Edit the PRIMARY, INDISK, OUTDISK, PRINTFILE fields to reflect your design. Uncomment and edit the EDTEXT line. fields in the Dracula deck to reflect your design entity.

Next, we need to process the netlist. Type LOGLVS. From LOGLVS, type case to turn on case sensitivity. Then type cir netlist to load in the netlist. Then type com to convert the netlist to the binary format read by Dracula. Type exit to quit.

Now copy the Pdracula run script from

```
/usr/tech/tsmc025/dracula/pdracula.lvs
```

Run PDRACULA

```
PDRACULA < pdracula.lvs > pdracula.out
```

Then run Dracula

```
./jxrun.com > run.log &
```

A file with a *.lvs extension will contains the LVS compare log.
Copy `drac025s3V5M.drc` and `pdracula.drc` from `/usr/tech/tsmc025/dacula` to your DRACULA directory. Edit the `drac025s3V5M.drc` file. In the DESCRIPTION block, set PRIMARY to `top_cell`. Set INDISK to `../STREAM/top_cell.str`. Set OUTDISK to `top_cell.out`.

We first need to run PDRACULA to create the Dracula run script for DRC.

Execute the following command:

```
PDRACULA <pdracula.drc >pdracula.out
```

You should not have a run script `jxrun.com`. Please execute this script:

```
./jxrun.com > output.log
```

The output data is in the file `drac.sum`. Look to see if there is anything listed under “Problem Geometry Error Listing.” More details on finding and debugging DRC errors through Dracula Inquiry and be found in `openbook` in the Dracula User’s Guide and Dracula Reference Manual.

LVS is a little more complex, since we need to have a schematic netlist as well as the layout data. To generate this netlist, you need to run the CDL netlist on your schematic. Create a CDL directory off of your home directory. Choose File → Export → CDL from the CIW. In the “CDL Out Run Form”, fill in the “Top Cell Name” and “Library Name” fields. In “Run Directory,” specify “/CDL. OK the form.

Change directory back to the Dracula run directory, “/DRACULA. We now need to create an EDTEXT file that contains all the pin locations at the top level. A Skill function has already been loaded with the CISTL setup to do this (but unfortunately, right now, it only works correctly with geometric pins; i.e., it doesn’t work with symbolic pins). To run this type:

```
createEdtext("libName" "cellName" "layout" "edtext")
```

where `libName` is the name of the library and `cellName` is the name of the cell. The EDTEXT file gets written into your run directory.

Now, we need to set up for the Dracula run. Copy

```
/usr/tech/tsmc025/dacula/drac025s.lvs
```

into your Dracula run directory. Copy the netlist from your “/CDL directory to your Dracula run directory. Copy the stream file from your “/STREAM directory to your Dracula run directory (if you haven’t done so already).
Columbia University  
Department of Electrical Engineering  
EE E4332. Tools tutorial #3.  
Dracula DRC/LVS for TSMC025

In this tutorial, we describe how to run Dracula DRC/LVS for layout verification. Dracula is a necessary part of our design flow because the fab (in this case TSMC) only “blesses” rule sets for Dracula. It is not possible to carry over all of the checks in Dracula in Diva, since Diva is a far less powerful tool. Full-chip DRC/LVS with Diva will also be virtually impossible, because of the limited capacity of Diva.

The “standard” for transferring layout data to the fab is GDSII format, a very old format that began with layout tools developed by GE Semiconductor in North Carolina in the early 1980’s. GE got rid of its semiconductor business when Jack Welch became CEO and most of the CAD-tool assets were acquired by what eventually became Cadence. Most of the CAD people from GE left the company, the most notable is Art De Geus, who then founded Synopsys, which is now the leading synthesis company (Art De Geus is still CEO of Synopsys). GDSII is sometimes also referred to as “stream” data, and we will generate this file from the Cadence layout database. Dracula will act directly on the stream data, adding another degree of checking, in that errors in the stream-out process would also be detected in DRC.

Running Dracula should probably be a more “automated” process, but for now, please follow the steps in this tutorial (carefully!).

To begin this process, we need to create a stream file for your layout. I will assume that the cell we are running is called top_cell in library tutorial. Create a directory called STREAM in your home directory. We will use this as our working directory. To create the stream file, choose File → Export → Stream from the CIW. On the Stream Out form, enter ~/STREAM as the “Run Directory.” Choose a “Library Name” of tutorial and a “Top Cell Name” of top_cell. Specify top_cell.str as the “Output File”. Hit the “User-Defined Options” button and you’ll get another form. As the “Layer Map Table”, please specify /usr/tech/tsmc025/gds2/opus.map. OK the form. OK the Stream form. The stream process should leave a GDSII file, top_cell.str in your STREAM directory.

Create a DRACULA directory in your home directory to act as your Dracula run space. Now, we need to set up the DRC run. cd to the DRACULA directory.