0;
    PLACEMENT: "Scan_Rcv", (18198, 62323), 180.0;
  FIXTURE: "ScanBd_Xmt", BDFAB, "Routing template";
    PLACEMENT: "ScanBd_Xmt", (30198, -57677);
$ENDFIXTURES
```

4.8.5 PANELS

The PANELS section references the boards and/or sub-panels that make up the panel along with other data that is pertinent to the panel itself. Included are the outer shape and any internal cutouts of the panel in terms of absolute x and y coordinates as viewed from the top or primary side. The PANELS section is required when the board assembly is panelized, and has the following form.

```
PANELS      ::=  $PANELS
            :>   1{
                  PANEL: <usedin_ref>;
            :>    {
                   1{
                      OUTLINE: <layer_ref>;
            :>         {
                          1{<polygon_builder>}n
                          & 0{THICKNESS: <dim_value>, <profiledesc_ref>;}1
            <:         }
                    }n
                   & 0{
                      CUTOUT: <cutout_name>, [<layer_ref>];
            :>           {
                                <shape_builder>
            <:           }
                    }n
                   & 0{
                      WELL: <well_name>, <nominal_angle>, <surface>, <cutmode>,
                      <rem_value>, [<barreldesc_ref>], [<profiledesc_ref>];
            :>           {
                                <shape_builder>
            <:           }
                    }n
                   & 0{
                      SLOT: <slot_name>, <slot_width>, [<layer_ref>];
            :>           <polyline_builder>
            <:      }n
                   & 0{
                      GROOVE: <groove_name>, <groove_width>, <nominal_angle>,
```

```
                                    <surface>, <cutmode>, <rem_value>,
                                    [<profiledesc_ref>];
                :>                      <polyline_builder>
                <:                  }n
                                  & 0{
                                    HOLE: <hole_name>, <hole_type>, <primitive_ref>,
                                      <barreldesc_ref>, [<profiledesc_ref>], [<layer_ref>], <xform>;
                                    }n
                                  & 0{
                                    TARGET: <target_ref>, [<layer_ref>], [<color_ref>], <xform>;
                                    }n
                                  & 0{
                                    KEEPOUT:<keepout_name>, <keepout_type>, [<layer_ref>];
                :>                      <polygon_builder>
                <:                  }n
                                  & 1{
                                    PLACEMENT:<usedin_ref>, <location>;
                                    }n
                <:                }
                <:          }n
                            $ENDPANELS

<usedin_ref>        ::=     string  - in PANEL statement references a <panel_name> in  $HEADER
                                    - in PLACEMENT statement references a board, assembly, or a
                                      subpanel
<layer_ref>         ::=     {<layersingle_ref> | <layerset_ref>}1  -  The default is ALL.
<layersingle_ref>   ::=     string
<layerset_ref>      ::=     string
<dim_value>         ::=     p_number
<profiledesc_ref>   ::=     string
<cutout_name>       ::=     string
<well_name>         ::=     string
<nominal_angle>     ::=     angle_units
<surface>           ::=     {TOP | BOTTOM | BOTH}1
<cutmode>           ::=     {REMOVE | REMAIN}1
<rem_value>         ::=     p_number
<barreldesc_ref>    ::=     string
<slot_name>         ::=     string
<slot_width>        ::=     p_number
<groove_name>       ::=     string
<groove_width>      ::=     p_number
<hole_name>         ::=     string
<hole_type>         ::=     {MECHANICAL | ELECTRICAL | TOOLING}1
<primitive_ref>     ::=     {<circle_ref> | <rectctr_ref> | <rectcnr_ref> | <rectcham_ref> |
                            <rectround_ref> |<oval_ref> | <dshape_ref> |<diamond_ref> |
                            <hexagon_ref> | <octagon_ref>}1
<circle_ref>        ::=     string
<rectctr_ref>       ::=     string
<rectcnr_ref>       ::=     string
<rectcham_ref>      ::=     string
<rectround_ref>     ::=     string
<oval_ref>          ::=     string
<dshape_ref>        ::=     string
<diamond_ref>       ::=     string
```

<hexagon_ref>	::=	string
<octagon_ref>	::=	string
<xform>	::=	<xy_ref>, [<rotation>],[<mirror>],[<scale_factor>]
<target_ref>	::=	string
<color_ref>	::=	string - color name identified in $HEADER; the default is "black"
<keepout_name>	::=	string
<keepout_type>	::=	{COMPONENT | VIA | ROUTE | TESTPIN | TESTPROBE | BOARD}1
<location>	::=	<xy_ref>, [<rotation>], [<mirror>]

4.8.5.1 PANEL

The PANEL statement is used to identify the printed board panel or assembly subpanel. The first parameter, <usedin_ref>, **shall** be the unique name of the panel or subpanel assigned in the HEADER of the GenCAM file.

4.8.5.2 OUTLINE

The OUTLINE statement is used to define the outer periphery of the panel, using one or more polygon shapes. The first parameter, <layer_ref>, is required and is used to reference a single layer or layer set associated with the panel. An OUTLINE statement **shall** be followed by either a POLYGON statement that references and locates a closed shape, or by sufficient GenCAM simple primitive statements to describe and locate the closed shape(s). The point of origin of the major outline **shall** be the location to which all the other panel characteristics, i.e., holes, cutouts, slots, etc. are referenced.

The optional THICKNESS statement is used to specify the thickness of the panel(s). The first parameter, <dim_value>, defines the thickness in dimension units as specified in the HEADER section. The second parameter, <profiledesc_ref>, references the tolerances on the thickness of the overall panel.

4.8.5.3 CUTOUT

The CUTOUT statement is used to name and define an internal area of the panel, which has had one or more layers cut away. The first parameter, <cutout_name>, is required. It is a freeform string that serves to uniquely name the cutout instance. The second parameter, <layer_ref>, is optional, and is used to reference the individual layer or layer set associated with the cutout. When omitted, a cutout is assumed to include all layers. A CUTOUT statement **shall** be followed by one or more in-line or referenced primitive definitions identified as being part of the set of <shape_builder> descriptions. If simple primitives are used they **shall** describe a closed shape.

4.8.5.4 WELL

The WELL statement is optional, and can be used to identify material removal from the panel. The first parameter, <well_name>, is required. It is a freeform string that **shall** serve to uniquely identify the well instance. The second parameter, <nominal_angle>, is required and **shall** define the nominal angle between the sidewalls of the well. An angle of zero **shall** be understood to define vertical, parallel walls. The third parameter, <surface>, is also required, and **shall** be one of the fixed fields TOP, BOTTOM or BOTH, and **shall** define the side from which the well is milled. The fourth parameter, <cutmode>, is also required, and **shall** be one of the fixed fields REMOVE or REMAIN. The fifth parameter, <rem_value> supplements the <cutmode> parameter, and provides the measure of the material removed or the measure of the material that remains. The six parameter, <barreldesc_ref>, is optional, and can reference a barrel description as defined in the PRIMITIVES section of the GenCAM file if the WELL is plated, coated or filled; the default is HOLLOW. The seventh parameter, <profiledesc_ref>, is optional. It can serve to define tolerances on any plating of the well, and when present, **shall** reference a finish profile description as defined in the PRIMITIVES section of the GenCAM file. A WELL statement **shall** be followed by one or more in-line or referenced primitive definitions identified as being part of the set of <shape_builder> descriptions. If simple primitives are used, they **shall** describe a closed shape.

4.8.5.5 SLOT

The SLOT statement is optional, and can be used to identify slots in the panel that may be used for de-panelization or other purposes. The first parameter, <slot_name>, is required. It is a freeform string that **shall** serve to uniquely identify the slot instance. The second parameter, <slot_width>, is also required, and **shall** define width of the slot. The default condition for a slot is vertical walls, completely through the fixture. The third parameter, <layer_ref>, is optional, and is used to reference the single layer or layer set associated with the slot. When omitted, a slot is assumed to include all layers.

The SLOT description **shall** be immediately followed by either a POLYLINE statement that references and locates the slot shape, or by one or more GenCAM Graphic Primitive statements which are used to describe the slot, from beginning to end, as a form of polyline. It is assumed that slots have rounded corners at their ends. The definition of the slot is to the center of the radius at the xy location ends of the polyline.

4.8.5.6 GROOVE

The GROOVE statement is optional, and can be used to identify grooves cut in the panel used for de-panelization. The first parameter, <groove_name>, is required. It is a freeform string that **shall** serve to uniquely identify the groove instance. The second parameter, <groove_width>, is also required, and **shall** define the width of the groove at its widest section. The third parameter of the GROOVE statement, <nominal_angle>, **shall** define the nominal angle between the sidewalls of the groove. An angle of zero **shall** be understood to define vertical, parallel walls. The fourth parameter, <surface>, is also required, and **shall** be one of the fixed fields TOP, BOTTOM or BOTH, and **shall** define the side from which the groove is ground. The fifth parameter, <cutmode>, is also required and **shall** be one of the fixed fields REMOVE or REMAIN. The sixth parameter, <rem_value> supplements the REMOVE or REMAIN parameters, and provides the measure of the material removed or the measure of the material that remains. The seventh parameter, <profiledesc_ref>, is optional. When present, it **shall** reference a profile description as defined in the PRIMITIVES section of the GenCAM file. It represents the tolerances on the material stated in <rem_value>. The GROOVE statement **shall** be follow by a POLYLINE statement that references and locates the center of the groove at its xy locations on the panel.

4.8.5.7 HOLE

The HOLE statement is optional, and can be used to name and define a drilled or punched feature of the panel. The first parameter, <hole_name>, is required. It is a freeform string that serves to uniquely name the hole instance. Each hole defined for the panel **shall** have a unique name within the panel description. The second parameter, <hole_type>, is required and **shall** identify the hole use as one of ELECTRICAL, MECHANICAL or TOOLING. The third parameter, <primitive_ref>, is required and **shall** reference a closed, standard or user defined shape, as defined in the PRIMITIVES section of the GenCAM file. The fourth parameter, <barreldesc_ref>, is required and **shall** reference a barrel description as defined in the PRIMITIVES section of the GenCAM file. The fifth parameter, <profiledesc_ref>, is optional. It can serve to define tolerances on the hole size and plating. The sixth parameter, <layer_ref>, is optional. It can define the individual layer, <layersingle>, or group of layers, <layerset>, through which the hole is drilled, as defined in the LAYERS section of the GenCAM file. If a <layer_ref> parameter is not included then all layers are assumed.

The remaining parameters define the placement of the hole. The x-y transform is required and **shall** define the relative position of the point of origin of the hole primitive with respect to the panel origin. The rotation parameter is optional, and can define the angle between the orientation of the primitive as defined, and its position on the panel, measured counter-clockwise. Any mirroring **shall** be done before the shape is rotated. The mirror parameter is optional, and can be used to indicate that the hole is mirrored before becoming part of the panel.

4.8.5.8 TARGET

The TARGET statement is optional, and can be used to name and define a z-axis alignment or x-y registration feature (e.g. fiducial), or a bad-board indicator. The first parameter, <target_ref>, is required. It is a freeform string that serves to reference the target definition in the ARTWORKS section of the GenCAM file. The second parameter, <layer_ref>, is optional. It can define the individual layer, <layersingle>, or group of layers, <layerset>, to which the TARGET should be applied, as defined in the LAYERS section of the GenCAM file. If a <layer_ref> parameter is not included then all layers are assumed. The third parameter, <color_ref> is optional, and can be used to reference a color as defined by the COLOR statement in the HEADER section of the GenCAM file.

The remaining parameters define the <xform> of the target. The <xform> is composed of an <xy_ref> parameter that is required and **shall** define the relative position of the point of origin of the target with respect to the panel origin. A <rotation> parameter is optional, and can define the angle between the position of the target as defined, and its orientation on the panel, measured counter-clockwise. Any mirroring **shall** be done before the target is rotated. The <mirror> parameter is optional, and can be used to indicate that the artwork is mirrored. The shape can be placed normally or can be mirrored before becoming part of the panel. The last parameter is an optional <scale_factor> multiplier that is applied to all dimensions of the target.

4.8.5.9 KEEPOUT

The KEEPOUT statement is optional, and can be used to name and define an area of the panel, inside which is not accessible for certain manufacturing or test uses. The shape defined for the keepout region must be closed. The first parameter, <keepout_name>, is required. It is a freeform string that serves to uniquely name the keepout instance. The second parameter, <keepout_type>, **shall** be one of the areas or physical features to which KEEPOUT applies; these are COMPONENT, VIA, ROUTE, TESTPIN, TESTPROBE, and BOARD. The third parameter, <layer_ref>, is optional. It can define the individual layer, <layersingle>, or group of layers, <layerset>, to which the keepout area applies, as defined in the LAYERS section of the GenCAM file. If a <layer_ref> parameter is not included then all layers are assumed. . A KEEPOUT statement **shall** be followed by a POLYGON statement that reference and locate the keepout shape, or by sufficient GenCAM simple primitive statements to describe and locate the closed shape(s).

4.8.5.10 PLACEMENT

The PLACEMENT statement is used to locate a board or a sub-panel within the manufacturing panel. The first parameter, <usedin_ref>, is required. It **shall** reference a PANEL or BOARD by its <usedin_name> as defined in the HEADER section of the GenCAM file. The second parameter, <location>, is required and provides the position and optional rotation and mirror characteristics. The x-y transform of the <location> parameter is required and **shall** define the relative position of the board or subpanel with respect to the fabrication PANEL origin. The rotation parameter is optional, and can define the angle between the orientation of the board or sub-panel as defined, and its position on the panel, measured counter-clockwise. Any mirroring **shall** be done before the shape is rotated. The mirror parameter is optional, and can be used to indicate that the shape can is mirrored before becoming part of the panel

A PANELS Section Example

```
$PANELS
  PANEL: "4UP_MODUS";
    OUTLINE: "All";
      LINE: (0, 0), (0, 175000);
      LINE: (0, 175000), (140000, 175000);
      LINE: (14000, 175000), (140000, 0);
      LINE: (140000, 0), (0, 0);
```

```
      THICKNESS: 150, "laminate4";
    PLACEMENT:"11354-66540RevA", (0, 0), 0.0;
    PLACEMENT:"11354-66540RevA", (8484, 1031), 180.0, MIRROR;
    PLACEMENT:"11354-66540RevA", (0, 86017), 0.0;
    PLACEMENT:"11354-66540RevA", (58484,87033), 180.0, MIRROR;
    HOLE: "Thole1", TOOLING, "Round1875", "B1875", , ,
          (41550, 59550);
    HOLE: "Thole2", TOOLING, "Round1875", "B1875", , ,
          (3550, 3550);
$ENDPANELS
```

4.8.6 BOARDS

The BOARDS section defines the outer shape and any internal cutouts of the printed circuit or printed wiring board in terms of absolute x and y coordinates as viewed from the top or primary side. The $BOARDS and $ENDBOARDS section keywords mark a BOARDS section of the GenCAM file. The BOARDS section is required and **shall** have the following form.

```
BOARDS      ::=  $BOARDS
            :>   1{
                  BOARD: <usedin_ref>;
            :>    {
                    1{
                      OUTLINE: <layer_ref>;
            :>         {
                         1{<polygon_builder>}n
                         & 0{THICKNESS: <dim_value>, <profiledesc_ref>;}1
            <:         }
                    }n
                    & 0{
                      CUTOUT: <cutout_name>, [<layer_ref>];
            :>          {
                              <shape_builder>
            <:          }
                    }n
                    & 0{
                      WELL: <well_name>, <nominal_angle>, <surface>, <cutmode>,
                        <rem_value>, [<barreldesc_ref>], [<profiledesc_ref>];
            :>          {
                              <shape_builder>
            <:          }
                    }n
                    & 0{
                      SLOT: <slot_name>, <slot_width>, [<layer_ref>];
            :>          <polyline_builder>
            <:        }n
                    & 0{
                      GROOVE: <groove_name>, <groove_width>, <nominal_angle>,
                        <surface>, <cutmode>, <rem_value>, [<profiledesc_ref>];
            :>          <polyline_builder>
            <:        }n
                    & 0{
                      HOLE: <hole_name>, <hole_type>, <primitive_ref>,
                        <barreldesc_ref>, [<profiledesc_ref>], [<layer_ref>], <xform>;
```

```
                    }n
                  & 0{
                    TARGET: <target_ref>, [<layer_ref>], [<color_ref>], <xform>;
                    }n
                  & 0 {
                    LOGO: <logo_ref>, [<layer_ref>], [<color_ref>], <xform>;
                    }n
                  & 0{
                    KEEPOUT:<keepout_name>, <keepout_type>, [<layer_ref>];
:>                      <polygon_builder>
<:                  }n
                  & BAREBOARDTEST: <continuity_voltage>, <continuity_current>,
                        <isolation_voltage>;
<:                }
<:          }n
            $ENDBOARDS
```

```
<usedin_ref>          ::=   string  - references a <board_name> in the HEADER section
<layer_ref>           ::=   {<layersingle_ref> | <layerset_ref>}1  -  The default is ALL
<layersingle_ref>     ::=   string
<layerset_ref>        ::=   string
<dim_value>           ::=   p_number
<profiledesc_ref>     ::=   string
<cutout_name>         ::=   string
<well_name>           ::=   string
<nominal_angle>       ::=   angle_units
<surface>             ::=   {TOP | BOTTOM | BOTH}1
<cutmode>             ::=   {REMOVE | REMAIN}1
<rem_value>           ::=   p_number
<barreldesc_ref>      ::=   string
<slot_name>           ::=   string
<slot_width>          ::=   p_number
<groove_name>         ::=   string
<groove_width>        ::=   p_number
<hole_name>           ::=   string
<hole_type>           ::=   {MECHANICAL | ELECTRICAL | TOOLING}1
<primitive_ref>       ::=   {<circle_ref> | <rectctr_ref> | <rectcnr_ref> | <rectcham_ref> |
                            <rectround_ref> |<oval_ref> | <dshape_ref> |<diamond_ref> |
                            <hexagon_ref> | <octagon_ref>}1
<circle_ref>          ::=   string
<rectctr_ref>         ::=   string
<rectcnr_ref>         ::=   string
<rectcham_ref>        ::=   string
<rectround_ref>       ::=   string
<oval_ref>            ::=   string
<dshape_ref>          ::=   string
<diamond_ref>         ::=   string
<hexagon_ref>         ::=   string
<octagon_ref>         ::=   string
<xform>               ::=   <xy_ref>, [<rotation>],[<mirror>],[<scale_factor>]
<target_ref>          ::=   string
<color_ref>           ::=   string  - color name identified in $HEADER; the default is "black"
<logo_ref>            ::=   string
<keepout_name>        ::=   string
```

<keepout_type>	::=	{COMPONENT | VIA | ROUTE | TESTPIN | TESTPROBE | BOARD}1	
<continuity_voltage>	::=	p_number	- Units **shall** be expressed in volts
<continuity_current>	::=	p_number	- Units **shall** be expressed in amps
<isolation_voltage>	::=	p_number	- Units **shall** be expressed in volts

4.8.6.1 BOARD

The BOARD statement is used to identify a printed board or printed board coupon. The first parameter, <usedin_ref>, shall be the unique name of the board assigned in the HEADER of the GenCAM file.

4.8.6.2 OUTLINE

The OUTLINE statement is used to define the outer periphery of the board, using one or more polygon shapes. The first parameter, <layer_ref>, is required and is used to reference a single layer or layer set associated with the board. An OUTLINE statement **shall** be followed by either a POLYGON statement that references and locates a closed shape, or by sufficient GenCAM simple primitive statements to describe and locate the closed shape(s). The point of origin of the major outline **shall** be the location to which all the other board characteristics, i.e., holes, cutouts, slots, etc. are referenced.

The optional THICKNESS statement is used to specify the thickness of the board(s). The first parameter, <dim_value>, defines the thickness in dimension units as specified in the $HEADER section. The second parameter, <profiledesc_ref>, references the tolerances on the thickness of the overall panel.

4.8.6.3 CUTOUT

The CUTOUT statement is used to name and define an internal area of the board, which has had one or more layers cut away. The first parameter, <cutout_name>, is required. It is a freeform string that serves to uniquely name the cutout instance. The second parameter, <layer_ref>, is optional, and is used to reference the individual layer, or layer set associated with the cutout. When omitted, a cutout is assumed to include all layers. A CUTOUT statement **shall** be followed by one or more in-line or referenced primitive definitions identified as being part of the set of <shape_builder> descriptions. If simple primitives are used they **shall** describe a closed shape.

4.8.6.4 WELL

The WELL statement is optional, and can be used to identify material removal from the board. The first parameter, <well_name>, is required. It is a freeform string that **shall** serve to uniquely identify the well instance. The second parameter, <nominal_angle>, is required and **shall** define the nominal angle between the sidewalls of the well. An angle of zero **shall** be understood to define vertical, parallel walls. The third parameter, <surface>, is also required, and **shall** be one of the fixed fields TOP, BOTTOM or BOTH, and **shall** define the side from which the well is milled. The fourth parameter, <cutmode>, is also required, and **shall** be one of the fixed fields REMOVE or REMAIN. The fifth parameter, <rem_value> supplements the <cutmode> parameter, and provides the measure of the material removed or the measure of the material that remains. The six parameter, <barreldesc_ref>, is optional, and can reference a barrel description as defined in the PRIMITIVES section of the GenCAM file if the WELL is plated, coated or filled; the default is HOLLOW. The seventh parameter, <profiledesc_ref>, is optional. It can serve to define tolerances on any plating of the well, and when present, **shall** reference a finish profile description as defined in the PRIMITIVES section of the GenCAM file. A WELL statement **shall** be followed by one or more in-line or referenced primitive definitions identified as being part of the set of <shape_builder> descriptions. If simple primitives are used, they **shall** describe a closed shape.

4.8.6.5 SLOT

The SLOT statement is optional, and can be used to identify slots in the board. The first parameter, <slot_name>, is required. It is a freeform string that **shall** serve to uniquely identify the slot instance. The

second parameter, <slot_width>, is also required, and **shall** define width of the slot. The default condition for a slot is vertical walls, completely through the board. The third parameter, <layer_ref>, is optional, and is used to reference the single layer or layer set associated with the slot. When omitted, a slot is assumed to include all layers.

The SLOT description **shall** be immediately followed by either a POLYLINE statement that references and locates the slot shape, or by one or more GenCAM Graphic Primitive statements which are used to describe the slot, from beginning to end, as a form of polyline. It is assumed that slots have rounded corners at their ends. The definition of the slot is to the center of the radius at the xy location ends of the polyline.

4.8.6.6 GROOVE

The GROOVE statement is optional, and can be used to identify grooves cut in the board for segmentation or other purposes. The first parameter, <groove_name>, is required. It is a freeform string that **shall** serve to uniquely identify the groove instance. The second parameter, <groove_width>, is also required, and **shall** define the width of the groove at its widest section. The third parameter of the GROOVE statement, <nominal_angle>, **shall** define the nominal angle between the sidewalls of the groove. An angle of zero **shall** be understood to define vertical, parallel walls. The fourth parameter, <surface>, is also required, and **shall** be one of the fixed fields TOP, BOTTOM or BOTH, and **shall** define the side from which the groove is ground. The fifth parameter, <cutmode>, is also required and **shall** be one of the fixed fields REMOVE or REMAIN. The sixth parameter, <rem_value> supplements the REMOVE or REMAIN parameters, and provides the measure of the material removed or the measure of the material that remains. The seventh parameter, <profiledesc_ref>, is optional. When present, it **shall** reference a profile description as defined in the PRIMITIVES section of the GenCAM file. It represents the tolerances on the material stated in <rem_value>. The GROOVE statement **shall** be follow by a POLYLINE statement that references and locates the center of the groove at its xy locations on the board.

4.8.6.7 HOLE

The HOLE statement is optional, and can be used to name and define a drilled or punched feature of the board. The first parameter, <hole_name>, is required. It is a freeform string that serves to uniquely name the hole instance. Each hole defined for the board **shall** have a unique name within the board description. The second parameter, <hole_type>, is required and **shall** identify the hole's use as one of ELECTRICAL, MECHANICAL or TOOLING. The third parameter, <primitive_ref>, is required and **shall** reference a closed, standard or user defined shape, as defined in the PRIMITIVES section of the GenCAM file. The fourth parameter, <barreldesc_ref>, is required and **shall** reference a barrel description as defined in the PRIMITIVES section of the GenCAM file. The fifth parameter, <profiledesc_ref>, is optional. It can serve to define tolerances on the hole size and plating. The sixth parameter, <layerset_ref>, is optional. It can define the individual layer, <layersingle>, or group of layers, <layerset>, through which the hole is drilled, as defined in the LAYERS section of the GenCAM file. If a <layer_ref> parameter is not included then all layers are assumed.

The remaining parameters define the placement of the hole. The x-y transform is required and **shall** define the relative position of the point of origin of the hole's primitive with respect to the board origin. The rotation parameter is optional, and can define the angle between the orientation of the primitive as defined, and its position on the board, measured counterclockwise. Any mirroring **shall** be done before the shape is rotated. The mirror parameter is optional, and can be used to indicate that the hole is mirrored before becoming part of the board.

4.8.6.8 TARGET

The TARGET statement is optional, and can be used to name and define a z-axis alignment or x-y registration feature (e.g. fiducial). The first parameter, <target_ref>, is required. It is a freeform string

that serves to reference the target definition in the ARTWORKS section of the GenCAM file. The second parameter, <layer_ref>, is optional. It can define the individual layer, <layersingle>, or group of layers, <layerset>, to which the TARGET should be applied, as defined in the LAYERS section of the GenCAM file. If a <layer_ref> parameter is not included then all layers are assumed. The third parameter, <color_ref> is optional, and can be used to reference a color as defined by the COLOR statement in the HEADER section of the GenCAM file.

The remaining parameters define the <xform> of the target. The <xform> is composed of an <xy_ref> parameter that is required and **shall** define the relative position of the point of origin of the target with respect to the board origin. A <rotation> parameter is optional, and can define the angle between the position of the target as defined, and its orientation on the board, measured counterclockwise. Any mirroring **shall** be done before the target is rotated. The <mirror> parameter is optional, and can be used to indicate that the artwork is mirrored. The last parameter is an optional <scale_factor> multiplier that is applied to all dimensions of the target.

4.8.6.9 LOGO

The LOGO statement is optional, and can be used to name and define the logo to be put onto the board. The first parameter, <logo_ref>, is required. It is a freeform string that serves to reference the logo definition in the ARTWORKS section of the GenCAM file. The second parameter, <layerset_ref>, is optional. It can define the individual layer, <layersingle>, or group of layers, <layerset>, to which the LOGO should be applied, as defined in the LAYERS section of the GenCAM file. If a <layer_ref> parameter is not included then all layers are assumed. The third parameter, <color_ref> is optional, and can be used to reference a color as defined by the COLOR statement in the HEADER section of the GenCAM file.

The remaining parameters define the <xform> of the logo. The <xform> is composed of an <xy_ref> parameter that is required and shall define the relative position of the point of origin of the target with respect to the board origin. A <rotation> parameter is optional, and can define the angle between the position of the target as defined, and its orientation on the board, measured counterclockwise. Any mirroring shall be done before the target is rotated. The <mirror> parameter is optional, and can be used to indicate that the artwork is mirrored. The last parameter is an optional <scale_factor> multiplier that is applied to all dimensions of the logo.

4.8.6.10 KEEPOUT

The KEEPOUT statement is optional, and can be used to name and define an area of the board, inside which is not accessible to test probes, components, etc. The shape defined for the keepout region must be closed. The first parameter, <keepout_name>, is required. It is a freeform string that serves to uniquely name the keepout instance. The second parameter is the keepout type, <keepout_type> and is a fixed field word identifying COMPONENT, VIA, ROUTE, TESTPIN, TESTPROBE and BOARD keepout areas. The third parameter, <layer_ref>, is optional. It can define the individual layer, <layersingle>, or group of layers, <layerset>, to which the keepout area applies, as defined in the LAYERS section of the GenCAM file. If a <layer_ref> parameter is not included then all layers are assumed. A KEEPOUT statement **shall** be followed by a POLYGON statement that reference and locate the keepout shape, or by sufficient GenCAM simple primitive statements to describe and locate the closed shape(s).

4.8.6.11 BAREBOARDTEST

The BAREBOARDTEST statement is required and is used to identify the test voltage and current to be used for board continuity testing. The first parameter, <continuity_voltage>, identifies the continuity voltage in volts. The second parameter, <continuity_current>, is the continuity current in amps. The last parameter, <isolation_voltage>, is the isolation voltage in volts.

A BOARDS Section Example:

```
$BOARDS
  BOARD: "66540";
    OUTLINE: "All";
      LINE: (0, 0), (12000, 20000);
      CIRCARC: (12000, 20000), (12000, 30000), (11800, 25000);
      LINE: (12000, 30000), (10000, 30000);
      THICKNESS: 150, "connector2";
      LINE: (10000, 30000), (0, 0);
  CUTOUT: "TransformerHole", "All";
    CIRCLE: "RoundCut", (80000, 140000);
  HOLE: "Thole1", "PlatedTooling1875", , (11550, 29550);
  HOLE: "Thole2", "PlatedTooling1875", , (550, 550);
    ATTRIBUTE: "MegaTool",  "board mill tool", "255";
$ENDBOARDS
```

4.8.7 DRAWINGS

The DRAWINGS section is used to identify and describe the drawing set associated with the assembly. All drawings that are included in the GenCAM file **shall** be described in the GenCAM language. Drawings in other formats can be referenced in this section for informational purposes only.

```
DRAWINGS    ::=  $DRAWINGS
                 1{
                   USEDIN: <usedin_ref> 0{,<usedin_ref>}n;
                   1{
                     DRAWING: <drawing_name>, [<url>], [<drawing_format>];
                      {
                       DWGTYPE: <drawing_type>;
                       & DWGSIZE: <drawing_size>;
                       & DWGREV: <drawing_revision>;
                       & DWGNUMBER: <drawing_number>;
                       & DWGDATE: <drawing_date>;
                       & 1{
                         SHEET: <sheet_number>, [<sheet_rev>];
                         {
                           FRAME: <artwork_ref>, <xform>;
                           &0{
                            INSERT: <item_ref>, <xform>;
                            0{
                            FILTER: <layersingle_ref>;
                            }n
                            }n
                           &0{
                            SYMBOL: <symbol_ref>, <comp_refdes>, <circuit_ref>,
                                    [<color_ref>], <xform>;
                              {
                            1{SYMPIN:<sympin_ref>, <net_ref>;}n
                           & 0 {TEXT: <text_string>, <p1>, <p2>, [<font_ref>],
                                    [<color_ref>], <xform>;}n
                              }n
                           }n
                           & 0{
                           NET:<net_ref>;
```

```
:>                  0{
                      <polyline_builder>
                     | TEXT: <text_string>, <p1>, <p2>, [<font_ref>],
                           [<color_ref>], <xform>;
<:                    }n
                    }n
                    & 1{<shape_builder>}n
                    & 0{ARTWORK: <artwork_ref>, [<color_ref>], <xform>;}n
                    & 0{LOGO: <logo_ref>, [<color_ref>], <xform>;}n
                    & 0{TEXT: <text_string>, <p1>, <p2>, [<font_ref>],
                           [<color_ref>], <xform>;}n
                  }
<:                 }n
<:              }
              | DWGREF: <url>, <drawing_format>;
<:          }n
<:      }n
        $ENDDRAWINGS
```

Note: The DWGREF statement references an externally defined drawing. This statement is not valid when the drawing is defined in-line in the GenCAM file.

<usedin_ref>	::=	string - a $HEADER fixture(s), panel(s), board(s) or assembly(ies)
<drawing_name>	::=	string
<url>	::=	string - According to the IETF Standard RFC-1738
<drawing_format>	::=	string - MIME type according to IETF Standards RFC-2045 – RFC-2049 - i.e., "text/IPCGenCAM", or "text/html", or "application/postscript"
<drawing_type>	::=	{FABRICATION | ASSEMBLY | SCHEMATIC | DETAIL | SPECIFICATION | PARTSLIST}1
<drawing_size>	::=	{ A | B | C | D | E | A0 | A1 | A2 | A3 | A4 }1
<drawing_revision>	::=	string
<drawing_number>	::=	string
<drawing_date>	::=	date_format
<sheet_number>	::=	p_integer
<sheet_rev>	::=	string
<artwork_ref>	::=	string
<item_ref>	::=	string - a reference to the name of an image in the PRIMITIVES, ARTWORKS, or PATTERNS section of the GenCAM file.
<layersingle_ref>	::=	{string | TOP | BOTTOM | BOTH}
<symbol_ref>	::=	string
<comp_refdes>	::=	string
<circuit_ref>	::=	string
<color_ref>	::=	string - <color name> identified in $HEADER; the default is "black"
<sympin_ref>	::=	string
<net_ref>	::=	string
<text_string>	::=	string
<p1>	::=	<xy_ref> - p1 and p2 define the corners of the rectangle
<p2>	::=	<xy_ref>
<font_ref>	::=	string
<logo_ref>	::=	string
<xform>	::=	<xy_ref>, [<rotation>], [<mirror>], [<scale_factor>]

The $DRAWINGS and $ENDDRAWINGS section keywords mark the DRAWINGS section of the GenCAM file. Each DRAWING description **shall** start with the DRAWING statement keyword.

4.8.7.1 USEDIN

The USEDIN statement associates the drawing statements that follow with the <usedin_name> of a board(s), panel(s), assembly(ies) or fixture(s) as defined in the HEADER section of the GenCAM file.

4.8.7.2 DRAWING

The DRAWING statement is used to identify a single sheet or multiple sheet drawing. The first parameter, <drawing_name>, contains a unique DRAWING name. It is a free field text string. This name can be used throughout the GenCAM file for all references to this DRAWING set. The second parameter, <url>, is optional, and identifies the location of the drawing information if it is not contained in the GenCAM file. The third parameter, <drawing_format>, is optional and defines the format for the drawing information in accordance with IETF standards RFC-2045 - RFC-2049, or as GenCAM, POSTSCRIPT, HTML. The default is GenCAM.

4.8.7.3 DWGTYPE

The DWGTYPE statement, <drawing_type>, is used to define the type of the drawing as one of FABRICATION, ASSEMBLY, SCHEMATIC, DETAIL, SPECIFICATION or PARTSLIST.

4.8.7.4 DWGSIZE

The DWGSIZE statement, <drawing_size>, is used to specify the paper size of the drawing as one of A, B, C, D, E, A0, A1, A2, A3 or A4.

4.8.7.5 DWGREV

The DWGREV statement, <drawing_rev>, is used to specify the revision of the drawing.

4.8.7.6 DWGNUMBER

The DWGNUMBER statement, <drawing_number>, is used to specify any number that is associated with the drawing.

4.8.7.7 DWGDATE

The DWGDATE statement, <drawing_date>, is used to specify the date code in the standard GenCAM date format.

4.8.7.8 SHEET

The SHEET statement is used to identify the drawing sheets. The first parameter, <sheet_number>, is required and provides the sheet number of a single or a multi-sheet drawing. The second parameter of the SHEET statement, <sheet_rev>, is optional and can be used to identify separate revisions for individual sheet numbers of multiple sheet drawings.

4.8.7.9 FRAME

The FRAME statement is used to specify a graphic template for drawings. The first parameter, <artwork_ref>, references an artwork outline for the drawing border, zone identifiers, and the drawing title block. The second parameter, <xform>, identifies the relationship with the drawing border and its constituents to the drawing size. Mirroring, rotation, and scaling are not required.

4.8.7.10 INSERT

The INSERT statement is used to be able to take any symbol previously defined and re-use it in any drawing. This includes primitives, as well as artworks or patterns. The first parameter, <item_ref>, identifies the PRIMITIVE, ARTWORK, or PATTERN to be inserted. The second parameter is <xform> and is required to locate the image and the relationship to the FRAME of the drawing. All parameters of <xform> apply.

4.8.7.11 FILTER

The FILTER statement is intended to define portions of various primitives, artworks, or patterns used within the drawing. The first parameter, <layersingle_ref>, is required and identifies the sheet's relationship of the image layers through multiple sheet drawings.

4.8.7.12 SYMBOL

The SYMBOL statement is used to instantiate the schematic graphics for device and device circuit rendering, cross-referencing the component reference designator as defined in the COMPONENTS section of the GenCAM file. The first parameter, <symbol_ref>, is required and refers to a SYMBOL described in the PATTERNS section. The second parameter, <comp_refdes>, is required and defines the reference designator of the component from the COMPONENT section. The third parameter, <circuit_ref>, is the reference as to which circuit within the component the symbol represents. The fourth parameter, <color_ref>, is optional, and is a reference to the color stated in the GenCAM HEADER section. The fifth parameter, <xform>, is required and identifies the location of the symbol within the frame of the schematic drawing.

4.8.7.13 SYMPIN

The SYMPIN statement is used to describe the net path relationship to the device or device circuit graphics, by cross-referencing a symbol pin identifier to the net name, as defined in the ROUTES section of the GenCAM file. The first parameter, <sympin_ref>, is required and identifies the symbol pin alphanumeric reference. The second parameter, <net_ref>, is mandatory, and identifies the net name of the signal as identified in the ROUTES section of the GenCAM file.

4.8.7.14 NET

The NET statement is used to cross-reference signals from the ROUTES section of the GenCAM file, to graphic representation of the schematic drawing paths and logic and schematic symbols that identify component characteristics. The first parameter, <net_ref>, is required and references a <net_name> assigned in the ROUTES section. The reference shall be followed by a number of <polyline_builder> statements and <text_strings> located within the FRAME of the drawing to connect SYMPIN images that correspond to the same <net_ref>.

4.8.7.15 Other Drawing Shapes

Additional drawing shapes are added to drawings as required using various graphic, simple, complex, and user primitives. These are defined in the BNF under the description of <shape_builder> and are required on every drawing. In addition, other features may be predefined in the ARTWORKS section and added to a drawing through an <artwork_ref>, or <logo_ref>. TEXT is used throughout all drawing descriptions to complete title blocks in the FRAME, and to add appropriate notes or annotation to the drawing.

A DRAWING Section Example

```
$DRAWINGS
  USEDIN: "ESD-VOCODER", "MSD-VOCODER", "APPLICATOR";
    DRAWING: "S23";
      DWGTYPE: SPECIFICATION;
```

```
        DWGSIZE: "A";
        DWGREV: "1.2";
        DWGDATE: 1997-08-07;
        DWGREF: "http://www.ipc.org/format/Asize",
                  "text/postscript";
    USEDIN: "MSD-VOICEOVER";
      DRAWING: "d23";
        DWGTYPE: SCHEMATIC;
        DWGSIZE: "D";
        DWGREV: "1.2";
        DWGDATE: 1997-08-07;
        SHEET: 1,"A";
          FRAME: "D-Size-Format", (0,0);
          LOGO: "DEC-24", ,"Black",(2.30, 0.50);
          ARTWORK: "2InputNAND", ,"Black", (12.6, 13.8);
          ARTWORK: "2InputNAND", ,"Black", (11.2, 16.3);
          ARTWORK: "syms";
            LINE: (11.1, 16.1), (12.3, 14.7), "Line1", "Black";
            LINE: (12.3, 14.7), (12.4, 13.6), "Line1", "Black";
          ARTWORK: "Notes for test", (0.0, 0.0);
$ENDDRAWINGS
```

4.8.8 PRIMITIVES

The PRIMITIVES section is provided for the user to extend the library of reusable geometry by naming and giving initial dimension to shapes, referencing the GenCAM Standard Primitives. As with predefined primitives, the definition provides only the shape. The PRIMITIVES section also contains named definitions of line and paint characteristics. Line characteristics can then be associated with unenclosed primitives, while either line or paint characteristics can be associated with closed, polygonal features. The PRIMITIVES section has the following form.

```
PRIMITIVES      ::=  $PRIMITIVES
                :>   1{
                       USEDIN: <usedin_ref> 0{,<usedin_ref>}n;
                :>     {
                        0{
                            LINEDESC:<linedesc_name>, <line_width>, [<line_end>],
                            [<color_ref>], [<line_type>], [<line_space>], [<line_length>],
                            [<line_mod>], [<mod_end>], [<dim_A>], [<dim_B>], [<dim_C>];
                        }n
                       & 0{
                            PAINTDESC: <paintdesc_name>, <paint_type>, [<color_ref>],
                                [<line_width>], [<pitch1>], [<angle1>], [<pitch2>], [<angle2>];
                            }n
                       & 0{
                            BARRELDESC: <barreldesc_name>, <barrel_type1>, <material1>,
                                [<min_thickness1>], [<barrel_type2>], [<material2>],
                                [<min_thickness2>], [<barrel_type3>], [<material3>],
                                [<min_thickness3>], [<fill_material>];
                            }n
                       & 0{
                            PROFILEDESC: <profiledesc_name>, <finished_LMC>,
                                <finished_MMC>, [<start_LMC>], [<start_MMC>];
                            }n
                       & 0{CIRCLE: <circle_name>, <diameter>;}n
```

```
                & 0{RECTCENTER: <rectangle_name>, <x_dimension>,
                        <y_dimension>;}n
                & 0{RECTCORNER: <rectangle_name>, <p1>, <p2>;}n
                & 0{RECTCHAM: <rectcham_name>, <width>, <height>, <chamfer>;}n
                & 0{RECTROUND: <rectround_name>, <width>, <height>, <radius>;}n
                & 0{OVAL: <oval_name>, <width>, <height>;}n
                & 0{DSHAPE: <dshape_name>, <end_shape>, <orientation>, <width>,
                        <height>, {<radius> | <chamfer>}1;}n
                & 0{DIAMOND: <diamond_name>, <width>, <height>;}n
                & 0{HEXAGON: <hexagon_name>, <point_to_point>;}n
                & 0{OCTAGON: <octagon_name>, <point_to_point>;}n
                & 0{THERMAL: <thermal_name>, <therm_shape>, <outer_diameter>,
                                <inner_diameter>, [<spoke_count>], [<spoke_width>],
                                [<spoke_start_angle>], [<spoke_end_shape>];}n
                & 0{POLYLINE: <polyline_name>;
:>                  1{
                        LINE: <start_xy>, <end_xy>;
                        | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>];
                        | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                                [<direction>];
<:                    }n
                  }n
                & 0{POLYGON: <polygon_name>;
:>                  2{
                        LINE: <start_xy>, <end_xy>;
                        | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>];
                        | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                                [<direction>];
<:                    }n
                  }n
<:              }n
<: }n
        $ENDPRIMITIVES
```

Note: A polygon definition **shall** define an enclosed shape whose edges do not cross.

The $PRIMITIVES and $ENDPRIMITIVES section keywords mark the PRIMITIVES section of the GenCAM file.

<usedin_ref>	::=	string - a $HEADER fixture(s), panel(s), board(s) or assembly(ies)
<linedesc_name>	::=	string
<line_width>	::=	p_number
<line_end>	::=	{NONE | ROUND | SQUARE}1 - The default is ROUND
<color_ref>	::=	string - <color_name> identified in $HEADER; the default is BLACK and BLACK or NONE if <paint_type> is_VOID.
<line_type>	::=	{SOLID | DOTTED | DASHED | CENTER | PHANTOM | ERASE}1 - The default is SOLID
<line_space>	::=	p_number - Distance between center line of dots and/or start and end of lines,
<line_length>	::=	p_number - Distance between start and end of dashes or lines. The diameter of a dot is equal to the line width.
<line_mod>	::=	{TP | TD | SL1 | SL2 | NONE}1 - The default is NONE.
<mod_end>	::=	{START | FINISH | BOTH}1
<dim_A>	::=	p_number - Distance from end of line to finish of modification,

			or radius of secondary round feature. (see 4.1.2.4)
<dim_B>	::=	p_number	- Distance from end of line to extension of modification.
<dim_C>	::=	p_number	- Radius joining curvature of land to line center. (see Figure 4-3)
<paintdesc_name>	::=	string	
<paint_type>	::=	{HOLLOW | HATCH | MESH | FILL | VOID}1	- The default is HOLLOW. VOID designates an opaque absence of FILL within or intersecting a FILL, HATCH, or MESH enclosure.
<pitch1>	::=	p_number	- Distance between line centers of HATCH or first set of MESH lines.
<angle1>	::=	<degree_value>	- (0 - 180) - 0 - 90 are only used with MESH paint type
<degree_value>	::=	number	
<pitch2>	::=	p_number	- Distance between line center of HATCH or second set of MESH lines.
<angle2>	::=	<degree_value>	- (90 - 180) - second line set for MESH
<barreldesc_name>	::=	string	
<barrel_type1>	::=	{HOLLOW | FILL | COAT | PLATE}1	- describes HOLE or WELL condition. The default is HOLLOW – no plating nor coating in hole.
<material1>	::=	string	- defines material type for the initial wall condition
<min_thickness1>	::=	p_number	- maximum dimension controlled using MMC of PROFILEDESC
<barrel_type2>	::=	string	- describes metallic plating or coating
<material2>	::=	string	- defines material type for the subsequent wall condition
<min_thickness2>	::=	p_number	- maximum dimension controlled using MMC of PROFILEDESC
<barrel_type3>	::=	string	- describes metallic plating or coating
<material3>	::=	string	- defines material type for the final wall condition
<min_thickness3>	::=	p_number	- maximum dimension controlled using MMC of PROFILEDESC
<fill_material>	::=	string	- identification of metallic and non-metallic materials for filling the hole.
<profiledesc_name>	::=	string	
<finished_LMC>	::=	p_number	- variation from nominal to least material condition
<finished_MMC>	::=	p_number	- variation from nominal to maximum material condition.
<start_LMC>	::=	p_number	- variation from nominal to least material condition prior to the addition of plating or coating (see BARRELDESC).
<start_MMC>	::=	p_number	- variation from nominal to maximum material condition prior to the addition of plating or coating (see BARRELDESC).
<circle_name>	::=	string	- see 6.6
<diameter>	::=	p_number	– dimension values for all primitives can be any convenient value. The actual size of a feature will be determined by applying the scale value of its transform to the defined dimension.
<rectangle_name>	::=	string	- see 6.6

<p1>	::=	<xy_ref>	– The coordinates p1 and p2 define the corners of the rectangle where p1 is the lower left hand corner and p2 is the upper right hand corner.
<p2>	::=	<xy_ref>	
<x_dimension>	::=	p_number	- The <x_dimension> is the length of the two parallel horizontal planes of the rectangle.
<y_dimension>	::=	p_number	- The <y_dimension> are the two parallel sides of the rectangle where the sides are perpendicular to the horizontal planes (the instance may then be rotated to any angle).
<rectcham_name>	::=	string	- see 6.7
<width>	::=	p_number	
<height>	::=	p_number	
<chamfer>	::=	p_number	- the chamfer size is the length of one leg of the chamfer. The chamfer is 45 degrees.
<rectround_name>	::=	string	- see 6.7
<radius>	::=	p_number	
<oval_name>	::=	string	- see 6.7
<dshape_name>	::=	string	- see 6.7
<end_shape>	::=	{ROUND | CHAMFER | FILLET}1	
<orientation>	::=	{NORTH | SOUTH}1	- the default is NORTH
<diamond_name>	::=	string	- see 6.7
<hexagon_name>	::=	string	- see 6.7
<point_to_point>	::=	p_number	- The dimensions represent the distance across the points of the hexagon or octagon
<octagon_name>	::=	string	- see 6.7
<thermal_name>	::=	string	- see 6.7
<therm_shape>	::=	{ROUND | SQUARE | OCTAGON}1	
<outer_diameter>	::=	p_number	
<inner_diameter>	::=	p_number	
<spoke_width>	::=	p_number	
<spoke_start_angle>	::=	<angular_measure>	
<angular_measure>	::=	number	
<spoke_count>	::=	p_integer	- default is 0 (donut)
<spoke_end_shape>	::=	{ROUND | SQUARE | PARALLEL}1	
<polyline_name>	::=	string	- see 6.8
<start_xy>	::=	<xy_ref>	
<end_xy>	::=	<xy_ref>	
	::=	<xy_ref>	
<focus1>	::=	<xy_ref>	
<focus2>	::=	<xy_ref>	
<direction>	::=	{CLKW | CCLKW}1	- Default is CCLKW, counterclockwise
<polygon_name>	::=	string	- see 6.8

4.8.8.1 USEDIN

The USEDIN statement associates the statements that follow with the <usedin_name> of a board(s), panel(s), assembly(ies) or fixture(s) as defined in the HEADER section of the GenCAM file. At least one <usedin_ref> is required.

4.8.8.2 LINEDESC

A LINEDESC statement serves to name and define line characteristics, which can then be referenced throughout the GenCAM file, and associated with both closed and non-closed shape primitives. The first parameter, <linedesc_name>, is required. It is a freeform string that serves to uniquely name the line

description. The second parameter, <line_width>, is required and **shall** define the line width in dimensioned units. The third parameter, <line_end>, is optional and can be used to describe the line end as one of SQUARE, ROUND or NONE. The default is ROUND. The fourth parameter, <color_ref>, is optional and references the <color_name> as defined in the GenCAM HEADER section. The fifth parameter, <line_type>, is optional and **shall** define the line type as one of SOLID, DOTTED, DASHED, CENTER, PHANTOM or (solid) ERASE. The default is SOLID. The sixth and seventh parameters, <line_space> and <line_length>, **shall** be used to define the relationship of DOTTED, DASHED, CENTER, and PHANTOM lines. The eighth parameter, <line_mod>, is optional and can be used to describe the line end modification as tear-drop (TD), tapered (TP) or one of two sub-land patterns (SL1, SL2). See Figure 4-3. The default is NONE. The ninth parameter, <mod_end>, serves to identify the application of the line end modifications. The application can be to the START end, the FINISH end, or BOTH. The default is BOTH. When Line modification is instantiated, the dimensional characteristics A, B, and C are the remaining parameters.

4.8.8.3 PAINTDESC

A PAINTDESC statement serves to name and define a set of characteristics that can be applied to closed shapes. The first parameter, <paintdesc_name>, is required. It is a freeform string that serves to uniquely name the paint description. The second parameter, <paint_type>, is required and **shall** associate texture using HOLLOW, HATCH, MESH, FILL or VOID. The default is HOLLOW. The third parameter, <color_ref>, is optional. Color definition can be done with at the time that the PAINTDESC is defined, or can be deferred until the enclosed region is instantiated. A defined PAINT color can also be over-ridden at the time of instantiation. The remaining parameters apply only to paint types of HATCH and MESH. The first of these, <line_width>, defines the width of the line. The second parameter, <pitch1> defines the distance between the first set of lines. The third parameter, <angle1> defines the angle of this set of lines. The last two parameters, <pitch2> and <angle2> apply similarly to the second set of lines for MESH.

4.8.8.4 PROFILEDESC

The PROFILEDESC statement defines the manufacturing tolerance range of any geometry. The first parameter, <profiledesc_name>, is required. It is a freeform string that shall serve to uniquely identify the profile description. The second parameter, <finished_LMC>, is required and defines the permitted variation from nominal to the least material condition (LMC) of a feature or the product. The third parameter, <finished_MMC>, is required and defines the permitted variation from nominal to the maximum material condition (MMC) of a feature or the product. The fourth parameter, <start_LMC>, is optional, and defines the permitted variation from nominal to the least material condition (LMC) of a feature or the product prior to plating, coating or additional material removal. The fifth parameter, <start_MMC>, is optional, and defines the permitted variation from nominal to the maximum material condition (MMC) of a feature or the product prior to plating, coating or additional material removal.

4.8.8.5 BARRELDESC

The BARRELDESC statement serves to further describe a HOLE or a WELL. The first parameter, <barreldesc_name>, is required. It is a freeform string that shall serve to uniquely identify the barrel or the wall description. The second parameter, <barrel_type1>, is required and **shall** be one of the GenCAM reserved words HOLLOW, FILL, COAT or PLATE. The third parameter, <material>, is required except for HOLLOW; it is a freeform string that defines the material for the hole or well. The fourth parameter, <min_thickness1>, is optional and defines a dimension for the minimum required thickness of material that would be within the limits of the <finished_LMC> of PROFILEDESC. The next three parameters are an optional set that starts with <barrel_type2>, and can be applied to FILL, COAT, or PLATE. The following two parameters, <material2>, and <min_thickness2> define the material and thickness of the secondary plating, or coating. If the secondary plating or coating is applied the minimum thickness becomes the determinator of the <finished_LMC> parameter; maximum thickness is controlled by <finished_MMC>. If the desire is to FILL the hole the third parameter <min_thickness> is not

applicable. The same rules apply for the next set of three parameters starting with <barrel_type3>, which pertains to a third plating or coating. In the case of multiple platings, or coatings, the order is significant, and **shall** follow the order of manufacturer. The last parameter, <fill_material>, is optional. It is a freeform string that can be used to identify the fill material of a hole or well that has been plated or coated.

A PRIMITIVES Section Example

```
$PRIMITIVES
  USEDIN: "ESD-VOCODER", "MSD-VOCODER", "APPLICATOR";
    LINEDESC: "Line1", 0.8, SOLID;
    LINEDESC: "Line2", 0.8, DASHED, 1.6, 3.2, SQUARE;
    PROFILEDESC: "Profile1", 46, 40, 36, 32;
    BARRELDESC: "CuSnPbFill", PLATE, "Copper", 0.7,
                PLATE, "TinLead", 0.5,FILL, "EPOXY";
    POLYGON: "P34";
      CIRCARC: (0, -50), (0, 50), (0, 0);
      LINE: (0, 50), (-100, 50);
      LINE: (-100, -50), (-100, 050);
      LINE: (-100, -50), (0, -50);
    CIRCLE: "circ2", 2.0;
    CIRCLE: "circ12", 12.0;
    CIRCLE: "circ8", 8.0;
    RECTCORNER: "rect6", (1240, 3370), (4535,2355);
    RECTCENTER: "rect3", 1.6, 2.8;
$ENDPRIMITIVES
```

4.8.9 ARTWORKS

The ARTWORKS section to the GenCAM file is used to describe a library of artwork, including logos (LOGO), features (FEATURE), targets (TARGET) and user text font (UFONT). ARTWORKS are defined referencing standard or user-defined primitives or other ARTWORKS.

```
ARTWORKS    ::=  $ARTWORKS
            :>   1{
                   USEDIN: <usedin_ref> 0{,<usedin_ref>}n;
                 :>  {
                       0{
                          ARTWORK: <artwork_name>;
                       :>    1{
                                  <shape_builder>
                                | TEXT: <text_string>, <p1>, <p2>, [<font_ref>], [<color_ref>],
                                        <xform>;
                                | LOGO: <logo_ref>, [<color_ref>], <xform>;
                                | ARTWORK: <artwork_ref>, [<color_ref>], <xform>;
                             <:    }n
                          }n
                        & 0{
                          TARGET: <target_name>, <function>, {<padstack_ref> |
                              <artwork_ref>}1, [<color_ref>];
                           | TARGET: <target_name>, <function>, [<color_ref>];
                       :>      {
                                     <shape_builder>
                             <:    }
                          }n
```

```
                    & 0{
                        FEATURE: <feature_name>, [<linedesc_ref>], [<paintdesc_ref>],
                            [<color_ref>];
                :>          {
                                <shape_builder>
                <:          }
                        }n
                    & 0{
                        LOGO: <logo_name>, [<color_ref>];
                :>      1{
                            <shape_builder>
                        | TEXT: <text_string>, <p1>, <p2>, [<font_ref>], [<color_ref>],
                            <xform>;
                <:        }n
                        }n
                    & 0{
                        UFONT: <font_name>;
                :>        1{
                            GLYPH: <char_code>, <p1>, <p2>;
                :>          {
                                <shape_builder>
                <:          }
                <:        }n
                        }n
                <:    }
                <:  }n
                    $ENDARTWORKS
```

<usedin_ref>	::=	string - a $HEADER fixture(s), panel(s), board(s) or assembly(ies)
<artwork_name>	::=	string
<text_string>	::=	string
<p1>	::=	<xy_ref> - p1 and p2 define the bounding box of the text string
<p2>	::=	<xy_ref>
<font_ref>	::=	string - references a standard true type/open type or user-defined font (see UFONT). When not specified, the font default is Helvetica.
<color_ref>	::=	string - <color_name> identified in $HEADER; the default is BLACK
<xform>	::=	<xy_ref>, [<rotation>], [<mirror>], [<scale_factor>]
<logo_ref>	::=	string
<artwork_ref>	::=	string
<target_name>	::=	string
<function>	::=	{REGISTRATION | ALIGNMENT | MARKER}1
<padstack_ref>	::=	string
<logo_name>	::=	string
<feature_name>	::=	string
<linedesc_ref>	::=	string
<paintdesc_ref>	::=	string
<font_name>	::=	string
<char_code>	::=	p_integer

The $ARTWORKS and $ENDARTWORKS section keywords mark the ARTWORKS section of the GenCAM file. Each artwork description **shall** start with the ARTWORK statement keyword.

4.8.9.1 USEDIN

The USEDIN statement associates the statements that follow with the <usedin_name> of a board(s), panel(s), assembly(ies) or fixture(s) as defined in the HEADER section of the GenCAM file.

4.8.9.2 ARTWORK

An ARTWORK statement serves to name and collect text and/or shapes, which are referenced to a common origin. Each <artwork_name> defined in the ARTWORKS section **shall** be unique. If the CAD system does not define artwork names, or just uses a sequenced order, then names such as artwork1, artwork2 etc **shall** be used. The only parameter required is <artwork_name>. It is a freeform string that serves to uniquely name the artwork. Statements immediately following the ARTWORK statement are optional and include or reference shape primitives as defined by the parameter <shape_builder>; TEXT, which consists of the <text_string>, the bounding box, (<p1>, <p2>), for the text, reference to a font, <font_ref>, an optional reference to color, <color_ref>, and the transformation, <xform>, of the text box; a reference to a pre-defined logo(s), <logo_ref>, with its color, <color_ref> and transformation <xform>; and a reference to a pre-defined artwork(s), <artwork_ref>, with its color, <color_ref> and transformation <xform>.

4.8.9.3 TARGET

A TARGET statement serves to name a special use collection of graphics that are referenced to a common origin. Each TARGET defined in the ARTWORKS section **shall** be unique. If the CAD system does not define target artwork names, or just uses a sequenced order, then names such as target1, target2 etc. **shall** be used. The first parameter, <target_name>, is required. It is a freeform string that serves to uniquely name the target. The second parameter, <function>, is required and **shall** name the target's special function as one of ALIGNMENT, REGISTRATION, or MARKER.

A TARGET can be defined in one of two ways. One method adds to first two parameters, either a reference to an artwork, <artwork_ref>, or a reference to a padstack, <padstack_ref>. The other method includes or references shape primitives as defined by the parameter <shape_builder>. In the first method, the third parameter is required. It is a freeform string that **shall** reference an artwork or a padstack as defined in their respective sections of the GenCAM file. For both methods, the last parameter is optional, and can be used to reference a color as defined in the HEADER section of the GenCAM file.

4.8.9.4 FEATURE

A FEATURE statement serves to name a special use collection of graphics that are referenced to a common origin. Each FEATURE defined in the ARTWORKS section **shall** be unique. The first parameter, <feature_name>, is required. It is a freeform string that serves to uniquely name the feature. The second parameter, <linedesc_ref>, is optional and references a pre defined line enhancement. The third parameter, <paintdesc_ref>, is optional and defines the filling texture of enclosed shapes. The fourth parameter <color_ref> is optional and references a color defined in the HEADER section. Statements immediately following the FEATURE statements **shall** define or reference shape primitives identified in <shape_builder>. This collection of shapes **shall** serve to define the feature.

4.8.9.5 LOGO

A LOGO statement serves to name and give special meaning to a collection of text and/or shapes that are referenced to a common origin. A LOGO is a special designation of an ARTWORK. The first parameter, <logo_name>, is required. It is a freeform string that serves to uniquely name the logo. The second parameter, <color_ref>, is an optional reference to a color defined in the HEADER section of the GenCAM file.

Statements immediately following the LOGO statements **shall** define or reference shape primitives identified in <shape_builder>. This collection of shapes **shall** serve to define the logo. An additional

optional statement, TEXT, may also be part of the logo description. The statement TEXT consists of the <text_string>, the bounding box for the text, (<p1>, <p2>), reference to a font, <font_ref>, an optional reference to color, <color_ref>, and the transformation, <xform>, of the text box;

4.8.9.6 UFONT

A UFONT statement serves to name and give special meaning to a collection of shapes, which are referenced to a common origin. A UFONT is a special designation of an ARTWORK made up of individual characters that are referenced in the description of a text string (TEXT). The first parameter, <font _name>, is required. It is a freeform string that serves to uniquely name the user font. The UFONT statement is immediately followed by a group of individually identified characters under the keyword statement GLYPH. Each GLYPH character **shall** have a unique identifier.

4.8.9.7 GLYPH

A GLYPH statement serves to name and give special meaning to a collection of individual characters that a user wishes to identify as his own unique font. The first parameter, <char_code>, is required. It is a positive integer intended to identify the character. The second parameter, <p1>, is required and is the xy coordinate of the lower left hand corner of the character box. The third parameter, <p2>, is also required and provides the xy coordinate of the upper right hand corner of the character box. Statements immediately following the GLYPH statements **shall** define or reference shape primitives identified in <shape_builder>. This collection of shapes **shall** serve to define the character. All shapes **shall** be so located that their <linedesc> enhancements are contained within the character bounding box, (<p1>, and <p2>).

4.8.9.8 TEXT

The optional TEXT statement can be used to define any text string, size and location that is attached to an ARTWORK or LOGO. The first parameter, <text_string>, is required and defines the text string. The second parameter, <p1>, is required and defines the lower left-hand xy coordinate of the box that **shall** contain the text. The third parameter is required and defines the upper right hand xy coordinate of the box that **shall** contain the text. All portions of the text, including the line width, must completely fit within the text box defined by <p1>, and <p2> parameters. The fourth parameter <font_ref> is optional. It is a free form string that identifies the font as any acceptable font known in the printing industry. The default is Helvetica. If the <font_ref> is a UFONT <font_name>, then the <text string> shall identify the text in terms of the individual <char_code> used to create the GLYPH characters. The fifth parameter, <color_ref>, is optional and is a reference to a color identified in the HEADER section. The last parameter, <xform>, is a set of parameters locating and positioning the text box.

The transform set includes, the xy coordinate, <xy_ref>, to locate the point of origin, <p1>, of the text box. The second parameter, <rotation>, is optional and defines the rotation of the text about the <p1> coordinate, measured counterclockwise. Any mirroring of the box, and therefore the text, must occur before the rotation is applied. The third parameter, <mirror>, is optional and defines the mirroring of the text. The fifth parameter, <scale>, of the transform set is optional and **shall** be a scale multiplier that is applied to the x and y dimensions of the text box.

AN ARTWORKS Section Example

```
$ARTWORKS
  USEDIN: "board 11352", "panel-66540", "subpanel 66550";
    ARTWORK: "PIN1_MARKER";
      ARTWORK: "PinDot", "White", (0, 6600);
      POLYGON: "Land13", "Paint1", "White", (0, 0);
    TARGET: "BadBoard", MARKER, "PS3";
      RECTCENTER: 0.30, 0.30, "Psolid", , (0, 0);
     LOGO: "Olympia";
```

```
      CIRCLE: 4.0, SOLID, 0.8, "Blue", (2.0, 3.0);
      CIRCLE: 4.0, SOLID, 0.8, "Green", (4.0, 3.0);
      CIRCLE: 4.0, SOLID, 0.8, "Yellow", (6.0, 3.0);
      CIRCLE: 4.0, SOLID, 0.8, "Red", (3.0, 1.0);
      CIRCLE: 4.0, SOLID, 0.8, "Black", (5.0, 1.0);
    ARTWORK: "Circle1";
      CIRCLE: 1.5, "Fill3", "Black";
    ARTWORK: "Land12";
      OVAL: 2.2, 1.5, "Fill3", "Black";
      LOGO: "Logo", "Paint1", "Blue", (0, 0), 0, MIRROR, 0.8;
$ENDARTWORKS
```

4.8.10 PATTERNS

The PATTERNS section is provided for the user to build a library of reusable pad and pad stack groupings by naming and giving layout data, referencing the GenCAM PADSTACKS definitions. The PATTERNS section has the following form.

```
PATTERNS    ::=  $PATTERNS
            :>   1{
                   USEDIN: <usedin_ref> 0{,<usedin_ref>}n;
            :>     {
                     0{
                       PATTERN: <pattern_name>, [<color_ref>];
            :>           {
                           1{PADSTACK: <padstack_ref>, <pin_physical_ref>, <xform>;}n
                            & 0{ARTWORK: <artwork_ref>, [<layer_ref>], [<color_ref>],
                                    <xform>;}n
                            & 0{TARGET: <target_ref>, [<layer_ref>], [<color_ref>],
                                    <xform>;}n
                            & 0{FEATURE: <feature_ref>, [<layer_ref>], [<color_ref>],
                                    <xform>;}n
            <:             }
                         }n
                       & 0{ PATTERN: <pattern_name>, [<color_ref>];
            :>             {
                                    <shape_builder>
            <:             }
                         }n
                        &0{ SYMBOL: <symbol_name>, [<color_ref>];
            :>           {
                           <shape_builder>
                         & 0 {ARTWORK: <artwork_ref>, [<color_ref>], <xform>;}n
                         & 0 {TEXT: <text_string>, <p1>, <p2>, [<font_ref>], [<color_ref>],
                                    <xform>;}n
                         & 0{ SYMPIN: <sympin_name>, <location>;}n
            <:           }
                       }n
            <:     }
            <:    }n
                $ENDPATTERNS
```

<usedin_ref>	::=	string - a $HEADER fixture(s), panel(s), board(s) or assembly(ies)
<xform>	::=	<xy_ref>, [<rotation>],[<mirror>],[<scale_factor>]
<pattern_name>	::=	string
<padstack_ref>	::=	string
<pin_physical_ref>	::=	string - Reference to pin name in PACKAGE section
<artwork_ref>	::=	string
<layer_ref>	::=	{<layersingle_ref> | <layerset_ref>}1 - The default is ALL
<layersingle_ref>	::=	string
<layerset_ref>	::=	string
<target_ref>	::=	string
<feature_ref>	::=	string
<color_ref>	::=	string - <color_name> identified in $HEADER; the default is "black"
<symbol_name>	::=	string
<text_string>	::=	string
<p1>	::=	<xy_ref> - p1 and p2 define the bounding box of the text string
<p2>	::=	<xy_ref>
<font_ref>	::=	string - references a standard true type/open type or user-defined font (see UFONT). When not specified, the font default is Helvetica.
<sympin_name>	::=	string
<location>	::=	<xy_ref>, [<rotation>],[<mirror>]

The $PATTERNS and $ENDPATTERNS section keywords mark the PATTERNS section of the GenCAM file. Each PATTERN description **shall** start with the PATTERN keyword; each SYMBOL description shall start with the SYMBOL keyword.

4.8.10.1 USEDIN

The USEDIN statement associates the statements that follow with the <usedin_name> of a board(s), panel(s), assembly(ies) or fixture(s) as defined in the HEADER section of the GenCAM file.

4.8.10.2 PATTERN

The PATTERN statement identifies a single pattern definition. The first parameter, <pattern_name>, is required. It is a freeform string that serves to uniquely name the pattern instance. The name is usually referenced in the COMPONENTS or DRAWINGS section of the GenCAM file. When a pattern is associated with a component, each PATTERN is defined by a grouping of PADSTACK references. Each PADSTACK is referenced to a pin number, <pin_physical_ref>, of a package. The ROUTES section of the GenCAM file provides the correlation of the interconnections for each PADSTACK in the PATTERN. When associated with a drawing, the pattern is similar to an ARTWORK and the relationship with a physical pin is not required. The second parameter, <color_ref>, is optional and provides a reference to a color statement in the HEADER section.

The PATTERN statements used for DRAWING insertion are immediately followed with statements that **shall** define or reference shape primitives identified in <shape_builder>. This collection of shapes **shall** serve to define the pattern. The PATTERN statements used for COMPONENT reference require the use of PADSTACK instances thus the PATTERN statements are followed by one or more PADSTACK statements, and as an option a number of ARTWORK, TARGET, or FEATURE statements.

4.8.10.3 PADSTACK

The PADSTACK statement references a padstack definition. The first parameter, <padstack_ref>, is required. It is a freeform string, and **shall** reference an entry in the PADSTACKS section of the GenCAM file. The second parameter, <pin_physical_ref>, is required. It is a freeform string that **shall** provide a cross-reference to the package PIN definitions. The third parameter, <xform>, is a

transformation set and is required. The transform set reestablishes the third parameter, <xy_ref>, as the relative position of the point of origin of the padstack with respect to that of the pattern.

Note that the PATTERN defined using the PADSTACK instances is associated with a PACKAGE definition in a DEVICE instance definition. Since the padstack locations defined here are coupled to the package shape that is defined in the PACKAGE section, the point of origin for these definitions **shall** coincide. PATTERN and DEVICE association is coupled in the COMPONENT section.

The fourth parameter, <rotation>, is optional and **shall** be the padstack rotation. The rotation is the angle between the padstack position as defined, and orientation of the padstack relative to the pattern origin measured counterclockwise. Any mirroring **shall** be done before the pad stack is rotated. The fifth parameter, <mirror>, **shall** be the pad stack mirror definition. The mirror parameter is optional, and can be used to indicate that the shape can is mirrored before becoming part of the pattern. The sixth parameter, <scale_factor>, is optional and changes all the <dim_value>, dimensions of the primitive references used for the PADS and the HOLES. The use of <scale_factor> should be avoided in the definition of padstacks for PATTERNS.

4.8.10.4 ARTWORK

The ARTWORK statement is optional and adds an artwork graphic to the pattern by referencing an artwork definition in the ARTWORKS section of the GenCAM file. The first parameter, <artwork_ref>, is required. It is a freeform string that serves to reference the artwork. The second parameter, <layer_ref>, is required. It **shall** define the layer to which the artwork is assigned, by referencing a single layer or layer set name as defined in the LAYERS section of the GenCAM file. The third parameter, <color_ref>, is optional. It is used to provide a reference to a color as defined in the HEADER section of the GenCAM file. The remaining parameters define the <xform> of the artwork. The <xform> is composed of an <xy_ref> parameter that is required and **shall** define the relative position of the point of origin of the artwork with respect to the pattern origin. A <rotation> parameter is optional, and can define the angle between the artwork as defined, and its orientation relative to the pattern origin, measured counterclockwise. Any mirroring **shall** be done before the artwork is rotated. The <mirror> parameter is optional, and can be used to indicate that the artwork is mirrored. The <scale_factor>, is optional and changes all the dimensions of the primitive references used for creating the artwork. Use of <scale_factor> should be avoided in the definition of artwork for PATTERNS.

4.8.10.5 TARGET

The TARGET statement is optional, and can be used to name and define a z-axis alignment or x-y registration feature (e.g. fiducial), or a bad-board indicator. The first parameter, <target_ref>, is required. It is a freeform string that serves to reference the target definition in the ARTWORKS section of the GenCAM file. The second parameter, <layer_ref>, is optional and can reference a single layer or layer set as identified in the LAYERS section. The <layer_ref> parameter should be used if the TARGET definition references a two-dimensional artwork, but is not a valid parameter if the TARGET definition references a pad stack. The next parameter, <color_ref> is optional, and can be used to reference a color as defined in the HEADER section of the GenCAM file.

The remaining parameters define the <xform> of the target. The <xform> is composed of an <xy_ref> parameter that is required and **shall** define the relative position of the point of origin of the target with respect to the pattern origin. A <rotation> parameter is optional, and can define the angle between the target as defined, and its orientation relative to the pattern origin, measured counterclockwise. Any mirroring **shall** be done before the target is rotated. The <mirror> parameter is optional, and can be used to indicate that the target is mirrored. The last parameter is an optional <scale_factor> multiplier that is applied to all dimensions of the target.

4.8.10.6 FEATURE

The FEATURE statement is optional and can add graphic symbology to the pattern by referencing a feature definition in the ARTWORKS section of the GenCAM file. The first parameter, <feature_ref>, is required. It is a freeform string that serves to reference the feature. The second parameter, <layer_ref>, is optional and can reference a single layer or layer set as identified in the LAYERS section. The third parameter, <color_ref>, is optional, and can be used to reference a color as defined in the HEADER section of the GenCAM file. The remaining parameters define the <xform> of the feature. The <xform> is composed of an <xy_ref> parameter that is required and **shall** define the relative position of the point of origin of the feature with respect to the pattern origin. A <rotation> parameter is optional, and can define the angle between the feature as defined, and its orientation relative to the pattern origin, measured counterclockwise. Any mirroring **shall** be done before the feature is rotated. The <mirror> parameter is optional, and can be used to indicate that the feature is mirrored. The last parameter is an optional <scale_factor> multiplier that is applied to all dimensions of the feature.

4.8.10.7 SYMBOL

A SYMBOL statement is optional and serves to name a special use collection of graphics that are referenced to a common origin. Each symbol defined in the PATTERNS section **shall** be unique. If the CAD system does not define the symbol names, or just uses a sequence order, then names such as "symbol1", "symbol2", etc. **shall** be used. SYMBOLS are patterns that define graphics. When a relationship to a PIN is established symbols are useful to add intelligence to Schematic or logic DRAWINGS. The first parameter, <symbol_name>, is required. It is a freeform string that serves to uniquely name the symbol. The second parameter, <color_ref>, is optional, and can be used to reference a color as defined in the HEADER section of the GenCAM file. The next set of statements are mandatory and consist of primitives defined under the description of <shape_builder>. ARTWORK statements are optional and reference predefined artwork. TEXT statements are optional and are used to provide appropriate text descriptions that become part of the symbol. The keyword statements SYMPIN are optional and provide a pin name and location on the symbol for each input or output pin. The additional statements and their parameters serve to make up the total definition of SYMBOL.

4.8.10.8 SYMPIN

The SYMPIN statement is optional and identifies the location of pin references to the symbol. The first parameter, <sympin_name>, is mandatory, and identifies the alphanumeric identifier of the pin. The relationship is established between the name assigned to the graphic symbol pin identifier and the net name assigned in the ROUTES section for each instance that a pin of a component is connected. These relationships are derived between the location of a component pin as defined in its PACKAGE description, its correlation to the part in the DEVICE section, and its instantiation when located on the assembly in the COMPONENTS section. At that point the <net_name> of the PATH or PLANE can be assigned to the symbol to provide intelligence to the drawing. The second parameter of SYMPIN, <location>, is also mandatory and identifies the xy location in relationship to the symbol artwork.

4.8.10.9 TEXT

The optional TEXT statement can be used to define any text string, size and location that is attached to a SYMBOL. The first parameter, <text_string>, is required and defines the text string. The second parameter, <p1>, is required and defines the lower left-hand xy coordinate of the box that **shall** contain the text. The third parameter is required and defines the upper right hand xy coordinate of the box that **shall** contain the text. All portions of the text, including the line width, must completely fit within the text box defined by <p1>, and <p2> parameters. The fourth parameter <font_ref> is optional. It is a free form string that identifies the font as any acceptable font known in the printing industry. The default is Helvetica. If the <font_ref> is a UFONT <font_name>, then the <text string> shall identify the text in terms of the individual <char_code> used to create the GLYPH characters. The fifth parameter, <color_ref>, is optional and is a reference to a color identified in the HEADER section. The last parameter, <xform>, is a set of parameters locating and positioning the text box.

The transform set includes, the xy coordinate, <xy_ref>, to locate the point of origin, <p1>, of the text box. The second parameter, <rotation>, is optional and defines the rotation of the text about the <p1> coordinate, measured counterclockwise. Any mirroring of the box, and therefore the text, must occur before the rotation is applied. The third parameter, <mirror>, is optional and defines the mirroring of the text. The fifth parameter, <scale>, of the transform set is optional and **shall** be a scale multiplier that is applied to the x and y dimensions of the text box.

A PATTERNS Section Example

```
$PATTERNS
  USEDIN: "board-44356","assembly-33450";
    PATTERN: "PAT34";
      PADSTACK: "stack2", "1", (0, 0);
      PADSTACK: "stack12", "3", (200, 0);
      PADSTACK: "stack6", "2", (0, 100);
      PADSTACK: "stack2", "4", (200, 100);
    ARTWORK: "Pin1", , (-100, -50);
    FEATURE: "Olympia","LayerTop", , (-50, +50);
$ENDPATTERNS
```

4.8.11 MECHANICALS

The MECHANICALS section is optional and can be used to describe non-electrical features of a fixture, board, panel, or assembly; e.g. fixing holes and mechanical components attached to the board. The MECHANICALS section has the following form.

```
MECHANICALS  ::=  $MECHANICALS
             :>   1{
                  USEDIN: <usedin_ref> 0{,<usedin_ref>}n;
             :>     1{
                      MECHANICAL:<part_name>, [<mfg_part_id>], [<part_revision>],
                                 [<assoc_comp_ref>];
             :>       {
                        0{HEIGHT: <dim_value>;}1
                      & 0{STANDOFF: <dim_value>;}1
                      & 0{ARTWORK:<artwork_ref>, [<color_ref>], <xform>;}n
                      & 0{HOLE: <hole_name>, <hole_type>, <primitive_ref>,
                           <barreldesc_ref>, [<profiledesc_ref>],[<layer_ref>], <xform>;}n
                      & 0{ALIAS: <mfg_part_id>, [<part_revision>];}n
             <:       }
             <:     }n
             <:   }n
             $ENDMECHANICALS
```

```
<usedin_ref>        ::=  string  - a $HEADER fixture(s), panel(s), board(s) or assembly(ies)
<part_name>         ::=  string
<mfg_part_id>       ::=  string
<part_revision>     ::=  string
<assoc_comp_ref>    ::=  string  - line item of an associated electrical part in the DEVICE section
<layer_ref>         ::=  {<layersingle_ref> | <layerset_ref>}1
<layersingle_ref>   ::=  string
<layerset_ref>      ::=  string
<dim_value>         ::=  p_number
```

<artwork_ref>	::=	string
<color_ref>	::=	string - <color_name> identified in $HEADER; the default is "black"
<xform>	::=	<xy_ref>, [<rotation>],[<mirror>],[<scale_factor>]
<hole_name>	::=	string
<hole_type>	::=	{MECHANICAL | ELECTRICAL | TOOLING}1
<primitive_ref>	::=	{<circle_ref>| <rectctr_ref> | <rectcnr_ref> | <rectcham_ref> | <rectround_ref> |<oval_ref> | <dshape_ref> |<diamond_ref> | <hexagon_ref> | <octagon_ref>}1
<circle_ref>	::=	string
<rectctr_ref>	::=	string
<rectcnr_ref>	::=	string
<rectcham_ref>	::=	string
<rectround_ref>	::=	string
<oval_ref>	::=	string
<dshape_ref>	::=	string
<diamond_ref>	::=	string
<hexagon_ref>	::=	string
<octagon_ref>	::=	string
<barreldesc_ref>	::=	string
<profiledesc_ref>	::=	string

The $MECHANICALS and $ENDMECHANICALS section keywords mark the MECHANICALS section of the GenCAM file. Any of the following statements can appear.

4.8.11.1 USEDIN

The USEDIN statement associates the statements that follow with the <usedin_name> of a fixture(s), board(s), panel(s), or assembly(ies) as defined in the HEADER section of the GenCAM file.

4.8.11.2 MECHANICAL

The MECHANICAL statement is required and is used to define mechanical components that have not been defined elsewhere in the GenCAM file (e.g., items that are usually not assigned an electrical function in the COMPONENTS section and are not pads, vias or test pins). The most common of these mechanical components are mounting hardware for power transistors or card extractor mechanisms. The first parameter, <part_name>, **shall** be the mechanical component name. This freeform string can contain a manufacturer's part name, a library part name, a stock name, or anything else that uniquely defines the mechanical component. The second parameter, <mfg_part_id>, is optional. It is a freeform string, and can be included to specify a part number as it might appear on a purchase order. The third parameter, <part_revision>, is optional and defines a revision level of the manufacturer's part, if appropriate. The fourth parameter, <assoc_comp_ref>, is optional. It is a freeform string that is included if the mechanical part has a relationship to an electrical part as might be appropriate for an IC heat sink. The string defines the electrical <part_name> assigned to the device in the DEVICES section.

4.8.11.3 HEIGHT

The HEIGHT statement is required and is used to identify the mechanical component's maximum height from the finished mounting surface. The only parameter, <dim_value>, is required. The <dim_value> is given as a positive number in the UNITS <dimension> parameter defined in the HEADER of the GenCAM file.

4.8.11.4 STANDOFF

The optional STANDOFF statement is used to identify the mechanical component's clearance from the finished mounting surface. The only parameter, <dim_value>, is required. The <dim_value> is given as a positive number in the UNITS <dimension> parameter defined in the HEADER of the GenCAM file.

4.8.11.5 ARTWORK

The ARTWORK statement is optional and adds an artwork graphic to the mechanical part by referencing an artwork definition in the ARTWORKS section of the GenCAM file. The first parameter, <artwork_ref>, is required. It is a freeform string that serves to reference the artwork. The second parameter, <color_ref>, is optional. It is used to provide a reference to a color as defined in the HEADER section of the GenCAM file. The remaining parameters define the <xform> of the artwork. The <xform> is composed of an <xy_ref> parameter that is required and **shall** define the relative position of the point of origin of the artwork with respect to other features of the mechanical part. A <rotation> parameter is optional, and can define the angle between the artwork as defined, and its orientation relative to the origin, measured counterclockwise. Any mirroring **shall** be done before the artwork is rotated. The <mirror> parameter is optional, and can be used to indicate that the artwork is mirrored. The <scale_factor>, is optional and changes all the dimensions of the primitive references used for creating the artwork.

4.8.11.6 HOLE

The HOLE statement is optional, and can be used to name and define a drilled, milled or punched feature of the mechanical part. The first parameter, <hole_name>, is required. It is a freeform string that serves to uniquely name the hole instance. Each hole defined for the mechanical part **shall** have a unique name within the part description. The second parameter, <hole_type>, is required and **shall** identify the hole use as one of ELECTRICAL, MECHANICAL or TOOLING. The third parameter, <primitive_ref>, is required and **shall** reference a closed, standard or user defined shape, as defined in the PRIMITIVES section of the GenCAM file. The fourth parameter, <barreldesc_ref>, is required and **shall** reference a barrel description as defined in the PRIMITIVES section of the GenCAM file. The fifth parameter, <profiledesc_ref>, is optional. It can serve to define tolerances on the hole size and plating. The sixth parameter, <layer_ref>, is optional. It can serve to reference a single layer or a layer set through which the hole is drilled, as defined in the LAYERS section of the GenCAM file. The <GenCAM_layer_type> is usually defined as COMPONENT. If a <layer_ref> parameter is not included then all layers are assumed. The last parameter, <xform>, defines the xy location of the hole. The x-y transform is required and **shall** define the relative position of the point of origin of the hole primitive with respect to the ARTWORK origin. The rotation parameter is optional, and can define the angle between the position of the primitive as defined, and its orientation on the part, measured counterclockwise. Any mirroring **shall** be done before the shape is rotated. The mirror parameter is optional, and can be used to indicate that the shape is mirrored.

4.8.11.7 ALIAS

The ALIAS statement is optional. It is used to establish an alternate to the original PART description. The first parameter, <mfg_part_id> is required. It is the alternate manufacturer's part identification number. The second parameter, <part_revision>, is optional and defines a revision level of the manufacturer's part, if appropriate.

A MECHANICALS Section Example

```
$MECHANICAL
  USEDIN: "34435";
    MECHANICAL: "Abracket", "B47-34";
      STANDOFF: 0.25;
      HEIGHT: 0.60;
      HOLE: "Mhole1", MECHANICAL, "Circle12", "BarrelHollow", ,
            "All",(1000, 1000);
      HOLE: "Mhole2", MECHANICAL, "Circle12", "BarrelHollow", ,
            "All",(1000, 3125);
      ARTWORK: "Bracket2", "Top", , (0, 0);
```

```
        MECHANICAL: "Terminal3", "T7.635","U23";
          ARTWORK: "Term3", "Top", ,(0, 0);
          ALIAS: "T14.635", "RevB";
    $ENDMECHANICALS
```

4.8.12 LAYERS

The LAYERS section of the GenCAM file is used to describe the CAD layers that make up the printed circuit board and to group the CAD layers into named layer sets. The LAYERS section has the following form.

```
LAYERS          ::=  $LAYERS
                :>   1{
                       USEDIN:<usedin_ref> 0{,<usedin_ref>}n;
                :>      1{
                          LAYERSINGLE:<layer_name>, <GenCAM_layer_type>, <material>,
                                      [<thickness>], [<profiledesc_ref>];
                <:        }n
                        & 1{
                          LAYERSET: <layerset_name>, [<material_code>], [<thickness>],
                                      [<profiledesc_ref>], [<impedance>];
                :>          1{LAYER: <layersingle_ref>;
                <:          }n
                        }n
                <:      }n
                   $ENDLAYERS

<usedin_ref>          ::=   string  - a $HEADER fixture(s), panel(s), board(s) or assembly(ies)
<layer_name>          ::=   {string | TOP | BOTTOM | BOTH}1
<GenCAM_layer_type>::=      {LEGEND
                            | GLUE
                            | SOLDERMASK
                            | COATINGCOND
                            | COATINGNONCOND
                            | CONDUCTOR
                            | DIELBASE
                            | DIELCORE
                            | DIELPREG
                            | DIELADHV
                            | SOLDERBUMP
                            | PASTEMASK
                            | HOLEFILL
                            | PIN
                            | COMPONENT
                            | RESISTIVE
                            | CAPACITIVE
                            | PROBE
                            | REWORK
                            | FIXTURE
                            }1

<material>            ::=   string
<thickness>           ::=   p_number
<profiledesc_ref>     ::=   string          - for LMC and MMC
<layerset_name>       ::=   string
<material_code>       ::=   string          - per IPC- 4101 designation
```

<layersingle_ref> ::= string
<impedance> ::= p_number - Units **shall** be expressed in ohms.

The $LAYERS and $ENDLAYERS section keywords mark the LAYERS section of the GenCAM file.

4.8.12.1 USEDIN

The USEDIN statement associates the statements that follow with the <usedin_name> of a board(s), panel(s), assembly(ies) or fixture(s) as defined in the HEADER section of the GenCAM file.

4.8.12.2 LAYERSINGLE

The LAYERSINGLE statement is modal, having two uses within the LAYERS section. First, prior to any LAYERSET statement, it is used to define a layer's physical characteristics, and to associate the CAD file layer name with GenCAM layer types. The LAYERSINGLE keyword statement is also used within a LAYERSET grouping to reference CAD layers that are defined elsewhere. When defining layers, the order of the layer references is significant, and must sequentially reflect the layer stack, from the primary (top) component side of the board, inward. The first parameter, <layer_name>, is required and is a freeform string that identifies the layer. This is typically the name assigned to the layer within the CAD system. Each layer **shall** have a unique name. The second parameter, <GenCAM_layer_type>, is required and **shall** be one of the GenCAM layer-type reserved keywords. The third parameter, <material>, is required and is a freeform string used to describe the material from which the layer is constructed. The fourth parameter, <thickness>, is an optional parameter used to define the thickness of the material using the nominal dimension in accordance with the UNITS shown in the HEADER of the GenCAM file. The fifth parameter, <profiledesc_ref>, is an optional parameter used to specify the acceptable tolerance (<finished_LMC> and <finished_MMC>) associated with the material thickness. If no thickness is provided then this parameter **shall** be omitted.

4.8.12.3 LAYERSET

The LAYERSET statement is used to define sets of layers, which can then be referenced for sequential lamination, cutouts, padstacks, vias, etc. Within a LAYERSET statement, the LAYER statement is used to reference CAD layers that make up the set, which are defined elsewhere. In this context, the LAYER statement can be the name of a CAD layer or can be a reference to a LAYERSINGLE or other LAYERSET identified in the GenCAM file. The order of the layer references is significant, and must sequentially reflect the layer stack, from the primary component side of the board, inward. The first parameter, <layerset_name>, is required and is a freeform string that identifies the layer set. The second parameter, <material_code>, is optional and is used to describe a layer in accordance with the IPC layer specification (IPC-4101) designation. The third parameter, <thickness>, is an optional parameter used to define the thickness of the material using the nominal dimension in accordance with the UNITS shown in the HEADER of the GenCAM file. The fourth parameter, <profiledesc_ref>, is an optional parameter used to specify the acceptable tolerance (<finished_LMC> and <finished_MMC>) associated with the material thickness. If no thickness is provided then this parameter **shall** be omitted. The fifth parameter, <impedance>, is optional, and can be used to provide a measurable impedance for the layer set.

A LAYERS Section Example

```
$LAYERS
  USEDIN: "BD4566-25", "PNL4566-29";
    LAYERSINGLE: "SilkscreenTop", LEGEND, "epoxy ink";
    LAYERSINGLE: "MaskTop", SOLDERMASK, "LPI", 1, "SMOBC1";
    LAYERSINGLE: "Coat1", COATINGCOND,"Tin Lead", 0.3, "SOLD1";
    LAYERSINGLE: "L1", CONDUCTOR, "copper", 0.7, "CU1";
    LAYERSINGLE: "Dielectric1", DIELCORE, "FR4", 6.2, "Diel6";
    LAYERSINGLE: "L2", CONDUCTOR, "copper", 1.4, "CU2";
    LAYERSINGLE: "Dielectric2", DIELCORE, "FR4", 6.2, "Diel7";
```

```
        LAYERSINGLE: "L3", CONDUCTOR, "copper", 1.4, "CU2";
        LAYERSINGLE: "Dielectric3", DIELCORE, "FR4", 6.2, "Diel6";
        LAYERSINGLE: "L4", CONDUCTOR, "copper", 0.7, "CU1";
        LAYERSINGLE: "MaskBottom", SOLDERMASK, "LPI", 1, "SMOBC1";
        LAYERSET: "ls1","L21 1500 C1/C1 A1A", 9.0, "Core12", 50;
          LAYER: "L2";
          LAYER: "Dielectric2";
          LAYER: "L3";
      USEDIN: "FX4566-20";
        LAYERSINGLE: "Stencil", FIXTURE, "Stainless Steel", 8,
                     "STN1";
    $ENDLAYERS
```

4.8.13 PADSTACKS

The PADSTACKS section is used to describe how a group of pads are arranged. The PADSTACKS section is required even if it contains only a single pad definition. The PADSTACKS section has the following form.

```
PADSTACKS      ::=  $PADSTACKS
               :>    1{
                      USEDIN: <usedin_ref> 0{,<usedin_ref>}n;
               :>     1{
                        PADSTACK:<padstack_name>;
               :>         {
                          0{
                            PAD: <layer_ref>, <pad_primitive_ref>, [<linedesc_ref>],
                              [<paintdesc_ref>], [<profiledesc_ref>], [<color_ref>], <xform>;
                           }n
                          & 0{
                              HOLE: <hole_name>, <hole_type>, <primitive_ref>,
                                    <barreldesc_ref>, [<profiledesc_ref>], [<layer_ref>],
                                    <xform>;
                            }n
               <:           }
               <:         }n
               <:       }n
                    $ENDPADSTACKS
```

```
<usedin_ref>          ::=   string  - a $HEADER fixture(s), panel(s), board(s) or assembly(ies)
<padstack_name>       ::=   string
<layer_ref>           ::=   {<layersingle_ref> | <layerset_ref>}1
<layersingle_ref>     ::=   string
<layerset_ref>        ::=   string
<linedesc_ref>        ::=   string
<paintdesc_ref>       ::=   string
<profiledesc_ref>     ::=   string
<color_ref>           ::=   string  - <color_name> identified in $HEADER; the default is "black".
<hole_name>           ::=   string
<hole_type>           ::=   {MECHANICAL | ELECTRICAL | TOOLING}1
<barreldesc_ref>      ::=   string
<primitive_ref>       ::=   {<circle_ref> | <rectctr_ref> | <rectcnr_ref> |
                             <rectcham_ref> | <rectround_ref> |<oval_ref> | <dshape_ref> |
                             <diamond_ref> | <hexagon_ref> | <octagon_ref>}1
```

<circle_ref>	::=	string
<rectctr_ref>	::=	string
<rectcnr_ref>	::=	string
<rectcham_ref>	::=	string
<rectround_ref>	::=	string
<oval_ref>	::=	string
<dshape_ref>	::=	string
<diamond_ref>	::=	string
<hexagon_ref>	::=	string
<octagon_ref>	::=	string
<xform>	::=	<xy_ref>, [<rotation>],[<mirror>],[<scale_factor>]
<pad_primitive_ref>	::=	{<circle_ref> | <rectctr_ref> | <rectcnr_ref> | <rectcham_ref> | <rectround_ref> |<oval_ref> | <dshape_ref> | <diamond_ref> | <hexagon_ref> | <octagon_ref> | <thermal_ref> | <polygon_ref>}1
<thermal_ref>	::=	string
<polygon_ref>	::=	string

The $PADSTACKS and $ENDPADSTACKS section keywords mark the PADSTACKS section of the GenCAM file. Each pad stack description **shall** start with the PADSTACK keyword. If the CAD system only defines the pad type by the layer type (plane, signal, power etc.) then a conversion to named layers has to be done before the GenCAM file is produced. Each hole in the PADSTACK section **shall** start with the HOLE keyword.

4.8.13.1 USEDIN

The USEDIN statement associates the statements that follow with the <usedin_name> of a board(s), panel(s), assembly(ies) or fixture(s) as defined in the HEADER section of the GenCAM file.

4.8.13.2 PADSTACK

The PADSTACK statement defines a name for each pad stack. The only parameter, <padstack_name>, is required. It is a free field text string that **shall** contain a unique pad stack name. This name **shall** be used throughout the GenCAM file for all references to this pad stack. If the CAD system does not define pad stack names, or just uses a sequenced order, then names such as padstack1, padstack2, etc. **shall** be used instead.

4.8.13.3 PAD

The PAD statement is used to define a pad shape in the pad stack. The PADSTACK includes not only conductive pads, but also the opening for the solder-mask of the appropriate layers (these may be the same size as the conductive pad, or be smaller or larger as required). Single pad PADSTACKS that require solder-mask openings **shall** be included. The first parameter, <layer_ref>, is required and **shall** reference the layer on which the pad is placed, as defined in the LAYERS section of the GenCAM file. The second parameter, <pad_primitive_ref>, is required and **shall** reference a primitive shape as defined in the PRIMITIVES section of the GenCAM file. The third parameter, <linedesc_ref>, is optional and defines the outline of the pad through a LINEDESC reference if an outline of the pad is required. The fourth parameter, <paintdesc_ref>, is optional and defines the texture of the pad through a PAINTDESC reference. Some pads require both LINEDESC and PAINTDESC; others require only one for the texture descriptions. The fifth parameter, <profiledesc_ref>, is optional and provides the LMC and MMC of the finished pad. The sixth parameter, <color_ref>, is optional and defines the color of the pad as referenced in the HEADER of the GenCAM file.

The remaining parameters define the <xform> of the pad. The <xform> is composed of an <xy_ref> parameter that is required and **shall** define the relative position of the point of origin of the pad with

respect to the panel, board, assembly, or fixture origin. A <rotation> parameter is optional, and can define the angle between the position of the pad as defined, and its orientation, measured counterclockwise. Any mirroring **shall** be done before the pad is rotated. The <mirror> parameter is optional, and can be used to indicate that the pad is mirrored. The pad can be placed normally or can be mirrored before becoming part of the product. The last parameter is an optional <scale_factor> multiplier that is applied to all dimensions of the pad.

4.8.13.4 HOLE

The HOLE statement is optional, and can be used to name and define a drilled or punched feature of the padstack. The relationship of the HOLE to the PAD is established under the <padstack_name>. The first parameter, <hole_name>, is required. It is a freeform string that serves to uniquely name the hole instance. Each hole within a single padstack **shall** have a unique name which assigns the hole instance to different layers, if appropriate. The second parameter, <hole_type>, is required and **shall** identify the hole use as one of ELECTRICAL, MECHANICAL or TOOLING. The third parameter, <primitive_ref>, is required and **shall** reference a closed, standard or user defined shape, as defined in the PRIMITIVES section of the GenCAM file. The fourth parameter, <barreldesc_ref>, is required and **shall** reference a barrel description as defined in the PRIMITIVES section of the GenCAM file. The fifth parameter, <profiledesc_ref>, is optional. It can serve to define tolerances on the hole size and plating. The sixth parameter, <layer_ref>, is optional. It can define the individual layer, <layersingle>, or group of layers, <layerset>, through which the hole is drilled, as defined in the LAYERS section of the GenCAM file. If a <layer_ref> parameter is not included then all layers are assumed.

The remaining parameters define the placement of the hole. The x-y transform is required and **shall** define the relative position of the point of origin of the hole primitive with respect to the padstack origin. The rotation parameter is optional, and can define the angle between the orientation of the primitive as defined, and its position in the padstack, measured counterclockwise. Any mirroring **shall** be done before the shape is rotated. The mirror parameter is optional, and can be used to indicate that the hole is mirrored before becoming part of the panel. The <scale_factor> is optional and should be avoided.

A PADSTACKS Section Example

```
$PADSTACKS
  USEDIN: "BD11357", "PNL66540";
    PADSTACK: "p_stack1";
      PAD: "L1", "p102_4VOID", , "Solid1", "Tol2", ,(0, 0),
180.0;
      PAD: "L5", "s102_4FILL", , "Solid1", "Tol2", ,(0, 0);
    PADSTACK: "p_stack2";
      PAD: "L2", "L1_Solid", , "r_r3FILL", , (0, 0), 180.0,
           MIRROR;
      PAD: "L3", "L1_Solid", , "r_r0VOID", , (0, 0), 180.0,
           MIRROR;
      PAD: "L4", "L1_Solid", , "r_r0FILL", , (0, 0), 180.0,
           MIRROR;
      HOLE: "PSH75", ELECTRICAL, "Round10", "CuPlate", "PS3",
           "L1toL3", (0, 0);
$ENDPADSTACKS
```

4.8.14 PACKAGES

The PACKAGES section to the GenCAM file is used to describe a library of component packages. The true physical dimensions of the package are required. One package can be used for many components. The PACKAGES section **shall** be included and have the following form.

```
PACKAGES        ::=     $PACKAGES
                :>      1{
                          USEDIN: <usedin_ref> 0{,<usedin_ref>}n;
                :>        1{
                            PACKAGE: <package_name>, [<package_type>];
                :>          {
                             0{BODY: <artwork_ref>;}1
                            & 0{HEIGHT: <dim_value>;}1
                            & 0{STANDOFF: <dim_value>;}1
                            & 0{TARGET: <target_ref>, [<layer_ref>], [<color_ref>],
                                  <xform>;}1
                            & 1{PIN: <pin_physical_name>, <pin_type>, <pin_exit>,
                                  [<pin_shadow>], <xform>;}n
                <:          }
                <:        }n
                <:      }n
                        $ENDPACKAGES

<usedin_ref>    ::=     string  - a $HEADER assembly(ies)
<package_name>  ::=     string
<package_type>  ::=     { CHIP
                        | TANTALUM
                        | MELF
                        | EMBEDDED
                        | SOT23
                        | SOT52
                        | SOT89
                        | SOT143
                        | SOD123
                        | SOIC
                        | SOPIC
                        | SSOIC
                        | TSOP
                        | CERAMIC_FLATPACK
                        | CERAMIC_QUAD_FLATPACK
                        | PGA
                        | PLASTIC_CHIP_CARRIER
                        | LEADLESS_CERAMIC_CHIP_CARRIER
                        | CERAMIC_DIP
                        | PLASTIC_DIP
                        | CERAMIC_SIP
                        | PLASTIC_SIP
                        | SQUARE_QUAD_FLATPACK
                        | RECTANGULAR_QUAD_FLATPACK
                        | SOJ
                        | PLASTIC_BGA
                        | CERAMIC_BGA
                        | MINI_BGA
                        | CHIP_SCALE
                        | BARE_DIE
                        | FLIPCHIP
                        | AXIAL_LEADED
                        | RADIAL_LEADED
                        | TO_TYPE
```

```
                    | MOLDED
                    | POWER_TRANSISTOR
                    | RELAY_SM
                    | RELAY_TH
                    | TRIMPOT_TH
                    | TRIMPOT_SM
                    | TRANSFORMER
                    | CONNECTOR_SM
                    | CONNECTOR_TH
                    | COIL
                    | CHOKE_SWITCH_SM
                    | SWITCH_TH
                    | HERMETIC_HYBRID
                    | MCM
                    | NETWORK
                    }1

<artwork_ref>       ::=  string
<dim_value>         ::=  p_number
<target_ref>        ::=  string
<layer_ref>         ::=  {<layersingle_ref> | <layerset_ref>}1
<layersingle_ref>   ::=  string
<layerset_ref>      ::=  string
<color_ref>         ::=  string   - <color_name> identified in HEADER; the default is "black"
<xform>             ::=  <xy_ref>, [<rotation>], [<mirror>], [<scale_factor>]
<pin_physical_name> ::=  string   -correlates to the <pin_physical_name> in the DEVICE section
<pin_type>          ::=  {
                          TH_ROUND
                         | TH_RIBBON
                         | TH_V
                         | TERMINATION
                         | GULLWING
                         | BUTTLEAD
                         | JLEAD
                         | SLEAD
                         | WRAPAROUND
                         | CASTELLATION
                         | BALL
                         | COLUMN
                         | LAND
                         }1
<pin_exit>          ::=  {EDGE | BOTTOM | TOP }1
<pin_shadow>        ::=  <artwork_ref>
<artwork_ref>       ::=  string
```

The $PACKAGES and $ENDPACKAGES section keywords mark the PACKAGES section of the GenCAM file. Each package description **shall** start with the PACKAGE keyword.

4.8.14.1 USEDIN

The USEDIN statement associates the statements that follow with the <usedin_name> of an assembly(ies) as defined in the HEADER section of the GenCAM file.

4.8.14.2 PACKAGE

The PACKAGE statement is used to define the device packages as defined in the CAD file. The first parameter, <package_name>, is required. It is a freeform string and serves to uniquely identify the package instance. The second parameter, <package_type>, is optional. It **shall** be a fixed field chosen from one of the GenCAM package type names. It is a reference to the package's BODY shape as defined in the ARTWORKS section of the GenCAM file.

4.8.14.3 BODY

The optional BODY statement is used to identify a reference to a PACKAGE artwork as defined in the ARTWORKS section of the GenCAM file. The first parameter, <artwork_ref>, is required and is a freeform string. Note that the PACKAGE body defined here is associated with a PATTERN definition through a COMPONENT instance definition. Since the package body is defined in the PACKAGE section, it is coupled with the pad stack locations in a PATTERN and the <pin_physical_ref>. The point of origin for these definitions **shall** coincide, and their pin identification **shall** match.

4.8.14.4 HEIGHT

The optional HEIGHT statement is used to identify the component's maximum height from the finished surface. The first parameter, <dim_value>, is required and is provided as a number in dimensional units identified in the HEADER of the GenCAM file.

4.8.14.5 STANDOFF

The optional STANDOFF statement is used to identify the component's clearance from the finished surface. The first parameter, <dim_value>, is required and is provided as a number in dimensional units identified in the HEADER of the GenCAM file.

4.8.14.6 TARGET

The TARGET statement is optional and defines fiducial targets on the package. The first parameter , <target_ref>, is required and **shall** be a reference to the target as described in the ARTWORKS section. The second parameter, <layer_ref>, is optional and **shall** be a reference to the surface of the package as defined in the LAYERS section of the GenCAM file; the <GenCAM_layer_type> **shall** be COMPONENT. The third parameter, <color_ref>, is optional and **shall** match the color as identified in the HEADER section of the GenCAM file.

The remaining parameters define the placement of the target. The x-y transform is required and **shall** define the relative position of the point of origin of the target artwork with respect to the package origin. The rotation parameter is optional, and can define the angle between the orientation of the target as defined, and its position on the package, measured counterclockwise. Any mirroring **shall** be done before the target is rotated. The mirror parameter is optional, and can be used to indicate that the target is mirrored before becoming part of the package. The <scale_factor> is optional and should be avoided.

4.8.14.7 PIN

The PIN statement is required and defines the characteristics of each package lead. The first parameter, <pin_physical_name>, is required. It is a freeform string and defines the physical pin identifier for the package that correlates to the PINDESC statement in the DEVICES section. The second parameter, <pin_type>, is required. It is a fixed field parameter describing the pin type as one of the GenCAM standard package pin types identified in JEDEC-1(IEC-30). The third parameter, <pin_exit>, is required. It is a fixed field identified as either EDGE, BOTTOM, or TOP. The fourth parameter, <pin_shadow>, is optional. It is a reference to an artwork that defines the two dimensional shape of the portion of the lead that extends beyond the package edge, looking down from the top of the package. The remaining parameters are required through the <xform> description. These define the position of the

<pin_shadow>, or the position of the lead (if no portion of the pin extends beyond the body of the package) with respect to the origin of the package.

The x-y transform is required and **shall** define the relative position of the point of origin of the <pin_shadow> with respect to the package origin. The rotation parameter is optional, and can define the angle between the orientation of the pin as defined, and its position on the package, measured counterclockwise. Any mirroring **shall** be done before the pin is rotated. The mirror parameter is optional, and can be used to indicate that the pin is mirrored before becoming part of the package. The <scale_factor> is optional and should be avoided.

A PACKAGES Section Example

```
$PACKAGES
  USEDIN: "ASSY11357-66540", "11356-33540";
    PACKAGE: "CAP_SUPPRESS_TYPE_24", "CERAMIC_DIP";
      BODY: "CAP3";
      HEIGHT: 1.3;
      STANDOFF: 0.05;
      PINS: 16, "TH_RIBBON", EDGE, "100", (0.5, 18.5);
      PINS: 14, "TH_RIBBON", EDGE, "100", (0.5, 16.0);
      PINS: 12, "TH_RIBBON", EDGE, "100", (0.5, 13.5);
      PINS: 10, "TH_RIBBON", EDGE, "100", (0.5, 11.0);
      PINS: 08, "TH_RIBBON", EDGE, "100", (0.5, 8.5);
      PINS: 06, "TH_RIBBON", EDGE, "100", (0.5, 6.0);
      PINS: 04, "TH_RIBBON", EDGE, "100", (0.5, 3.5);
      PINS: 02, "TH_RIBBON", EDGE, "100", (0.5, 1.0);
      PINS: 15, "TH_RIBBON", EDGE, "100", (8.0, 18.5);
      PINS: 13, "TH_RIBBON", EDGE, "100", (8.0, 16.0);
      PINS: 11, "TH_RIBBON", EDGE, "100", (8.0, 13.5);
      PINS: 09, "TH_RIBBON", EDGE, "100", (8.0, 11.0);
      PINS: 07, "TH_RIBBON", EDGE, "100", (8.0, 8.5);
      PINS: 05, "TH_RIBBON", EDGE, "100", (8.0, 6.0);
      PINS: 03, "TH_RIBBON", EDGE, "100", (8.0, 3.5);
      PINS: 01, "TH_RIBBON", EDGE, "100", (8.0, 1.0);
     TARGET: "FIDUC_X", COMPONENT, "White", (1.0, 2.0);
$ENDPACKAGES
```

4.8.15 FAMILIES

The FAMILIES section is optional and is used to describe the logic families of the components attached to the board. The FAMILIES section has the following form.

```
FAMILIES    ::=  $FAMILIES
            :>   1{
                   USEDIN: <usedin_ref> 0{, <usedin_ref>}n;
            :>      1{
                      FAMILY: <family_name>;
            :>          {
                            LOGIC: <drive_high>, <drive_low>, <receive_high>,
                                   <receive_low>, <open_input_logic>;
                          & RISETIME: <edge_speed>;
                          & LOAD: <load>;
            <:          }
```

```
<:          }n
<:        }n
        $ENDFAMILIES
```

<usedin_ref>	::=	string - a $HEADER assembly(ies)
<family_name>	::=	string
<drive_high>	::=	number
<drive_low>	::=	number
<receive_high>	::=	number
<receive_low>	::=	number
<open_input_logic>	::=	{ONE | ZERO | X}1
<edge_speed>	::=	p_number
<load>	::=	{UP | DOWN | NONE}

The $FAMILIES and $ENDFAMILIES section keywords mark the FAMILIES section of the GenCAM file. Each family description **shall** start with the FAMILY keyword.

4.8.15.1 USEDIN

The USEDIN statement associates the statements that follow with the <usedin_name> of an assembly(ies) as defined in the HEADER section of the GenCAM file.

4.8.15.2 FAMILY

The FAMILY statement defines a name for the family feature for reference. Each <family_name> defined in the FAMILIES section **shall** be unique.

4.8.15.3 LOGIC

The LOGIC statement is used to describe the expected logical voltage characteristics of the family. The first parameter,<drive_high>, is required and defines the minimum voltage level, which a device of the family will source from an output when the expected state is a logical 1. The second parameter, <drive_low> is required and defines the maximum voltage level that a device of the family will source from an output when the expected state is a logical 0. The third parameter, <receive_high>, is required and defines the minimum voltage level, which must be driven for a device of the family to be expected to recognize a logical 1. The fourth parameter, <receive_low>, is required and defines the maximum voltage level for which a device of the family can be expected to recognize a logical 0. The fifth parameter, <open_input_logic>, is required and is the default logic-state exhibited by unconnected (floating) input pins. It is a fixed field parameter of either ONE, ZERO, or X.

4.8.15.4 RISETIME

The RISETIME statement defines the edge speed of a device or pin of a family. The only parameter, <edge_speed>, is required and is expressed in volts per nanosecond.

4.8.15.5 LOAD

The LOAD statement expresses whether a device or pin of a family expects to have its output pins pulled up, pulled down, or left floating by the surrounding circuitry. The only parameter, <load>, is required and is a fixed field parameter of either UP, DOWN, or NONE.

A FAMILIES Section Example

```
$FAMILIES
 USEDIN: "11357-66540";
  FAMILY: "CMOS";
    LOGIC: 4.0, 0.8, 3.6, 1.3, X;
```

```
    RISETIME: 50;
    LOAD: NONE;
  FAMILY: "TTL";
    LOGIC: 3.5, 0.2, 2.0, 0.8, 1;
    RISETIME: 100;
    LOAD: UP;
$ENDFAMILIES
```

4.8.16 DEVICES

The DEVICES section holds the device descriptions for all the components used on the board. These statements apply to the device itself and do not depend upon any characteristic of the printed circuit board (i.e. are independent of PACKAGE). The DEVICES section **shall** be included and can use some or all of the statements described below, in any sequence.

```
DEVICES     ::=  $DEVICES
            :>   1{
                   USEDIN: <usedin_ref> 0{, <usedin_ref>}n;
            :>      1{
                     DEVICE: <part_name>;
            :>         {
                       PACKAGE: <package_ref>;
                     & 0{PART: <mfg_part_id>, [<part_revision>];}1
                     & 0{ALIAS: <mfg_part_id>, [<part_revision>];}n
                     & TYPE: <type_desc>;
                     & 0{FAMILY: <family_ref>;}1
                     &  0{PINDESC: <pin_physical_name>, [<pin_funct>],
                            [<circuit_number>], [<sympin_ref>], [<symbol_ref>],
                            [<family_ref>];}n
                     & 0{
                         VALUE: <measured_value>, <value_type>;
            :>            {
                            {    TOL: <p_dim_value>;
                              | { NTOL: <p_dim_value>; & PTOL: <p_dim_value>;}
                            }
                            & 0{DESC: <description>;}1
            <:             }
                        }n
            <:       }
            <:     }n
            <:  }n
                $ENDDEVICES

<usedin_ref>      ::=  string  - a $HEADER assembly(ies)
<part_name>       ::=  string
<package_ref>     ::=  string
<mfg_part_id>     ::=  string
<part_revision>   ::=  string
<type_desc>       ::=  {RES          - Used for two terminal resistors
                      | VRES         - Used for variable resistors
                      | FABRES       - Used for embedded resistors
                      | RPCK         - Used for resistor pack
                      | DPCK         - Used for diode pack
                      | LEDPCK       - Used for LED pack
```

		| HYBRID	- Used for AtoD, DtoA
		| CAP	- Used for two terminal non-polarized capacitors
		| VCAP	- Used for variable capacitors
		| PCAP	- Used for two terminal polarized capacitors
		| TCAP	- Used for tantalum capacitors
		| FABCAP	- Used for embedded capacitors
		| CPCK	- Used for capacitor packs
		| IND	- Used for inductors
		| VIND	- Used for variable inductor
		| XFMR	- Used for transformer
		| DIODE	- Used for diodes, including Schottkey
		| DIAC	- Used for diacs
		| ZENER	- Used for zener diodes
		| BRIDGE	- Used for silicone bridge rectifier
		| PNP	
		| NPN	- Used for transistors, unijunctions and darlingtons.
		| NFET	
		| PFET	
		| NJFET	
		| PJFET	- Used for FET families
		| TRIAC	- Used for triacs
		| SCR	- Used for Thyristors
		| VR	- Used for voltage regulators
		| OPTO	- Used for opto-isolators
		| LED	- Light Emitting Diode
		| OPAMP	- Used for operational amplifier ICs
		| XTAL	- Used for crystals
		| RELAY	- Used for relays
		| SWITCH	- Used for switches
		| FUSE	- Used for fuses
		| JUMPER	- Used for jumpers
		| CONN	- Used for connectors
		| SOCKET	- Used for sockets
		| LOGIC	- Used for all logic devices
		| ANALOG	- Used for analog ICs
		| OTHER	- Used when none of the standard types are sufficient
		}1	
<family_ref>	::=	string	
<pin_physical_name>	::=	string	- Establishes the pin identifier for the device
<pin_funct>	::=	{DRIVER	- Used for outputs which drive a net.
		| RECEIVER	- Used for inputs
		| BIDIRECTIONAL	- Used for bi-directional logic
		| ANALOGIN	
		| ANALOGOUT	- Used for analog ICs
		| NCLOSED	
		| NOPEN	- Used for relays and switches
		| POWER	- Used for power supply pins
		| GROUND	- Used for ground pins
		| ANALOG	- Used for analog pins
		| DIGITAL	- Used for any family of logic pins
		| INACTIVE	- Used for resistors, capacitors, etc. pins
		| ANODE	- Used for diodes, zeners, unijunctions, thyristors etc.
		| CATHODE	- Used for diodes, zeners, unijunctions, thyristors etc.
		| BASE	- Used for transistors

| COLLECTOR | - Used for transistors (including IGBTs)
| EMITTER | - Used for transistors (including IGBTs)
| SOURCE | - Used for FETs
| DRAIN | GATE | - Used for FETs
| WIPER | - Used for variable components
| CASE | - Used for connection to device screen or can
| CLOCK | - Used for clock
| ENABLE | DISABLE | - Used for device enable/disable
| TDI | TDO | TMS | TCK | TRST – Used for Boundary Scan
| INTNC | - Used for pins that are internally disconnected
}1

<circuit_number> ::= string
<sympin_ref> ::= string
<symbol_ref> ::= string
<measured_value> ::= number
<value_type> ::= {OHM | FARAD | HENRY | VOLT | AMP | WATT | HERTZ | JOULE | LUMEN}1
<p_dim_value> ::= p_number
<description> ::= string

The $DEVICES and $ENDDEVICES section keywords mark the DEVICES section of the GenCAM file. Each device description **shall** start with the DEVICE keyword.

4.8.16.1 USEDIN

The USEDIN statement associates the statements that follow with the <usedin_name> of an assembly(ies) as defined in the HEADER section of the GenCAM file.

4.8.16.2 DEVICE

The DEVICE statement defines a name for each device type. The first parameter, <part_name>, is required. It is a freeform string that uniquely identifies the device.

4.8.16.3 PACKAGE

The PACKAGE statement serves to correlate the device to a package type. Exactly one PACKAGE statement **shall** be included and must reference a package as defined in the PACKAGES section of the GenCAM file. The only parameter, <package_ref>, is required. It is a freeform string that serves to reference the package definition by name.

4.8.16.4 PART

The PART statement is an optional free field for the user to define a manufacturer's part number. It has no cross -reference to any other GenCAM section. This PART name does not have to be unique to any one part. The first parameter, <mfg_part_id> is required. It is the manufacturer's part identification number. The second parameter, <part_revision>, is optional and defines a revision level of the manufacturer's part, if appropriate.

4.8.16.5 ALIAS

The ALIAS statement is optional. It is used to establish an alternate to the original PART description. The first parameter, <mfg_part_id> is required. It is the alternate manufacturer's part identification number. The second parameter, <part_revision>, is optional and defines a revision level of the manufacturer's part, if appropriate.

4.8.16.6 TYPE

The TYPE statement is required and is a fixed field. The only parameter, <type_desc>, identifies the fixed field for the user to define one of the GenCAM device types.

4.8.16.7 FAMILY

The FAMILY statement is optional and can be used to associate the device with a logic family as defined in the FAMILIES section of the GenCAM file. If the device pins exhibit multiple logic family characteristics, then the FAMILY statement should be omitted, and the logic family association should be added to each PINDESC statement. The only parameter, <family_ref>, is required and is a freeform string.

4.8.16.8 PINDESC

The PINDESC statement is optional, and can be used to describe the functionality of each pin of the device. This statement should be used to provide CAD data that is required for tester input data. Any CAD data that might be relevant to tester input should be held in the PINDESC statement. Particularly useful is the naming of the anode and cathode on diodes and zeners; collector, base and emitters on transistors; and gate, source, and drain on fets. The first parameter, <pin_physical_ref>, is required and **shall** correlate to the pin physical name assigned in the PACKAGES section of the GenCAM file. Every component and pattern defined in the COMPONENTS section use the same pin designator. Only one PINDESC statement is allowed for each pin for any one device. The second parameter, <pin_funct>, is one of the fixed fields identified as GenCAM pin functions. The third parameter, <circuit_number>, is a freeform string and is an optional reference to the circuit number used to describe the circuit number in the IC package. The fourth parameter, <sympin_ref>, is optional, It is a reference to the symbol pin in the logic symbol and is a freeform string. The fifth parameter, <symbol_ref>, is an optional reference to the symbol that uses that physical pin. The sixth parameter, <family_ref>, is an optional reference to the logic FAMILY. If the FAMILY of the device is consistent across all signal pins, then it is not necessary to define each pin's FAMILY, but in the case where the device has multiple family characteristics, it can be necessary to specifically associate a logic family with each input and output.

4.8.16.9 VALUE

The VALUE statement is optional and provides the user an opportunity to give the device a value. The type of value entered **shall** always be the same for all the same type of devices. For example, if the value used for resistors is resistance, then all resistors **shall** have values that are resistance. The first parameter, <measured_value>, is required and is a number that expresses the value. The second parameter, <value_type>, is a fixed field parameter and is either OHM, FARAD, HENRY, VOLT, AMP, WATT, HERTZ, JOULE, or LUMEN.

4.8.16.10 TOL

The TOL statement is optional and allows the user to specify a ± tolerance for the device. If the positive and negative tolerance values are different, then the statements NTOL and PTOL **shall** be used instead of TOL. The first parameter, <p_dim_value>, is required and is a positive number that describes the permitted variation from the value. Any string can be entered as a positive floating point percentage value.

4.8.16.11 NTOL

The NTOL statement is optional and allows the user to specify a negative (minimum) tolerance for the device. Any string can be entered as a positive floating point percentage value. A negative sign is implicit in the definition of NTOL and is ignored on reading the file and is never inserted on writing a GenCAM file.

4.8.16.12 PTOL

The PTOL statement is optional allows the user to specify a positive (maximum) tolerance for the device. Any string can be entered as a positive, floating-point percentage value.

4.8.16.13 DESC

The DESC statement is optional and the <description> parameter can be used to describe the device.

A DEVICES Section Example

```
$DEVICES
  USEDIN: "ASSY11357", "ASSY66540";
  DEVICE: "89-1N4148";
    PART: "1N4148";
    TYPE: DIODE;
    PACKAGE: "AXIAL2";
    PINDESC: "PIN1", ANODE;
    PINDESC: "PIN2", CATHODE;
    VALUE: 0.6;
    TOL: 33.0;
    DESC: "Diode 1N4148 bandoleer reverse voltage 100V";
$ENDDEVICES
```

4.8.17 COMPONENTS

The COMPONENTS section holds all the information pertinent to each component on the board. This information includes component positions on the board and a reference to the device description for both electrical and mechanical devices. The COMPONENTS section **shall** be included and have the following form.

```
COMPONENTS  ::=  $COMPONENTS
            :>   1{
                   USEDIN: <usedin_ref> 0{, <usedin_ref>}n;
            :>      1{
                      COMPONENT: <ref_desig>, [<layer_ref>], [<location>];
            :>          {
                          {DEVICE: <part_ref>, [<xform>];
                        | MECHANICAL: <part_ref>, [<xform>];}1
                        & 0{PATTERN: <pattern_ref>, [<xform>];}1
                        & 0{ARTWORK: <artwork_ref>, <layer_ref>, [<color_ref>],
                                [<xform>];}n
                        & 0{PROGRAM: <program_name>, <program_rev>,
                                [<program_date>];}1
                        & 0{PREPARATION: <comp_prep_name>, <comp_prep_type>;
            :>                  {
                                    <shape_builder>
            <:                      }
                        & 0{PRESENTATION: <delivery_system>, [<configuration>],
                                [<other>];}1
                          }n
                        & 0{
                            KEEPOUT: <keepout_name>, <keepout_type>,
                                [<layer_ref>];
            :>              1{
                            <polygon_builder>
```

```
<:                    }n
                    }n
                & 0{SHEET: <sheet_ref>;}1
                & 0{ONFAIL: <text>;}1
<:              }
<:          }n
<:      }n
        $ENDCOMPONENTS
```

```
<usedin_ref>          ::=   string  - a $HEADER assembly(ies)
<ref_desig>           ::=   string
<layer_ref>           ::=   {<layersingle_ref> | <layerset_ref>}1
<layersingle_ref>     ::=   string
<layerset_ref>        ::=   string
<location>            ::=   <xy_ref>, [<rotation>], [<mirror>]
<part_ref>            ::=   string
<xform>               ::=   <xy_ref>, [<rotation>], [<mirror>], [<scale_factor>]
<pattern_ref>         ::=   string
<artwork_ref>         ::=   string
<color_ref>           ::=   string  - <color_name> identified in $HEADER; the default is BLACK
<program_name>        ::=   string
<program_rev>         ::=   string
<program_date>        ::=   date_format
<comp_prep_name>      ::=   string
<comp_prep_type>      ::=   {LEADLENGTH | LEADBEND | CLINCH | CLINCHLENGTH |
                            PREPCODE | OTHER}1
<delivery_system>     ::=   {REEL | PACK | BULK | TRAY | TUBE | CARTRIDGE}1
<configuration>       ::=   string  - i.e. "size", "matrix", "width"
<other>               ::=   string
<keepout_name>        ::=   string
<keepout_type>        ::=   {COMPONENT | VIA | ROUTE | TESTPIN | TESTPROBE |
                            BOARD}1
<sheet_ref>           ::=   string
<text>                ::=   string
```

The $COMPONENTS and $ENDCOMPONENTS section keywords mark the COMPONENTS section of the GenCAM file. Each component description **shall** start with the COMPONENT keyword, followed by the other statements in any sequence.

4.8.17.1 USEDIN

The USEDIN statement associates the statements that follow with the <usedin_name> of an assembly(ies) as defined in the HEADER section of the GenCAM file.

4.8.17.2 COMPONENT

The COMPONENT statement defines one component instance. The first parameter, <ref_desig>, is a freeform string for the user to define a unique name for either the electrical component (DEVICE) or the mechanical component (MECHANICAL), typically the CAD component reference designator. If the CAD system does not define component names, or just uses a sequenced order, then names such as u1, u2, u3, etc. must be used instead. Examples of component name are U32, C202, R51 or PLA132. The second parameter, <layer_ref>, is optional and **shall** indicate on which side of the board the component is mounted, or which layer of the board the part is embedded. The surface parameter TOP, EDGE or BOTTOM, or a reference to a layer as defined in the LAYERS section, **shall** be used. Specifying that the surface is BOTTOM does not imply that the package is mirrored, nor does it imply that pads that were on

the TOP are now on BOTTOM. The third parameter, <location>, is optional and is to be included to give the absolute x and y coordinates of the origin of the component in relation to the board or panel origin, as well as rotation and mirroring.

The origin of the component is defined by its PACKAGE, PATTERN, or MECHANICAL references, which describe the package/mechanical body shape and pin locations. The fourth parameter, <xy_ref> is required and establishes the relation of the component to the board or panel to which the component is being mounted. The fifth parameter, <rotation> is optional and can be included to give the rotation of the component on the board relative to that defined by its PACKAGE or MECHANICAL reference. The angle is measured counterclockwise from the defined position in the PACKAGE or MECHANICAL section to the position on the board when viewed looking down onto the top of the board. Mirroring takes affect before the component is rotated. Any pins defined with the package will be rotated with the package.

4.8.17.3 DEVICE

Exactly one DEVICE or MECHANICAL statement **shall** be provided for each component. The first parameter, <part_ref>, is required. It is a freeform string for the user to associate the component designator name to a device description in the DEVICES section of the GenCAM file. The second parameter, <xform> is optional, and can be used to describe the location of the device package origin on the board, when it is different than that of the component placement.

4.8.17.4 MECHANICAL

The MECHANICAL statement is required if no DEVICE statement is present. The first parameter, <mechanical_ref>, is required and is a freeform string that references a <part_name> in the MECHANICALS section. The second parameter, <xform> is optional, and can be used to describe the location of the mechanical part origin on the board, when it is different than that of the component placement.

4.8.17.5 PATTERN

Exactly one PATTERN statement **shall** be provided for each device listed as a component; a PATTERN statement for a MECHANICAL is optional. The first parameter, <pattern_ref>, is required. It is a freeform string for the user to associate the component designator to a pattern description as defined in the PATTERNS section of the GenCAM file. The second parameter, <xform>, is optional and can be used to describe the location of the pattern origin on the board or panel, when it is different than that of the component placement.

4.8.17.6 ARTWORK

The ARTWORK statement is optional and is used to give the component any artwork feature that has been defined in the ARTWORKS section of the GenCAM file. Components may use different artworks at different locations on the board. The first parameter, <artwork_ref>, is required and **shall** be the artwork name as defined in the ARTWORKS section in the GenCAM file. The second parameter, <layer_ref>, is required and **shall** be a reference to a layer or layer set as defined in the LAYERS section of the GenCAM file. The third parameter, <color_ref>, is optional and defines the color referenced in the HEADER of the GenCAM file.

The remaining parameters are defined by the <xform> and are optional if the artwork differs from the component location. The fourth parameter, <xy_ref>, is required and establishes the relative position of the artwork item origin with respect to the origin of the component. The fifth parameter, <rotation>, is optional and **shall** be the rotation of the artwork feature about the artwork origin. The rotation is the angle between the artwork feature, as defined in the ARTWORKS section and its position on the board or panel, measured counterclockwise. Any mirroring **shall** be done before the ARTWORK feature is rotated. The sixth parameter, <mirror>, is optional and **shall** be the artwork feature mirror definition.

The artwork feature can be placed normally or can be mirrored before becoming part of the component definition. When an artwork feature is mirrored all the artwork feature's items are mirrored but stay on the same layer(s). The last parameter is an optional scaling multiplier to apply to the artwork dimensions.

4.8.17.7 PROGRAM

The PROGRAM statement is optional and can be used to define the software download of the component. The first parameter, <program_name>, is required. It is a freeform string that names the program. The second parameter, <program_ref>, is required and defines the program revision as a freeform string. The third parameter, <program_date>, is optional and **shall** provide the program date code in the GenCAM date code format.

4.8.17.8 PREPARATION

The PREPARATION statement is optional and provides information on component lead conditions, both before and after insertion. The first parameter, <comp_prep_name>, is required and is a unique string to identify the lead preparation. The second parameter, <comp_prep_type>, is required and uses one of the reserved fixed parameters: LEADLENGTH, LEADBEND, CLINCH, CLINCHLENGTH, PREPCODE, or OTHER. The <comp_prep_type> parameter will be immediately followed by the <shape_builder> to describe the physical lead bend requirements.

4.8.17.9 PRESENTATION

The PRESENTATION statement is optional and provides information on the delivery system packaging media for the component. The first parameter, <delivery_system>, is a required fixed field parameter specifying the part delivery presentation as being one of REEL, PACK, BULK, TRAY, TUBE, or CARTRIDGE. The second parameter, <configuration>, is optional. It is a string used to define characteristics of the media used. The third parameter, <other>, is optional and provides additional information regarding presentation.

4.8.17.10 KEEPOUT

The KEEPOUT statement is optional and is used to name and define areas associated with the component areas that are not accessible to test or insertion machines. If the CAD system does not provide a unique name for each keepout, then the GenCAM file **shall** be provided with names such as keepout1, keepout2 etc. The first parameter, <keepout_name>, is required and is a freeform string and assigns a name to the keepout area. The second parameter, <keepout_type>, is required and is a fixed field parameter of either COMPONENT, VIA, ROUTE, TESTPIN, TESTPROBE, or BOARD. The third parameter, <layer_ref>, is optional and **shall** define the surface or layer set to which the keepout region is applied. When omitted, all layers are assumed. The third parameter is followed immediately by one or more <polygon_builder> that define the keepout area.

4.8.17.11 SHEET

The SHEET statement is optional and can be used to define the schematic sheet number, a zone or anything else that is a location property of the component. The only parameter, <sheet_ref>, is required and is a text free string referencing a sheet in the DRAWINGS section of the GenCAM file.

4.8.17.12 ONFAIL

The ONFAIL statement is optional. The first parameter, <text>, is required and is a freeform string that can be used to send the failure reporting device/file when an in-circuit test for the component fails.

A COMPONENTS Section Example

```
$COMPONENTS
  USEDIN: "ASSY11357", "ASSY66540";
```

```
        COMPONENT: "D102", TOP, (1200, 1800), 90.0;
          DEVICE: "1N4148";
          PATTERN: "2-400";
          ARTWORK: "ORIGIN_MARKER18", "SilkscreenTop", "Yellow",
               (0,0), 0.0, MIRROR;
          ONFAIL: "Check output enable from U5 pin 4";
          SHEET: "12_B3";
        MECHANICAL: "Heatsink2";
    $ENDCOMPONENTS
```

4.8.18 POWER

The POWER section is optional and can be used to describe power injection to the board. The POWER section has the following form.

```
POWER          ::=  $POWER
                :>  1{
                      USEDIN: <usedin_ref> 0{, <usedin_ref>}n;
                :>      1{
                          SUPPLY: <voltage>, <current_limit>, <net_ref>, <refnet_ref>,
                                [<supply_name>];
                <:          }n
                <:  }n
                    $ENDPOWER
```

<usedin_ref>	::=	string - a $HEADER fixture(s), panel(s), board(s) or assembly(ies)
<voltage>	::=	number - in volts
<current_limit>	::=	p_number - in amps
<net_ref>	::=	string
<refnet_ref>	::=	string
<supply_name>	::=	string

The $POWER and $ENDPOWER keyword statements mark the POWER section of the GenCAM file. Each power injection description **shall** start with the keyword SUPPLY.

4.8.18.1 USEDIN

The USEDIN statement associates the statements that follow with the <usedin_name> of a board(s), panel(s), assembly(ies) or fixture(s) as defined in the HEADER section of the GenCAM file.

4.8.18.2 SUPPLY

The SUPPLY statement is required and defines test system power supply parameters, and associates the parameters with two board signal names. The first parameter, <voltage>, is required and is a number (in volts) defining the voltage level to be attained. The second parameter, <current_limit>, is required and is a number (in amps) defining the maximum current limit. The third parameter, <net_ref>, is required, and is a freeform string referencing an electrical network (power injection net) as named in the ROUTES section of the GenCAM file. The fourth parameter, <refnet_ref>, is required and is a freeform string referencing an electrical network (typically board ground) as named in the ROUTES section of the GenCAM file. The last parameter, <supply_name>, is an optional freeform string that can be used to identify the test system power supply.

A POWER Section Example

```
$POWER
  USEDIN: "ASSY11357", "ASSY66540";
   SUPPLY: 5.0, 2.0, "VCC", "GND";
   SUPPLY: 3.3, 0.5, "LOW_V", "DCOM", "COM12";
$ENDPOWER
```

4.8.19 ROUTES

The ROUTES section is used to define the board signals and describe the track routing and all electrically significant features of the board. The ROUTES section **shall** be included, and have the following form:

```
ROUTES      ::=  $ROUTES
            :>   1{
                  USEDIN: <usedin_ref> 0{,<usedin_ref>}n;
            :>    1{
                   ROUTE:<net_name>, <net_class>, [<refnet_ref>], [<impedance>],
                         [<impedance_tol>], [<capacitance>], [<inductance>];
            :>          {
                          1{PATH: <layer_ref>, <linedesc_ref>, [<profiledesc_ref>],
                                  [<impedance>], [<impedance_tol>], [<capacitance>],
                                  [<inductance>];
            :>                      1{
                                       <polyline_builder>
            <:                       }n
                           }n
                            & 0{PLANE: <layer_ref>, <paintdesc_ref>, [<impedance>],
                                       [<capacitance>], [<inductance>];
            :>                      1{
                                       <polygon_builder>
            <:                       }n
                           }n
                          & 0{ VIA: <via_name>, <padstack_ref>, [<testpoint_xy>],
                                  [<access_desc>], <xform>;}n
                          & 0{ TESTPAD: <testpad_name>, <padstack_ref>, [<testpoint_xy>],
                                  [<access_desc>], <xform>;}n
                          & 0{ COMPPIN: <component_ref>, <pin_physical_ref>,
                                  [<testpoint_xy>], [<access_desc>];}n
                          & 0{ CONPIN: <connector_ref>, <pin_physical_ref>,
                                  [<testpoint_xy>], [<access_desc>];}n
                          & 0{HIGHPOTTEST: <refnet_ref>, <bias_voltage>,
                                  <leakage_current>, <dwell_time>, [<rise_time>],
                                  [<fall_time>];}1
            <:           }
            <:         }n
            <:       }n
                   $ENDROUTES

<usedin_ref>      ::=   string  - a $HEADER panel(s), board(s) or assembly(ies)
<net_name>        ::=   string        - User-defined string or GENCAMEXTNC for
                                        unconnected features
<net_class>       ::=   {SIGNAL | CLK | FIXED}1  - the default is SIGNAL
<refnet_ref>      ::=   string
```

<impedance>	::=	p_number	- Units **shall** be expressed in ohms
<impedance_tol>	::=	p_number	- Units **shall** be expressed in percentage
<capacitance>	::=	p_number	- Units **shall** be expressed in farads
<inductance>	::=	p_number	- Units **shall** be expressed in Henrys
			- Values associated with a PATH override those associated with a ROUTE. ROUTE requirements should be left blank if individual paths have properties.
<layer_ref>	::=	{<layersingle_ref> | <layerset_ref>}1	
<layersingle_ref>	::=	string	
<layerset_ref>	::=	string	
<linedesc_ref>	::=	string	
<profiledesc_ref>	::=	string	
<paintdesc_ref>	::=	string	
<via_name>	::=	string	
<padstack_ref>	::=	string	
<xform>	::=	<xy_ref>, [<rotation>],[<mirror>],[<scale_factor>]	
<testpoint_xy>	::=	<xy_ref>	
<access_desc>	::=	{NOACCESS | NOPROBE | CRITICAL | AVAILABLE}1	- the default is NOACCESS
<testpad_name>	::=	string	
<component_ref>	::=	string	- Reference to component by <ref_desig> in COMPONENTS section
<pin_physical_ref>	::=	string	- Reference to <pin_physical_name> in PACKAGES section
<connector_ref>	::=	string	- Reference to connector identified by <ref_desig> in COMPONENTS section
<bias_voltage>	::=	number	
<leakage_current>	::=	number	- Units **shall** be expressed in amps
<dwell_time>	::=	number	- Units **shall** be expressed in seconds
<rise_time>	::=	number	
<fall_time>	::=	number	- Units **shall** be expressed in seconds

The $ROUTES and $ENDROUTES section keywords mark the ROUTES section of the GenCAM file. Each route description **shall** start with a ROUTE keyword.

4.8.19.1 USEDIN

The USEDIN statement associates the statements that follow with the <usedin_name> of a board(s), panel(s), assembly(ies) or fixture(s) as defined in the HEADER section of the GenCAM file.

4.8.19.2 ROUTE

The ROUTE statement is required and identifies the signal (net) name. The first parameter, <net_name> is required and is a freeform string used to specify the net name. The second parameter, <net_class>, is required and is used to specify nets that may require special consideration at in-circuit test, and uses the reserved words of SIGNAL (default) for standard nets, CLK for fast edge-speed nets, and FIXED for non-driven nets with either injected power, or fixed voltage levels which are derived from the circuit. The third parameter, <refnet_ref>, is optional and is a freeform string to identify electrical networks that should be referenced to the named net. The fourth and fifth parameters, <impedance> and <impedance_tol>, are optional and can be used to specify the impedance and impedance tolerance limit of the trace in ohms. The sixth and seventh parameters, <capacitance> and <inductance>, are used to identify capacitance (in Farads) and inductance (in Henrys). GenCAM reserves the net name GENCAMEXTNC as a null net name. All component pins and connector pins that are otherwise not connected to any net, **shall** appear in a ROUTE with this reserved name. Note that this does not imply that these pins are electrically connected together.

4.8.19.3 PATH

The PATH statement is required and identifies the conductor (track) width, width tolerance limit and layout description. The coordinates given for the trace path **shall** always be in the center of the trace. The PATH keyword can appear at any time within a route description where a trace changes width. The first parameter, <layer_ref>, is required and **shall** reference a layer that is defined in the LAYERS section of the GenCAM file. The <layer_ref> keyword can appear at any time within a net or route where the route changes layers. The second parameter, <line_desc>, is required and describes the width of the conductor (trace). The third parameter, <profiledesc_ref>, is optional and defines the tolerances on the conductors. The fourth and fifth parameters, <impedance> and <impedance_tol>, are optional and can be used to specify the impedance and impedance tolerance limit of the trace in ohms. The sixth and seventh parameters, <capacitance> and <inductance>, are used to identify capacitance (in Farads) and inductance (in Henrys).

4.8.19.4 PLANE

The PLANE statement is optional and is used to define an enclosed area of copper using GenCAM Standard primitives, or a reference to a shape PRIMITIVE. A ROUTE statement **shall** appear before a PLANE statement to define a signal name. If component pins and pads are found within the area defined (on the same layer), then a PATH statement **shall** also appear. The copper area is assumed to be clear of the pins by a non copper trace of graphic description LINDESC using the ERASE <line_type> parameter. Component pins are only connected to the copper plane if the ROUTES section defines the component pins as being on the same signal as the plane. Clearances between the PLANE and other signals are not assumed. The first parameter after the keyword statement, <layer_ref>, is required as is used to assign the plane a unique layer. If the CAD system does not provide unique names then names such as plane1, plane2, etc. must be used for the ROUTE name. The second parameter, <paintdesc_ref>, is required and defines a paint requirement when a second plane can be described with x-y coordinates inside another plane on the same layer. The PAINT mode of VOID **shall** be used when this inner shape is not copper. The third parameter, <impedance>, is optional and can provide the impedance requirements as a number in ohms for the plane. The fourth parameter, <capacitance> is optional and is a number in Farads. The fifth parameter, <inductance>, is optional and is a number in Henrys.

4.8.19.5 VIA

The VIA statement is optional and is used to describe a via in the ROUTES section. The first parameter, <via_name>, is required and **shall** be a unique name for the via. If the CAD system does not name the vias then names such as via1, via2, etc. must be used. The second parameter, <padstack_ref>, is required and **shall** be the pad stack name as defined in the PADSTACKS section. The third parameter, <testpoint_xy>, is optional, and **shall** be the absolute coordinates of the via PADSTACK. The fourth parameter, <access_desc>, is optional and is a fixed field parameter of NOACCESS, NOPROBE, CRITICAL, or AVAILABLE; the default is NOACCESS. To be test-probe accessible the via must be on the top or bottom surfaces and should not be covered by the solder mask. The fifth parameter, <xform>, is required and is used to position the via with respect to the origin of the board or panel. The transformation includes the <xy_ref>, <rotation>, <mirror>, and <scale_factor>.

4.8.19.6 TESTPAD

The TESTPAD statement is optional and defines the location of a pad, its type and orientation. Such pads are usually for test. The first parameter, <testpad_name>, is required and **shall** be a unique name for the test pad. If the CAD system does not name the test pads then names such as tp1, tp2, etc. **shall** be used. The second parameter, <padstack_ref>, is required and **shall** be the pad stack name as defined in the PADSTACKS section. The third parameter, <testpoint_xy>, is optional and can define the absolute coordinates for probing the test pad if this location differs from the origin of the pad. The fourth parameter, <access_desc>, is optional and defines the access of the TESTPAD as a fixed field parameter of NOACCESS, NOPROBE, CRITICAL, or AVAILABLE; the default is NOACCESS. To be test-probe

accessible, the testpad must be be on the top or bottom surfaces and should not be covered by the solder mask. The fifth parameter, <xform>, is required and is used to position the test pad with respect to the origin of the board or panel. The transformation includes the <xy_ref>, <rotation>, <mirror>, and <scale_factor>.

4.8.19.7 COMPPIN

The COMPPIN statement is optional and is used to identify a component pin on the net. One COMPPIN statement **shall** be included for each component pin that is associated with the net. The first parameter, <component_ref>, is required and identifies a unique name of the associated component. The second parameter, <pin_physical_ref>, is required and establishes the unique name of the associated pin as identified in the PACKAGE section. The third parameter, <testpoint_xy>, is optional and can define the absolute coordinates for probing the COMPPIN if this location differs from the origin of the pad stack. The fourth parameter, <access_desc>, is optional and can define a special test probe access type as one of the fixed field description of NOACCESS, NOPROBE, CRITICAL, or AVAILABLE; the default is NOACCESS.

4.8.19.8 CONPIN

The CONPIN statement is required and is used to identify a fabricated connector pin (e.g. board edge connector pin) on the net. One CONPIN statement **shall** be included for each fabricated connector pin that is associated with the net. The first parameters, <connector_ref>, is required and is a unique name associated with the connector identified by its reference designator, <ref_desig>, in the COMPONENTS section. The second parameter, <pin_physical_name>, is required and establishes a unique name of the edge-board connector. The third parameter, <testpoint_xy>, is optional and can define the absolute coordinates for probing the CONPIN. The fourth parameter, <access_desc>, is optional and can define a special test probe access type as one of the fixed field description of NOACCESS, NOPROBE, or CRITICAL.

4.8.19.9 HIGHPOTTEST

The HIGHPOTTEST statement is optional and can be used to indicate a route that must be tested using high voltage and current to test board performance capability. The first parameter, <refnet_ref>, is required and is a freeform string that defines the net to be tested. The second parameter, <bias_voltage>, is required and identifies the maximum voltage that shall be used for the highpot test. The third parameter, <leakage_current>, is required and identifies the maximum current used in the highpot test. The fourth parameter, <dwell_time>, is required and establishes the duration of the test. The fifth parameter, <rise_time>, is optional and establishes the rise time for the voltage/current exposure. The sixth parameter, <fall_time>, is optional and establishes the fall time conditions.

A ROUTES Section Example

```
$ROUTES
  USEDIN: "BD11357", "BD66540";
    ROUTE: "CPU_PAGE_12_U23_3", ICT;
      PATH: "C1", 20, 7;
        LINE: (1200, 1420), (1200, 1600);
        LINE: (1200, 1600), (1320, 1600);
        LINE: (1320, 1600), (1320, 1815);
        LINE: (1320, 1815), (1460, 1815);
        VIA: "TV1", "CS1COMMON", , NOPROBE, (1460, 1815);
    ROUTE: "C3","HCLK", CLK, 750, 50;
      PATH: 10, 3;
        LINE: 1300, 1400, 1200, 2200;
        TESTPAD: "p106_2", "PS14", 1200, 2200, 0, MIRROR;
    ROUTE: GND;
```

```
        PATH: "GND1", 20, 7;
        PLANE: "shield_gnd";
          POLYGON: "bottom_irregular_plane";
        COMPPIN: "ic2", "7", 9045, 1775;
    $ENDROUTES
```

4.8.20 TESTCONNECTS

The TESTCONNECTS section is optional and can be used to define the tester resource connections between the board, optional fixture-electronics, and the test system. The placement of the test probes can be done manually or automatically, and can be on component pins, connector pins, test pads or vias. The TESTCONNECTS section has the following form.

```
TESTCONNECTS ::= $TESTCONNECTS
             :> 1{
                  USEDIN: <usedin_ref> 0{,<usedin_ref>}n;
             :>   1{
                    TESTCONNECT: <net_ref>, [<group_name>], [<group_type>];
             :>     {
                      0{TESTPIN: <testpin_id>, [<testconnect_id>];
             :>         0{
                            LOCATION: <layer_ref>, [<xy_ref>];
             <:           }n
                        }n
                      & 0{FIXELEC: <fixelec_signal_name>, [<xy_ref>]; }n
                      & 0{
                           TESTPROBE: <testprobe_name>, <probe_type>, <probe_size>,
                             <spring>, <recept_depth>, <layer_ref>, [<tip_type>],
                             [<probe_attach>];
                         }n
                      & 0{VIA:<via_ref>, [<testpoint_xy>];}n
                      & 0{TESTPAD: <testpad_ref>, [<testpoint_xy>];}n
                      & 0{COMPPIN: <component_ref>, [<testpoint_xy>];}n
                      & 0{CONPIN: <connector_ref>, [<testpoint_xy>];}n
             <:     }
             <:   }n
             <: }n
                $ENDTESTCONNECTS
```

<usedin_ref>	::=	string - a $HEADER fixture(s), panel(s), board(s), or assembly(ies)
<net_ref>	::=	string
<group_name>	::=	string
<group_type>	::=	{TWISTED | CABLED | BUSSED}1
<testpin_id>	::=	string -Tester assigned resource pin reference number.
<testconnect_id>	::=	string
<xy_ref>	::=	<x_offset>, <y_offset>
<fixelec_signal_name>	::=	string
<testprobe_name>	::=	{string | NOTASSIGNED}1 - Tester assigned probe reference - use NOTASSIGNED to indicate the test probe reference is not assigned
<probe_type>	::=	{ICT | DUALSTAGE | TRANSFER}1
<probe_size>	::=	p_number
<spring>	::=	string

<recept_depth>	::=	p_number
<layer_ref>	::=	{<layersingle_ref> | <layerset_ref> | <surface>}1
<layersingle_ref>	::=	string
<layerset_ref>	::=	string
<surface>	::=	{TOP | BOTTOM | BOTH}1
<tip_type>	::=	{SPEAR | CHISEL | CROWN | TULIP4 | TULIP3 | CASTLE | RADIUS | OTHER}1
<probe_attach>	::=	{CAPACITIVE | INDUCTIVE | MATING | OTHER}1
<via_ref>	::=	string
<testpoint_xy>	::=	<xy_ref>
<testpad_ref>	::=	string
<component_ref>	::=	string
<connector_ref>	::=	string

The $TESTCONNECTS and $ENDTESTCONNECTS section keywords are used to mark the TESTCONNECTS section in the GenCAM file.

4.8.20.1 USEDIN

The USEDIN statement associates the statements that follow with the <usedin_name> of a board(s), panel(s), assembly(ies) or fixture(s) as defined in the HEADER section of the GenCAM file.

4.8.20.2 TESTCONNECT

The TESTCONNECT statement is required and is used to define a connection between a test probe and a tester resource pin, or between either of these and a fixture-electronics signal. Fixture-electronics is used here to describe any electronic circuitry which is neither part of the test system nor part of the board under test, but which is connected to one or the other of these. The first parameter, <net_ref>, is required and **shall** reference a <net_name> located in the ROUTES section on the board, panel, or assembly under test. The second parameter, <group_name>, is an optional string and is used to name a connection grouping. All TESTCONNECTS referencing the same grouping name will be associated in the manner prescribed by the grouping type. The third parameter, <group_type>, is optional and fixed field names are used to define a connection grouping type as either TWISTED, CABLED or BUSSED. A TWISTED type grouping must be referenced in exactly two (2) TESTCONNECT statements, each referencing a different signal name. CABLED and BUSSED type grouping must be referenced in at least two (2) TESTCONNECT statements, each referencing a different signal name.

4.8.20.3 TESTPIN

The TESTPIN statement is optional and establishes a unique name for the pin and correlates that to the tester. The pin name may be the Tester Assigned number or the Tester Interface number. The first parameter, <testpin_id>, is required and is a string that represents the name of the test pin. The second parameter, <testconnect_id>, is an optional parameter and is a string that establishes the identification or location (absolute coordinates) of the tester interface.

4.8.20.4 LOCATION

The LOCATION statement is optional and immediately follows the TESTPIN statement. There **shall** be a separate LOCATION statement for each layer of a fixture starting with the layer closer to the tester and ending with the layer closest to the board, panel, or assembly under test. The first parameter, <layer_ref>, is required and is a layer of the fixture as identified in the LAYERS section of the GenCAM file. The second parameter, <xy_ref>, is optional and is a reference to the exact location of the HOLE in the fixture through which the testpin will pass.

4.8.20.5 FIXELEC

The FIXELEC statement is optional and is used to define a fixture-electronics terminal. The first parameter, <fixelec_signal_name>, is required and is used to define a unique fixture-electronics signal name. The second parameter, <xy_ref>, is optional and is used to define the absolute coordinates of the fixture electronics terminal in reference to the tooling origin of the fixture.

4.8.20.6 TESTPROBE

The TESTPROBE statement is optional and is used to establish the characteristics for the test probe. The first parameter, <testprobe_name>, is required and **shall** be a unique name for the probe. The name can be a string or the fixed field, NOTASSIGNED, to indicate the test probe reference is not assigned. The second parameter, <probe_type>, is required and **shall** use a fixed field statement to identify either ICT, DUALSTAGE, or TRANSFER probes. The third parameter, <probe_size>, is required and **shall** be a number in terms of the UNITS identified in the HEADER that establishes the probe tip maximum diameter. The fourth parameter, <spring>, is required and is a freeform string that establishes the receptacle spring force in terms of ounces or grams as appropriate. The fifth parameter, <recept_depth>, is required and is a dimensional value indicating the recess (negative value) or raise (positive value) to be applied to the receptacle insertion in the units of measure defined in the HEADER section. The sixth parameter, <layer_ref>, is required and **shall** identify the fixture probe plate into which the probe is inserted as one of TOP or BOTTOM. The seventh parameter, <tip_type>, is an optional fixed field parameter indicating the probe type for ICT or DUALSTAGE probes. The fixed field identifiers are SPEAR, CHISEL, CROWN, TULIP4, TULIP3, CASTLE, RADIUS, or OTHER. The eighth parameter, <probe_attach>, is optional and is a fixed field that establishes the manner in which the probe is connected to the tester. The fixed field identifiers are CAPACITIVE, INDUCTIVE, MATING, or OTHER.

4.8.20.7 VIA

The VIA statement is optional and is used to refer to a via on a board, panel, or assembly intended to be accessed for test. The first parameter, <via_ref>, is required and references a via name established in the ROUTES section. The second parameter, <testpoint_xy>, is optional and is used to define the absolute xy coordinates where the via should be probed.

4.8.20.8 TESTPAD

The TESTPAD statement is optional and is used to refer to a testpad on a board, panel, or assembly intended to be accessed for test. The first parameter, <testpad_ref>, is required and references a testpad name established in the ROUTES section. The second parameter, <testpoint_xy>, is optional and is used to define the absolute xy coordinates where the testpad should be probed.

4.8.20.9 COMPPIN

The COMPPIN statement is optional and is used to refer to a component pin on an assembly intended to be accessed for test. The first parameter, <component_ref>, is required and references a unique component name, <ref_desig>, established in the COMPONENTS section. The second parameter, <testpoint_xy>, is optional and is used to define the absolute xy coordinates of the pin, <pin_physical_ref>, identified in the PACKAGES section for the particular component.

4.8.20.10 CONPIN

The CONPIN statement is optional and is used to refer to a connector pin on a board, panel, or assembly intended to be accessed for test. The first parameter, <connector_ref>, is required and references a unique connector name, <ref_desig>, established in the COMPONENTS section. The second parameter, <testpoint_xy>, is optional and is used to define the absolute xy coordinates of the pin, <pin_physical_ref>, identified in the PACKAGES section for the particular connector.

A TESTCONNECTS Section Example

```
$TESTCONNECTS
  USEDIN: "ICTFX-66540";
    TESTCONNECT: "ENABLE_BAR";
      TESTPIN: "C2R1", 7767;
        LOCATION: "fixture_top", (2.3, 16.4);
        LOCATION: "fixture_bottom", (2.5, 16.1);
      TESTPROBE: "P1", ICT, 100, "8oz", 0.250, BOTTOM, SPEAR;
      COMPPIN: "U134", (3.0, 4.6);
    TESTCONNECT: "HS_RECEIVE";
      TESTPIN: "C4R6", 300;
        LOCATION: "fixture_top", (2.8, 17.3);
        LOCATION: "fixture_bottom", (2.6, 17.1);
      TESTPROBE:"P2", ICT, 75, "8oz", 0.25, BOTTOM, CROWN;
      VIA: "TV16", (4.9, 6.3);
    TESTCONNECT: "RN_COMP";
      FIXELEC: "FN02";
      TESTPROBE: "J3-1", ICT, 100, "8oz", 0.00, BOTTOM, CROWN;
    TESTCONNECT: "CLK_DIV";
      TESTPIN: "C02R06", (1550, 14250);
        LOCATION: "fixture_top", (6.3, 14.4);
        LOCATION: "fixture_bottom", (6.5, 14.1);
      FIXELEC: "FN01";
    TESTCONNECT: "U3TJ1";
      TESTPIN:"C78R23", (30198, -7767);
      TESTPROBE: "TJ1", ICT, 100, "8oz", 0.00, TOP, , CAPACITIVE;
$ENDTESTCONNECTS
```

4.8.21 CHANGES

The CHANGES section can be used to define and track a sequential list of changes to the main GenCAM file. A file of changes **shall** be a separate file, and **shall** consist of only a HEADER section and a CHANGES section that defines the details of various sections in which changes occur. It is assumed that all of the changes have been or will be implemented in the main GenCAM file. Changes are identified by name and all parameters tied to that name are also changed. The CHANGES section can have some or all of the keywords listed below.

```
CHANGES     ::=  $CHANGES
            :>   1{
                   USEDIN: <usedin_ref> 0{,<usedin_ref>}n;
            :>     1{
                     CHANGE: <change_name>, <section_name>, <action>,
                       [<history_increment>];}n
            :>       1{
                       <statement_keyword>: <old_parameter_list>;}n
                     1{
                       <statement_keyword>: <new_parameter_list>;}n
                     0{COMMENT: <comment>;}n
            <:     }n
            <:   }n
            <:   }n
                 $ENDCHANGES

<usedin_ref>     ::=   string  - a $HEADER fixture(s), panel(s), board(s) or assembly(ies)
```

<usedin_ref>	::=	string
<change_name>	::=	string - an identifier for the change i.e. "SAM1"
<section_name>	::=	{HEADER | ADMINISTRATION | FIXTURES | PANELS | BOARDS | DRAWINGS | PRIMITIVES | ARTWORKS | PATTERNS | MECHANICALS | LAYERS | PADSTACKS | PACKAGES | FAMILIES | DEVICES | COMPONENTS | POWER | ROUTES | TESTCONNECTS}1
<action>	::=	{ADD | DELETE | REPLACE}1
<history_increment>	::=	p_integer
<statement_keyword>	::=	fixed field string -- only the keyword statements (Appendix B) from the section specified in the CHANGE statement are allowed.
<old_parameter_list>	::=	acceptable parameter value
<new_parameter_list>	::=	acceptable parameter value
<comment>	::=	string

The $CHANGES and $ENDCHANGES section keywords mark the CHANGES section of the GenCAM file. Each change description **shall** start with the CHANGE keyword.

4.8.21.1 USEDIN

The USEDIN statement associates the statements that follow with the <usedin_name> of a board(s), panel(s), assembly(ies) or fixture(s) as defined in the HEADER section of the GenCAM file.

4.8.21.2 CHANGE

The CHANGE statement can be used to add a single change or multiple changes to the GenCAM file. The first parameter, <change_name>, is required. It is a freeform string that serves to uniquely name the change. The second parameter, <section_name>, is required. It is the beginning section keyword (see Appendix A) identifying the section in which the change occurs. The third parameter, <action>, is a fixed field, and **shall** be one of ADD, DELETE or REPLACE, describing the nature of the change. The fourth parameter, <history_increment>, is optional. It can provide a history increment number that serves to group changes, and to provide an index for future reference.

4.8.21.3 STATEMENT

The *Statement,* <statement_keyword>, serves to identify the keyword statement(s) to which the changes applies; only the official GenCAM keyword statements identified in the standard are permissible. When the change type is ADD, only the line(s) to be added, with all their appropriate parameters, is required. This may be a single keyword statement or multiple keyword statements. When the change type is DELETE, only the single line, or multiple lines whose keyword statement and parameters match exactly, an existing line(s) is required. When the change type is REPLACE, the first keyword statement and all its associated keyword statement(s), are expected to match an existing line or line set (i.e. POLYGON) of the GenCAM section. The following statement is required to indicate the new information to replace that which is being removed.

4.8.21.4 COMMENT

The optional COMMENT statement can be used to add other information that is considered relevant to the CHANGE. The only parameter is a single quoted string, though the comment string can span multiple lines.

4.8.21.5 Changes to the HEADER Section

Some of the HEADER statement parameters can be changed by indicating the <section_name> HEADER in the CHANGES section. The syntax must be the same as that described in the original HEADER

section. The GenCAM statement should not be changed unless the GenCAM specification changes are reflected in the file. The GENERATEDBY statement may be changed through a REPLACE action; the parameters of <software_package> and <software_rev> are required.

ASSEMBLY, BOARD, PANEL, and FIXTURE should not be changed except for the REVISION parameter. The keyword statement would maintain all parameters except that a new REVISION would replace the old one. The UNITS and the ANGLEUNITS statement can be changed through a REPLACE action, however there may be rounding errors on dimensions. The HISTORY statement cannot be changed in the CHANGES section. Only the host system can update this parameter, however the <history_increment> must match the original file.

4.8.21.6 Changes to the ADMINISTRATION Section

New information may be added to the ADMINISTRATION Section for any of the keyword statements contained therein Information should be defined as ADD or REPLACE actions; DELETE actions are only appropriate for superfluous characteristics. Since the TRANSACTION keyword statement defines the transaction as a PO, RFQ, RFP, RFA, or CO, the CO **shall** be used if changing the details of an existing GenCAM file.

4.8.21.7 Changes to a FIXTURES Section

New information may be added to the FIXTURES section for any of the keyword statements contained therein. This may be through an ADD, DELETE, or REPLACE <action>. A fixture name can be changed through reference of the <usedin_ref> parameter of the FIXTURE statement. FIXTURE parameters can also be changed or removed from the FIXTURES section by redefining the entire FIXTURE, using the same <fixture_name>, in a REPLACE <action>.

4.8.21.8 Changes to a PANEL Section

A PANEL can be added to a FIXTURE by defining the PANEL, using a unique <panel_name>, in an ADD <action> of the CHANGES section and then adding a reference to the panel in the FIXTURE Section using an ADD <action>. New information may be added to the PANELS section for any of the keyword statements contained therein. This may be through an ADD, DELETE, or REPLACE <action>. A panel name can be changed through reference of the <usedin_ref> parameter of the PANEL statement. PANEL parameters can also be changed or removed from the PANELS section by redefining the entire PANEL, using the same <panel_name>, in a REPLACE <action>.

4.8.21.9 Changes to the BOARD Section

A BOARD can be added to a PANEL or directly to a FIXTURE by defining the BOARD, using a unique <board_name>, in an ADD <action> of the CHANGES section and then adding a reference to the board in the PANEL or FIXTURE sections using an ADD <action>. New information may be added to the BOARDS section for any of the keyword statements contained therein. This may be through an ADD, DELETE, or REPLACE <action>. A board name can be changed through reference of the <usedin_ref> parameter of the BOARD statement. BOARD parameters can also be changed or removed from the BOARDS section by redefining the entire BOARD, using the same <board_name>, in a REPLACE <action>.

4.8.21.10 Changes to the DRAWINGS Section

New information may be added to the DRAWINGS section for any of the keyword statements contained therein. New drawings can be defined using a unique <drawing_name> and descriptions in a DRAWING block using an ADD <action>. Existing drawing descriptions can also be changed by redefining the drawing, using the same <drawing_name>, with a REPLACE <action>. Drawings can be deleted by entering a DRAWING statement and a DELETE <action> and referencing the existing <drawing_name>. Referencing the <drawing_name> will delete the optional <url> and <drawing_format>, if they exist.

4.8.21.11 Changes to the PRIMITIVES Section

New shape primitives can be added using a unique <primitive_name> in a PRIMITIVE definition using an ADD <action>. Parameters of an existing primitive can only be changed by redefining the entire shape, using the original <primitive_name>, and a REPLACE <action>. Primitive definitions can be deleted by entering a PRIMITIVE statement and a DELETE <action> and referencing the existing <primitive_name>. All GenCAM sections that reference a PRIMITIVE that is to be deleted must be updated, removing the reference before the PRIMITIVE definition is removed.

4.8.21.12 Changes to the ARTWORKS Section

New artworks can be added using a unique <artwork_name> in an ARTWORK definition using an ADD <action>. Parameters of an existing artwork can only be changed by redefining the entire artwork, using the same <artwork_name>, and a REPLACE clause <action>. Artwork definitions can be deleted by entering an ARTWORK statement and a DELETE <action> and referencing the existing <artwork_name>. All GenCAM sections that reference an ARTWORK that is to be deleted must be updated, removing the reference before the ARTWORK definition is removed.

4.8.21.13 Changes to the PATTERNS Section

New pattern descriptions can be added using a unique <pattern_name> in a PATTERN definition using an ADD <action>. Parameters of an existing pattern can only be changed by redefining the entire pattern, using the same <pattern_name>,and a REPLACE <action>. Pattern definitions can be deleted by entering a PATTERN statement and a DELETE <action> and referencing the existing <pattern_name>. All GenCAM sections (COMPONENTS, DRAWINGS, ROUTES, etc.) that reference a PATTERN that is to be deleted must be updated, removing the reference before the PATTERN definition is removed.

4.8.21.14 Changes to the MECHANICALS Section

New MECHANICAL descriptions can be added by using a unique <part_name> in with an ADD <action>. Existing MECHANICAL descriptions can only be modified by completely redefining the part in a MECHANICAL block of a REPLACE <action> referencing the original <part_name>. MECHANICAL definitions can be deleted by entering the MECHANICAL statement and a DELETE <action> and referencing the original <part_name>. New HOLE definitions can be added by using a unique location and layers using an ADD <action>. Existing HOLE parameters can only be modified by redefining the HOLE using a REPLACE <action>, referencing the original location and layers. HOLE definitions can be removed by entering the HOLE statement and a DELETE <action> and referencing the original location and layers.

4.8.21.15 Changes to the LAYERS Section

New layers can be added using a unique LAYER or LAYERSET in a complete definition statement using an ADD <action>. Parameters of an existing layer or layer set can only be changed by redefining the entire layer, using the original information and a REPLACE <action>. The layers that make up a LAYERSET can only be changed by redefining the entire set, using the original <layer_set_name>, in a LAYERSET description using a REPLACE <action>. LAYERSET definitions can be deleted by entering a LAYERSET statement and a DELETE <action> and referencing the existing <layer_set_name>.

4.8.21.16 Changes to the PADSTACKS Section

New padstacks can be defined using a unique <padstack_name> and keyword descriptions in a PADSTACK block using an ADD <action>. Existing pad stack parameters can only be changed by redefining the whole pad stack description , using the same <padstack_name> and a REPLACE <action>. Pad stack definitions can be deleted by entering a PADSTACK statement and a DELETE <action> and referencing the existing <padstack_name>. All GenCAM sections (COMPONENTS, DRAWINGS,

ROUTES, PATTERNS, etc.) that reference a PADSTACK that is to be deleted must be updated, removing the reference before the PADSTACK definition is removed.

4.8.21.17 Changes to the PACKAGES Section

New package descriptions can be added using a unique <package_name> in a PACKAGE definition using an ADD <action>. Parameters of an existing package can only be changed by redefining the entire package, using the same <package_name>, and a REPLACE <action>. Package definitions can be deleted by entering a PACKAGE statement and a DELETE <action> and referencing the existing <package_name>. All COMPONENT or DRAWING definitions that reference a PACKAGE that is to be deleted must be updated, removing the reference before the PACKAGE definition is removed.

4.8.21.18 Changes to the FAMILIES Section

New logic families can be defined using a unique <family_name> and keyword descriptions in a FAMILY block using an ADD <action>. Existing logic family parameters can only be changed by redefining the FAMILY, using the same <family_name>, and a REPLACE <action>. Logic family definitions can be deleted by entering a FAMILY statement and a DELETE <action> and referencing the existing <family_name>. All DEVICE definitions that reference a FAMILY that is to be deleted must be updated, removing the reference before the FAMILY definition is removed.

4.8.21.19 Changes to the DEVICES Section

New device descriptions can be added using a unique <part_name> in a DEVICE definition using an ADD <action>. Parameters of an existing device can only be changed by redefining the entire device, using the same <part_name>, and a REPLACE <action>. Device definitions can be deleted by entering a DEVICE statement and a DELETE <action> and referencing the existing <part_name>. All COMPONENT definitions that reference a DEVICE that is to be deleted must be updated, removing the reference before the DEVICE.

4.8.21.20 Changes to the COMPONENTS Section

New component descriptions can be added using a unique <ref_desig> in a COMPONENT definition using an ADD <action>. The reference designator for a component can be changed ***globally*** by putting a COMPONENT statement pair and a REPLACE <action>. Parameters of an existing component can only be changed by redefining the entire component, using the same <ref_desig>, and a REPLACE <action>. Component definitions can be deleted by entering a COMPONENT statement and a DELETE <action>, and referencing the existing <ref_desig>. All ROUTE definitions that reference a COMPONENT that is to be deleted must be updated, removing the reference before the COMPONENT definition is removed.

4.8.21.21 Changes to the POWER Section

New power supply data can be added using a unique <supply_name> and description in an SUPPLY definition using an ADD <action>. Parameters of an existing supply can only be changed by redefining the SUPPLY, using the same <supply_name>, and a REPLACE <action>. Power supply definitions can be deleted by entering a SUPPLY statement and a DELETE <action> and referencing the original <supply_name>.

4.8.21.22 Changes to the ROUTES Section

New route descriptions can be added using a unique <net_name> and a ROUTE definition using an ADD <action>. The signal name for a route can be changed ***globally*** by indicating a ROUTE statement pair and a REPLACE <action>. Parameters of an existing route can only be changed by redefining the entire route, using the same <net_name>, and a REPLACE <action>. Routes can be deleted by entering a ROUTE statement and a DELETE <action> and referencing the existing <net_name>. This action causes all references in TESTCONNECTS to change.

4.8.21.23 Changes to the TESTCONNECTS Section

New test connections can be created using new or existing test pins test probes and fixture electronics terminals using an ADD <action>. Existing connections can only be changed by redefining the entire TESTCONNECT, using the same <net_name>, <testpin_id>, and <testprobe_name> and a REPLACE <action>. Test connections can be deleted by entering the entire TESTCONNECT description, using the same <net_name>, <testpin_id>, and <testprobe_name> and a DELETE <action>.

4.8.21.24 Changes to the CHANGES Section

The CHANGE sections themselves can NEVER be changed and must be completely redefined with a new <history_increment>.

A CHANGE Section Example

```
$CHANGES
  USEDIN: "4566-25";
    CHANGE:  "CoatingChange1", LAYERS, REPLACE;
      LAYER: "Coat1", COATINGCOND, "Tin Lead", 0.3, "SOLD1";
      LAYER: "Coat1", COATINGCOND, "Gold Lead", 0.3, "SOLD1";
      COMMENT: "customer decided to plate with gold";
    CHANGE:  "LayerSetChange1", LAYERS, REPLACE;
      MODIFIED: "HarryParkinson", 1998-07-03, "The wrong layer
                was specified";
      LAYERSET: "Is1";
        LAYER: "L2";
        LAYER: "Dielectric2";
        LAYER: "L3";
      LAYERSET: "Is1";
        LAYER: "L2";
        LAYER: "Dielectric2";
        LAYER: "L4";
$ENDCHANGES
```

5 MODELING

The data sections of GenCAM may be mapped to the information models. Information models are in development to ensure that complete mapping is capable between the information provided within GenCAM characteristics. The correlation is provided in the activity models shown in IPC-2519.

All data of GenCAM activities are based on activity models. The activity models covered by CAD and CAM include the engineering, design, administrative, and fabrication and assembly characteristics. Each of these sections are intended to be detailed into various levels of activity much like layers of information needed to perform a particular manufacturing process.

Figure 5-1 shows the activity needed to develop an electronic assembly. There are five major sections of activity. These include: Activity models start with the desire to develop an electronic assembly. That is the highest level of activity modeling and has, as it's input, the system requirements and the assembly partitioning and has as it's output, the activities involved for engineering, design, administration, fabrication, and assembly.

Figure 5-1 shows a preliminary relationship of the five activities necessary to develop the electronic assembly, provides some insight into the inputs, and the type of outputs that are necessary to clearly communicate the needs of the design to manufacturing disciplines. These instances of activities are further explored into different layers. Details are provided on each of the characteristics. These are then

used to perform interface relationships with the source of the information and the destination clearly defined in a fully documented activity model.

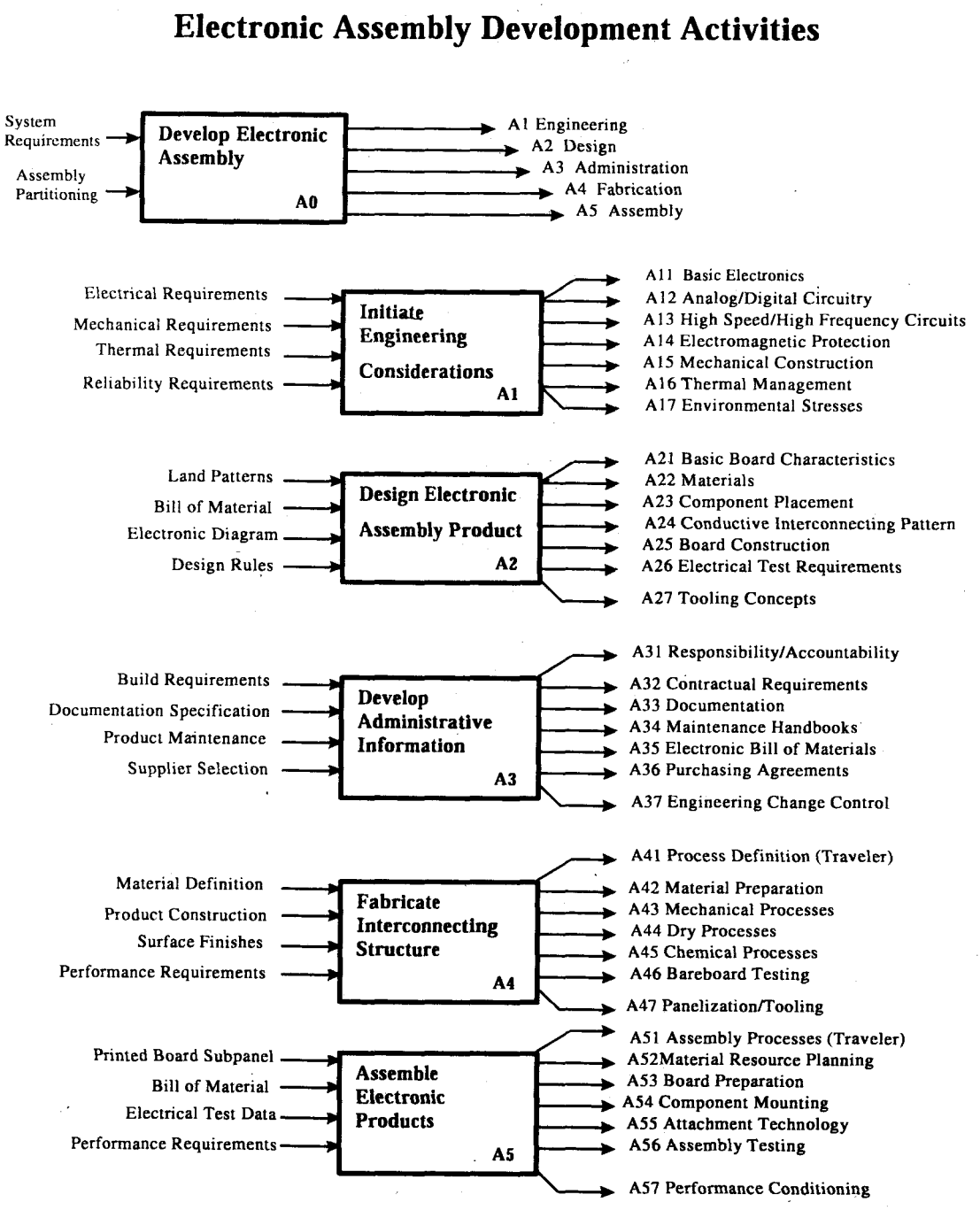


Figure 5-1 Relationship to GenCAM

The data files of GenCAM are mapped to information models. The effort ensures that complete mapping is capable between information models provided with the same characteristic as the flat ASCII file.

Design deals with design issues, manufacturing deals with manufacturing activities, nevertheless, the format is in ASCII text and includes total data necessary for product characterization.

5.1 Activity and Information Models

Activity and information models are contained in IPC-2519. Information is correlated to the attributes of GenCAM as well as to the activity models used to design, order and manufacture printed boards and printed board assemblies.

6 STANDARD SEMANTIC RULES

The following data format represent syntax rules, parameters or descriptions. These may be populated from information models or from CAM workstations. The syntax and semantics **shall** serve to create Compliance Test Modules (CTMs) for the data contain in the GenCAM file.

6.1 The GenCAM file

The GenCAM data set **shall** be contained within a single file. The GenCAM file can have any file name that the user chooses provided that it is acceptable to the operating system(s) being used. A GenCAM file **shall** have exactly one of each section as defined in the GenCAM standard. If a section is not necessary, the section indicators are not required, however every GenCAM file **shall** have as a minimum, a $HEADER section and a $ADMINISTRATION section.

It is understood that GenCAM files are a product of the data that they intend to support. The sectional standards of the IPC-2510 series (IPC-2512 through 2518) identify those sections of the GenCAM file that must be included at the creation of the file as well as the GenCAM sections that **shall** be included when the file has gone through its manufacturing sequences. In each of the sectional standards, the letter "M" defines a mandatory requirement, the "O" is an optional requirement, and a dash signifies that the section is not necessary.

Figure 6-1 shows the GenCAM file structure. The simplest example is a file that includes only changes, thus must only have a $HEADER, $ADMINISTRATION, and $CHANGES section. It should be understood that when multiple fixtures, boards, panels, are identified within a single GenCAM file, they **shall** be segmented by the "USEDIN" parameter, clearly identified which keywords and which parameters relate to the particular USEDIN name assigned in the HEADER section.

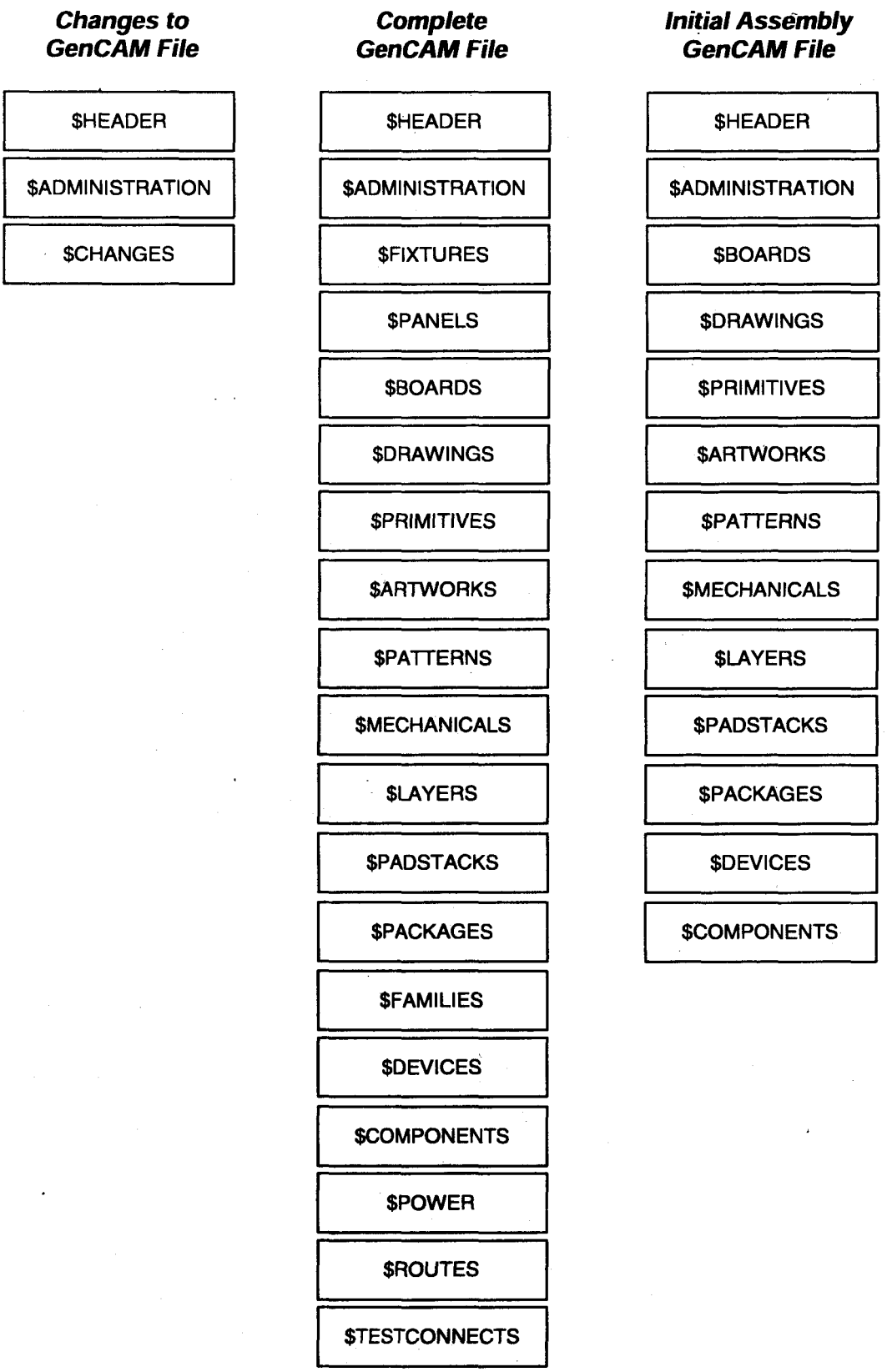


Figure 6-1 The GenCAM Files [Change File, Complete File, Initial Assembly File]

6.2 Alpha-numeric Parameters

string ::= SPACE (ASCII decimal 32) to ~ (ASCII decimal 126) inclusive
This is the default character set and its use is assumed, unless another has been declared using the CHARACTERSET statement in the HEADER section of the GenCAM file.

number ::= $3.4x10^{-38} \leq$ value $\leq 3.4x10^{38}$
where value can be positive, negative, integer or floating point, with 7 digit precision. Numbers are assumed to be positive but can be explicitly designated as positive by a preceding + (ASCII decimal 43) character. Negative numbers must be explicitly designated as negative by a preceding - (ASCII decimal 45) character.
Exponentiation format **shall** be [±][0-9]*.[0-9]*[Ee][0-9]*

p_number ::= $0.0 \leq$ value $\leq 3.4x10^{38}$

p_integer ::= 1{0 1 2 3 4 5 6 7 8 9}n

date_format ::= yyyy-mm-dd (numeric) - yyyy = year, mm = month, dd = day i.e., 1997-02-23

6.3 GenCAM Parameter Units

<angular_unit> ::= {RADIANS | DEGREES}1
<angular_measure> ::= { <radian_value> | <degree_value> }1
<radian_value> ::= 0.0 <= number <= 2π
<degree_value> ::= 0.0 <= number <= 360.0

<dimension> ::= {
MM - Millimeter(s)
| UM - Micrometer(s)
| MM100 - hundredths of millimeter
| INCH - Inch
| THOU - Inch/1,000
| THOU10 - Inch/10,000
| USERCM - Units to the Centimeter
| USERMM - Units to the Millimeter
| USERUM - Units to the Micrometer
| USERINCH - Units to the Inch
}1

<grid_value> ::= p_integer - A dimensional value to determine the grid system

6.4 Description of GenCAM Standard Modifiers

<xform> ::= <xy_ref>, [<rotation>], [<mirror>], [<scale_factor>]
<xy_ref> ::= (<x_offset>, <y_offset>)
<rotation> ::= <angular_measure>
<mirror> ::= MIRROR - indicates that all x-values go to minus x-values.
<scale_factor> ::= p_number - Scale multiplies all dimensions proportionally.
<x_offset> ::= number
<y_offset> ::= number

<location> ::= <xy_ref>, [<rotation>], [<mirror>]
<xy_ref> ::= (<x_offset>, <y_offset>)
<rotation> ::= <angular_measure>
<mirror> ::= MIRROR - indicates that all x-values go to minus x-values.
<x_offset> ::= number
<y_offset> ::= number

<color_ref> ::= COLOR: <color_name>, <R>, <G>, <B>;
<color_name> ::= string
<R> ::= p_integer – R, G, B values must be in the range 0-255
<G> ::= p_integer
<B> ::= p_integer

<attribute_def> ::= ATTRIBUTE: <attribute_group>, <attribute_type>, <attribute_value>;
may be used to provide additional information or reference to external data attached to the most immediate prior key word.
<attribute_group> ::= string
<attribute_type> ::= string
<attribute_value> ::= string

<modified_def> ::= MODIFIED: <person_ref>, <mod_date>, <mod_text>;
may be used to provide notification of changes
made to data contained in the previous key word.
<person_ref> ::= string - person identified in the ADMINISTRATION section
<mod_date> ::= date_format
<mod_text> ::= string

<audit_def> ::= AUDIT: <audit_type>, <person_ref>, <point_loss>, [<explanation>];
may be used to evaluate quality, completeness, design rule adherence, or DfM principles.
<audit_type> ::= {BDFAB | BDTST | BDASM | ASEMT}1
<person_ref> ::= string - identified in the ADMINISTRATION Section
<point_loss> ::= p_number - must be a value between 1 and 10
<explanation> ::= string

6.5 Description of GenCAM Standard Graphic Primitives

The GenCAM Standard Graphic Primitives are line and arc segments, and are located by the x-y points of their definitions.

<line_def> ::= LINE: <start_xy>, <end_xy>;

<circular_arc_def> ::= CIRCARC: <start_xy>, <end_xy>, , [<direction>];
<start_xy> ::= <xy_ref>
<end_xy> ::= <xy_ref>
 ::= <xy_ref>
<direction> ::= {CLKW | CCLKW}1 - Default is CCLKW, counterclockwise

<ellipse_arc_def> ::= ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>, [<direction>];
<start_xy> ::= <xy_ref>
<end_xy> ::= <xy_ref>
<focus1> ::= <xy_ref>
<focus2> ::= <xy_ref>

<direction> ::= {CLKW | CCLKW}1 - Default is CCLKW, counterclockwise

<text> ::= TEXT: <text_string>, <p1>, <p2>, [<font_ref>], [<color_ref>], <xform>;

<text_string> ::= string
<p1> ::= <xy_ref> - p1 and p2 define the bounding box of the text string
<p2> ::= <xy_ref>
<font_ref> ::= string - references a standard or user-defined font (see UFONT). When not specified, the font default may be any font acceptable for the intended purpose
<color_ref> ::= COLOR: <color_name>, <R>, <G>, <B>;

6.6 Description of GenCAM Standard Simple Primitives

GenCAM Standard Simple Primitives are common, closed shapes. Unlike the GenCAM Standard Graphic Primitives, their definitions do not explicitly define their location.

<circle_def> ::= CIRCLE: <circle_name>, <diameter>;
<circle_name> ::= string
<diameter> ::= p_number – dimension values for all primitives can be any convenient value. The actual size of a feature will be determined by applying the scale value of its transform to the defined dimension.

<rectangle_corner_def> ::= RECTCORNER: <rectangle_name>, <p1>, <p2>;
<rectangle_name> ::= string
<p1> ::= <xy_ref> - The coordinates p1 and p2 define the corners of the rectangle where p1 is the lower left hand corner and p2 is the upper right hand corner.
<p2> ::= <xy_ref>

<rectangle_center_def> ::= RECTCENTER: <rectangle_name>, <x_dimension>, <y_dimension>;
<rectangle_name> ::= string
<x_dimension> ::= p_number - The <x_dimension> is the length of the two parallel horizontal planes of the rectangle.
<y_dimension> ::= p_number - The <y_dimension> are the two parallel sides of the rectangle where the sides are perpendicular to the horizontal planes (the instance may then be rotated to any angle).

6.7 Description of GenCAM Standard Complex Primitives

GenCAM defines a set of complex primitives that includes predefined shapes used for holes, lands, or thermal relief features.

<rectangle_chamfer_def>::= RECTCHAM:<rectcham_name>, <width>, <height>, <chamfer>;
The chamfer size is the length of one leg of the chamfer.
The chamfer is 45 degrees.
<rectcham_name> ::= string
<width> ::= p_number
<height> ::= p_number
<chamfer> ::= p_number

<rectangle_round_def> ::= RECTROUND: <rectround_name>, <width>, <height>, <radius>;

<rectround_name> ::= string
<width> ::= p_number
<height> ::= p_number
<radius> ::= p_number

<oval_def> ::= OVAL: <oval_name>, <width>, <height>;
<oval_name> ::= string
<width> ::= p_number
<height> ::= p_number

<dshape_def> ::= DSHAPE: <dshape_name>, <end_shape>, <orientation>, <width>, <height>, {<radius> | <chamfer>}1;

<dshape_name> ::= string
<end_shape> ::= {ROUND | CHAMFER | FILLET}1
<orientation> ::= {NORTH | SOUTH}1 - the default is NORTH
<width> ::= p_number
<radius> ::= p_number
<chamfer> ::= p_number

<diamond_def> ::= DIAMOND: <diamond_name>, <width>, <height>;
<diamond_name> ::= string
<width> ::= p_number
<height> ::= p_number

<hexagon_def> ::= HEXAGON: <hexagon_name>, <point_to_point>;
<hexagon_name> ::= string
<point_to_point> ::= p_number - The dimensions represent the distance across the points of the hexagon

<octagon_def> ::= OCTAGON: <octagon_name>, <point_to_point>;
<octagon_name> ::= string
<point_to_point> ::= p_number - The dimensions represent the distance across the points of the octagon

<thermal_def> ::= THERMAL: <thermal_name>, <therm_shape>, <outer_diameter>, <inner_diameter>, [<spoke_count>], [<spoke_width>], [<spoke_start_angle>], [<spoke_end_shape>];

<thermal_name> ::= string
<therm_shape> ::= {ROUND | SQUARE | OCTAGON}1
<outer_diameter> ::= p_number
<inner_diameter> ::= p_number
<spoke_width> ::= p_number
<spoke_start_angle> ::= <angular_measure>
<spoke_count> ::= p_integer - default is 0 (donut)
<spoke_end_shape> ::= {ROUND | SQUARE | PARALLEL}1

6.8 GenCAM Description of User Primitives

GenCAM permits the user to describe unique primitives that are reused in various sections of the GenCAM file. These include polygon (closed shapes), and polyline (non-closed shapes). User primitives may be constructed using any of the standard graphic primitives.

```
<polyline_def>      ::=  POLYLINE: <polyline_name>;
                    :>       1{
                                LINE: <start_xy>, <end_xy>;
                                | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>];
                                | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                                        [<direction>];
                    <:       }n

<polyline_name>     ::=  string

<polygon_def>       ::=  POLYGON: <polygon_name>;
                    :>       2{
                                LINE: <start_xy>, <end_xy>;
                                | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>];
                                | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                                        [<direction>];
                    <:       }n

<polygon_name>      ::=  string
```

6.9 GenCAM User Enhanced "In-Process" Primitives

GenCAM permits the user to define polylines and polygons that are part of a specific GenCAM section. The polyline and polygons are defined using the transform of the specific section. They may be named if they belong to another keyword, i.e. artwork, slot, path, keepout. The rules for "in-process" primitives are:

```
<polyline_builder>  ::= 1{
                           POLYLINE: <polyline_ref>, <linedesc_ref> , [<color_ref>], <xform>;
                         | POLYLINE: [<linedesc_ref>], [<color_ref>];
                    :>       1{
                             LINE: <start_xy>, <end_xy>, [<linedesc_ref>], [<color_ref>];
                              | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>],
                                  [<linedesc_ref>], [<color_ref>];
                              | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                                  [<direction>], [<linedesc_ref>], [<color_ref>];
                    <:       }n
                            }n

<polygon_builder> ::= 1{
                        POLYGON: <polygon_ref>, [<linedesc_ref>], [<paintdesc_ref>],
                                [<color_ref>], <xform>;
                      | POLYGON: [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>];
                    :>       2{
                             LINE: <start_xy>, <end_xy>, [<linedesc_ref>], [<color_ref>];
                              | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>],
                                  [<linedesc_ref>], [<color_ref>];
                              | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                                  [<direction>], [<linedesc_ref>], [<color_ref>];
```

<: }n
}n

<shape_builder> ::= {
LINE: <start_xy>, <end_xy>, [<linedesc_ref>], [<color_ref>];

| CIRCARC: <start_xy>, <end_xy>, , [<direction>], [<linedesc_ref>],
[<color_ref>];

| ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>, [<direction>],
[<linedesc_ref>], [<color_ref>];

| CIRCLE: <diameter>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| CIRCLE: <circle_ref>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| RECTCENTER: <x_dimension>, <y_dimension>, [<linedesc_ref>],
[<paintdesc_ref>], [<color_ref>], <xform>;

| RECTCENTER: <rectangle_ref>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| RECTCORNER: <p1>, <p2>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| RECTCORNER: <rectangle_ref>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| RECTCHAM: <width>, <height>, <chamfer>, [<linedesc_ref>],
[<paintdesc_ref>], [<color_ref>], <xform>;

| RECTCHAM: <rectcham_ref>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| RECTROUND: <width>, <height>, <radius>,
[<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| RECTROUND: <rectround_ref>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| OVAL: <width>, <height>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>],<xform>;

| OVAL: <oval_ref>, [<linedesc_ref>], [<paintdesc_ref>],
[<color_ref>], <xform>;

| DSHAPE: <end_shape>, <orientation>, <width>, <height>,
{<radius> | <chamfer>}1, [<linedesc_ref>],
[<paintdesc_ref>], [<color_ref>], <xform>;

```
                | DSHAPE: <dshape_ref>, [<linedesc_ref>], [<paintdesc_ref>],
                                [<color_ref>], <xform>;

                | DIAMOND: <width>, <height>, [<linedesc_ref>], [<paintdesc_ref>],
                                [<color_ref>], <xform>;

                | DIAMOND: <diamond_ref>, [<linedesc_ref>], [<paintdesc_ref>],
                                [<color_ref>], <xform>;

                | HEXAGON: <point_to_point>, [<linedesc_ref>], [<paintdesc_ref>],
                                [<color_ref>], <xform>;

                | HEXAGON: <hexagon_ref>, [<linedesc_ref>], [<paintdesc_ref>],
                                [<color_ref>], <xform>;

                | OCTAGON: <point_to_point>, [<linedesc_ref>], [<paintdesc_ref>],
                                [<color_ref>], <xform>;

                | OCTAGON: <octagon_ref>, [<linedesc_ref>], [<paintdesc_ref>],
                                [<color_ref>], <xform>;

                | POLYGON: <polygon_ref>, [<linedesc_ref>], [<paintdesc_ref>],
                                [<color_ref>], <xform>;

                |POLYGON: [<linedesc_ref>], [<paintdesc_ref>], [<color_ref>];
                :>              2{
                                LINE: <start_xy>, <end_xy>, [<linedesc_ref>], [<color_ref>];
                                | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>],
                                    [<linedesc_ref>], [<color_ref>];
                                | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                                    [<direction>], [<linedesc_ref>], [<color_ref>];
                <:              }n

                | POLYLINE: <polyline_ref>, [<linedesc_ref>], [<color_ref>], <xform>;

                | POLYLINE: [<linedesc_ref>], [<color_ref>];
                :>              1{
                                LINE: <start_xy>, <end_xy>, [<linedesc_ref>], [<color_ref>];
                                | CIRCARC: <start_xy>, <end_xy>, <center>, [<direction>],
                                    [<linedesc_ref>], [<color_ref>];
                                | ELLIPARC: <start_xy>, <end_xy>, <focus1>, <focus2>,
                                    [<direction>], [<linedesc_ref>], [<color_ref>];
                <:              }n
                        }n

<line_desc_ref>         ::=     string
<paintdesc_ref>         ::=     string
<polyline_ref>          ::=     string
<polygon_ref>           ::=     string
<circle_ref>            ::=     string
<oval_ref>              ::=     string
<dshape_ref>            ::=     string
<diamond_ref>           ::=     string
<hexagon_ref>           ::=     string
```

<octagon_ref> ::= string
<rectangle_ref> ::= string
<rectcham_ref> ::= string
<rectround_ref> ::= string

6.10 Description of GenCAM Standard Primitive Enhancers

Several types of primitive enhancements have been identified for usage with primitives. These descriptors provide texture and substance to primitive features. Each one is assigned a unique name which is coupled with the use of the primitive as needed in the various sections of the GenCAM file. Primitive enhancement descriptors include line descriptions, paint descriptions, barrel descriptions, and profile descriptions.

<linedesc_def> ::= LINEDESC: <linedesc_name>, <line_width>, [<line_end>], [<color_ref>], [<line_type>], [<line_space>], [<line_length>], [<line_mod>], [<mod_end>], [<dim_A>], [<dim_B>], [<dim_C>]; Optional parameters not required for line description; definitions are ignored even though a value may appear.
<linedesc_name> ::= string
<line_width> ::= p_number
<line_end> ::= {NONE | ROUND | SQUARE}1 - The default is ROUND
<color_ref> ::= string - <color name> identified in $HEADER; the default is "black"
<line_type> ::= {SOLID | DOTTED | DASHED | CENTER | PHANTOM | ERASE}1 - The default is SOLID
<line_space> ::= p_number - Distance between center line of dots and/or start and end of lines,
<line_length> ::= p_number - Distance between start and end of dashes or lines. The diameter of a dot is equal to the line width.
<line_mod> ::= {TP | TD | SL1 | SL2 | NONE}1 - The default is NONE.
<mod_end> ::= {START | FINISH | BOTH}1
<dim_A> ::= p_number - Distance from end of line to finish of modification, or radius of secondary round feature. (see 4.1.2.4)
<dim_B> ::= p_number - Distance from end of line to extension of modification.
<dim_C> ::= p_number - Radius joining curvature of land to line center. (see Figure 4-3)

<paintdesc_def> ::= PAINTDESC: <paintdesc_name>, <paint_type>, [<color_ref>], [<line_width>], [<pitch1>], [<angle1>], [<pitch2>], [<angle2>]; Optional attributes not required for paint description; definitions are ignored even though a value may appear.

<paintdesc_name> ::= string
<paint_type> ::= {HOLLOW | HATCH | MESH | FILL | VOID}1 -The default is HOLLOW. VOID designates an opaque absence of FILL within or intersecting a FILL, HATCH, or MESH enclosure.
<color_ref> ::= string - <color name> identified in $HEADER; the default is BLACK or NONE if <paint_type> is VOID.
<line_width> ::= p_number - refers to line width for MESH or HATCH lines.
<pitch1> ::= p_number - Distance between line centers of HATCH or first set of MESH lines.
<angle1> ::= <degree_value>- (0 - 180) - 0 - 90 are only used with MESH paint type

<pitch2>	::=	p_number	- Distance between line center of HATCH or second set of MESH lines.
<angle2>	::=	<degree_value>	- (90 - 180) - second line set for MESH
<barreldesc_def>	::=	BARRELDESC: <barreldesc_name>, <barrel_type1>, <min_thickness1>, [<barrel_type2>], [<min_thickness2>], [<barrel_type3>], [<min_thickness3>], [<fill_material>];	
<barreldesc_name>	::=	string	
<barrel_type1>	::=	{HOLLOW | FILL | COAT | PLATE}1	- describes HOLE or WELL condition. The default is HOLLOW – no plating nor coating in hole.
<material1>	::=	string	- defines material type for the initial wall condition
<min_thickness1>	::=	p_number	- maximum thickness is controlled by PROFILEDESC of hole; at MMC is smallest hole
<barrel_type2>	::=	string	- describes metallic plating or coating
<material2>	::=	string	- defines material type for the subsequent wall condition
<min_thickness2>	::=	p_number	- maximum thickness is controlled by PROFILEDESC of hole; at MMC is smallest hole
<barrel_type3>	::=	string	- describes metallic plating or coating
<material3>	::=	string	- defines material type for the final wall condition
<min_thickness3>	::=	p_number	- maximum thickness is controlled by PROFILEDESC of hole; at MMC is smallest hole
<fill_material>	::=	string	- identification of metallic and non-metallic materials for filling the hole.
<profiledesc_ref>	::=	PROFILEDESC: <profiledesc_name>, <finished_LMC>, <finished_MMC>, [<start_LMC>], [<start_MMC>];	
<profiledesc_name>	::=	string	
<finished_LMC>	::=	p_number	- variation from nominal to least material condition.
<finished_MMC>	::=	p_number	- variation from nominal to maximum material condition.
<start_LMC>	::=	p_number	- variation from nominal to least material condition prior to the addition of plating or coating (see BARRELDESC).
<start_MMC>	::=	p_number	- variation from nominal to maximum material condition prior to the addition of plating or coating (see BARRELDESC).

7 REPORT GENERATORS

Each of the sections of the GenCAM format has various report generators that industry uses to provide the user with hard copy of the GenCAM data file. Some of them are preferred based on industry preferences, others are mainly examples. The detailed report generators are described in each of the seven sections of the sectional documents i.e. IPC-2512 - 2518 .

8 REFERENCE INFORMATION

The following sections define reference documents that are useful in clarifying the products or process of the industry or provide additional insight into the subject of data modeling or released information models.

8.1 IPC (1)

IPC-T-50	Terms and Definitions
IPC-D-275	Design Standard for Rigid Printed Boards and Rigid Printed Board Assemblies

IPC-D-300	Printed Board Dimensions and Tolerances
IPC-D-310	Guidelines for Artwork Generation and Measurement Techniques for Printed Circuits
IPC-D-325	Documentation Requirements for Printed Boards, Assemblies and Support Drawings

8.2 American National Standards Institute (2)

ANSI X3/TR-1-77	American National Dictionary for Information Processing
ANSI X3.12	Subroutine Record Format Standardization
ANSI Y14.5	Dimensioning and Tolerancing for Engineering Drawing
ANSI Y32.1	Logic Diagram Standards
ANSI Y32.16	Electrical and Electrical Reference Designators
ANSI Z210.1	Metric Practice Guide (ASTM 380-72)

8.3 Department of Defense (3)

DoD-STD-100	Engineering Drawings

8.4 Electronic Industries Association (4)

EDIF 4 0 0	Electronic Data Interchange Format

8.5 International Organization for Standards (ISO)

ISO STEP Documentation

ISO 10303-AP210	Electronic Assembly, Interconnect, and Packaging Design
ISO 10303-AP212	Electrotechnical Design & Installation
AP220	Process Planning, Manufacturing, and Assembly of Layered Electronic Products
AP221	Process Plant Functional Data & Schematic Representation

8.6 Internet Engineering Task Force (IETF) Standards

(ftp://NIS.NSF.NET/internet/documents/rfc)

RFC-1738	Berners-Lee, T., Masinter, L. and M. McCahill, "Uniform Resource Locators" (URL), RFC 1738, December 1994
RFC-2045	Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, December 1996
RFC-2046	Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, December 1996
RFC-2047	Moore, K., "Multipurpose Internet Mail Extensions (MIME)Part Three: Representation of Non-ASCII Text in Internet Message Headers", RFC 2047, December 1996
RFC-2048	Freed, N., Klensin, J. and J. Postel, "Multipurpose Internet Mail Extensions (MIME) Part Four: Mime Registration Procedures", RFC 2048, December 1996
RFC-2049	Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC 2049, December 1996

APPENDIX A

Beginning and Ending Section Keywords

Word	Definition	Usage	Page(s)
ADMINISTRATION	Start ADMINISTRATION section	GenCAM	52
ARTWORKS	Start ARTWORKS section	GenCAM	82
BOARDS	Start BOARDS section	GenCAM	69
CHANGES	Start CHANGES section	GenCAM	119
COMPONENTS	Start COMPONENTS section	GenCAM	107
DRAWINGS	Start DRAWINGS section	GenCAM	72
DEVICES	Start DEVICES section	GenCAM	102
ENDADMINISTRATION	End ADMINISTRATION section	GenCAM	55
ENDARTWORKS	End ADMINISTRATION section	GenCAM	82
ENDBOARDS	End BOARDS section	GenCAM	68
ENDCHANGES	End CHANGES section	GenCAM	119
ENDCOMPONENTS	End COMPONENTS section	GenCAM	107
ENDDEVICES	End DEVICES section	GenCAM	106
ENDDRAWINGS	End DRAWINGS section	GenCAM	71
ENDFAMILIES	End FAMILIES section	GenCAM	99
ENDFIXTURES	End FIXTURES section	GenCAM	57
ENDHEADER	End HEADER section	GenCAM	50
ENDLAYERS	End LAYERS section	GenCAM	90
ENDMECHANICALS	End MECHANICALS section	GenCAM	87
ENDPACKAGES	End PACKAGES section	GenCAM	95
ENDPADSTACKS	End PADSTACKS section	GenCAM	92
ENDPANELS	End PANELS section	GenCAM	61
ENDPATTERNS	End PATTERNS section	GenCAM	86
ENDPOWER	End POWER section	GenCAM	108
ENDPRIMITIVES	End PRIMITIVES section	GenCAM	75
ENDROUTES	End ROUTES section	GenCAM	109
ENDTESTCONNECTS	End TESTCONNECTS section	GenCAM	113
FAMILIES	Start FAMILIES section	GenCAM	98
FIXTURES	Start FIXTURES section	GenCAM	56
HEADER	Start HEADER section	GenCAM	49
LAYERS	Start LAYERS section	GenCAM	90
MECHANICALS	Start MECHANICALS section	GenCAM	87
PACKAGES	Start PACKAGES section	GenCAM	95
PADSTACKS	Start PADSTACKS section	GenCAM	92
PANELS	Start PANELS section		60
PATTERNS	Start PATTERNS section	GenCAM	83
POWER	Start POWER section	GenCAM	108
PRIMITIVES	Start PRIMITIVES section	GenCAM	74
ROUTES	Start ROUTES section	GenCAM	109
TESTCONNECTS	Start TESTCONNECTS section	GenCAM	113

APPENDIX B

Statement Keywords

Word	Definition	Usage	Page(s)
ACCEPT	Person in Receiving Dept.	Administration	53, 55
ALIAS	Alternate part number	Devices	87, 89
ANGLEUNITS	Angular measure	Header	50, 52
ARTWORK	Shape, Logo, Font, Target desc.	Artworks, Drawings	71
ASSEMBLY	Printed board assembly id	Header, Administration	49, 51
ATTRIBUTE	Ref to external information	General	52
AUDIT	Ratings system for GenCAM file	General	49
BAREBOARDTEST	Electrical testing	Boards	66, 68
BARRELDESC	Plating requirements for holes and other surfaces	Primitives	74, 76
BILLTO	Individual to pay for service	Administration	53, 55,
BOARD	Board description	Boards, Administration	65, 67
BODY	Outline of standard parts	Packages	95, 97
BUYER	Person resp. for order	Administration	53, 55
CHANGE	Identification of changes	Changes	117, 118
CHARACTERSET	ISO character reference	Header	49, 51
CIRCARC	Graphic primitive	Boards, Panels, Artworks, Drawings, etc	75
CIRCLE	Round primitive	Primitives, Artworks, Patterns	75
COLOR	Color intensity description	Header	50, 52
COMMENT	Text for additional information	Administration, Changes	53, 55
COMPPIN	Component pin identification	Routes, Testconnects	108, 111
COMPONENT	Component description	Components	104, 105
CONPIN	Connector pin identification	Routes, Testconnects	109, 112
CUSTOMERSERVICE	Person resp. for service	Administration, Header	53, 55
CUTOUT	Material removed definition	Boards, Panels, Fixtures	56, 58
DESC	Description	Devices	100, 104
DESIGNER	Person resp. for design	Administration	53, 55
DEVICE	Device identification	Components, Devices	100, 102
DIAMOND	Standard simple primitive	Primitives	75
DSHAPE	Complex primitive	Primitives, Artworks, Drawings, etc.	75
DRAWING	Drawing identification	Drawings	70, 72
DWGDATE	Drawing date	Drawings	70, 72
DWGNUMBER	Drawing number	Drawings	70, 72
DWGREF	Reference to external file (URL)	Drawings	71
DWGREV	Drawing revision	Drawings	70, 72
DWGSIZE	Drawing size	Drawings	70, 72
DWGTYPE	Drawing type	Drawings	70, 72
ELLIPARC	Graphic primitive	Boards, Primitives	75
ENGINEER	Engineer resp. for order	Administration	53, 55
FAMILY	Reference to family name	Devices	98, 99
FEATURE	Description of special features	Artworks	80, 81
FILTER	Removal of extraneous data from insert	Drawings	70, 72
FIXELEC	Special fixture electrical parameter	Testconnects	113, 115
FIXTURE	Fixture identification	Fixtures	56, 58
FRAME	Drawing format	Drawings	70, 72
GENCAM	Reference to format	Header	49, 51
GENERATEDBY	Software operating sys creating file	Header	51, 53
GLYPH	Identification of user font	Artworks	80, 82
GROOVE	Cutout in board or panel	Panels, Boards	62, 63

APPENDIX B

Statement Keywords

Word	Definition	Usage	Page(s)
HEIGHT	Distance of a part from surface	Mechanicals, Packages	87, 89
HEXAGON	Standard simple primitive	Primitives, Artworks, Patterns, Fixtures, Boards	75
HIGHPOTTEST	Bare board stress testing	Routes	109, 112
HISTORY	Increment of file modifications	Header	50, 52
HOLE	Instance of material removal	Mechanicals, Padstacks, Panels, Boards, etc.	56, 58
INSERT	Reference to pre-defined feature	Drawings	70, 73
KEEPOUT	Area where no access is permitted	Fixtures, Mechanicals, Components	105, 107
LAYER	Identification of single layer	Layers, Routes	91
LAYERSET	Multiple layers	Layers	90, 91
LAYERSINGLE	Individual layer	Layers	90, 91
LINE	Conductor or drawing lines and user constructions	Routes, Panels, Boards, Drawings, etc.	75
LINEDESC	Texture for line descriptions	Primitives	74, 77
LOAD	Output pins pulled up or down	Families	99
LOCATION	Testpin coordinates	Testconnects	113, 114
LOGIC	Logical voltage characteristics	Families	97, 99
LOGO	Identification of company	Artworks	66, 69
MECHANICAL	Non-electrical part descriptions	Mechanicals	87, 88
MODIFIED	Identification of person	General	48, 49
NET	Reference to route net names	Drawings	71, 73
NTOL	Negative tolerance	Devices	100, 104
OCTAGON	Standard simple primitive	Primitives, Artworks, Patterns	75
OHM			102, 103
ONFAIL	Related failure reports	Components	105, 108
OUTLINE	Periphery identification using graphic primitives	Fixtures, Boards, Panels, Primitives, Artworks, Patterns	56, 58
OVAL	Standard simple primitive	Primitives	75
PACKAGE	Component description	Devices, Packages	95, 97
PAD	Feature of padstack	Padstacks	92, 93
PADSTACK	Multiple pads in an array	Padstacks, Patterns	92, 93
PAINTDESC	Description of texture for closed primitives	Primitives	74, 76
PANEL	Identification and part number	Header, Panels	60, 62
PART	Reference to part description	Devices	100, 102
PATH	Conductor topology	Routes	109, 111
PATTERN	Combination of padstacks	Patterns	104, 106
PERSON	Individual name and address	Administration	53, 54
PINDESC	Identification of pin and description	Devices	100, 103
PIN	Outline of leads and terminations	Packages	97
PLANE	Conductive large area	Routes	109, 111
PLACEMENT	Position of boards within a fixture	Fixtures	61, 64
POLYGON	User-defined primitive	General	75
POLYLINE	User-defined unclosed primitive	General	75
PREPARATION	Component lead prep	Components	104, 107
PRESENTATION	Component delivery system	Components	104, 107
PROFILEDESC	Tolerance on any feature	General	74, 78

APPENDIX B

Statement Keywords

Word	Definition	Usage	Page(s)
PROGRAM	Programming of components	Components	104, 107
PTOL	Positive tolerance	Devices	100, 104
RECEIVED	Person receiving GenCAM output	Administration	53, 55
RECTCENTER	Standard simple primitive	Primitives, Artworks, Patterns	75
RECTCORNER	Standard simple primitive	Primitives, Artworks, Patterns	75
RECTCHAM	Standard simple primitive	Primitives, Artworks, Patterns	75
RECTROUND	Standard simple primitive	Primitives, Artworks, Patterns	75
RISETIME	Signal time to reach maximum	Families	99
ROUTE	Definition of conductor topology	Routes	109, 110
SCHEDULE	Requirements for receipt of GenCAM output	Administration	53, 55
SENT	Person resp. for sending file	Administration	53, 54
SHEET	Ref. to drawing sheet identification	Components, Drawings	70, 72
SLOT	Cutout in board or panel	Fixtures, Panels, Boards	57, 59
STANDOFF	Distance/clearance from board	Mechanicals, Packages	87, 88
SUPPLY	Power supply description	Power	108
SYMBOL	Identification of schematic/logic	Drawings, Patterns	70, 73
SYMPIN	Identification of pins on symbol	Drawings, Patterns	70, 73
TARGET	Complex primitive/Artwork	Primitives, Panels, Boards, Packages	79, 81
TESTPAD	Identification of point to be probed	Testconnects	109, 111
TESTPIN	Identification of pin in test fixture	Testconnects	113, 114
TESTPROBE	Test probe identifier	Testconnects	113, 115
TEXT	Description of words	Primitives	79, 82
THERMAL	Standard complex primitive	Primitives, Artworks, Patterns	75
THICKNESS	Definition of thickness characteristics	Boards, Panels	65
TOL	Tolerance on electrical parameters	Devices	103
TRANSACTION	Type of request using GenCAM file	Administration	53, 55
TYPE	Component device type	Devices	100, 103
UFONT	User font description	Artworks	80, 82
UNITS	Numerical characteristics	Header	50, 51
USEDIN	Identification of GenCAM products	General	70, 72
VALUE	Electrical value	Devices	100, 103
VIA	Plated through-hole	Routes, Testconnects	109, 111
VOLT	Electrical power	Devices	102, 103
WELL	Depression cut into product	Fixtures, Boards, Panels	60, 62

APPENDIX C

General Fixed Field Parameters

Word	Definition	Usage	Page(s)
A	U.S. drawing size	<drawing_size>	71
A0	Int'l drawing size	<drawing_size>	71
A1	Int'l drawing size	<drawing_size>	71
A2	Int'l drawing size	<drawing_size>	71
A3	Int'l drawing size	<drawing_size>	71
A4	Int'l drawing size	<drawing_size>	71
ADD	Addition to GenCAM file	<action>	117
ALIGNMENT	Purpose of target	<function>	80
AMP	Device value type	<value_type>	102
ASEMT	Audit of assembly test data Fixture application	<audit_type> <fixture_app>	49 57
ASSEMBLY	Type of drawing	<drawing_type>	71
AVAILABLE	Access availability for test probing	<access_desc>	110
B	U.S. drawing size	<drawing_size>	71
B	Blue color intensity	<color_name>	50
BDASM	Board assembly data audit Board assembly fixture application	<audit_type> <fixture_app>	49 57
BDFAB	Board fabrication data audit Board fabrication fixture application	<audit_type> <fixture_app>	49 57
BDTST	Board test data audit Board test fixture application	<audit_type> <fixture_app>	49 57
BLACK	Line color	<color_ref>	75
BOARD	Keepout description	<keepout_type>	105
BOTH	Surface identification Line modification	<surface> <mod_end>	114 75
BOTTOM	Pin exit description Surface identification	<pin_exit> <surface>	96 57
BULK	Presentation media	<delivery_system>	105
BUSSED	Type of test cable group	<group_type>	113
C	U.S. drawing size	<drawing_size>	71
CABLED	Grouping of test cable	<group_type>	113
CAPACITIVE	Capacitive layer Probe attachment	<GenCAM_layer_type> <probe_attach>	90 114
CARTRIDGE	Presentation media	<delivery_system>	105
CASTLE	Type of probe pin tip	<tip_type>	114
CCLKW	Counterclockwise	<direction> (ELLIPARC, CIRCARC)	126
CENTER	Type of line	<line_type>	75
CHAMFER	Definition of D-shape end	<end_shape>	77
CHISEL	Type of probe pin tip	<tip_type>	114
CLINCH	Lead clinching requirement	<comp_prep_type>	105
CLINCHLENGTH	Lead clinching length requirement	<comp_prep_type>	105
CLK	Clock	<net_class>	110
CLKW	Clockwise	<direction> (ELLIPARC, CIRCARC)	126
CO	Change order	<trans_type>	53
COAT	Conductive/non-conductive coating	<barrel_type1> <barrel_type2> <barrel_type3>	76 76 76
COATINGCOND	Conductive coating layer	<GenCAM_layer_type>	90
COATINGNONCOND	Non-conductive coating layer	<GenCAM_layer_type>	90
COMPONENT	Layer type Keepout description	<GenCAM_layer_type>, <keepout_type>	90
CONDUCTOR	Conductor image layer	<GenCAM_layer_type>	90

APPENDIX C

General Fixed Field Parameters

Word	Definition	Usage	Page(s)
CRITICAL	Electrical test probing	<access_desc>	110
CROWN	Type of probe pin tip	<tip_type>	114
D	U.S. drawing size	<drawing_size>	71
DASHED	Type of line	<line_type>	75
DEGREES	Increments of 360 segments	<angular_unit>	50
DELETE	Change action of GenCAM file	<action>	117
DETAIL	Drawing type	<drawing_type>	71
DIELADHV	Non-conductive adhesive	<GenCAM_layer_type>	90
DIELBASE	Dielectric base material	<GenCAM_layer_type>	90
DIELCORE	Composite of dielectric materials	<GenCAM_layer_type>	90
DIELPREG	Resin pre-impregnated reinforcement	<GenCAM_layer_type>	90
DOTTED	Type of line	<line_type>	75
DOWN	Pull down LOAD parameter	<load>	99
DUALSTAGE	Type of probe pin	<probe_type>	114
E	U.S. drawing size	<drawing_size>	71
EDGE	Pin exit description	<pin_exit>	96
ELECTRICAL	Purpose of hole	<hole_type>	57
ERASE	Line erasure (SOLID)	<line_type>	75
FABRICATION	Type of printed board drawing	<drawing_type>	71
FARAD	Device value description	<value_type>	102
FILL	Hole filling requirement	<barrel_type1>	76
		<barrel_type2>	76
		<barrel_type3>	76
	Texture for closed shapes	<paint_desc>	76
FILLET	D-shaped end description	<end_shape>	77
FINISH	Modification to line end	<mod_end>	75
FIXED	Type of electrical net	<net_class>	110
FIXTURE	Fixture layer description	<GenCAM_layer_type>	90
G	Green color intensity	<color_name>	50
GENCAMEXTNC	Neutral net name	<net_name>	110
GLUE	Glue dot layer	<GenCAM_layer_type>	90
HATCH	Cross-hatching of closed area	<paint_type>	76
HENRY	Device value description	<value_type>	102
HERTZ	Device value description	<value_type>	102
HOLEFILL	Hole filling layer	<GenCAM_layer_type>	90
HOLLOW	No plating in hole	<barrel_type1>	76
		<barrel_type2>	76
		<barrel_type3>	76
	No texture in closed area	<paint_type>	76
ICT	In-circuit test	<probe_type>	114
INCH	Dimensional unit	<dimension>	50
INDUCTIVE	Type of probe attachment	<probe_attach>	114
JOULE	Device value description	<value_type>	102
LEADBEND	Lead preparation	<comp_prep_type>	105
LEADLENGTH	Lead length preparation	<comp_prep_type>	105
LEGEND	Reference designation layer	<GenCAM_layer_type>	90
LUMEN	Device value description	<value_type>	102

APPENDIX C

General Fixed Field Parameters

Word	Definition	Usage	Page(s)
MARKER	Purpose of target	<function>	80
MATING	Type of probe attachment	<probe_desc>	114
MECHANICAL	Purpose of holes	<hole_type>	57
MESH	Single line angled texture	<paint_type>	76
MIRROR	Mirroring of image	<mirror>	7
MM	Millimeters	<dimension>	50
MM100	hundredths of millimeter	<dimension>	50
NOACCESS	No electrical test access	<access_desc>	110
NONE	Line end characteristics Line modification characteristics Load characteristics	<line_end> <line_mod> <load>	75 75 99
NOPROBE	No probing access	<access_desc>	110
NORTH	Direction of D-shape	<orientation>	77
OCTAGON	Description of thermal	<therm_shape>	26
OHM	Device value description	<value_type>	102
ONE	Logic description	<open_input_logic>	99
OTHER	Type of probe tip Type of component preparation Type of probe attachment	<tip_type> <comp_prep_type> <probe_attach>	114 105 114
PACK	Presentation media	<delivery_system>	105
PARALLEL	Parallel spoke end shape	<spoke_end_shape>	77
PARTSLIST	List of materials	<drawing_type>	71
PASTEMASK	Solder paste mask layer type	<GenCAM_layer_type>	90
PHANTOM	Type of line	<line_type>	75
PIN	Pin layer	<GenCAM_layer_type>	90
PLATE	Conductive plating	<barrel_type1> <barrel_type2> <barrel_type3>	76 76 76
PO	Purchase order	<trans_type>	53
PREPCODE	Component preparation	<comp_prep_type>	105
PROBE	Probing layer point	<GenCAM_layer_type>	90
R	Red color intensity	<color_name>	50
RADIANS	Increments of 2 π	<angular_unit>	50
RADIUS	Type of probe type tip	<tip_type>	114
REEL	Presentation media	<delivery_system>	105
REGISTRATION	Purpose of target	<function>	80
REMAIN	Well depth	<cut_mode>	57
REMOVE	Well depth	<cut_mode>	57
REPLACE	Change action to GenCAM file	<action>	117
RESISTIVE	Resistive material layer	<GenCAM_layer_type>	90
REWORK	Layer to be or having been reworked	<GenCAM_layer_type>	90
RFA	Request for audit	<trans_type>	53
RFP	Request for proposal	<trans_type>	53
RFQ	Request for quote	<trans_type>	53
ROUND	End of line End of D-shape Description of thermal End of thermal	<line_end> <end_shape> <therm_shape> <spoke_end_shape>	75 77 26 77
ROUTE	Keepout description	<keepout_type>	105
SCHEMATIC	Type of drawing	<drawing_type>	71
SELF	Description of self employment	<enterprise>	53

APPENDIX C

General Fixed Field Parameters

Word	Definition	Usage	Page(s)
SIGNAL	Type of route	<net_class>	110
SL1	Line modification (key hole)	<line_mod>	75
SL2	Line modification (key hole)	<line_mod>	75
SOLDERBUMP	Solid solder deposition	<GenCAM_layer_type>	90
SOLDERMASK	Solder resist layer	<GenCAM_layer_type>	90
SOLID	Type of line	<line_type>	75
SOUTH	Direction for D-shape primitive	<orientation>	77
SPEAR	Type of probe tip	<tip_type>	114
SPECIFICATION	Type of drawing	<drawing_type>	71
SQUARE	End of line characteristics Thermal space characteristics Description of thermal	<line_end> <spoke_end_shape> <therm_shape>	75 77 76
START	Modification of beginning of line	<mod_end>	75
TD	Tear drop line modification	<line_mod>	75
TP	Tapered line modification	<line_mod>	75
TESTPIN	Keepout description	<keepout_type>	105
TESTPROBE	Keepout description	<keepout_type>	105
THOU	Inch/1,000	<dimension>	50
THOU10	Inch/10,000	<dimension>	50
TOOLING	Purpose of holes	<hole_type>	57
TOP	Pin exit description Surface description	<pin_exit> <surface>	96 57
TRANSFER	Type of probe characteristics	<probe_type>	114
TRAY	Presentation media	<delivery_system>	105
TUBE	Presentation media	<delivery_system>	105
TULIP3	Type of probe tip	<tip_type>	114
TULIP4	Type of probe tip	<tip_type>	114
TWISTED	Type of group cabling	<group_type>	113
UM	Micrometer (µm)	<dimension>	50
UP	Pull up LOAD	<load>	99
USERINCH	Grid inch-based units	<dimension>	50
USERCM	Grid Centimeter units	<dimension>	50
USERMM	Grid millimeter units	<dimension>	50
USERUM	Grid micrometer units	<dimension>	50
VIA	Type of keepout	<keepout_type>	105
VOID	Removal of texture in filled, mesh, or hatched objects	<paint_type>	76
VOLT	Device value description	<value_type>	102
WATT	Device value description	<value_type>	102
X	Open input logic description	<open_input_logic>	99
ZERO	Open input logic description	<open_input_logic>	98

APPENDIX D

Device Type Parameters

Word	Definition	Usage	Page(s)
ANALOG	Analog ICs	Devices	101
BRIDGE	Bridge Rectifiers	Devices	101
CAP	Two terminal non-polarized capacitors	Devices	101
CONN	Connectors	Devices	101
CPCK	Capacitor packs	Devices	101
DIAC	DIAC	Devices	101
DIODE	Diodes including Schottkey	Devices	101
DPCK	Diode package	Devices	101
FABCAP	Used for embedded capacitors	Devices	101
FABRES	Used for embedded resistors	Devices	101
FUSE	Fuses	Devices	101
HYBRID	A-to-D or D-to-A converter	Devices	101
IND	Inductor	Devices	101
JUMPER	Jumpers (shunts)	Devices	101
LED	Light Emitting Diode	Devices	101
LEDPCK	LED package	Devices	101
LOGIC	Used for logic	Devices	101
NPN	Transistors	Devices	101
NFET	Transistors	Devices	101
NJFET	Transistors	Devices	101
OPTO	Opto-isolators	Devices	101
OPAMP	Operational Amplifiers	Devices	101
OTHER	Used for undefined parts	Devices	101
PCAP	Two terminal polarized capacitors	Devices	101
PFET	Transistors	Devices	101
PJFET	Transistors	Devices	101
PNP	Transistors	Devices	101
RELAY	Relays	Devices	101
RES	Two terminal fixed value resistors	Devices	101
RPCK	Resistor packs	Devices	101
SCR	Thyristors	Devices	101
SOCKET	Sockets	Devices	101
SWITCH	Switch	Devices	101
TCAP	Tantalum capacitors	Devices	101
TRIAC	TRIAC	Devices	101
VCAP	Variable capacitor	Devices	101
VR	Voltage regulator	Devices	101
VIND	Variable inductor	Devices	101
VRES	Variable resistor	Devices	101
XFMR	Transformer	Devices	101
XTAL	Crystals	Devices	101
ZENER	Zener Diode	Devices	101

APPENDIX E

Device Pin Type Parameters

Word	Definition	Usage	Page(s)
ANALOG	Used for analog pins	Devices	102
ANALOGIN	Used for inputs for analog ICs	Devices	101
ANALOGOUT	Used as outputs for analog ICs	Devices	101
ANODE	Used for diodes, zeners, unijunctions, thyristors etc.	Devices	102
BASE	Used for transistors	Devices	102
BIDIRECTIONAL	Used for bi-directional logic	Devices	102
CASE	Used for connection to device screen or can	Devices	102
CATHODE	Used for diodes, zeners, unijunctions, thyristors etc.	Devices	102
CLOCK	Used for clock	Devices	102
COLLECTOR	Used for transistors (including IGBTs)	Devices	102
DIGITAL	Used for any family of logic pins	Devices	102
DRAIN I GATE	Used for FETs	Devices	102
DRIVER	Used for outputs which drive a net.	Devices	102
EMITTER	Used for transistors (including IGBTs)	Devices	102
ENABLE I DISABLE	Used for device enable/disable	Devices	102
GROUND	Used for ground pins	Devices	101
INACTIVE	Used for resistors, capacitors, etc. pins	Devices	102
INTNC	Used for pins that are internally disconnected	Devices	102
NCLOSED	Closed relay or switches	Devices	101
NOPEN	Open relays and switches	Devices	101
POWER	Used for power supply pins	Devices	101
RECEIVER	Used for inputs	Devices	101
SOURCE	Used for FETs	Devices	102
TDI I TDO I TMS I TCK	Used for 1149.1	Devices	102
WIPER	Used for various components	Devices	102

APPENDIX F

Package Type Parameters

Word	Definition	Usage	Page(s)
AXIAL_LEADED		Packages	96
BARE_DIE		Packages	96
CERAMIC_BGA		Packages	95
CERAMIC_DIP		Packages	95
CERAMIC_FLATPACK		Packages	95
CERAMIC_QUAD_FLATPACK		Packages	95
CERAMIC_SIP		Packages	95
CHIP		Packages	95
CHIP_SCALE		Packages	96
CHOKE SWITCH_SM		Packages	96
COIL		Packages	95
CONNECTOR_SM		Packages	96
CONNECTOR_TH		Packages	96
EMBEDDED		Packages	95
FLIPCHIP		Packages	96
HERMETIC_HYBRID		Packages	96
LEADLESS_CERAMIC_CHIP_CARRIER		Packages	96
MCM		Packages	96
MELF		Packages	95
MINI_BGA		Packages	95
MOLDED		Packages	96
NETWORK		Packages	96
PGA		Packages	95
PLASTIC_BGA		Packages	95
PLASTIC_CHIP_CARRIER		Packages	95
PLASTIC_DIP		Packages	95
PLASTIC_SIP		Packages	95
POWER_TRANSISTOR		Packages	96
RADIAL_LEADED		Packages	96
RECTANGULAR_QUAD_FLATPACK		Packages	96
RELAY_SM		Packages	96
RELAY_TH		Packages	96
SOD123		Packages	95
SOIC		Packages	95
SOJ		Packages	95
SOPIC		Packages	95
SOT143		Packages	95
SOT23		Packages	95
SOT52		Packages	95
SOT89		Packages	95
SQUARE_QUAD_FLATPACK		Packages	95
SSOIC		Packages	95
SWITCH_TH		Packages	96

APPENDIX F

Package Type Parameters

Word	Definition	Usage	Page(s)
TANTALUM		Packages	95
TO_TYPE		Packages	96
TRANSFORMER		Packages	96
TRIMPOT_SM		Packages	96
TRIMPOT_TH		Packages	96
TSOP		Packages	95

APPENDIX G

Package Pin Type Parameters

Word	Definition	Usage	Page(s)
BALL		Packages	98
BUTTLEAD		Packages	98
CASTELLATION		Packages	98
COLUMN		Packages	98
GULLWING		Packages	98
JLEAD		Packages	98
LAND		Packages	98
SLEAD		Packages	98
TERMINATION		Packages	98
TH_RIBBON		Packages	98
TH_ROUND		Packages	98
TH_V		Packages	98
WRAPAROUND		Packages	98

APPENDIX H

Drawing Type Parameters

Word	Definition	Usage	Page(s)
A DRAWING SIZE	8 ½" x 11"	Drawings	73
A0 DRAWING SIZE	840mm x any length	Drawings	73
A1 DRAWING SIZE	594mm x 840mm	Drawings	73
A2 DRAWING SIZE	420mm x 594mm	Drawings	73
A3 DRAWING SIZE	297mm x 420mm	Drawings	73
A4 DRAWING SIZE	210mm x 297mm	Drawings	73
ASSEMBLY	Assembly drawing definition	Drawings	73
B DRAWING SIZE	11" x 17"	Drawings	73
C DRAWING SIZE	17" x 22"	Drawings	73
D DRAWING SIZE	22" x 34"	Drawings	73
DETAIL	Detail part drawings	Drawings	73
E DRAWING SIZE	34" x 44"	Drawings	73
FABRICATION	Printed board fabrication	Drawings	73
PARTSLIST	List of materials for printed board assembly	Drawings	73
SCHEMATIC	Schematic or logic diagram	Drawings	73
SPECIFICATION	Specification or source control	Drawings	73

Bureau of Indian Standards

BIS is a statutory institution established under the *Bureau of Indian Standards Act,* 1986 to promote harmonious development of the activities of standardization, marking and quality certification of goods and attending to connected matters in the country.

Copyright

Review of Indian Standards

Amendments are issued to standards as the need arises on the basis of comments. Standards are also reviewed periodically; a standard along with amendments is reaffirmed when such review indicates that no changes are needed; if the review indicates that changes are needed, it is taken up for revision. Users of Indian Standards should ascertain that they are in possession of the latest amendments or edition by referring to the latest issue of 'BIS Catalogue' and 'Standards: Monthly Additions'.

This Indian Standard has been developed from Doc : No. LTD 17 (1937).

Amendments Issued Since Publication

Amend No.	Date of Issue	Text Affected

BUREAU OF INDIAN STANDARDS

Headquarters :

Manak Bhavan, 9 Bahadur Shah Zafar Marg, New Delhi 110 002
Telephones : 323 01 31, 323 33 75, 323 94 02

Telegrams : Manaksanstha
(Common to all offices)

Regional Offices :		Telephone
Central	: Manak Bhavan, 9 Bahadur Shah Zafar Marg, NEW DELHI 110 002	323 76 17, 323 38 41
Eastern	: 1/14 C.I.T. Scheme VI M, V. I. P. Road, Kankurgachi, KOLKATA 700 054	337 84 99, 337 85 61, 337 86 26, 337 91 20
Northern	: SCO 335-336, Sector 34-A, CHANDIGARH 160 022	60 38 43, 60 20 25
Southern	: C.I.T. Campus, IV Cross Road, CHENNAI 600 113	254 12 16, 254 14 42, 254 25 19, 254 13 15
Western	: Manakalaya, E9 MIDC, Marol, Andheri (East), MUMBAI 400 093	832 92 95, 832 78 58, 832 78 91, 832 78 92

Branches : AHMEDABAD. BANGALORE. BHOPAL. BHUBANESHWAR. COIMBATORE. FARIDABAD. GHAZIABAD. GUWAHATI. HYDERABAD. JAIPUR. KANPUR. LUCKNOW. NAGPUR. NALAGARH. PATNA. PUNE. RAJKOT. THIRUVANANTHAPURAM. VISAKHAPATNAM.

Printed by Dee Kay Printers.