Tutorial for Design Compiler

STEP 1: Login to the Linux system on Linuxlab server. Start a terminal (the shell prompt). (If you don't know how to login to Linuxlab server, look at here)

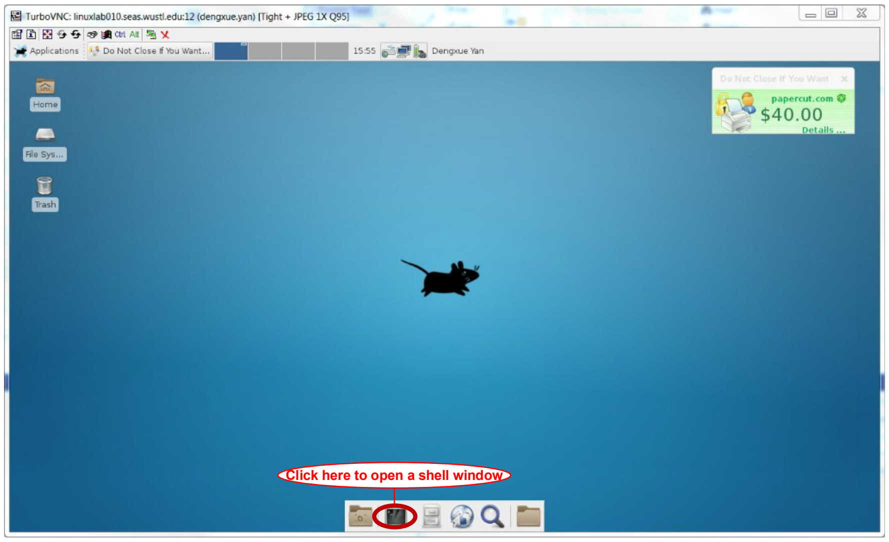

TurboVNC: linuxlab010.seas.wustl.edu:12 (dengxue.yan) [Tight + JPEG 1X Q95]
Applications
Do Not Close If You Want...
15:55
Dengxue Yan
Home
File Sys...
Trash
papercut.com
$40.00
Details ...
Click here to open a shell window

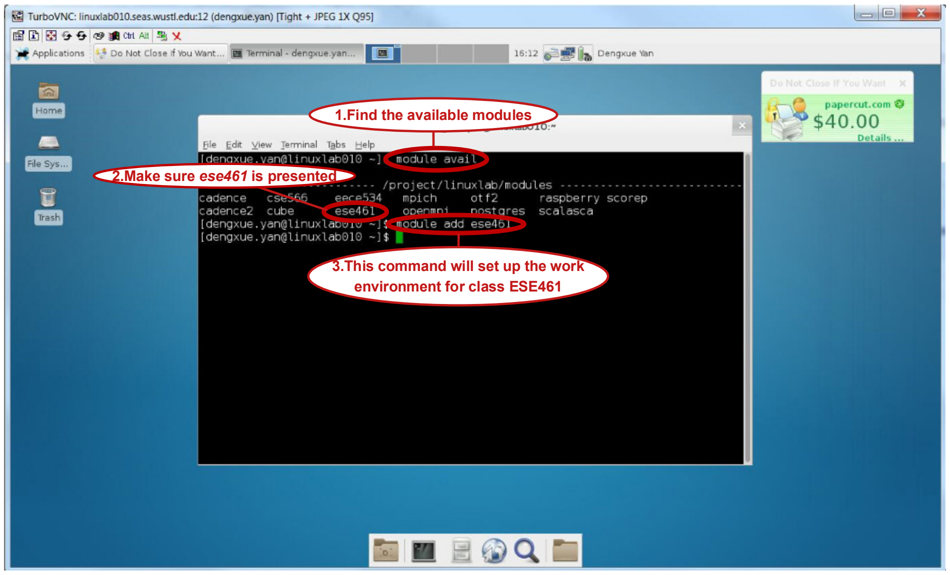
Fig. 1 The screen when you login to the Linuxlab through equeue

STEP 2: Build work environment for class ESE461.

In the terminal, execute the following command:

module add ese461

This command will build work environment for class ESE461. You could perform "*module avail*" in the terminal to find the available modules on Linuxlab before you do "*module add ese461*" . Make sure ese461 is presented when you execute this command.



```
[dengxue.yan@linuxlab010 ~]$ module avail

------------------------------ /project/linuxlab/modules ------------------------------
cadence   cse566    eece534   mpich     otf2      raspberry scorep
cadence2  cube      ese461    openmpi   postgres  scalasca
[dengxue.yan@linuxlab010 ~]$ module add ese461
[dengxue.yan@linuxlab010 ~]$
```

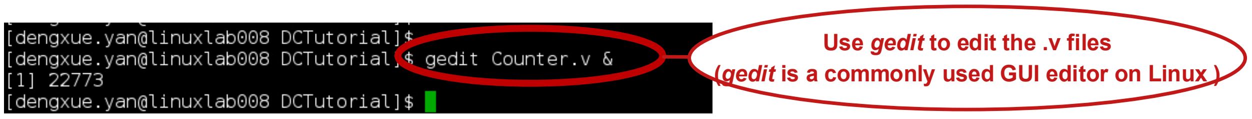
TurboVNC: linuxlab010.seas.wustl.edu:12 (dengxue.yan) [Tight + JPEG 1X Q95]
Applications
Do Not Close If You Want...
Terminal - dengxue.yan...
16:12
Dengxue Yan
File Edit View Terminal Tabs Help
1.Find the available modules
2.Make sure ese461 is presented
3.This command will set up the work environment for class ESE461
papercut.com
$40.00

Fig. 2 Build work environment for class ESE461 using module

STEP 3: Getting started with Verilog

- **Creating a new folder (better if you have all the files for a project in a specific folder).**
- **Enter into this new folder and start writing your Verilog script in a new file (.v file). Example code for modeling an counter is [here](#)**

Note: We do not need tech bench here, because test bench is for simulation, but here we do RTL synthesis, which is the next step after simulation.



```
[dengxue.yan@linuxlab008 DCTutorial]$
[dengxue.yan@linuxlab008 DCTutorial]$ gedit Counter.v &
[1] 22773
[dengxue.yan@linuxlab008 DCTutorial]$
```

Use *gedit* to edit the .v files
(*gedit* is a commonly used GUI editor on Linux)

Fig. 3 Edit module file using gedit

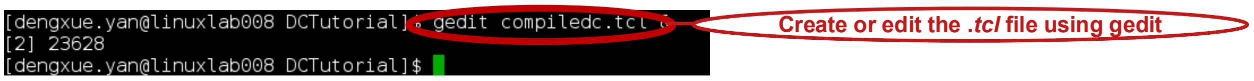
STEP 4: Writing constraints file - ".tcl" file

Example *tcl* file for *[counter](#)* above is [here](#).

Based on the example *tcl* file:

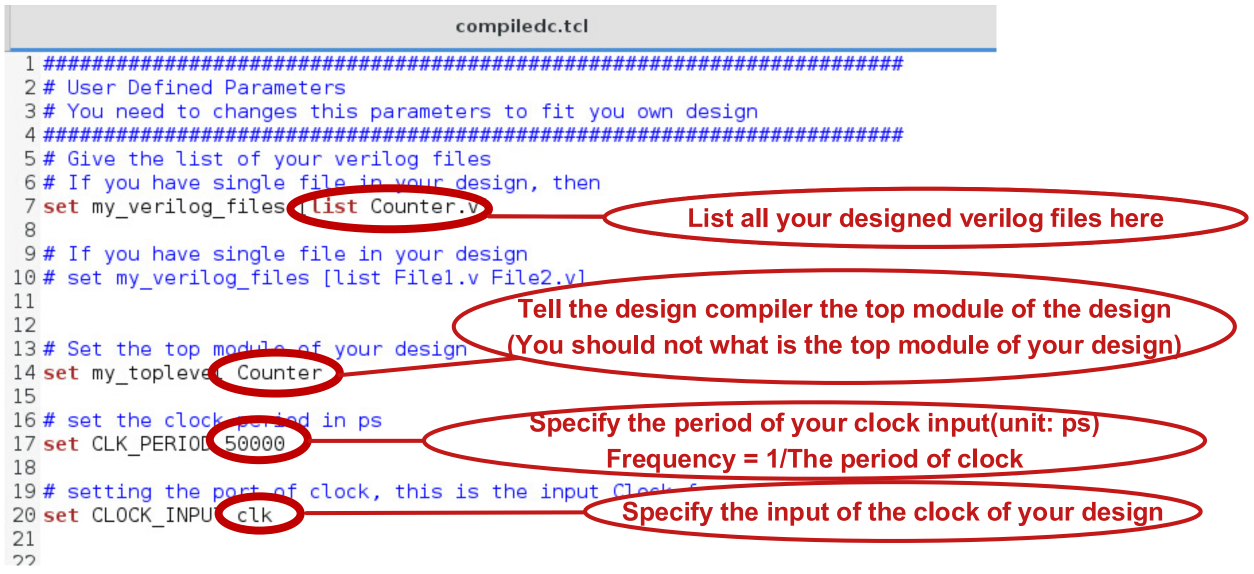
- **Lines where modifications are required specific to model:**
  - **Set Path to Verilog files**
  - **Top module of the design have to be specified (To specify the Hierarchy)**
  - **Specifying the clock port used in the model**
  - **Clock frequency at which the model is operated**

Note: Please go through the provided .*tcl* file to check other options by yourself.



```
[dengxue.yan@linuxlab008 DCTutorial]$ gedit compiledc.tcl &
[2] 23628
[dengxue.yan@linuxlab008 DCTutorial]$
```

Fig. 4 Edit tcl file using gedit



```
########################################################################
# User Defined Parameters
# You need to changes this parameters to fit you own design
########################################################################
# Give the list of your verilog files
# If you have single file in your design, then
set my_verilog_files [list Counter.v]

# If you have single file in your design
# set my_verilog_files [list File1.v File2.v]


# Set the top module of your design
set my_toplevel Counter

# set the clock period in ps
set CLK_PERIOD 50000

# setting the port of clock, this is the input Clock f
set CLOCK_INPUT clk

```

List all your designed verilog files here
Tell the design compiler the top module of the design
(You should not what is the top module of your design)
Specify the period of your clock input(unit: ps)
Frequency = 1/The period of clock
Specify the input of the clock of your design

Fig. 5 Lines that modifications are required

STEP 5: Compile your code

- In the terminal, change the directory to where your model (*Counter.v*) and tcl(*compiledc.tcl*) files are present by using this command:
  cd <path>
  For example:
  cd ~/ESE461/DCTutorial/
  (Remark: '~' means home directory on Linux)
- Compile the files by typing in the terminal:
  dc_shell-t -f <file>.tcl
  In the above example, it should be:
  dc_shell-t -f compiledc.tcl
  After run this command, there might be some warning but no error presented in the terminal. Otherwise you need to check your code or *tcl* file and correct them according to the related messages.
  Note: The oldest messages might be lost due to too many lines printed on the terminal. In this situation you could redirect the output to a file:
  dc_shell-t -f compiledc.tcl > compiledc.log
  Then you can open *compiledc.log* using *gedit* and search in it. The ">" here is the sign for *Output Redirection* in Linux shell.

After successfully done STEP 1~ STEP 5, you will find the following report files(name of the files are defined in *tcl* file above) in your project folder:

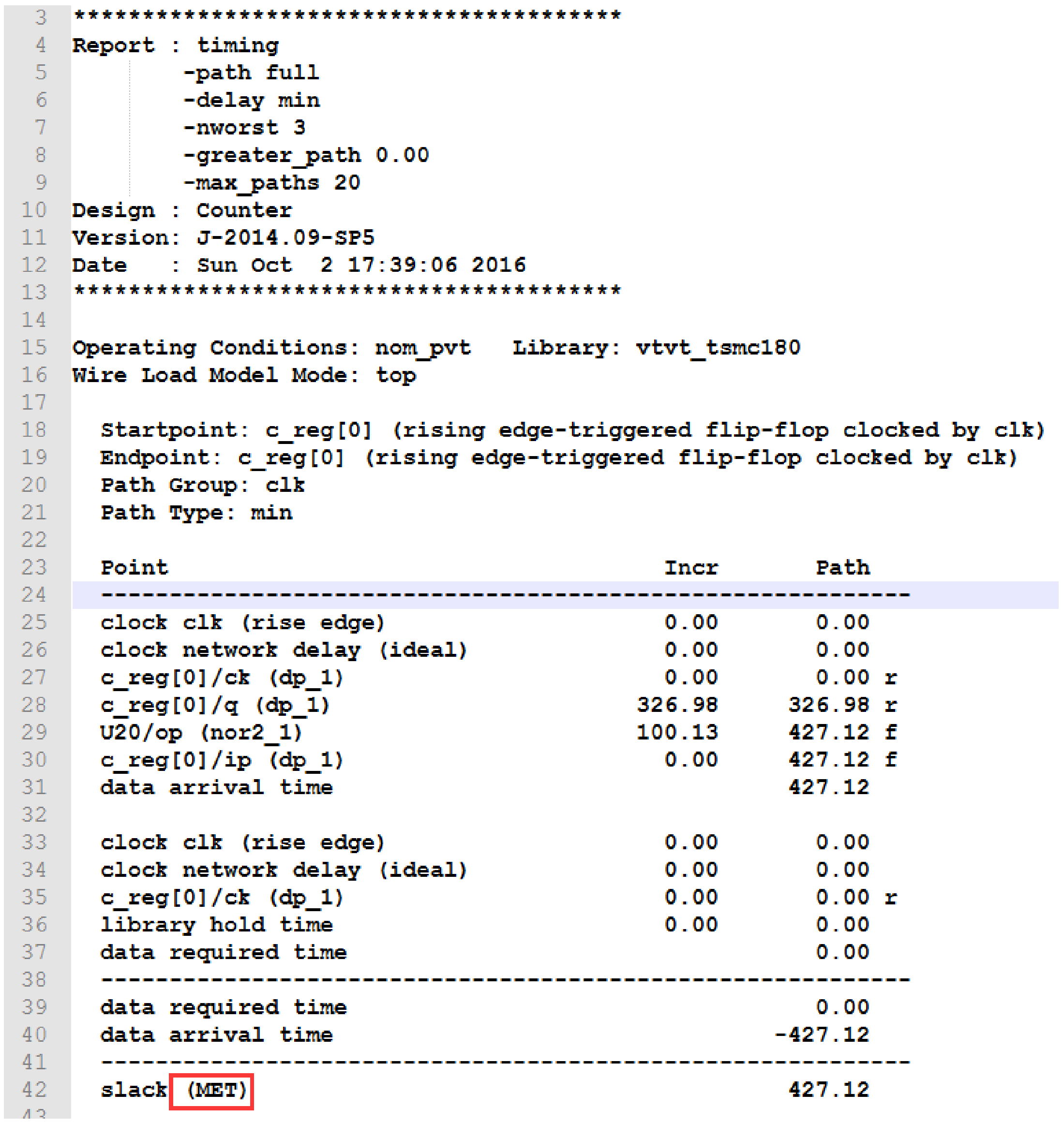
Note: *my_toplevel* following is defined in *tcl* file. In the above example *my_toplevel* is *Counter*

1.Timing Analysis reports:

- ***<my_toplevel>_min_timing.repC***
- ***<my_toplevel>_max_timing.repC***
- ***<my_toplevel>_out_min_timing.repC***

Make sure the timing report requirements are MET. You can observe which module in the design is giving the maximum delay and optimize accordingly.

Example:



```
****************************************
Report : timing
        -path full
        -delay min
        -nworst 3
        -greater_path 0.00
        -max_paths 20
Design : Counter
Version: J-2014.09-SP5
Date   : Sun Oct  2 17:39:06 2016
****************************************

Operating Conditions: nom_pvt   Library: vtvt_tsmc180
Wire Load Model Mode: top

  Startpoint: c_reg[0] (rising edge-triggered flip-flop clocked by clk)
  Endpoint: c_reg[0] (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: min

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  c_reg[0]/ck (dp_1)                       0.00       0.00 r
  c_reg[0]/q (dp_1)                      326.98     326.98 r
  U20/op (nor2_1)                        100.13     427.12 f
  c_reg[0]/ip (dp_1)                       0.00     427.12 f
  data arrival time                                 427.12

  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  c_reg[0]/ck (dp_1)                       0.00       0.00 r
  library hold time                        0.00       0.00
  data required time                                  0.00
  -----------------------------------------------------------
  data required time                                  0.00
  data arrival time                                -427.12
  -----------------------------------------------------------
  slack (MET)                                       427.12
```

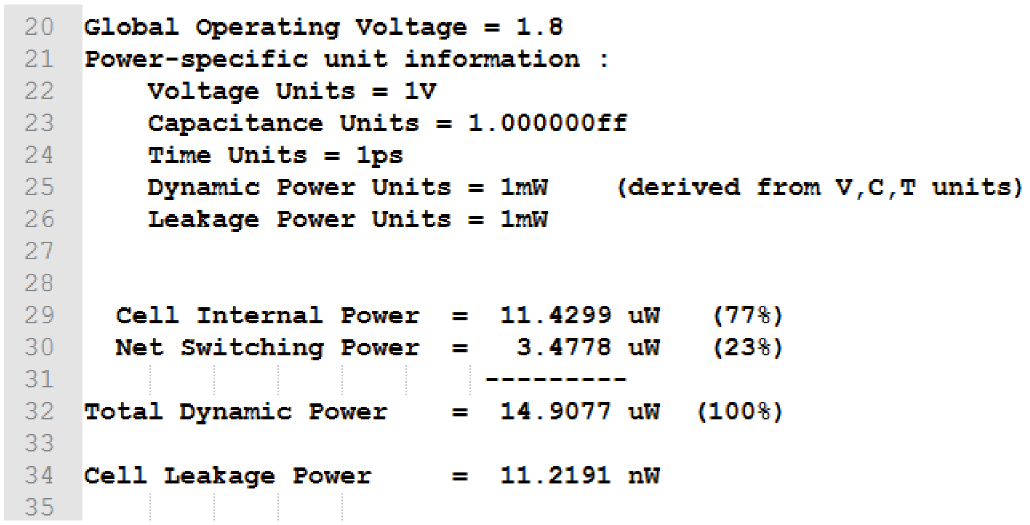
Fig. 6 Partial of timing report(Counter_min_timing.repC)

2. Power Analysis reports:

- *<my_toplevel>_power.repC*

Design Compiler gives the detailed information about the static and dynamic power.

Example:



```
Global Operating Voltage = 1.8
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000ff
    Time Units = 1ps
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1mW


  Cell Internal Power  =  11.4299 uW   (77%)
  Net Switching Power  =   3.4778 uW   (23%)
                         ---------
Total Dynamic Power    =  14.9077 uW  (100%)

Cell Leakage Power     =  11.2191 nW

```

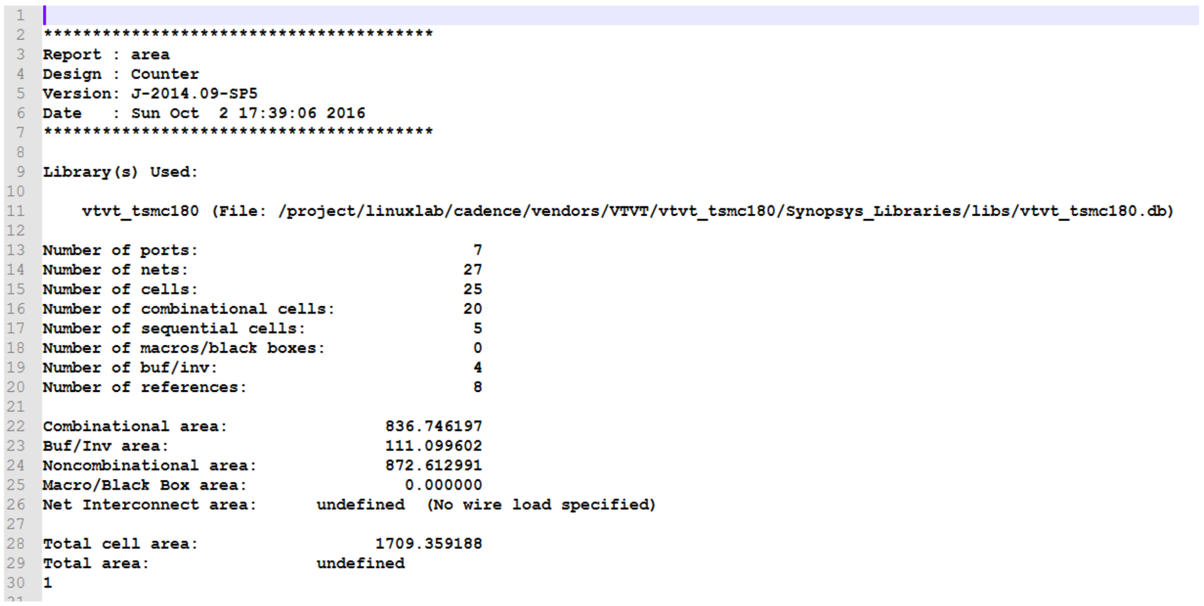
Fig. 7 Partial of power report(Counter_power.repC)

3. Area Analysis reports:

- *<my_toplevel>_area.repC*

Area report file generated using design compiler contains detail information about the size of each cell used for this model. Units for Virginia tech library are micro meters.

Example:



```

****************************************
Report : area
Design : Counter
Version: J-2014.09-SP5
Date   : Sun Oct  2 17:39:06 2016
****************************************

Library(s) Used:

    vtvt_tsmc180 (File: /project/linuxlab/cadence/vendors/VTVT/vtvt_tsmc180/Synopsys_Libraries/libs/vtvt_tsmc180.db)

Number of ports:                            7
Number of nets:                            27
Number of cells:                           25
Number of combinational cells:             20
Number of sequential cells:                 5
Number of macros/black boxes:               0
Number of buf/inv:                          4
Number of references:                       8

Combinational area:                836.746197
Buf/Inv area:                      111.099602
Noncombinational area:             872.612991
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:                  1709.359188
Total area:                 undefined
1
```

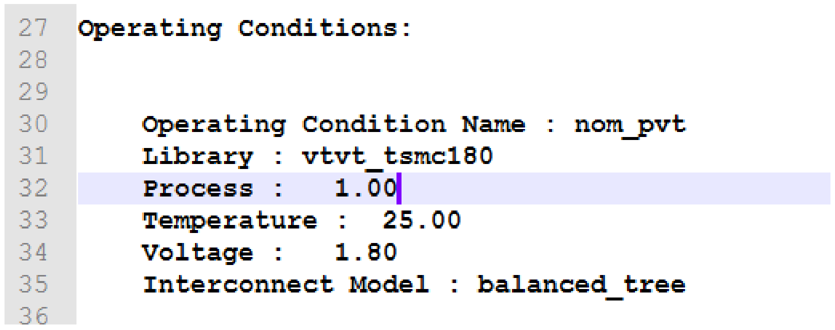
Fig. 8 Partial of area report(Counter_area.repC)

4. Design Analysis reports:

Important information found in this report is the operating conditions. VT library only support normal conditions for operation. But in technology libraries provided by the vendor there are some extreme conditions (WCCO – worst case conditions) available for getting the maximum values to verify the performance of the model.

Example:

- *<my_toplevel>_area.repC*

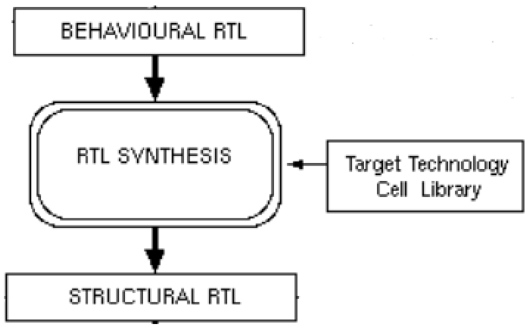


```
Operating Conditions:


    Operating Condition Name : nom_pvt
    Library : vtvt_tsmc180
    Process :   1.00
    Temperature :  25.00
    Voltage :   1.80
    Interconnect Model : balanced_tree

```

Fig. 9 Partial of design report(Counter_area.repC)

Additional Information for Design Compiler:

The process that Design Compiler does is RTL synthesis. This means, converting a gate level logic Verilog file to transistor level Verilog with the help of Technology library provided by the foundry.



Useful links:

http://lyle.smu.edu/~manikas/SDC_help/dvtut.pdf

http://www.ee.ncu.edu.tw/~jfli/vlsi21/lecture/dc.pdf