**The CIF File Format**

The Cal (Tech) Intermediate Format (CIF) was developed at Cal Tech in the earliest days of design automation. The format, such as it is (there are many dialects), is public domain. Though possibly still used in educational and research environments, it is ususual in current commercial IC engineering.

The format used in native cell files and the device library file is an extension of the CIF file format. Through extension, this format is robust enough to meet the needs of *Xic* while retaining the syntactic simplicity of the original format. This section outlines the basic syntax of CIF, while the next section will provide details about the extensions used by *Xic*.

In CIF, ``lines'' are terminated with semicolons. The line feed and carriage return characters are taken as white space and ignored, and may not even be present, so the ``lines'' are actually logical only.

Comments in CIF are enclosed in parentheses. Comments are ignored in CIF, however *Xic* uses special comment lines for various purposes, as will be seen in the next section.

(This is an example comment);

Note that this (and all) CIF lines must be terminated with `;'.

The first one or two non-whitespace characters of a line (i.e., following `;') are used as a command key. In strict CIF, this key is a letter, though numbers have been adopted as widely-used extensions.

Historically, in CIF the word ``symbol'' has been used to refer to what in current terminology is referred to as a cell. When describing CIF, the terms ``symbol'' and ``cell'' are used as synonyms.

The DS (define symbol) directive begins a symbol (cell) definition.

DS *symnum A B*;

In strict CIF, symbols do not have names, but are referenced by symbol number. The assigned symbol number (an integer) follows DS. The remaining two parameters are for scaling. Each coordinate in the symbol is scaled by *A*/*B*. The use of two integers rather than a single floating point value was once considered a speed optimization. *Xic* never uses the *symnum* or scaling factors in native files:

DS 0 1 1;

The definition of a symbol is terminated with a DF line,

DF;

Within the symbol definition, there are layer directives, followed by geometry specifications, and subcell calls. A layer directive consists of a line with the form

L *layername*;

where *layername* is a name for a layer. This name is alphanumeric and is four characters in length or fewer. All geometry which follows a layer declaration will be assigned to that layer, until the next layer declaration.

There are three types of CIF geometric objects used by *Xic*: boxes, polygons, and wires. Boxes have the form

B *width height x y* [*rx ry*];

where the first two parameters are the box width and height, and the second two parameters are the coordinates of the midpoint of the box. The last two parameters are optional, and indicate a rotation. The two numbers define a vector with respect to the origin, and the angle represents the angle by which the box should be rotated. *Xic* never uses the rotation parameters for boxes. Non-Manhattan rotated boxes are converted to polygons.

Ordinarily, boxes are rendered according to the attributes of the layer on which the box is defined. In *Xic* electrical mode, boxes on the SCED layer use that attribute, however boxes on other layers are rendered as a dotted outline with no fill. The SCED layer defaults to solid fill, other layers default to empty fill. All physical layers default to empty fill.

Polygons are specified with P followed by x-y coordinate pairs. The first and last coordinates must be the same, indicating closure of the polygon.

P *x0 y0 x1 y1 ... xN yN x0 y0*;

The polygon is rendered using the fill attributes of the layer on which the polygon is defined. There should be at least four pairs of coordinates defined for a polygon.

Wires are fixed-width paths. A wire is specified with W followed by the width, which is followed by x-y coordinate pairs representing the path.

W *width x0 y0 x1 y1 ... xN yN*;

In electrical mode, the basic line primitive is a zero width wire. In physical mode, wires are defined with finite width as a physical necessity. The coordinates will form the vertices of the path. A wire can technically consist of a single vertex, which will be rendered as a box with the width of the wire. This construct is disallowed in *Xic*, and should be avoided. Wires are rendered with the fill attribute of the layer on which the wire is defined.

A symbol call (subcell) is indicated with a C, followed by the symbol number, followed by a transformation specification. The transformation if made up of components representing translation, rotation, and reflection. Translation is indicated by T followed by the translation:

T *x y*

Rotation is specified by R, followed by two numbers which represent a vector with respect to the origin, the angle of which is the angle of rotation. Many parsers recognize only orthogonal rotations. *Xic* recognizes angles that are multiples of 45 degrees.

R *rx ry*

Mirroring about the y-axis is specified with

MX

and about the x-axis with

MY

The transformation specification is a concatenation of these directives, which are evaluated in sequence to obtain the coordinate mapping from the cell coordinates in the symbol being instantiated to the cell coordinates of the parent of the instance. The overall syntax is

C *symnum transform*;

where an example would be

C 0 R 1 0 T -1000 0;

The parsing is terminated with an end directive:

END

This line need not be terminated with a semicolon.

The base coordinate system specified for CIF uses 100 units per micron.

## CIF Format Extensions

There have been numerous extensions to the CIF syntax used to enhance the capabilities of the original format. Some of these extensions have been accepted widely and have become essentially part of the standard. *Xic* uses these extensions, plus some further extensions, in native format files and in files converted to CIF without the **Strip For Export** button active. These extensions to the basic CIF syntax are enumerated below. Unless stated otherwise, the extension is applied identically in native cell files and CIF output.

1. The DS (define symbol) line is always followed by a cell name extension line of the form   
   9 *symbol\_name*;   
   This extension is widely used, and is a standard means for including the symbol names within the CIF framework.

In native cell files, however, the DS line is *preceded* by the symbol name line.

1. In *Xic* releases prior to 3.0.0, the symbol number part of an instance call was set to 0, i.e., the call sequence was always

C 0 ...;

when cell name extensions were used. Since cell names were provided through the extensions, the cell numbering is unneeded. In current releases, the cell numbering is retained and will appear in the instance calls, in all CIF output.

In CIF, the name of the cell being instantiated may precede the ``C ...'' (symbol call) line, using the same format as associated with the DS line, i.e.

9 *master\_name*;   
C *N* ...;

This is redundant, since the master name can be obtained from the symbol number.

In native cell files, this line *follows* the C line. This line is required in native cell files, as there is no symbol numbering.

In native cell files only, instead of a cell name, the string can contain a full path to a layout file in one of the supported formats, which must use one of the recognized file extensions. For example:

C 0 ...;   
9 /path/to/directory/containing/myfile.gds;

When the cell file is read into the *Xic* main database, the archive will be read in as well, and the instantiation master will become the top-level cell (there must be one only) in the archive file. A table is retained of the top-level cell to file path associations, and the path is used (rather than the top-level cell name) it the parent cell is later saved as a single native cell file.

1. Labels are specified with a unique syntax:   
   94 «*label string*» *x y code width height*;   
   This is a further extension of a widely-used extension for labels, which does not have the *code*, *width*, or *height* fields and the delimiters around the label. The original extension also required that the string contain no white space.

The *width* and *height* are the dimensions of the untransformed bounding box of the label. The label will be stretched to fill this area. The label is surrounded by « ». The *x* and *y* are the reference coordinates, which by default is the lower left corner of the bounding box. The *code* entry specifies transformations applied to the label at the reference point, and other rendering information, as shown in the table.

|  |  |
| --- | --- |
| bits 0-1 | rotate the text about *x*,*y* |
|  | 00 no rotation |
|  | 01 90 degrees |
|  | 10 180 degrees |
|  | 11 270 degrees |
| bits 2-3 | bit 2 = 1, mirror y after rotation |
|  | bit 3 = 1, mirror x after mirror y |
| bit 4 | bit 4 = 1, shift rotations to 45, 135, 225, 315 degrees |
| bits 5-6 | horizontal justification, 00, 11 left, 01 center, 10 right |
| bits 7-8 | vertical justification, 00, 11 bottom, 01 center, 10 top |
| bits 9-10 | GDSII font number |

1. Cells and instances can be preceded by properties of the general form   
   5 *prop\_num prop\_string*;   
   The property number *prop\_num* is an arbitrary integer. The property string begins with the first non-space character following the integer, and ends with the semicolon (the semicolon is not included). The string can contain any alphanumeric, punctuation or white space but not `;' for obvious reasons. There are a number of properties used by *Xic*, particularly in electrical mode. This extension is widely used.

*Xic* writes the electrical information in a second symbol definition which immediately follows the physical cell definition in native files, but after the terminating token of the physical cell. Similarly, when *Xic* writes a CIF file without the **Strip For Export** function active, the electrical CIF representation immediately follows the physical CIF data, after the termination token.

In *Xic* releases prior to 3.0.0, the cell terminator was the single character E. This was used in both native cell files and unstripped CIF. In the present release, the cell terminator is always ``End'' in CIF, ``E'' in native cell files..

Whether or not these extensions are used when writing CIF output is controlled by a set of flags, which can be individually set from the CIF page of the **Set Export Parameters** panel. Actually, there are two banks of these flags, one bank is used when **Strip For Export** is set, the other bank is used when **Strip for Export** is unset. In the case of **Strip For Export** set, the flags all default to 0, so no extensions are used. In the case of **Strip For Export** unset, the flags all default to 1, so all extensions are used.

The user can set these flags individually through the **Extension Flags** menu in the CIF page of the **Set Export Parameters** panel. The bank of flags being set is determined by the state of the **Strip For Export** button and variable.

The flags in the menu have the following effects.

**scale extension**   
Traditional CIF has a fixed resolution of 100 units per micron. This extension will add a comment of the form

(RESOLUTION *nnn*);

near the top of the file, and use *nnn* as the file resolution. The CIF reader must check for this comment and scale numerical values accordingly.

*Xic* normally uses internal units in unstripped CIF and native files, signaled with the addition of a comment line ahead of the first symbol definition something like:

(RESOLUTION 1000);

*Xic* will look for this comment, and interpret the coordinates accordingly. If no comment is found, the CIF default of 100 units per micron is assumed. *Xic* will always use internal units when writing a CIF file when this extension is enabled, and 100 units otherwise.

**cell properties**   
Properties may be applied to cell definitions, ahead of the DS.

**inst name comment**   
Comments of form

(SymbolCall *cellname*);

are added ahead of instance `C' calls.

**inst name extension**   
Text in the form

9 *cellname*;

is added ahead of instance `C' calls.

**inst magn extension**   
Cell instance `C' calls can be preceded by a magnification extension of the form

1 Magnify *magn*;

where *magn* is a magnification factor. All internal structure of the cell will be scaled by the given factor, which is a floating point number greater than zero. This extension will appear in physical cell descriptions only. It is unique to *Xic*.

**inst array extension**   
Cell instance `C' calls can be preceded by an array extension of the form

1 Array *x dx ny dy*;

where *nx* and *ny* are the number of cells to array in the x and y directions, and *dx* and *dy* are the spacing between cells. This extension was used in earlier CAD programs.

**inst bound extension**   
Cell instance `C' calls can be preceded by a bounding box extension of the form

1 Bound *left bottom right top*;

The *left*, *bottom*, *right*, *top* are the coordinates of the parent cell defining the bounding box of the subcell. This extension is not currently used, though it is written into the files. It is unique to *Xic*.

**inst properties**   
Properties may be added ahead ahead of instance `C' calls.

**obj properties**   
Properties may be added ahead of B (boxes), P (polygons), and W (wires).

**wire extension**   
The end style of wires is not part of traditional CIF. In this extension, text of the form

1 7033 PATHTYPE *n*;

may be added ahead of wires to specify an end style. The values of *n* are 0 (flush ends), 1 (rounded ends), or 2 (extended ends, the default).

This extension was used in *Xic* prior to 2.5.23. It has been superseded by **wire extension new**, which will have precedence if both extensions are enabled.

**wire extension new**   
This overrides **wire extension**, wires include an end-style designation:

W0 | W1 | W2 *width* *x-y data*;

The end style of wires is not part of traditional CIF. In this extension, the wire end style 0-2 immediately follows the `W', with the rest of the line as in traditional CIF. The end style is the same as the GDSII path type: 0 for flush ends, 1 for rounded ends, and 2 for extended square ends.

This extension was introduced in release 2.5.23. Older releases of *Xic* are not compatible with this extension.

**text extension**   
Label string text is enclosed in « », and may include white space. Without this extension, white space characters in the label text will be replaced with underscores. In both cases, semicolons are replaced with underscores. This extension applies with any of the label format choices.