Exercise

1. Setup EDA Environment

% setdt vcs // set vcs for simulation
% setdt syn // set design compiler for logic synthesis

2. Simulate your design and generate switching activity information file.

a) Copy and Extract the file **/js1/songch/DA_VLSI/DA_VLSI_Lab5.tgz** to your home directory.

Select one from the designs in verilog/ and go on the following exercise.

b) Modify the testbench module to generate **.SAIF** file.

The following example shows the use of system tasks, in test bench module, to generate SAIF file for YOUR MODULE.

```
module your_module_test;
…      your_module #(16)   your_module_imp( …
      initial begin
            ……
            $set_gate_level_monitoring("rtl_on");
            $set_toggle_region(your_module_test.your_module_imp);
                  // module instance name: testbench_name.module_instance_name
            $toggle_start();
            ……
      end
    …
      always @ (posedge clk) begin
            if(done==0) begin
                  $toggle_stop();
                  $toggle_report("your_module.saif",1.0e-10,"your_module_test.yo
              ur_module_imp");
                  ……
                   $finish();
            end
        …
```

c) Build a simulator and watch waveforms.

i. Type the compile command of vcs tool

% vcs -full64 -PP +lint=all +v2k -debug -line **verilog/your_module.v**

ii. Type the simulate command of vcs tool

% ./simv

3. Use Tcl-scripts to synthesis your design.

Learn Tcl script syntax from Reference : Using Tcl With Synopsys Tools.
Link : **http://202.38.80.152/dc_doc/tclug.pdf**

a) Look through the following files:

rm_setup/common.tcl,
rm_setup/dc_setup.tcl, rm_setup/
dc_setup_filenames.tcl
rm_dc_scripts/dc.tcl

If you don't know the meaning of the commands, pls. refer to DC document "**Synthesis Tool Command**" http://202.38.80.152/dc_doc/syn2.pdf

b) Modify related Tcl script. (Refer the following Tutorial)

In **common_setup.tcl**:
DESIGN_NAME => **top module name** of your design
# The name of the top-level design
TARGET_LIBRARY_FILES => "saed32rvt_tt1p05v125c.db"
# Target technology logical libraries, we use Synopsys 32 nm standard cell library
DESIGN_INSTANCE => ""
# Your instance name, which you can find it in **your_module.saif.**
E.g. **your_module_test/your_module_imp**

In **dc_setup.tcl**:
RTL_SOURCE_FILES => " "
# Enter the list of source RTL files if reading from RTL

In **your_module.constraints.tcl**:
my_clock_pin => clock signal name of your design
# Clock Pin Name in your module
my_clk_freq_MHz => Target performance (frequency)
# Work Frequency for your module
input_delay => 1
# Input Delay for synthesis
output_delay => 1
# Output Delay for synthesis

Example for a timing constraints script:

```
set my_clock_pin
set my_clk_freq_MHz

set my_period [expr 1000 / $my_clk_freq_MHz]

set find_clock [ find port [list $my_clock_pin] ]
if {    $find_clock != [list] } {
    set clk_name $my_clock_pin
    create_clock -period $my_period $clk_name
} else {
    set clk_name vclk
    create_clock -period $my_period -name $clk_name
}

set input_delay
set output_delay
```

In **dc.tcl**:

Line 157: **set_leakage_optimization true**

Line 158: **set_dynamic_optimization true**

\# Power constraints for your design

c) Execute the script for synthesis.

DA_VLSI_Lab5> dc_shell-t -f rm_dc_scripts/dc.tcl | tee dc.log

Tips: Make sure **your_module.saif** and **your_module.constraints.tcl** exists in your Lab directory.

d) Check the "results" and "reports", and **dc.log for what happened, and "warning" and "errors"**.

Tips: Enter "reports" folder, and write down relative information such as **Cells, Slack, Area** and **Power.**

4. Modify Tcl-scripts to check the impact of key options on the synthesis, e.g., -flatten -no_autoungroup, -gate_clock, etc.

a) View the manual information for the command "compile_ultra", modify Tcl-scripts

In **dc.tcl**:

-Line 378: compile_ultra # Change the compile options for synthesis as you like

b) Pls. refer to Exercise 2, re-synthesis your module, and compare the synthesized results.

5. Add the option `-topographical` for the dc_shell.
   Pls. make yourself understand what will happen if you use "-topographical".
   Pls refer to the Design compiler user guide chapter 10, http://202.38.80.152/dc_doc/dcug.pdf

   DA_VLSI_Lab5> dc_shell-t -topographical_mode -f rm_dc_scripts/dc.tcl | tee dc.log

6. Increase the clock frequency from 200, 500, 1000 Mhz, check your synthesized results.

Supplemental Exercise: Learn to use DesignWare Building Block IPs (Optional)
The following is an example for use of the IP blocks in Synopsys Designware

```
module pipelined_multiplier(in1, in2, control, clk, product);
      parameter wordlength1 = 8, wordlength2 = 8;
      input [wordlength1-1:0] in1;
      input [wordlength2-1:0] in2;
      input control;
      input clk;
      output [wordlength1+wordlength2-1:0] product;
      // instance DW02_mult_5_stage
      DW02_mult_5_stage #(wordlength1, wordlength2) U1(in1, in2, control, clk, product);
endmodule
```

Copy the files for the instantiation and test of a pipelined multiplier in the following directory:

Build a simulator and check the simulation results
Note: you have to add the following options when building a simulator.
-y /soft1/synopsys/synthesis/H-2013.03-SP2/dw/sim_ver +libext+.v+
Synthesize the pipelined simulator.

For more building block IPs, please refer to the following documents in the course website
http://staff.ustc.edu.cn/~songch/dc_doc.htm
DesignWare Building Block IP Quick Reference
DesignWare Building Block IP User Guide
DesignWare Building Block IP Application Notes

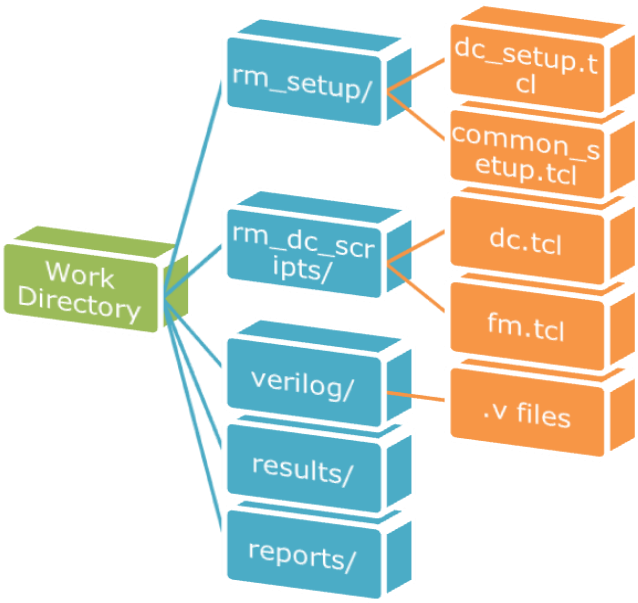
Tips:
Use "man command_name", in dc_shell promote, to view more information about the commands

Please refer to **http://staff.ustc.edu.cn/~songch/da-ug.htm** for the following resources
1. Verilog-basics for the Verilog HDL.
2. the Linux tutorial and simple vi manual, respectively, for the simple Linux command and use of **vi/emacs/gedit(recommended)** editor.
Please read the **error** information carefully if any.

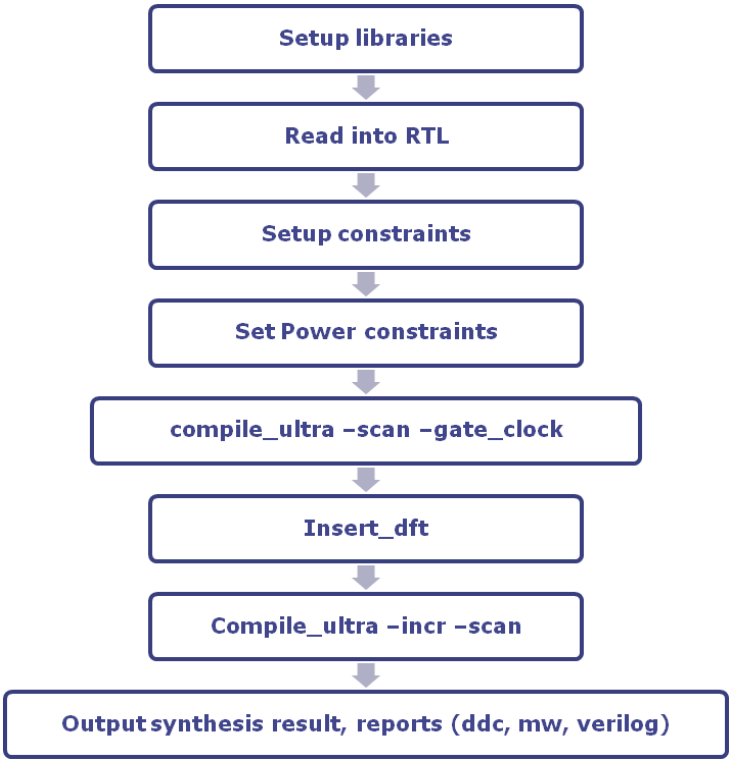
Tutorial on Synthesis using Synopsys

Reference design flow

DC Reference Methodology



DC-RM Script flow



Execution of the DC-RM

1. Modify the setup files to set library files, input data

common_setup.tcl → dc_setup.tcl

2. Customize the script for your design

rm_dc_scripts/dc.tcl

3. Execute the customized script

```
dc_shell –f rm_dc_scripts/dc.tcl | tee dc.log
```

Confirm the synthesized results

reports/ → results/