VLSI Design Flow: RTL to GDS

Dr. Sneh Saurabh
Department of Electronics and Communication Engineering
IIIT-Delhi

Tutorial 5
Lecture 22
Logic Synthesis using Yosys

Hello everyone, my name is Amina Haroon. I am a PhD student at IIIT Delhi and I will be your TA for the course VLSI Design Flow. In the previous tutorial, we learned about the simulation based verification on a digital circuit implemented in Verilog and also the code coverage. In this tutorial, we will see how to perform RTL synthesis and mapping the synthesized netlist to a library. I will be using an open source tool called Yosys Open Synthesis Suite, but you can use any other tool as well. I have listed all the steps in a tutorial sheet that I will be referencing in this demo and that sheet will be shared with you as well.

First, I will be demonstrating the steps to install the tool. For that, you should have a Linux distribution installed on your system. I am using Ubuntu, so let's begin. Go to the GitHub page of Yosys and install the required dependencies.

```
$ sudo apt-get install build─essential clang bison flex libreadline-dev gawk tcl-dev libffi-dev git graphviz xdot pkg─config python3 libboost-system-dev libboost-python-dev libboost-filesystem-dev zlib1g-dev
```

The dependencies are installed now. Clone the GitHub repository using the command git clone and the link to the repository.

```
$ gitclone https://github.com/YosysHQ/yosys.g i t
```

It is done now. Let's check the contents of the directory by typing in the command

```
$ ls
```

Change into the Yosys directory and compile the source code by typing the command

```
$ make
```

This will take a while. Make is done now. The next step is to do

$ sudo make install

You can launch the tool by typing

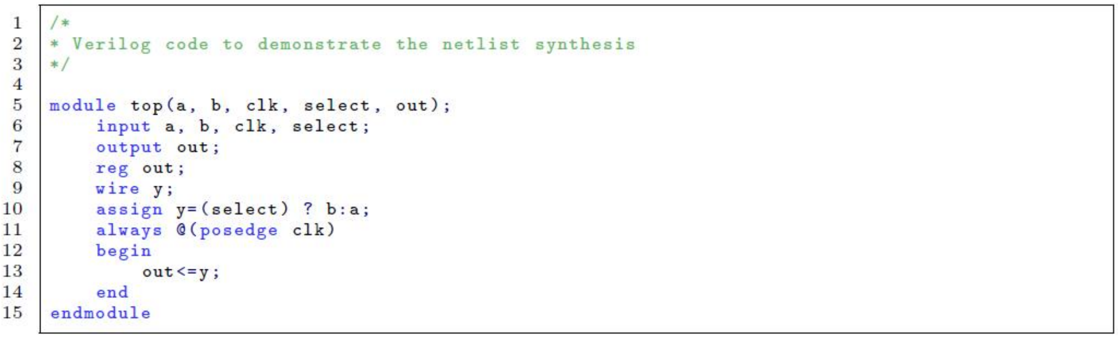
$ ./yosys

We will see how to use the tool with an example later in the demo.

We also need a technology library file. I am using Silvaco Open-Cell 45 nanometer free PDK libraries. To get the file, go to this link. The link is provided in the tutorial sheet as well. First, you need to fill the form.

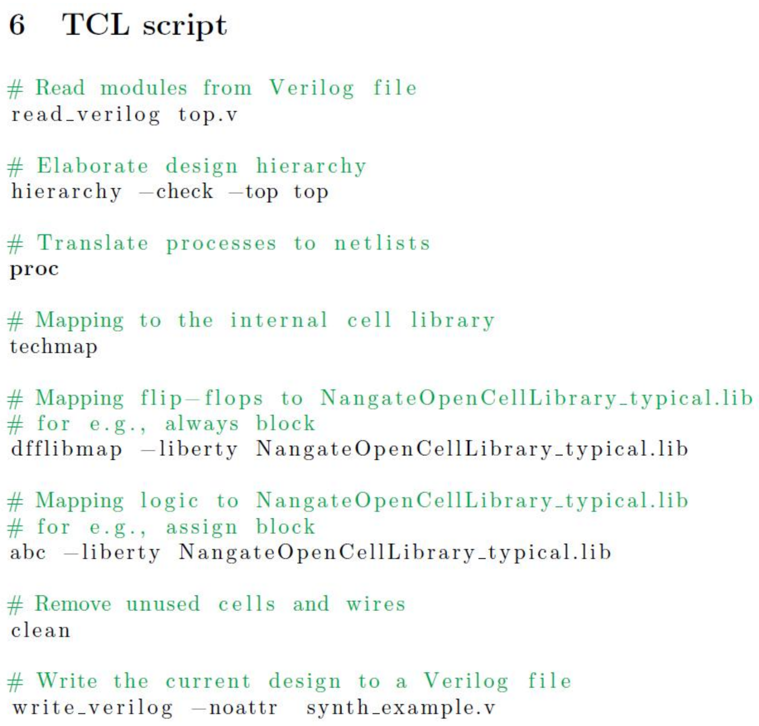
Please fill the form correctly because these details are verified. So your name, the organization you are associated with, the company or the university name, the address to your university or the company, your email address that is associated with the organization, the phone number, choose the SI2 membership status and finally select the FreePDK45 library. Once you submit the form, you will get the download link via email within a day. It's just valid for three days. So please download it as soon as you get it.

5 Example code [2]



```verilog
/*
* Verilog code to demonstrate the netlist synthesis
*/

module top(a, b, clk, select, out);
    input a, b, clk, select;
    output out;
    reg out;
    wire y;
    assign y=(select) ? b:a;
    always @(posedge clk)
    begin
        out<=y;
    end
endmodule
```



6 TCL script

```
# Read modules from Verilog file
read_verilog top.v

# Elaborate design hierarchy
hierarchy -check -top top

# Translate processes to netlists
proc

# Mapping to the internal cell library
techmap

# Mapping flip-flops to NangateOpenCellLibrary_typical.lib
# for e.g., always block
dfflibmap -liberty NangateOpenCellLibrary_typical.lib

# Mapping logic to NangateOpenCellLibrary_typical.lib
# for e.g., assign block
abc -liberty NangateOpenCellLibrary_typical.lib

# Remove unused cells and wires
clean

# Write the current design to a Verilog file
write_verilog -noattr  synth_example.v
```

Let's see how we can use Yosys to generate a netlist. I have saved the required files in a directory. So I will change to yosys_codes. Let's see the contents of the directory. So there is a library file, a Verilog design called top.v and a TCL script that has all the commands required for the synthesis. The Verilog file and the TCL script are provided in the tutorial sheet as well. Let's see the Verilog design. So there is input a, b, clk, select and the output out and it has a combinational logic part defined using assign block and the sequential logic that is always block. For select is equal to 0, y is assigned a.

For select is equal to 1, y is assigned b. This behavior is similar to 2 to 1 mux. At the positive edge of the clock, y is latched to out. Otherwise out holds the previous value. This behavior is similar to a flip flop.

Next is the library file. You can find it in the NLDM folder. Let's see the TCL script. Here the design constraint that is an SDC file is not provided. So the generated netlist will be mapped for minimum area constraints.

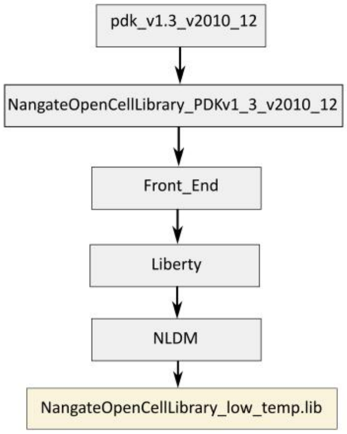

pdk_v1.3_v2010_12
NangateOpenCellLibrary_PDKv1_3_v2010_12
Front_End
Liberty
NLDM
NangateOpenCellLibrary_low_temp.lib

Figure 1: Libaray file location

The first step is to read the Verilog design that is top.v. The techmap command maps to the internal library. The dfflibmap maps sequential logic part of the design to a flip flop. abc -liberty maps the assign block.

The unused extra wires and cells are removed using the command clean. Finally the generated netlist is exported in.v format. The details of all the commands can be found here. Now launch the Yosys tool by typing in the command.

```
$ yosys
```

The command to run the TCL script is

```
Yosys> script yosys_commands.tcl
```

This is the name of the TCL script in the working directory. All the commands are executed. Now exit the yosys with Ctrl Z. Let's check the contents of the directory.

Let's see the generated netlist. The input and output pins are mapped to wires and a mux and a D flipflop are instantiated from the Nangate library file. Finally the netlist will look like the image shown here. This ends the demonstration of Yosys. Thank you very much.