

A Very Short Tutorial to Timing Analysis using Encounter RTL Compiler

by Stelios Neophytou,

This tutorial aims in giving a brief introduction on how to use Cadence RTL Compiler to perform simple timing analysis actions. This is neither a full tutorial, nor a full manual about Timing Analysis using the RTL Compiler. For a full reference regarding timing analysis in RTL Compiler please see `rc_ta.pdf` located in `$CDSDIR/SOC5.2/doc/rc_ta/` directory.

1. Preparing the Analysis Instance

First take all the steps described in the RTC tutorial you have from the previous laboratory, to set the library, load a design and its subdesigns. For this tutorial, the files you are going to need are `Full_Adder.vhd` and `Adder8.vhd`, located in `/ECE407/rtc2/` directory. Elaborate both designs.

Now use the following commands to set some constraints on the inputs and the outputs of the circuit examined:

```
rc:/> define_clock -name clk1 -period 100000 [find / -port prots_in/*]
```

This command defines a clock-like periodic signal driving all inputs of your design, having a 100 ns period (100 000 ps).

```
rc:/> external_delay -output 200000 [find / -port ports_out/*]
```

This command specifies the output delay, i.e. the delay until the outputs are available after applying the inputs. Here, this delay is set to 200 ns (200 000 ps).

Next, synthesize the two designs without giving any optimization parameters, for the provided technology mapping (library `osu018_stdcells`). Note the various circuit characteristics after synthesis. Try to use the **report** command and explore the various options. Give special consideration to *timing*, *datapath*, *power* and *area* options.

2. Including and/or Excluding Logic Components

The **set_attribute** command can be used to assign values to various parameters used for the synthesis, elaboration and optimization procedure. Here, we show a couple of usages of this command. First, we will use it for excluding the usage of a specific component from the synthesis procedure:

```
rc:/> set_attribute avoid true FAX1
```

By this command, we ask the RTL compiler not to include the FAX1 cell in the synthesis procedure. FAX1 is a description of a Full Adder located in the library osu018. If you push the TAB button after this command, you will get the full path of the library cell:

```
rc:/> set_attribute avoid true /libraries/osu018_stdcells/libcells/FAX1/
```

The later can be read as “avoid using FAX1 cell, located in the standard cells library named osu018_stdcells”. Try to re-synthesize the Adder design. Observe that when you double-click in the Full_Adder component, you get a different schematic description than before. In order to be able to compare among the two synthesis outputs, you can re-load the design and elaborate it. Observe that you get a second instance of the Adder8 design. Remove the exclusion constraint, by giving the following command:

```
rc:/> set_attribute avoid false FAX1
```

Synthesize the second design, with the mapped technology option, by giving the name of this second design (probably Adder8_8). Compare the two schematics in terms of implementation, area, power and the size of the largest path. Note, that this way you can apply the synthesis process using different components, targeting different goals, like minimizing the overall area, minimizing the propagation delay of the circuit and so on. To help you explore as much options as possible, the following command does the opposite of the previous one, i.e. considers a specific component (cell) to be enforced in the synthesis process:

```
rc:/> set_attribute preserve false <libcell_name>
```

```
rc:/> set_attribute avoid false <libcell_name>
```

These two commands should be used in conjunction, to allow enforcing of the specific cell, in the synthesis process.

If you want to remove any designs from your workspace, apply the **rm** command:

```
rc:/> rm <design_name>
```

Note that design_name has been now removed from your gui window. By giving

```
rc:/> rm /designs/*
```

you can delete all the previous compiled designs, illustrated in the gui window and located into RTC memory.

Take some time to explore the various options of the **set_attribute** command. Try to optimize anyone of the circuit characteristic (area, longest path etc) by excluding or enforcing the different components in the library.

■