Lab for Physical Design

% setdt syn

% setdt icc

2. Use Tcl-scripts to synthesize your design.

Reference for Tcl script: Using Tcl With Synopsys Tools.

Link : **http://202.38.80.152/dc_doc/tclug.pdf**

a) Copy and Extract the file **/js1/songch/DA_VLSI/DA_VLSI_PD.tgz** to your home directory.Please do 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 **rm_setup/**

icc_setup.tcl

rm_setup/Makefile_zrt

If you cannot under the meaning of the commands, pls. refer to DC /ICC document "**Synthesis Tool Command**"

Links: **http://202.38.80.152/dc_doc/syn2.pdf** **http://202.38.80.152/icc_doc/icc2.pdf**

b) Pls. simply modify the script to **synthesize a mips_processor** in the directory v**erilog/**:

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

Pls. make sure you have the file for constraining your design, xxxx.constraints.tcl

(rm_setup/dc_setup_filenames.tcl: $DESIGN_NAME.constraints.tcl)

Pls. check the file dc.log for warning&errors to ensure your design has been well synthesized after you perform the above command.

You may choose to generate .saif file for switch power optimization or not.

How much is the timing slack?

How many cells do the design have?_____

How much is the power? _____

_____

c) Physical design.

Pls. look through the file **rm_setup/Makefile_zrt,**

and check the file **rm_setup/icc_setup.tcl** to confirm the following setup.

ICC_INIT_DESIGN_INPUT "MW"

// the file $DESIGN_NAME.mapped.ddc/.sdc, etc., obtained in Synthesis stage would be the inputs to the physical design (back-end design)

ICC_FLOORPLAN_INPUT "CREATE"

a). Run the following command to start your back-end design

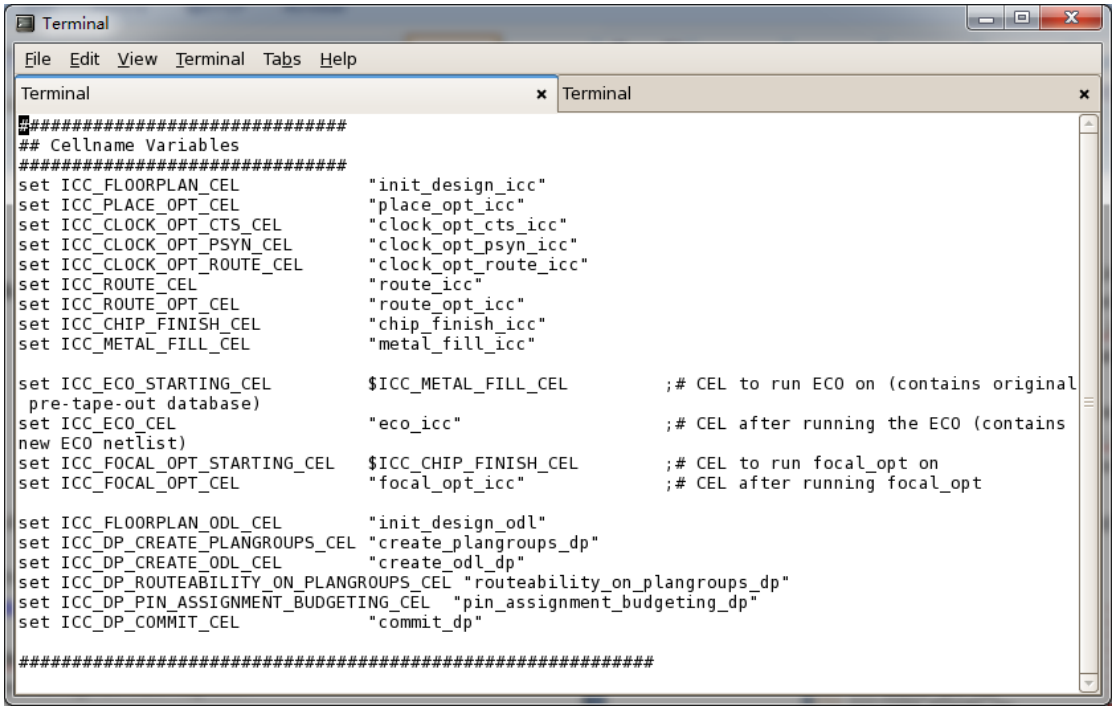
DA_VLSI_PD> make init_design_icc -f rm_setup/Makefile_zrt

Pls. use **'ls -l' to view dirs** generated in your current directory, and check the log file for errors or warnings.

Tips: If you want to redo this step, please execute the command

"make -f rm_setup/Makefile_zrt clean" first, or just remove the file 'init_design_icc', and run again the above command.

Some intermediate cells created during the physical design procedure.



View the file **rm_setup/Makefile_zrt,**

~%make ic will generate a cell with everything done. (**sign_off_drc** is the final step, metal_fill_icc is the final cell)

To check error, you can do step by step, for example,

~%make init_design_icc will get your design data ready.

Execute the script file: icc_scripts/init_design_icc.tcl

~%make place_opt_icc will generate a cell with standard cell placement done

Execute the script file: icc_scripts/place_opt_icc.tcl

….

~%make route_icc will generate a cell with routing done

Execute the script file: icc_scripts/route_icc.tcl

……

[Check the directory log_zrt/ for log files in every stage, warning and errors. If there are some errors, generally, you have to check and correct the setup files/scripts, and run the command again.]

After you "make ic", please check the timing reports.

If everything goes well, you can invoke icc_shell and view the layout.

~%icc_shell

icc_shell> source icc_setup.tcl

icc_shell> open_mw_lib library_name // open the milkway library, which is used to store layout. In this exercise, the directory with a suffix _LIB is generally a directory for Milkway library.

icc_shell> open_mw_cel cell_name // you can open any cells you created before.

icc_shell>start_gui // in the gui, you can try something.

Hints:

You can also invoke icc_shell in gui mode, by "icc_shell –gui"

icc_shell> source icc_setup.tcl

File->Open Library [The libraries are highlighted and end with _LIB]

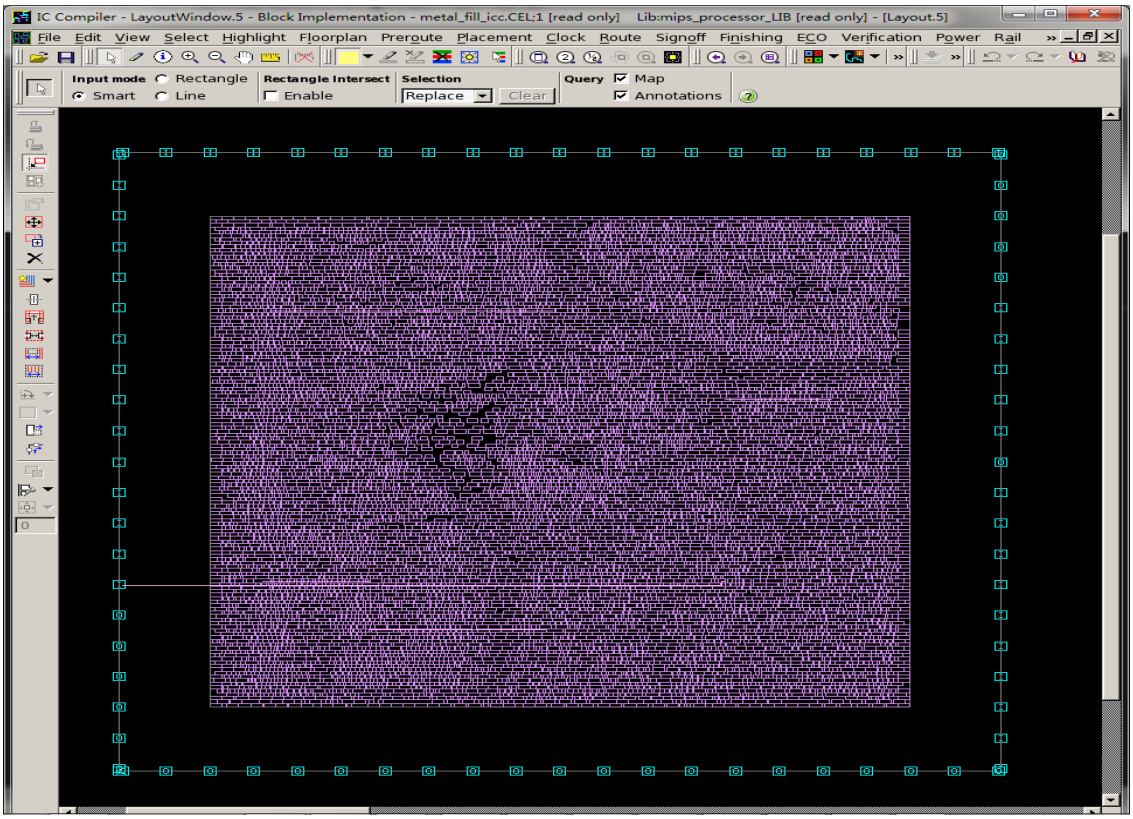
http://staff.ustc.edu.cn/~songch/da-ug.htm -> IC Compiler Documents, for documents of IC compiler: Command manuals, and error messages, etc.

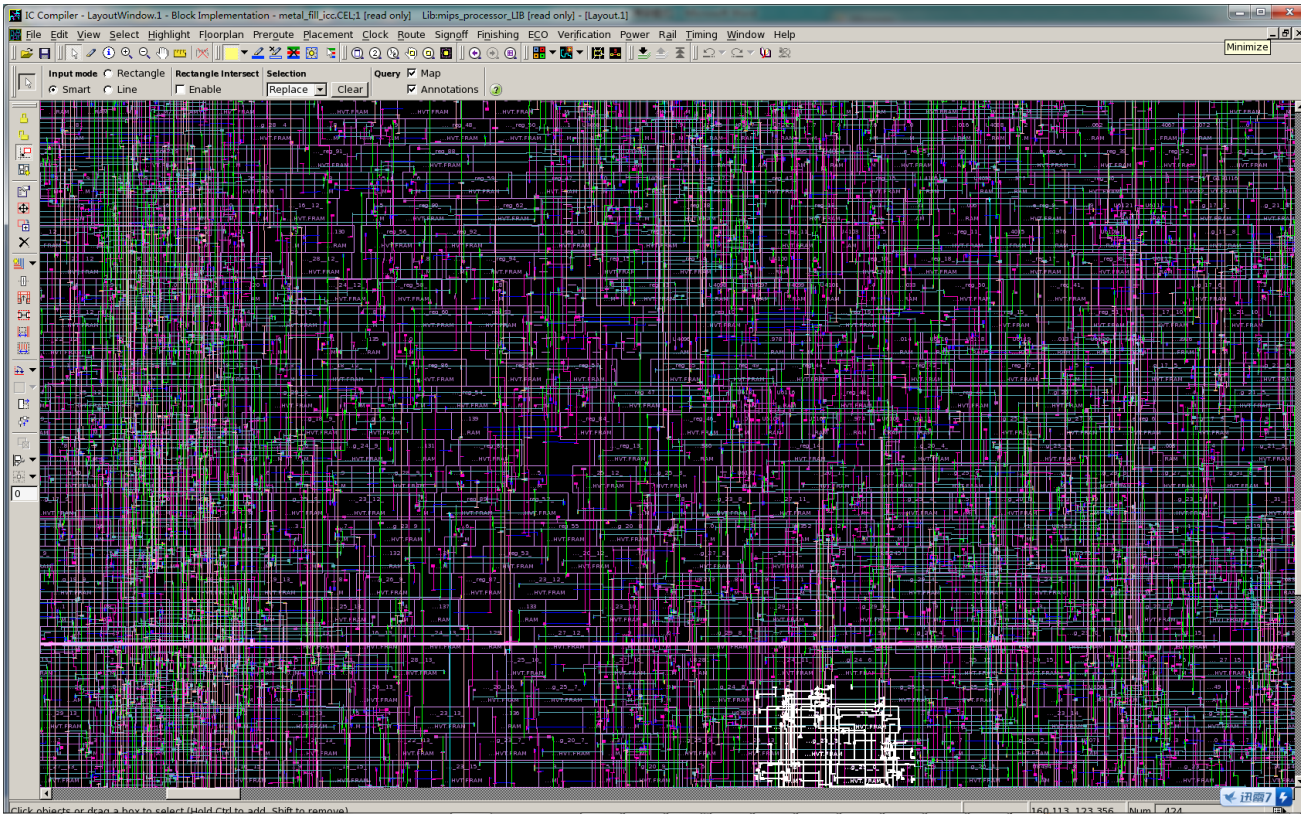
Or, Use "help command_name" to view the usage of the commands and "man command_name" to view the detailed explanation of the commands.

How much is the timing slack? _____

How many cells do the design have? _____

How much is the power? _____


IC Compiler - LayoutWindow.5 - Block Implementation - metal_fill_icc.CEL;1 [read only] Lib:mips_processor_LIB [read only] - [Layout.5]
File Edit View Select Highlight Floorplan Preroute Placement Clock Route Signoff Finishing ECO Verification Power Rail
Input mode Rectangle Smart Line
Rectangle Intersect Enable
Selection Replace Clear
Query Map Annotations


IC Compiler - LayoutWindow.1 - Block Implementation - metal_fill_icc.CEL;1 [read only] Lib:mips_processor_LIB [read only] - [Layout.1]
File Edit View Select Highlight Floorplan Preroute Placement Clock Route Signoff Finishing ECO Verification Power Rail Timing Window Help
Minimize
Input mode Rectangle Smart Line
Rectangle Intersect Enable
Selection Replace Clear
Query Map Annotations

3. (Optional)

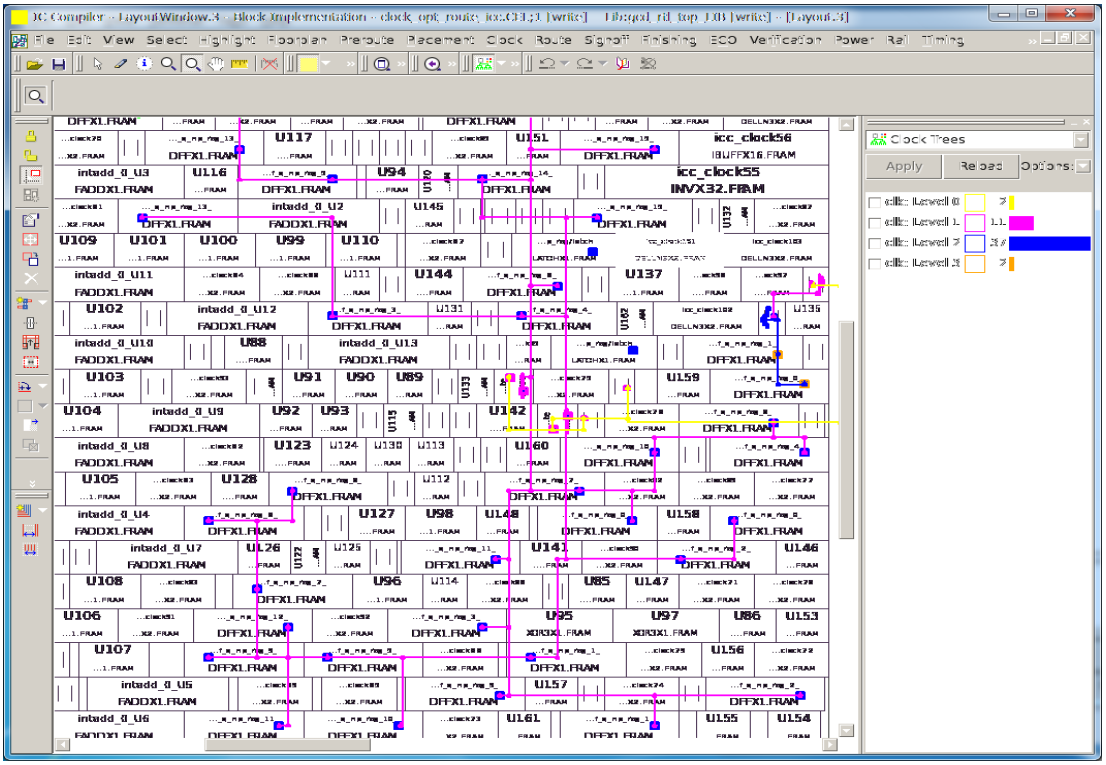
DA_VLSI_PD> make -f rm_setup/Makefile_zrt dp

For Flat Floorplanning.

4. Open the cell saved just after the clock routing stage and view the clock tree and found how many **clock buffers** are used in the design, and how much is the **clock skew, clock propagation delay**? _____

File->Open Design -> clock_opt_route_icc

// The cell saved during layout just after the clock routing



icc_shell> report_clock_timing –type skew // view the maximum clock skew, for more information see the manual of 'report_clock_timing'.

5. Check the timing report and simply summarize the timing change as the physical design goes on.

xxxx.timing.rpt

xxxx.max.tim

xxxx.min.tim

6. Simulate and synthesize Signal Controller (No response to technical questions)

*Module is in sig_control.v