Lab 10: Digital system Synthesis Using Synopsys Design Analyzer

Part 1: Introduction

Synopsys Design Compiler is a widely used Logic Synthesis and Optimization tool. Logic synthesis translates textual circuit descriptions like Verilog or VHDL into gate-level representations. Optimization minimizes the area of the synthesized design and improves the design's performance. The HDL description can be synthesized into a gate-level net-list composed of instances of the standard cells.

In this laboratory you will synthesize an Arithmetic Logic Unit (ALU) in Synopsys. After you have successfully synthesized the ALU, you will import the schematic generated by Synopsys into Cadence. You will also create an optimized Verilog description of the ALU from Synopsys that you will use with SOC Encounter to create a layout of the ALU. You will then use LVS to compare the schematic generated by Synopsys and the layout created by SOC Encounter.

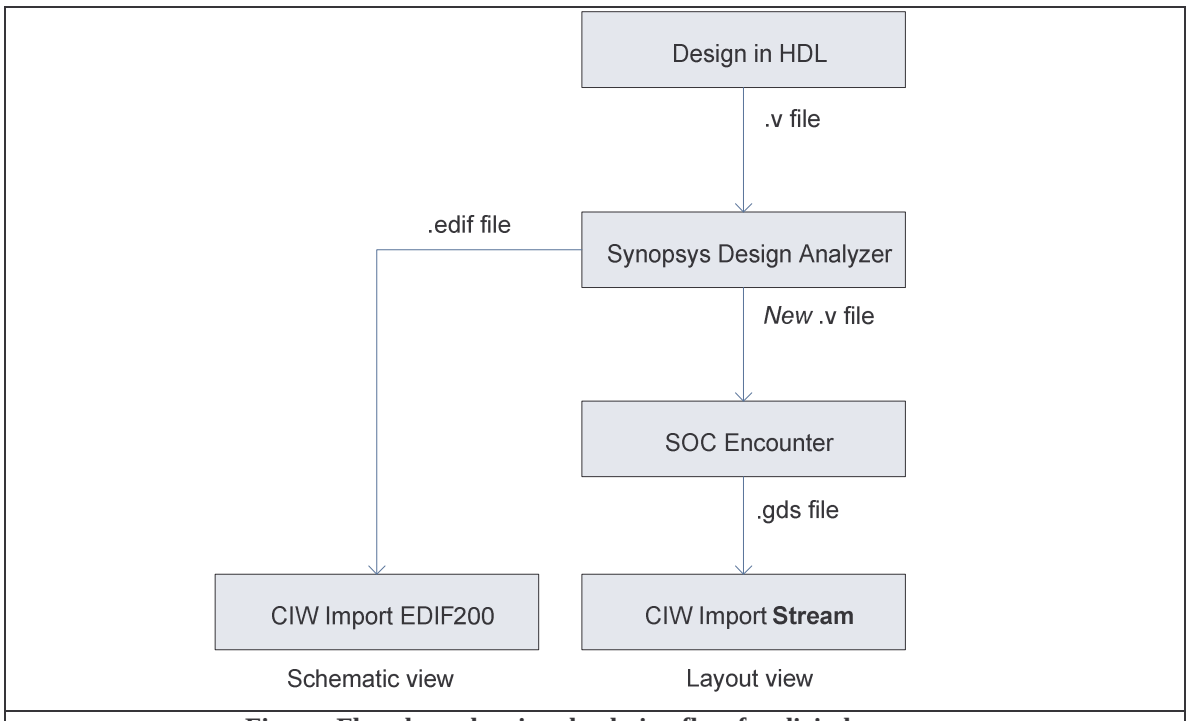

Figure: Flowchart showing the design flow for digital systems

Part 2: Environment Setup

You will need to add the Synopsys suite to the list of commands in your path. You can do that by editing your ~/`.software` file and adding a new line that contains the string `synopsys`. You will need to log out and log back in again so the tools will be loaded in your login profile.

We would like to keep our Synopsys work in a separate directory. In your ~/*ee434* directory, create a *synopsys* subdirectory for this purpose. Change into this new directory, copy the file `$ISU_LOCAL/synopsys.tar` here, and uncompress it. You can customize the Synopsys tool and

store the settings in a file named *.synopsys_dc.setup*. The Synopsys tool reads this file upon startup and applies the customizations. Open this file with your favorite editor to a few options. First, change the designer name to your own name. Next, find the field *edifout_designs_library_name* and change its value from "ckt_synopsys" to "alu". When a schematic generated by Synopsys is imported into icfb, the design are imported into the name of the library specified here. If a library by this name is not already present and visible in the Library Manager, one is automatically created for you.

Part 3: Synthesis

Now you are ready to synthesize a design. Run the following command from the *synopsys* directory.

design_analyzer &

The Design Analyzer window should appear.

1 - Command Window

Begin by verifying the environment has been initialized correctly. Click on **Setup→Defaults.** A *default* window appears. Verify that the information is pre-filled. Close this window by hitting *Cancel*. If you do not see your own name here, either you did not start design_analyzer from the correct `~/ee434/synopsys` directory or you did not make the customizations in Part 2. Make sure you correct this before proceeding further.

The next step is to analyze the source file that contains the HDL description of your design. Before proceeding, open the *Command Window* by choosing **Setup→Command Window**. *Command Window* is where all warnings and errors incurred during synthesis are displayed. All information displayed by various reports and menu windows are reflected in the *Command Window*.

Most importantly all commands entered via the menus of Design Analyzer are echoed, in proper Synopsys syntax, which means the same action can be performed by either using the pull-down menus or by typing appropriate commands in the *Command Window*. Pay attention to the information displayed in the *Command Window* at all times.

2 - Analyze Design

Choose **File→Analyze** and select the design that you want to synthesize. For our case, select `alu_4.v`. However, the file `alu_4.v` includes instantiations of modules contained in the files `add_4.v`, `sub_4.v`, `or_4.v`, and `xor_4.v`. In order to correctly synthesize our design, we need to analyze the main file () and its dependencies. In the Analyze form, hold down *shift* and click on all the files mentioned above. The file names you click will appear in the File Name window. Below the file selection, you can choose the library to analyze the file in, choose *WORK*. Make sure the format of the input files set to *Verilog*. Click OK.

The *Analyze* command checks HDL syntax and Synopsys rule checking. The intermediate files will be saved to the design library (the *WORK* directory). Click *Cancel* to dismiss the *Analyze* window.

3 - Elaborate the design

The next step is to elaborate the Verilog files read earlier. The Elaborate command reads the intermediate files created by Analyze. Click on **File→Elaborate** to open the Analyze form. Select the *WORK* library and choose the design you have just analyzed (`alu_4(verilog)`) . Click OK and

then on *Cancel* to dismiss the Elaborate window. The Design Analyzer window now displays the icons for the elaborated blocks. The buttons on the left side of the Design Analyzer window can be used to select different views for a design. Clicking on a block and then the view buttons on the left side of the Design Analyzer window allows you to see the different levels of each design.

The Analyze & Elaborate commands can be performed in a single step by the **File→Read** command.

4 - Compile the Design

After doing the Analyze and Elaborate steps, you will get an intermediate schematic representation of your design for each file you read in earlier. After Compiling, Design Analyzer generates the final design that uses the standard cells used in icfb. The final design should satisfy any constraints specified by the user and can be imported into IC. To compile the design, first double click on the `alu_4` to switch to its "Symbol view" (Check the bottom-right corner of the window to see the current view). Double click on the symbol view to see a preliminary schematic. While viewing the Schematic view, type `compile` in the *Command Window*. Once the compilation is successful, you will see a 1 just before the command prompt in your *Command Window* signifying success. Simultaneously, the schematic in the main window will change to use only the gates available in our standard cells library, isucells.

5 – Exporting the design in Verilog format

After the design has been compiled, an optimization of the logic has been performed. To create a layout of this synthesized design, we need to export the design in a format that can be read by a Place-and-Route tool such as SoC Encounter (introduced in a separate lab). Since our target is Encounter, we will export the synthesized and optimized design in the Verilog Format. Note that the Verilog file that we will now save will use actual gates from the standard cell library (isucells) instead of the behavioral Verilog we read into Synopsys. The Verilog file can be generated using the option of **File→Save As.** A save file window will pop up. In this window, changed to File format to Verilog and type any new name for the target file (e.g. `alu_4new.v`). Click OK. This new Verilog file can be used by Encounter to generate a layout.

6 – Exporting the design to ICFB

EDIF is the file format in which you can import schematics to Cadence. After making sure that you are in the "Schematic View", click on **File→Save As**. In the Save as window, change the file format to *EDIF* and type the filename to be `alu_4.edif`. Click *OK*. You can now close the Synopsys Design Analyzer window.

7 - Importing the design into ICFB

Open the icfb environment. At the CIW select **File→Import→EDIF200**. In the import EDIF200 window, enter the complete path to the EDIF file you generated in the previous step in the **Input Files field.** In the **Sheet Symbol Library**, change the entry to **NCSU_Sheets_8ths**. Click OK. You will get some warnings but should not get any errors. If you do not get any errors, the importing process was successful. The imported design is in the library "alu" (or a different name that you specified in Part 2). Locate this target library in Library Manager, right click on the name, click on

Properties…, and make sure that the pop-up form shows that it is attached to the ami06u process. If the library is attached to a different process, attach it to the ami0.6u process.

Part 4: Layout Versus Schematic verification

We will use SOC Encounter again to create the layout for the synthesized Verilog code. The steps to follow are found in the previous lab. Repeat the procedure outlined in that lab to generate the layout. Note that the top cell name for use in Encounter is still alu_4, not alu_4new. You may have saved the file by the name alu_4new but the top cell in the design was always called alu_4.

Note: Take into account the size of the ALU when doing the placement. Use your own judgment in choosing a number of rows or columns. You might have to run this step many times until you get an acceptable placement.

LVS

You should perform an LVS on the layout and schematic of alu_4. In order to achieve a netlist match, you must correct all of the DRC errors in the layout imported from SOC Encounter. Pay special attention to pins and pin names and the well spacing between instantiated standard cells.