Synopsys Custom Designer Tutorial
for a chip integration using
the University of Utah Standard Cell Libraries
In ON Semiconductor 0.5u C5 CMOS

Version 6.0
Overview
This tutorial will take you through the following steps:

1) Create a new library and add standard cells and copy a pad frame. The copy of the pad frame will serve as the final chip view.

2) import your GDS file and verilog netlist from IC Compiler – referred to as the “core” - into the new library.

3) add pins to the core for connectivity

4) instantiate the core into the top-level chip in both layout and schematic

5) wire the core and pad frame together in layout and schematic

6) run LVS on the chip

7) run DRC on the chip

8) generate a chip level GDS to be sent to MOSIS for fabrication
**Start the tool**
Create a working directory – I made one called “custom”

Add cds.lib, lib.def and color map files into the working directory.

Invoke the tool with “cdesigner &”

Open the library manager and create a new library.
**Fill your user’s library**
Select and copy all cells from the UofU_Digital_v1_2 library into your new library using Library Manager. Right click on the cells to copy.

When copying libraries you may see a dialog box about replacing cells. These are all the same so just re-copy and replace any redundant cells.
**Fill your user’s library**
Also copy the UTFSM_pad_frame cell into your library but rename it “final_chip”.

You will edit this cell by including your core and connecting it in both layout and schematic. This will be the final design to be fabricated.
**Import GDS – step 1**

From Custom Designer Console we can import GDS. From the Console, choose File > Import > Stream. Under the Main tab, specify required input details: Run Directory; Input GDS Stream file and top cell.

Output details: Output Library, view: Layout

Under the Options tab, specify optional details. Under the Map Files tab, specify optional map file details. See the attached map file *(UofU_TechLib_ami06_version2.layermap)*

Click OK.
Import GDS – step 2
Select the correct layer map. (*UofU_TechLib_ami06_version2.layermap* – included in the chip tutorial zip files).
Import Verilog to create Symbol and Netlist view for the Core in Custom Designer

From Custom Designer Console
• Go to File → Import → Text

Select this option to import Verilog to symbol & schematic
Select the language as Verilog

Browse for your post-place-and-route netlist. Remove all fill cells first.

Use the following unix/linux command to remove fill:
grep -v FILL netlist.v > netlist.nofill.v

Choose the library that you created previously
Import Verilog: Choose Input Netlist File to create a symbol and schematic for your core. This step can a minute to perform if you have many gates in your core.

Creates a symbol for your chip level schematic and a schematic for LVS. This is your working library after you copied in all the standard cells. This can be either schematic or symbol. I prefer symbol.
Import Verilog: View of created symbol with port names

Go to your core cell view and double click the symbol to see something like this.

This will be used in your chip level schematic to represent and to connect to your core.
Import Verilog: View of created schematic necessary for LVS checking at the end

Don’t worry about red fly lines in the schematic. I’m investigating this but have found no problems.
Fix your core ports *(read and perform carefully.)*

The GDS import will have included text for the pins of your core but pins must be instantiated to tell LVS where connections in layout will be made. We lost this info when we converted to and from GDS. I hope to update the methodology to fix this problem but have not found the solution as of December 1st.

1) Click the text name near the port, press “q” for properties and change the layer to “text drawing”.
2) Click the metal square at the end of the port, press “q” and update the net name to match the text – however change square brackets to triangle brackets for the indices.
3) Press “create pin”, type the name of the net into the tool bar input, press enter and select two opposing corners of the pin to define the connection point. You can easily do many pins consecutively. Remember to select the correct metal (metal 2 or 3). This is symbolic for the tool and provides information as where your core should be contacted.
4) Select the newly created pin, press “q” and change the “inputOutput” to either “input” or “output”.

I would recommend doing each step above for all pins at once.

Again, I am working to eliminate this annoying step. We are losing info when we convert to and from GDS. I am trying to read the ICC libraries directly into Custom Designer.
**Edit your chip layout**

Use “i” (or edit->add instance) to create an instance. Select your core and place in the center.
**Layout hints**

Press “z” and draw a rectangle to zoom to area

Press “shift f” to see internals of lower level cells.

Press “control f” to hide internals of lower level cells.

Press “f” to fit the view.

Press “Ctrl-r” to redraw after changing selection and visibility of layers.

When “creating interconnect”, start with “p” and use “Ctrl-v” and “Shift-v” to switch metal layers through a via up (1 to 2 or 2 to 3) or down (3 to 2 or 2 to 1) respectively.

Press “q” to see the properties of an object.

Double click an object to descend the hierarchy and Ctrl-E to return.

Almost everything you will do will be in the metal layers, so turn on visibility for metal 1, 2, 3, via, via 2 and text.
Add connections to layout
“Create Interconnect” or press p to add wires to make IO connections, shift-v and ctrl-v to switch metal layers through a via as you connect from one pin to the other. Here is a picture of some connections between the core and padring.
Add all connections to layout -1

1) To connect two pins together, start by pressing “p” and single click next to the pin. I try to cover the entire pin with the new metal.

2) Single click to make a turn.

3) Double click or press enter to finish the operation. Cover the entire destination pin.
Add all connections to layout - 2
Use “shift-v” to go up a metal layer or “control-v” to descend a layer in order to avoid a collision with the same metal layer.
**Add all connections to layout**
Each IO has an “enable” which should be tied to ground (inputs) or vdd (outputs). Tie the signal to either a !gnd pin (purple box in the inner IO ring) or !vdd pin (purple box in the second ring from the inside).

If IO is an output, tie the “DataOutput” signal to the appropriate pin on your core. Let the two “DataInput” signals float.

If IO is an input, tie the “DataInput” signal to the appropriate pin on your core. Tie the “DataOutput” pin to ground. “DataInputN” can float.

On all four sides we need to connect the power ring of the IO to the power ring of the core. We need similar connections between the ground rings as well.

When you press “p” for creating interconnect, a “width” option appears in the menu bar. **Use 0.9u for signals and 4.5u for power.**

Put at least one ground and vdd connection per side of the core. More is better.
Add connections to layout – Input Example
Here a IO is configured as an Input. The input out signal is on the top right and the bottom shows the enable being grounded. Only metal 2 is shown. The purple boxes are metal 2 pin that designate an intentional connection.

DataInput, Output of IO metal 2, used when enable is tied low as in this case.

DataInputN, Output of IO metal 2, Don’t use this.

DataOutput, input to IO metal 2, used when enable is high

Enable, input to IO metal 2, tie high for Output Tie low for Input. Tied low here. See connection to purple ground pin
Add connections to layout – Example Output
Here the enable is brought by metal 3 to the Vdd ring and the dataout signal is connected in metal 2. Metal 3 is required to “jump over” the metal 2 ground bus.
Add connections to layout – Connecting the ground rings

Use a thick metal 2 line between ground in a inner pad ring of your core to the inner ring of your chip to connect ground. Do this on all four sides for robustness. Experiment with the width field in the menu bar as you create this bus. I had success with 4.5u which matched the power rings of my core.
Add connections to layout – Connecting the Vdd rings
Add connections to layout
Example of all connections with Metals 2 and 3. In this example I only had one ground and power connection as this is a tiny circuit. Please use more.
**Edit your chip Schematic**
In your chip level schematic view, you will see the 28 pin pad frame. You need to instantiate your core in the middle and connect to the padframe.

“i” will bring up the instantiation window, “w” will start to draw wires, “L” to add wire name. You can make all connections through labeling. This approach is less likely to introduce errors through making wrong connections.

Instantiate gnd and vdd symbols from the UofU_Analog_Parts library.

The connections in the schematic should match the layout for LVS later.

Save and check the design in the end.
Edit your chip Schematic
Run LVS

Currently, MOSIS only supports the LVS and DRC tools from Cadence and Mentor. So we need to send the completed design to UTEP for final checks with these tools. I am working on converting the rules for Synopsys, but this may not be done by the tape out date.

Go to your working directory and you should see a sub-directory that has the same name as your library. Use the following command to create a zip file to send to me at UTEP.

```bash
tar –cvf your_name_library_name.tar library_name
gzip your_name_library_name.tar
```
Currently, MOSIS only supports the LVS and DRC tools from Cadence and Mentor. So we need to send the completed design to UTEP for final checks with these tools. I am working on converting the rules for Synopsys but this may not be done by the tape out date.

Go to your working directory and you should see a sub-directory that has the same name as your library. Use the following command to create a zip file to send to me at UTEP.

```
tar -cvf your_name_library_name.tar library_name
gzip your_name_library_name.tar
```
**Export Final GDS to be Fabricated**

To export design data in GSDII format:
From the Console, choose File > Export > Stream.
In Main tab
   Specify the output: Enter Run Directory and GDS file in Stream file name.
   Specify the input: Browse the library and cell that you want to export and leave the view as layout. Select the cell that includes the pad ring and core connected.
Choose the Options tab and specify option details. We can change the options I left it default.
Choose the Map Files tab and specify mapfile details: Browse the layer map file.
   The layer map should be different than the one used to read your core. OA2GDS.layermap
Click OK.

Send me ([emac@utep.edu](mailto:emac@utep.edu)) the zipped version of you design library and the final gds file.

I will run the final checks and report problems. If no problems, the GDS file is sent for fabrication.