Magic, CIF, and GDS Files,

Magic Tech Files, MOSIS Tech Codes,

***(and things that go* bump *in the night*)**

This document covers some important topics relating to sending our chip designs to MOSIS for fabrication. It includes brief discussions of how to generate files in the 'CIF' format that MOSIS requires, tips on final checking of the design, and a brief look at magic and CIF file syntax.

Magic Files versus CIF (and GDS) File Formats

When you create your layouts, your design is stored in a file format particular to the specific EDA tool you are using. In our case, this is Magic, and the geometry information is output into '.mag' files when you use the 'save' command. For other tools, other formats are used. IC foundaries however require a standard format independent of the particular design tool.

The most common file formats accepted by IC foundaries are CIF and GDS (aka 'Calma'). The main difference is that CIF is ASCII (human readable) while GDS is binary. While GDS is preferred by many people due to it's smaller file size, the format of CIF is easier to understand. This is the format we will use.

Some examples of Magic and CIF files are shown on the attached page. For detailed information on Magic's file format, please see the on-line user and maintenance manuals for Magic (especially Magic ***Tutorial #9: Format Conversion for CIF and Calma***). A good explanation of CIF and GDS (Calma) file formats can be found in an out-of-print text book by Steven Rubin. He has placed this book on the Web for free access at ***http://www.rulabinsky.com/cavd***. See Appendix B for CIF and Appendix C for GDS format descriptions.

Exporting to CIF (or GDS) Files from Magic

Magic can export it's drawing data to either CIF or GDS formats. To export to CIF, do:

```
:cif ostyle lambda=0.6(nwell)
:cif write <filename>
```

The first command tells magic how to scale the design during the export process and that the target process is nwell. The second command creates the output file (appending '.cif' to the name you supply). These are the commands we will use.

There are many options to these commands. For instance, if you wish to export GDS instead of CIF, then substitue `:calma write <filename>` for the second command. Magic will write the file in GDS (calma) format, appending the suffix .strm. Or, if you target a different process, you will need to supply different parameters to the ostyle command. (See on-line magic manuals for more info.)

Importing CIF or GDS Files into Magic

Magic can also convert files in the opposite direction. To import a CIF file, invoke magic with no filename and then do:

```
:cif istyle lambda=0.6(nwell)
:cif read <filename>
```

Important Final Checks before Submitting a Design

It is extremely wise to export your design to CIF, and then read it back into magic before doing final checks and submitting it for fabrication. Magic does some 'sneaky' things behind the scenes, and the only way to be sure that what you send to MOSIS is correct is to look at the design after it goes through the export process. (Typical problems encountered are behind-the-scenes modifications of wells that create design rule violations, merging of wells that you don't want merged, and 'abutment' problems created where subcells overlap.)

Before you submit your company's design to me, you should export it to CIF, import it back into magic, and then carefully check it for DRC errors. A good way to do the last step is to put magic's edit box around the whole chip after importing, and then type `:drc why` (or hit the 'y' key).

Another critical check that should be made is to use the 's' key command on each pad to check that all pads are properly connected to circuits inside the chip, and that power and ground are not shorted!

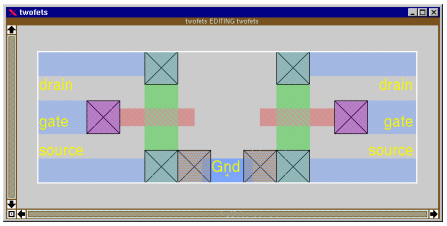
Tech Files, Tech Codes, and Other Important Things

The conversion of magic's internal data format into the CIF or GDS formats is controlled by the Technology File that you specify when you invoke magic. Traditionally, this has been the 'scmos' tech file that you have been using, and this is the tech file that we will use. However, if you ever need to design for a process not supported by this tech file (such as one of the newer submicron processes now offered by MOSIS), you will need to use a different tech file and understand issues such as MOSIS's 'Tech Codes'. Please see the MOSIS web pages and the Magic user and maintenance manuals for information on these topics.

And Finally... Things that go *bump* in the night

Submitting a chip design to MOSIS is (and should be) a scary experience. Similar to software design, a single mistake (like a short between Vdd and Gnd in the final layout) can cause a whole chip to fail. However, unlike software design, the time required to fix a chip is months instead of minutes, and the cost of remaking the chip is infinitely higher. Hence, it is important to do as much checking and testing as possible within the limits of the design tools. Check your design, and then recheck it again. Do this to the last minute, so that you can rest easy during the cold dark nights of Winter while your chip is being fabricated, and you can be confident of working circuits in the Spring!

Here are some examples of magic and CIF file formats. The boxes below show the files corresponding to the hierarchical design at the right which consists of two copies of a cell called ‘fet’ (with one flipped sideways), and a small piece of metal 1 with label ‘Gnd’ in the top-level cell ‘twofets’.



File ‘twofets.mag’

```
magic
tech scmos
timestamp 944505883
<< metal1 >>
rect 20 0 26 3
use fet fet_0
timestamp 944505650
transform 1 0 13 0 1 1
box -13 -1 8 15
use fet fet_1
timestamp 944505650
transform -1 0 33 0 1 1
box -13 -1 8 15
<< labels >>
rlabel metal1 23 1 23 1 1 Gnd
<< end >>
```

File ‘fet.mag’

```
magic
tech scmos
timestamp 944505650
<< polysilicon >>
rect -3 6 0 8
rect 4 6 6 8
<< ndiffusion >>rect 0 8 4 11
rect 0 3 4 6
<< metal1 >>
rect -13 12 0 15
rect -13 5 -7 9
rect -13 -1 0 2
<< ntransistor >>
rect 0 6 4 8
<< polycontact >>
rect -7 5 -3 9
<< ndcontact >>
rect 0 11 4 15
rect 0 -1 4 3
<< psubstratepcontact >>
rect 4 -1 8 3
<< labels >>
rlabel metal1 -13 12 -13 15 4
drain
rlabel metal1 -13 5 -13 8 3 gate
rlabel metal1 -13 -1 -13 2 2
source
<< end >>
```

File ‘twofets.cif’ generated using ‘scmos’ tech file with ‘cif ostyle lambda=0.6(nwell)’.

```
( @@user : wkuhn );
( @@machine : wireless.eece.ksu.edu );
( @@source : twofets.mag );
( @@tool : Magic 6.5 );
( @@patch : 0 );
( @@patchnames : release-6.5b1, linux1,
( @@compiled : Thu Jul 29 17:12:51 CDT
( @@technology : scmos );
( @@version : 8.2.8 );
( @@techdesc : MOSIS Scalable CMOS
( @@style : lambda=0.6(nwell) );
( @@date : Mon Dec  6 13:08:15 1999 );
DS 1 30 2;
9 twofets;
L CMF;
    B 24 12 92 6;
94 Gnd 92 4 CMF;
C 2 R 1 0 T 52 4;
C 2 MX R 1 0 T 132 4;
DF;
DS 2 30 2;
9 fet;
L CMF;
    B 68 12 -18 54;
    B 16 4 8 46;
    B 40 16 -32 28;
    B 32 4 16 10;
    B 84 12 -10 2;
L CPG;
    B 16 4 -20 34;
    B 52 8 -2 28;
    B 16 4 -20 22;
L CAA;
    B 16 48 8 36;
    B 32 16 16 4;
L CCA;
    B 8 8 8 52;
    B 8 8 8 4;
L CCA;
    B 8 8 24 4;
L CCP;
    B 8 8 -20 28;
L CSN;
    B 32 56 8 40;
    B 24 24 4 0;
L CSP;
    B 16 8 32 16;
    B 24 24 28 0;
94 drain -52 54 CMF;
94 gate -52 26 CMF;
94 source -52 2 CMF;
DF;
C 1;
End
```