Lab 3: Logical Synthesis

COMPENG 4DV4: Very Large Scale Integration System Design

February 24, 2026

Objective

This document describes the steps to:

1. Configure Synopsys Design Compiler (DC) to use open source libraries.
2. Synthesize the RTL ALU to a gate-level netlist and produce SDF, SDC, power and area reports.
3. Simulate the gate-level netlist with SDF annotation using NCVerilog.

1 Tools Used

- Synopsys Design Compiler

2 Environment Setup and Data Preparation

2.1 Extracting Lab Files

Upload `Lab3.tar` to your working directory and extract it:

```
tar -xvf Lab3.tar
```

2.2 Loading Environment Scripts

Load the Synopsys Design Compiler environment in your working directory:

```
source /CMC/scripts/synopsys.syn.2022.12.csh
```

3 Convert from .lib to .db file for synthesis

Create a /db directory in your working directory, then write the following commands in the terminal:

```
dc_shell

enable_write_lib_mode
```

```
set asap7_lib "<.lib files location>"


read_lib $asap7_lib/asap7sc7p5t_AO_LVT_TT_nldm_211120.lib
read_lib $asap7_lib/asap7sc7p5t_AO_SLVT_TT_nldm_211120.lib
read_lib $asap7_lib/asap7sc7p5t_INVBUF_LVT_TT_nldm_220122.lib
read_lib $asap7_lib/asap7sc7p5t_INVBUF_SLVT_TT_nldm_220122.lib
read_lib $asap7_lib/asap7sc7p5t_OA_LVT_TT_nldm_211120.lib
read_lib $asap7_lib/asap7sc7p5t_OA_SLVT_TT_nldm_211120.lib
read_lib $asap7_lib/asap7sc7p5t_SEQ_LVT_TT_nldm_220123.lib
read_lib $asap7_lib/asap7sc7p5t_SEQ_SLVT_TT_nldm_220123.lib
read_lib $asap7_lib/asap7sc7p5t_SIMPLE_LVT_TT_nldm_211120.lib
read_lib $asap7_lib/asap7sc7p5t_SIMPLE_SLVT_TT_nldm_211120.lib


list_libs


write_lib asap7sc7p5t_AO_LVT_TT_nldm_211120 -f db -o $asap7_lib/db/
    asap7sc7p5t_AO_LVT_TT_nldm_211120.db

write_lib asap7sc7p5t_AO_SLVT_TT_nldm_211120 -f db -o $asap7_lib/db/
    asap7sc7p5t_AO_SLVT_TT_nldm_211120.db

write_lib asap7sc7p5t_INVBUF_LVT_TT_nldm_211120 -f db -o $asap7_lib/db/
    asap7sc7p5t_INVBUF_LVT_TT_nldm_211120.db

write_lib asap7sc7p5t_INVBUF_SLVT_TT_nldm_211120 -f db -o $asap7_lib/db/
    asap7sc7p5t_INVBUF_SLVT_TT_nldm_211120.db

write_lib asap7sc7p5t_OA_LVT_TT_nldm_211120 -f db -o $asap7_lib/db/
    asap7sc7p5t_OA_LVT_TT_nldm_211120.db

write_lib asap7sc7p5t_OA_SLVT_TT_nldm_211120 -f db -o $asap7_lib/db/
    asap7sc7p5t_OA_SLVT_TT_nldm_211120.db

write_lib asap7sc7p5t_SEQ_LVT_TT_nldm_220123 -f db -o $asap7_lib/db/
    asap7sc7p5t_SEQ_LVT_TT_nldm_220123.db

write_lib asap7sc7p5t_SEQ_SLVT_TT_nldm_220123 -f db -o $asap7_lib/db/
    asap7sc7p5t_SEQ_SLVT_TT_nldm_220123.db

write_lib asap7sc7p5t_SIMPLE_LVT_TT_nldm_211120 -f db -o $asap7_lib/db/
    asap7sc7p5t_SIMPLE_LVT_TT_nldm_211120.db

write_lib asap7sc7p5t_SIMPLE_SLVT_TT_nldm_211120 -f db -o $asap7_lib/db/
    asap7sc7p5t_SIMPLE_SLVT_TT_nldm_211120.db
```

close dc_shell.

4 Prepare a DC Tcl script (dc_run.tcl)

Create a file named `dc_run.tcl` with the following contents. This script:

- sets the libraries

- reads the Verilog RTL
- applies constraints (clock, I/O delays)
- checks and compiles the design
- writes gate-level outputs and SDF/SDC

```
set search_path {<path to db folder>}

set link_library "asap7sc7p5t_AO_LVT_TT_nldm_211120.db
    asap7sc7p5t_AO_SLVT_TT_nldm_211120.db asap7sc7p5t_INVBUF_LVT_TT_nldm_211120.db
    asap7sc7p5t_INVBUF_SLVT_TT_nldm_211120.db asap7sc7p5t_OA_LVT_TT_nldm_211120.db
    asap7sc7p5t_OA_SLVT_TT_nldm_211120.db asap7sc7p5t_SEQ_LVT_TT_nldm_220123.db
    asap7sc7p5t_SEQ_SLVT_TT_nldm_220123.db asap7sc7p5t_SIMPLE_LVT_TT_nldm_211120.db
    asap7sc7p5t_SIMPLE_SLVT_TT_nldm_211120.db"
set target_library "asap7sc7p5t_AO_LVT_TT_nldm_211120.db
    asap7sc7p5t_AO_SLVT_TT_nldm_211120.db asap7sc7p5t_INVBUF_LVT_TT_nldm_211120.db
    asap7sc7p5t_INVBUF_SLVT_TT_nldm_211120.db asap7sc7p5t_OA_LVT_TT_nldm_211120.db
    asap7sc7p5t_OA_SLVT_TT_nldm_211120.db asap7sc7p5t_SEQ_LVT_TT_nldm_220123.db
    asap7sc7p5t_SEQ_SLVT_TT_nldm_220123.db asap7sc7p5t_SIMPLE_LVT_TT_nldm_211120.db
    asap7sc7p5t_SIMPLE_SLVT_TT_nldm_211120.db"

read_file {/<path to verilog designs folder>} -autoread -recursive -format verilog
    -top mips_cpu


create_clock -name "clk" -period 200 -waveform {"0" "100"} {"clk"}
set_dont_touch_network [find clock clk]
set_fix_hold clk

set_dont_touch_network [get_nets rst_n]
set_dont_touch u_alu
set_dont_touch u_regfile
set_dont_touch u_ctrl
set_dont_touch u_imem
set_dont_touch u_sign_extend
set_dont_touch u_alu_ctrl
set_dont_touch u_dmem

set_boundary_optimization "*"
set_fix_multiple_port_nets -all -buffer_constant
set_max_area 0

check_design

compile -map_effort medium

report_timing -path full -delay max -max_paths 1 -nworst 1 > ./reports/MIPS.timing
report_power > ./reports/MIPS.power
report_area -nosplit > ./reports/MIPS.area

write -hierarchy -format ddc
write_sdc ./outputs/MIPS.sdc
write_sdf -version 1.0 ./outputs/MIPS.sdf
write -format verilog -hierarchy -output ./outputs/MIPS_syn.v
```

```
exit
```

5 Run Design Compiler

Create two folders `outputs` and `reports` first:

```
mkdir outputs
mkdir reports
```

Start DC and run the Tcl script. Example:

```
# start dc_shell
dc_shell -f dc_run.tcl
```

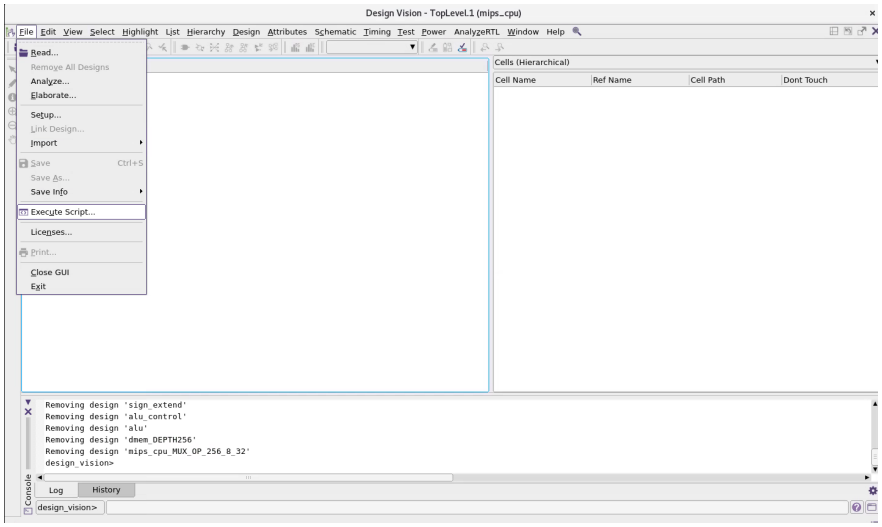
Fix any errors or warnings DC prints (e.g., unsupported constructs, inferred latches). Edit your RTL if there are latches or non-synthesizable constructs.

6 Design Vision

You can use the gui interface to run your script and to view the schematic of the generated netlist. write in the terminal:

```
design_vision
```

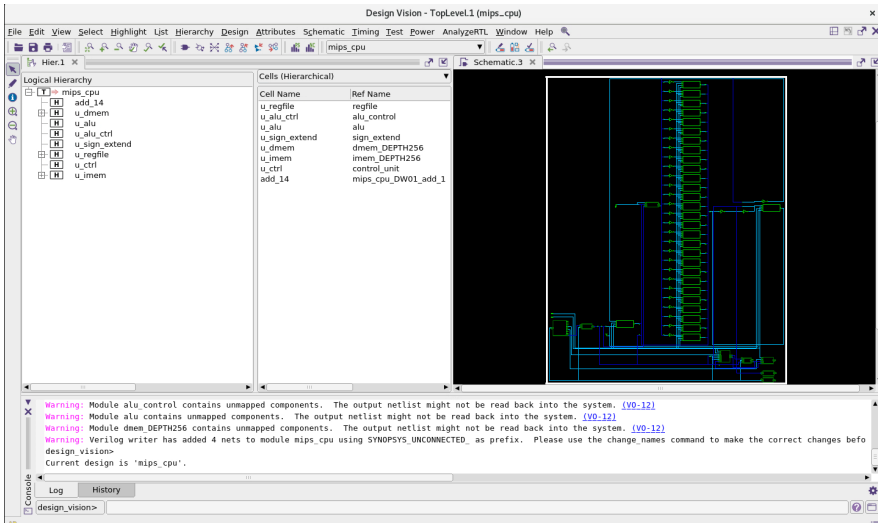
A gui interface will appear. From file select `Execute Script` and choose the .tcl script.



After it finishes, it will show you the hierarchy. click on the top module, right click then schematic view. You can double click the schematic view to see the design.

7 Checkpoints

After a successful run you should have:



1. `ALU_syn.v` – gate-level Verilog netlist (for gate simulation) 20%
2. `ALU.sdf` – timing delays 20%
3. `ALU.sdc` – the constraints file 20%
4. `ALU.area`, `ALU.timing`, `ALU.power` – reports 20%
5. Write a report about your results and show the netlist schematic 20%

Open `ALU.timing` and make sure the top critical path slack is non-negative.

8 Submission

- A report with all the screenshots stated in the checkpoints and lab steps with annotation
- Name it `lab3_studentID`
- Upload it to its corresponding Dropbox on avenue