ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT,
UNIVERSITY OF CYPRUS

ECE 407 Computer Aided Design for VLSI

Cadence Encounter™ RTL Compiler Ultra

Table of Contents

This tutorial aims in giving you the basic knowledge to start using **Cadence Encounter™ RTL Compiler Ultra.** This is an introductory tutorial and it does not replace the user guide that can be found in $ece407dir/RTLcomp/rc_user.pdf. To access the file, copy it to your working directory and transfer it to your local machine. Viewing this file from the remote machine is not recommended, since it may result to memory deficiency in the remote machine.

RTL Compiler Ultra is a powerful tool for logic synthesis and analysis for digital designs. It is fully compatible with all other Cadence Tools and especially with Cadence Encounter which is mainly used for physical design automation (floorplanning, placement and rooting).

1. Before Starting

Make a new directory **rtl_tut** in your working directory and copy all the files located in the **$ece407dir/RTL_comp/rtl_tut/** directory.

2. Getting Started

Set your current directory to *rtl_tut* and launch the RTL Compiler by typing in a terminal prompt: *rc –gui* with no & sign in the end. You will get this screen:

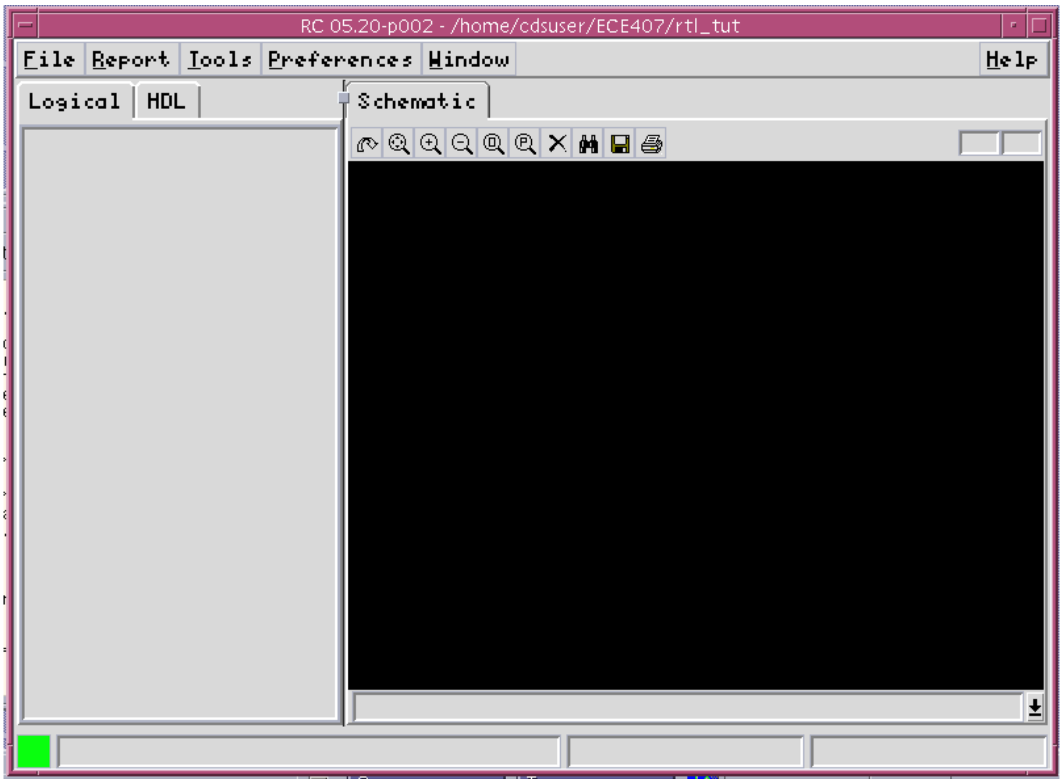


Fig. 1 RTL Compiler GUI

Unlike other GUI interfaces, the console this time is the initial window from which you launched *rc* (that's why it had to be launched in the foreground). In this console you can give commands that have effect in the GUI interface and view all the error and warning messages regarding the synthesis procedure.

3. Loading and Synthesizing

In order to read your design file you have to run *read_hdl* command in the console, followed by your design name. Add a *–vhdl* argument to specify that vhdl is the description language used for the RTL description. For this tutorial we will try to synthesize an 8-bit full adder consisting of 8 1-bit full-adders. Load your design by giving to the console prompt the following command:

```
        rc:/ >  read_hdl –vhdl Adder8.vhd
```

If the file exists in your working directory you will get a “file loaded” message in the GUI’s status bar. Next, you have to elaborate and synthesize your design. First run the following commands to set the library of standard cells to be used with your design.

```
rc:/ >  set_attribute  lib_search_path  /ECE407/RTLcomp/osu_stdcells/lib/tsmc018/signalstorm
rc:/ > set_attribute library {osu018_stdcells.lib}
```

Now, apply elaboration to your design by giving the *elaborate* command in the console. After elaborating you can see the netlist of your design in the GUI window (Fig.2).

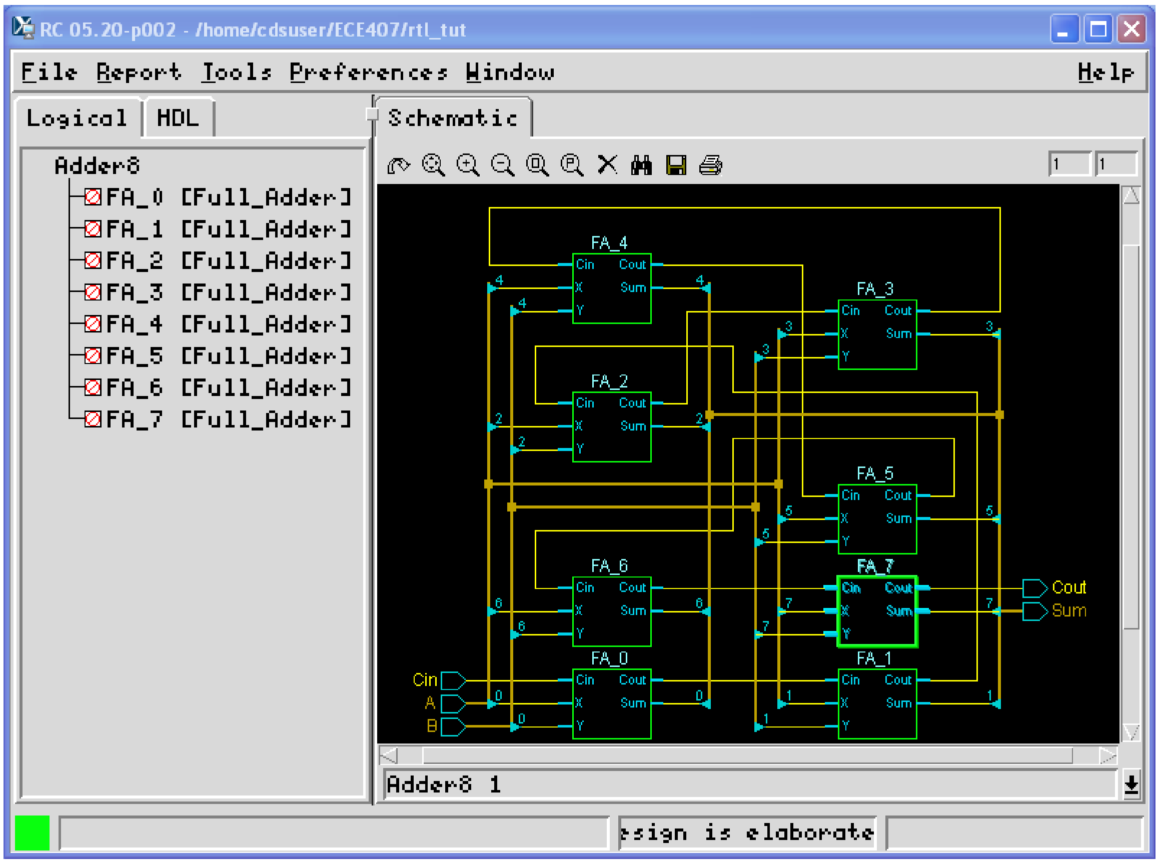


Figure 2. An 8-bit adder design after elaboration

Note that the 8-bit adder design consists of 8 1-bit Full Adders which, however, appear with a forbid sign on the side . This indicates that the Full Adder design has not been elaborated prior to the elaboration of the 8-bit Adder. Run through the loading and elaboration actions described above to build the main component of the 8-bit Adder contained in the Full_Adder.vhd file. Note that when reading the Full_Adder.vhd file you will get some warnings about some part of the Adder's code that cannot be synthesized. The tool informs you that it ignores all the time information contained in this file, in other words, all the variables of type TIME and all the "after" statements. To override these warnings, appropriately modify the VHDL file by removing all the unsensitizable code. The GUI window has no more forbid signs next to the Full Adder designs, and you can freely navigate through the netlist, by zooming in and out, selecting components or even inspect designs at lower levels by double-clicking on them. Observe that by clicking on the HDL tab on the GUI window, you can get the source code of design selected.

You are now ready to synthesize the 8-bit Adder. To do so run the following command:

```
rc:/ > synthesize -to_generic
```

Despite the fact that a synthesis succession message appears in the console, the synthesis process continues for some more time. Wait until no progress bar is active in the GUI window. If you try to zoom into a FA component of the design you will get the same circuit as before synthesis. That occurs because elaboration is essentially a synthesis process with generic library cells. If you run the next command, instead:

```
rc:/ > synthesize -to_mapped
```

you will have the design mapped to the specified technology defined in the library attribute you set before. The name of the library for this tutorial is osu018_stdcells.lib, which is a description of the TCMS standard cell library with 0.18 micro technology. If you try to zoom now in some of the components you will get a totally different netlist for the selected component, since the technology mapping found a Full Adder component in the standard cell library. Try to explore more options in the *synthesize* procedure by giving the command: *man synthesize*. As in a UNIX shell, the command **man** displays help on the selected command.

You can write your synthesized design in a Verilog description, which is suitable for importing the design in the Encounter tool. You will be introduced to the Encounter tool later in this class. The command used for writing a file in the synthesized Verilog format is:

```
rc:/ > write_hdl  –generic Adder8 >Adder8_synth.v
```

or for the technology mapped description :

```
rc:/ > write_hdl  –mapped Adder8 >Adder8_synth.v
```

If you don't put the > [filename] option, the synthesized Verilog description is shown in the console display.

Going through this loading and synthesize procedure from the command line, is not always efficient. Especially for larger designs with many **hdl** files, the procedure becomes very time-consuming and error-prone. The RTL Compiler provides an easy and efficient way of batching commands into TCL files (TCL is a popular scripting language). An example file, called **rtl.tcl** has been given to you. Open the file and see the description of the various commands. It is nothing more than all the commands given in the previous procedure. To execute ALL the commands in that file, go to the GUI window and click on menu *File* → *Source Script*. Then select the **rtl.tcl** file and click OK. You will see that all the steps you have done before are now performed in a single execution. The resulting window is like the one in Figure 3.

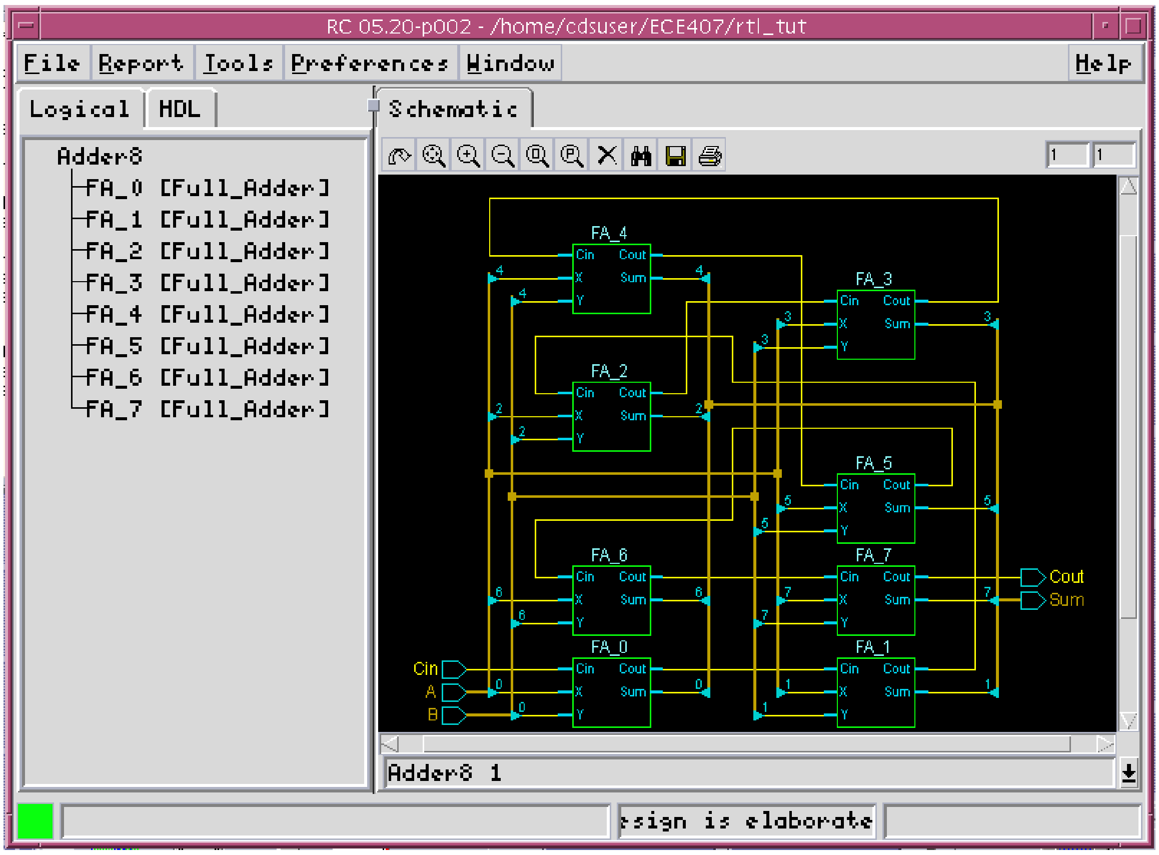


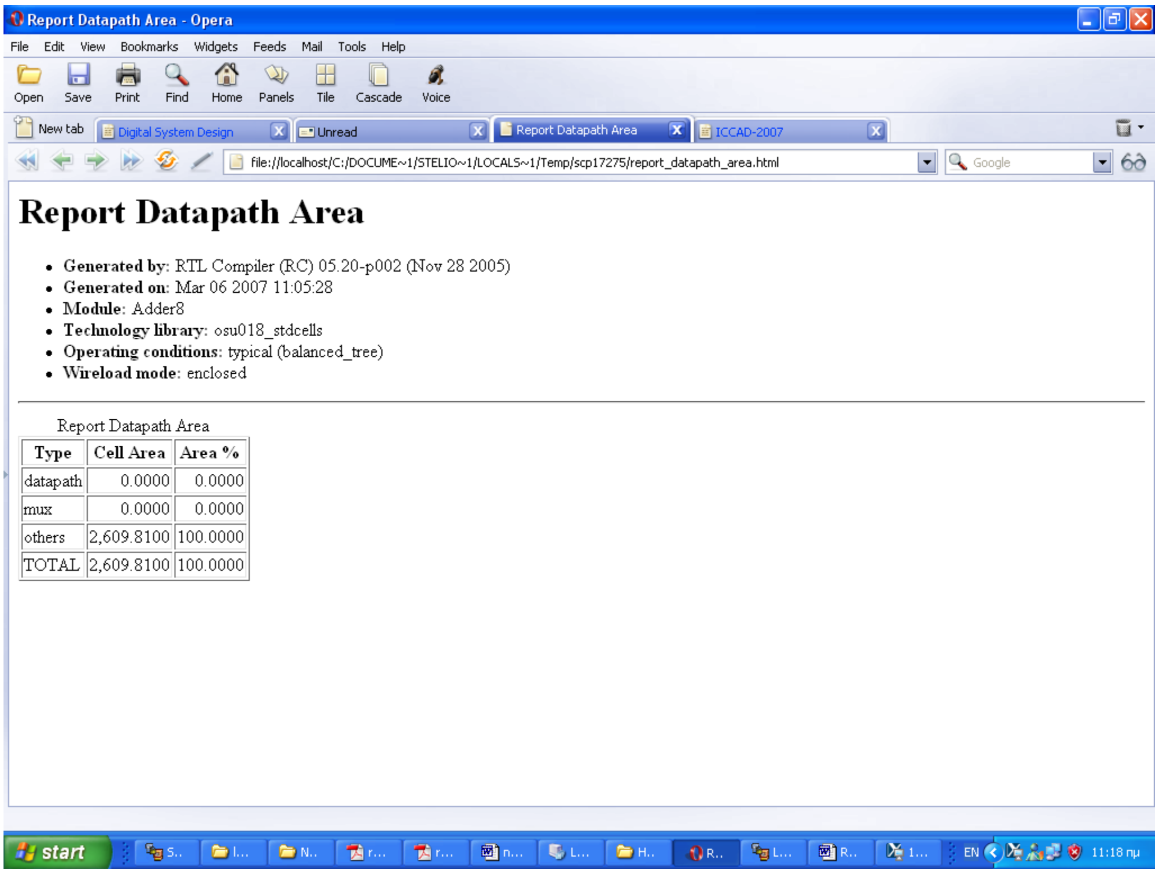
Figure 3. Design after full synthesis procedure

Try to change the various options in the commands located in the tcl file. Be careful not to re-run the command from the beginning. A second loading of the same file will import a second instance of the same design. To modify the existing design exclude the *read_hdl* command from the tcl file. You will use such tcl files in the assignments.

4. Report and Analysis

The RTL Compiler GUI, visualizes some of the various reporting and design analysing features of the tool. Start by clicking the menu *Tools* → *Object Browser*. You will get a new window reporting all the designs that are active in the current instance of the tool, the HDL libraries and the user defined libraries. Take some time to explore the various reported values and information. Give special consideration to the design synthesized at the previous Section.

Next, click on menu *Report*. Here, you can find information about your design. Start with the option *Netlist* → *Statistics* and *Netlist* → *Area (Press OK in Report Area Window)*. Here you can see some information regarding the area occupied by your design, the logic used or other components. The second option provides analytical area information per component. You can even export your design statistics in HTML (*exported in current directory*) format by clicking the corresponding Button (Fig 4). These reporting features are extremely useful when trying to do optimizations in your design. You can compare the different synthesis results of the same design when having different optimization parameters. Take some time to explore the other reporting options of the tool and interpreter their significance in the overall synthesis procedure. Leave the tool at any time by clicking *File* → *Exit*.



Report Datapath Area - Opera
File Edit View Bookmarks Widgets Feeds Mail Tools Help
Open Save Print Find Home Panels Tile Cascade Voice
New tab
Digital System Design
Unread
Report Datapath Area
ICCAD-2007
file://localhost/C:/DOCUME~1/STELIO~1/LOCALS~1/Temp/scp17275/report_datapath_area.html
Google
start
11:18 πμ

Report Datapath Area

- **Generated by:** RTL Compiler (RC) 05.20-p002 (Nov 28 2005)
- **Generated on:** Mar 06 2007 11:05:28
- **Module:** Adder8
- **Technology library:** osu018_stdcells
- **Operating conditions:** typical (balanced_tree)
- **Wireload mode:** enclosed

Report Datapath Area

Type	Cell Area	Area %
datapath	0.0000	0.0000
mux	0.0000	0.0000
others	2,609.8100	100.0000
TOTAL	2,609.8100	100.0000

Figure 4. An HTML Report generated by RTL Compiler tool.

5. The whole picture

You have seen the basic procedure of performing synthesis and analysis using the RTL Compiler tool. Synthesis must take place after extensive simulation of your design, in order to minimize any runtime or logic errors during design. Use the tool **NCLaunch**, introduced in a previous lab before doing the synthesis and analysis procedure, described here.

■