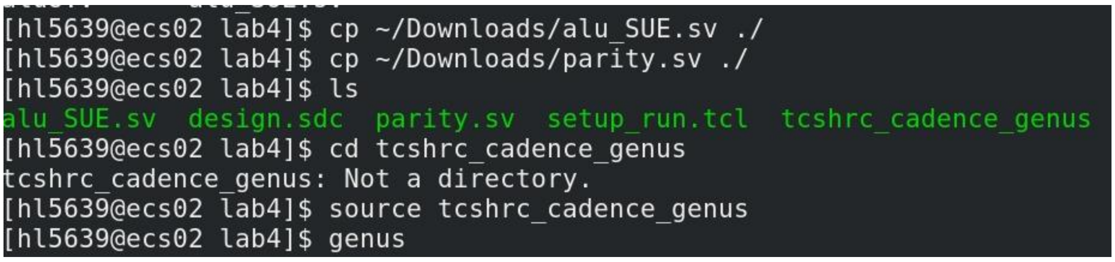
Lab 4

hl5639 Hongrui Liu

ALU:

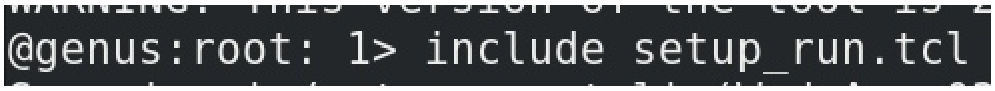
Synthesis Steps:

1. Copy all files into the work space from Brightspace by using cp ./
2. Sourcing the source file to setup the environment for genus.
3. Launch genus without gui to make sure running stably,



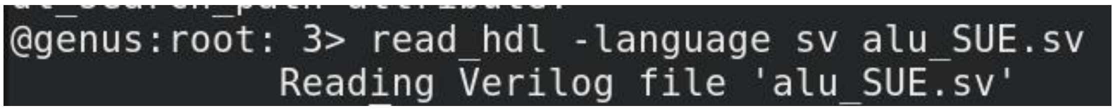
```
[hl5639@ecs02 lab4]$ cp ~/Downloads/alu_SUE.sv ./
[hl5639@ecs02 lab4]$ cp ~/Downloads/parity.sv ./
[hl5639@ecs02 lab4]$ ls
alu_SUE.sv  design.sdc  parity.sv  setup_run.tcl  tcshrc_cadence_genus
[hl5639@ecs02 lab4]$ cd tcshrc_cadence_genus
tcshrc_cadence_genus: Not a directory.
[hl5639@ecs02 lab4]$ source tcshrc_cadence_genus
[hl5639@ecs02 lab4]$ genus
```

4. Set up the environment in genus with the tcl files to match the asap7 library.



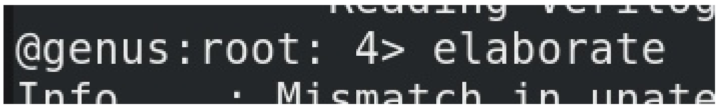
```
@genus:root: 1> include setup_run.tcl
```

5. Read the RTL files for synthesis with specify the language as SystemVerilog.



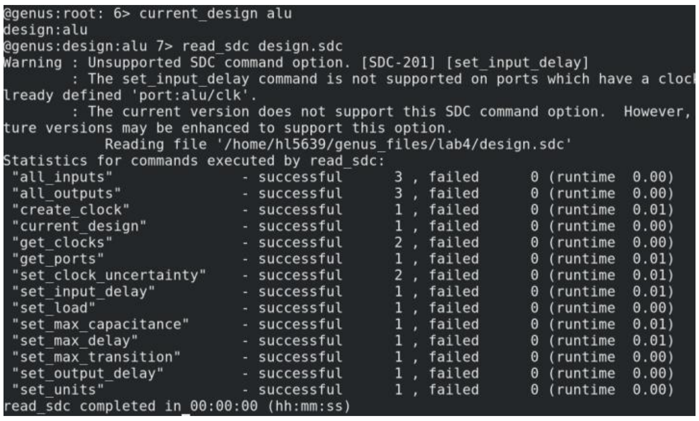
```
@genus:root: 3> read_hdl -language sv alu_SUE.sv
            Reading Verilog file 'alu_SUE.sv'
```

6. Launch the elaboration step which used to generate our initial circuit with high level blocks based on the RTL files provided.



```
@genus:root: 4> elaborate
```

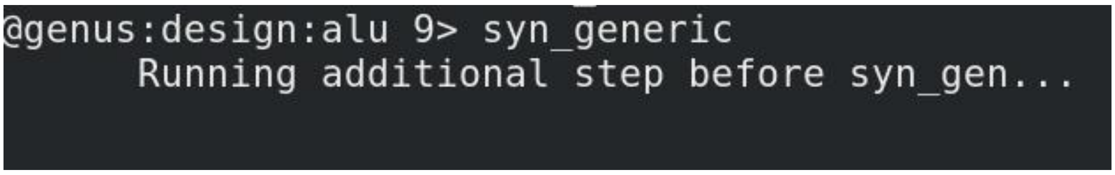
7. Specify the top module to be simulated which I named alu for this alu module and then includes the design constraint files into the workspace. With all commends showing successful means it is ready to move to the next step.



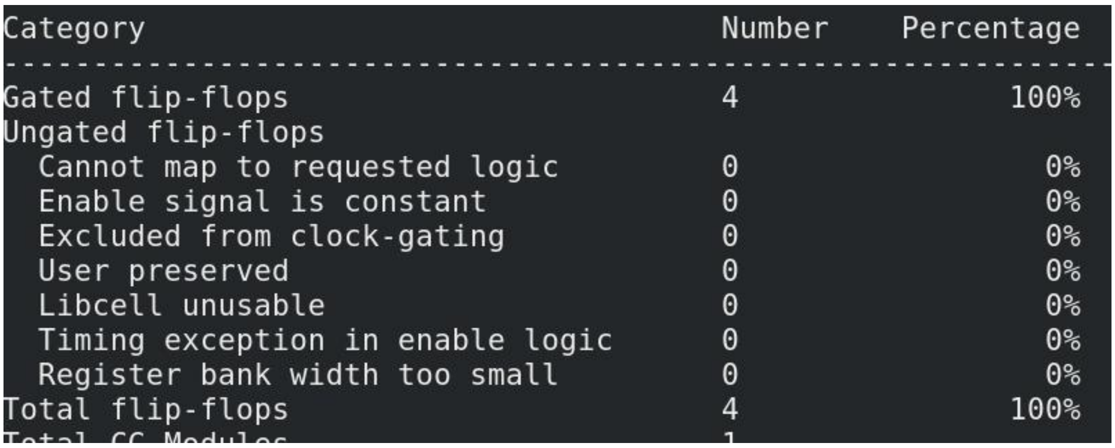
```
@genus:root: 6> current_design alu
design:alu
@genus:design:alu 7> read_sdc design.sdc
Warning : Unsupported SDC command option. [SDC-201] [set_input_delay]
        : The set_input_delay command is not supported on ports which have a clock
lready defined 'port:alu/clk'.
        : The current version does not support this SDC command option.  However,
ture versions may be enhanced to support this option.
            Reading file '/home/hl5639/genus_files/lab4/design.sdc'
Statistics for commands executed by read_sdc:
 "all_inputs"              - successful      3 , failed      0 (runtime  0.00)
 "all_outputs"             - successful      3 , failed      0 (runtime  0.00)
 "create_clock"            - successful      1 , failed      0 (runtime  0.01)
 "current_design"          - successful      1 , failed      0 (runtime  0.00)
 "get_clocks"              - successful      2 , failed      0 (runtime  0.00)
 "get_ports"               - successful      1 , failed      0 (runtime  0.01)
 "set_clock_uncertainty"   - successful      2 , failed      0 (runtime  0.01)
 "set_input_delay"         - successful      1 , failed      0 (runtime  0.01)
 "set_load"                - successful      1 , failed      0 (runtime  0.00)
 "set_max_capacitance"     - successful      1 , failed      0 (runtime  0.01)
 "set_max_delay"           - successful      1 , failed      0 (runtime  0.01)
 "set_max_transition"      - successful      1 , failed      0 (runtime  0.00)
 "set_output_delay"        - successful      1 , failed      0 (runtime  0.00)
 "set_units"               - successful      1 , failed      0 (runtime  0.00)
read_sdc completed in 00:00:00 (hh:mm:ss)
```

8. Then, we can synthesis our design. Firstly, it can be synthesis generally by default

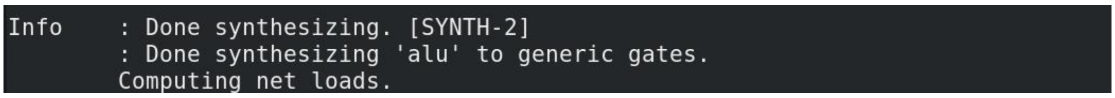
which are the general gates not corresponding to any components in the library that we include.



```
@genus:design:alu 9> syn_generic
      Running additional step before syn_gen...
```



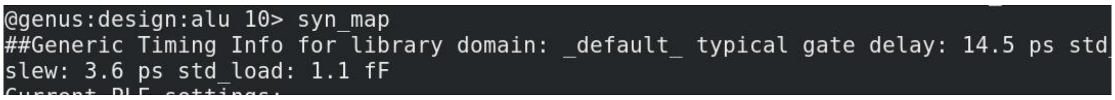
```
Category                                Number    Percentage
-------------------------------------------------------------
Gated flip-flops                        4               100%
Ungated flip-flops
  Cannot map to requested logic         0                 0%
  Enable signal is constant             0                 0%
  Excluded from clock-gating            0                 0%
  User preserved                        0                 0%
  Libcell unusable                      0                 0%
  Timing exception in enable logic      0                 0%
  Register bank width too small         0                 0%
Total flip-flops                        4               100%
```



```
Info    : Done synthesizing. [SYNTH-2]
        : Done synthesizing 'alu' to generic gates.
        Computing net loads.
```

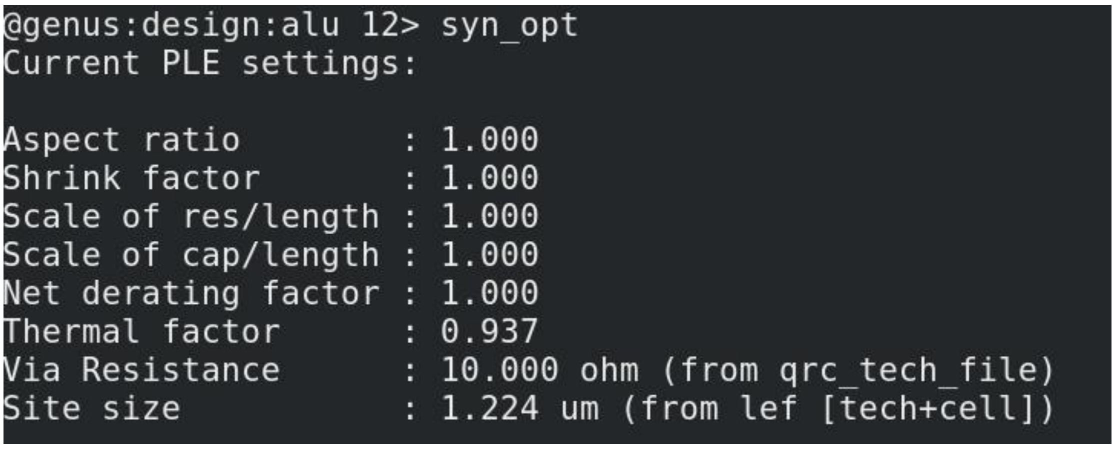
We can see from the output that there four FF are used for generic gates and just the basic gate that performances.

9. Next, synthesis the RTL mapped to our asap7 library.



```
@genus:design:alu 10> syn_map
##Generic Timing Info for library domain: _default_ typical gate delay: 14.5 ps std_
slew: 3.6 ps std_load: 1.1 fF
```

10. And for further synthesis to make more optimization on our design.

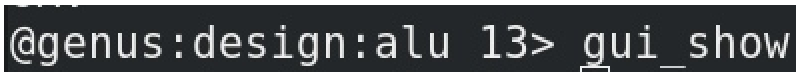


```
@genus:design:alu 12> syn_opt
Current PLE settings:

Aspect ratio        : 1.000
Shrink factor       : 1.000
Scale of res/length : 1.000
Scale of cap/length : 1.000
Net derating factor : 1.000
Thermal factor      : 0.937
Via Resistance      : 10.000 ohm (from qrc_tech_file)
Site size           : 1.224 um (from lef [tech+cell])
```

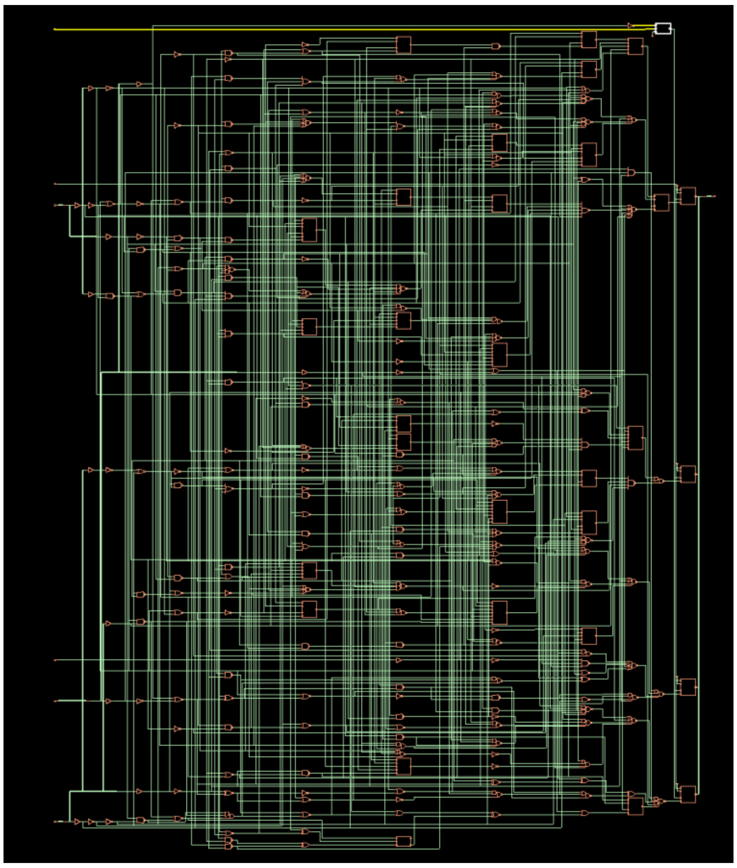
And we can see that from the output it generally optimizes focusing on sizing,

11. Then, we can perform the visualize synthesis.



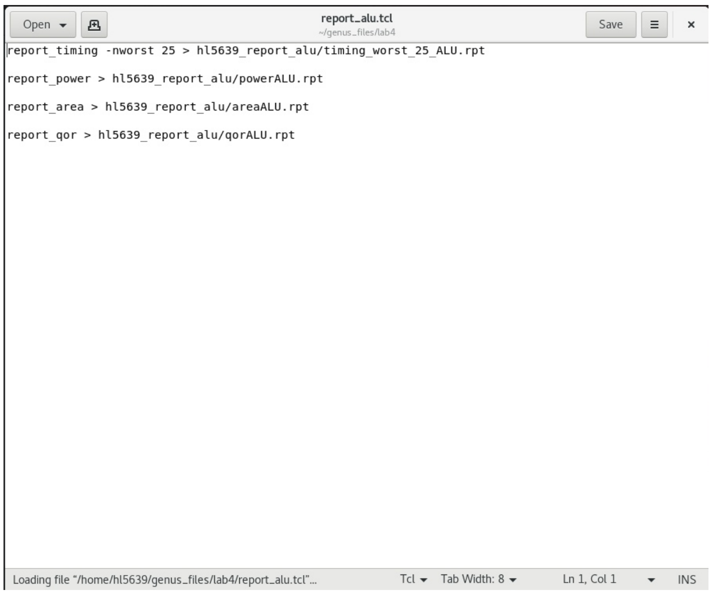
```
@genus:design:alu 13> gui_show
```

Finally, we having the final Schematics:



Reports:

1. Creating a tcl files with containing all the report commends inside and created a folder to save them:



```
report_timing -nworst 25 > hl5639_report_alu/timing_worst_25_ALU.rpt

report_power > hl5639_report_alu/powerALU.rpt

report_area > hl5639_report_alu/areaALU.rpt

report_qor > hl5639_report_alu/qorALU.rpt
```

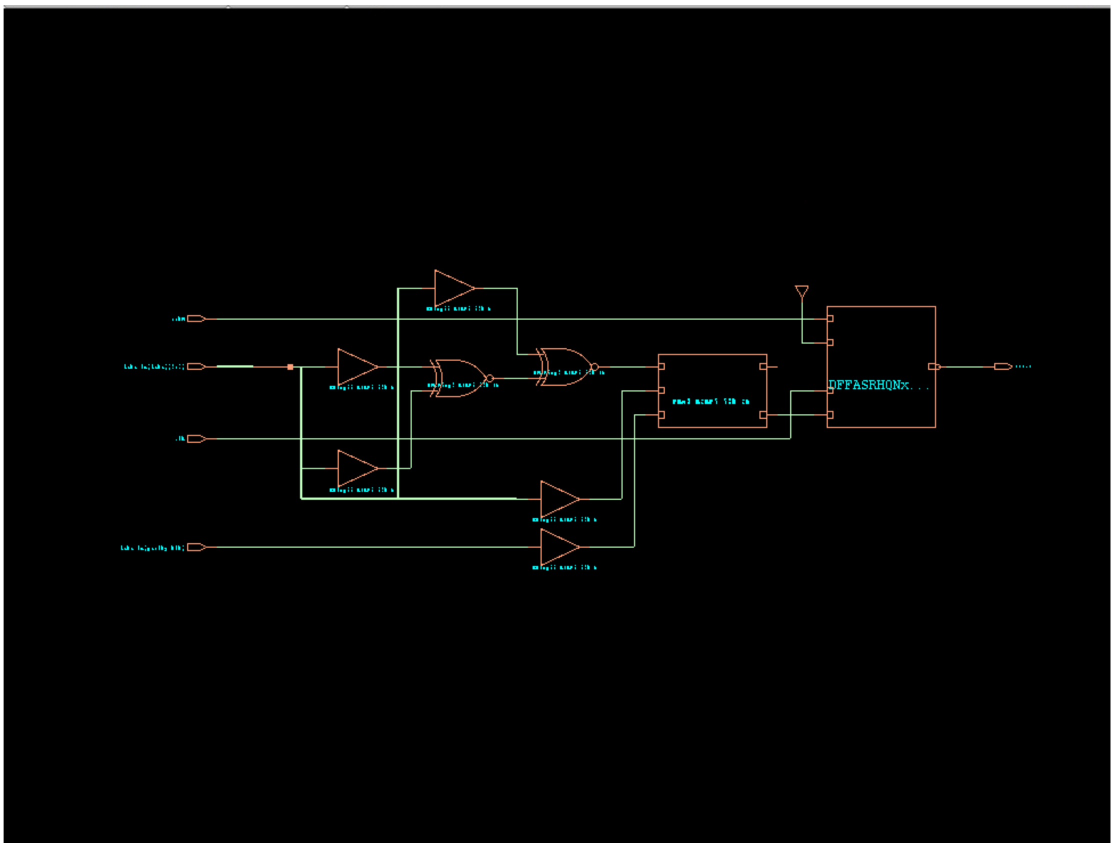
Loading file "/home/hl5639/genus_files/lab4/report_alu.tcl"...
Tcl
Tab Width: 8
Ln 1, Col 1
INS

2. And then source the tcl file to generate

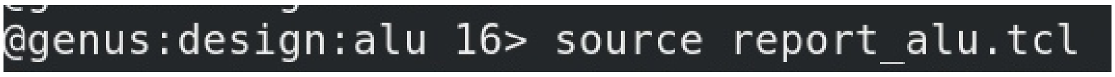
Parity:

Synthesis Steps:

For parity all the steps are the same and here is the final Schematic:



3. the report (which will not appear in this document, and will includes in zip files)



```
@genus:design:alu 16> source report_alu.tcl
```