

Computer-Aided VLSI System Design

Lab2: Waveform Debugging

TA: 羅宇呈 f08943129@ntu.edu.tw

Introduction

In this lab, you will learn:

1. How to dump FSDB waveform file through NC-Verilog simulator
2. Waveform debugging by using nWave
3. SDF annotation for gate-level simulation after synthesis

Data Preparation

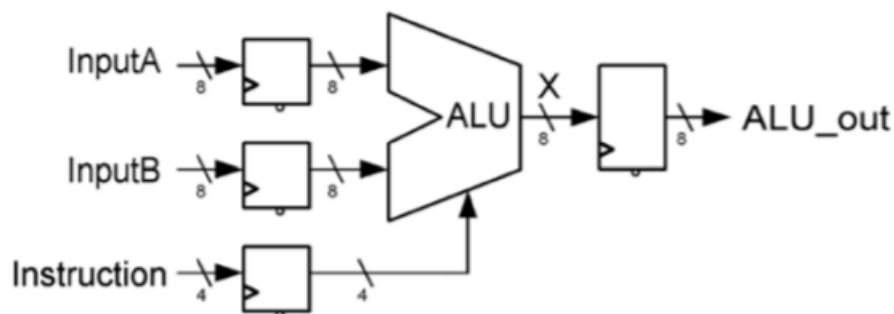
1. Upload your files (Lab2.tar) to your work directory.
2. Decompress Lab2.tar with following command:

```
tar -xvf Lab2.tar
```

3. Lab2 files are shown as below:

Files/Folder	Description
Lab2_alu.v	RTL code of ALU
Lab2_test_alu.v	Testbench for ALU
Lab2_alu_run.f	File list for RTL simulation
Lab2_alu_s.v	Gate-level netlist of ALU
Lab2_alu_run_s.f	File list for gate-level simulation
Lab2_alu_s.sdf	SDF file for gate-level timing annotation

4. The circuit diagram of the ALU used for this lab is illustrated as follows:



5. Enter Lab2 directory:

```
cd lab2
```

Environment Setup

1. Source the default **cshrc** file:

```
source /usr/cad/cadence/cshrc
source /usr/spring_soft/CIC/verdi.cshrc
```

Lab 2.1. Generating the VCD & FSDB Waveform

Value Change Dump (VCD) format:

- Indigenously supported by most simulators
- Using ASCII text for waveform recording (larger file size)

```
$dumpfile("filename.vcd");
$dumpvars;
```

Fast Signal Database (FSDB) format:

- Defined by Verdi debugging system
- More compact format, small file size

```
$fsdbDumpfile("filename.fsdb");
$fsdbDumpvars;
```

1. **Run the RTL simulation using ncverilog:**

```
ncverilog Lab2_test_alu.v Lab2_alu.v
```

or

```
ncverilog -f Lab2_alu_run.f
```

2. You would only see the text telling you "congratulations" but not knowing what's going on inside the circuit. Now we would like to generate VCD file.

Please open the testbench Lab2_test_alu.v, **add following lines in the initial block** in testbench to generate vcd file:

```
$dumpfile("Lab2_alu.vcd");
$dumpvars;
```

3. Now re-run the simulation, note we should add **"*+access+rw*"** to enable vcd file dumping:

```
ncverilog +access+rw -f Lab2_alu_run.f
```

4. Please check if there exists a file called "**Lab2_alu.vcd**"
5. Now we would like to generate FSDB file. Please open the testbench again.

6. Add following lines in the **initial** block in Lab2_test_alu.v, you can also remove the two lines added in step 2:

```
$fsdbDumpfile("Lab2_alu.fsdb");  
$fsdbDumpvars;
```

7. Now re-run the simulation, note we should add "+access+rw" to enable FSDB file dumping:

```
ncverilog +access+rw -f Lab2_alu_run.f
```

8. Please check if there exists a file called "Lab2_alu.fsdb"

Lab 2.2. Using nWave for Waveform Debugging

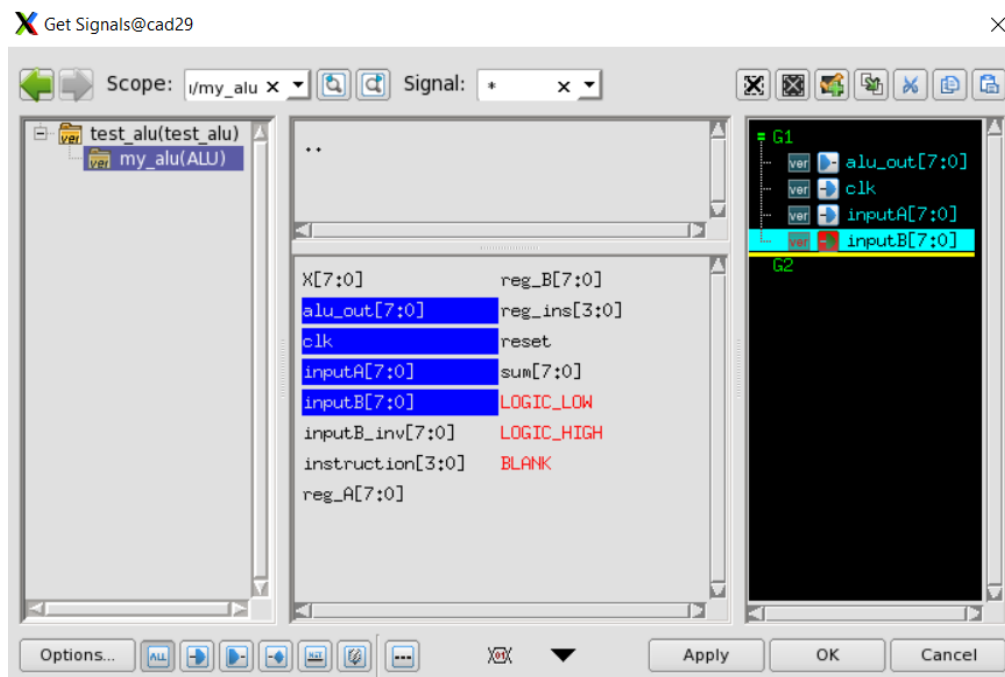
1. Start nWave. "&" enables you to use the terminal while nWave is running in the background:

```
nWave &
```

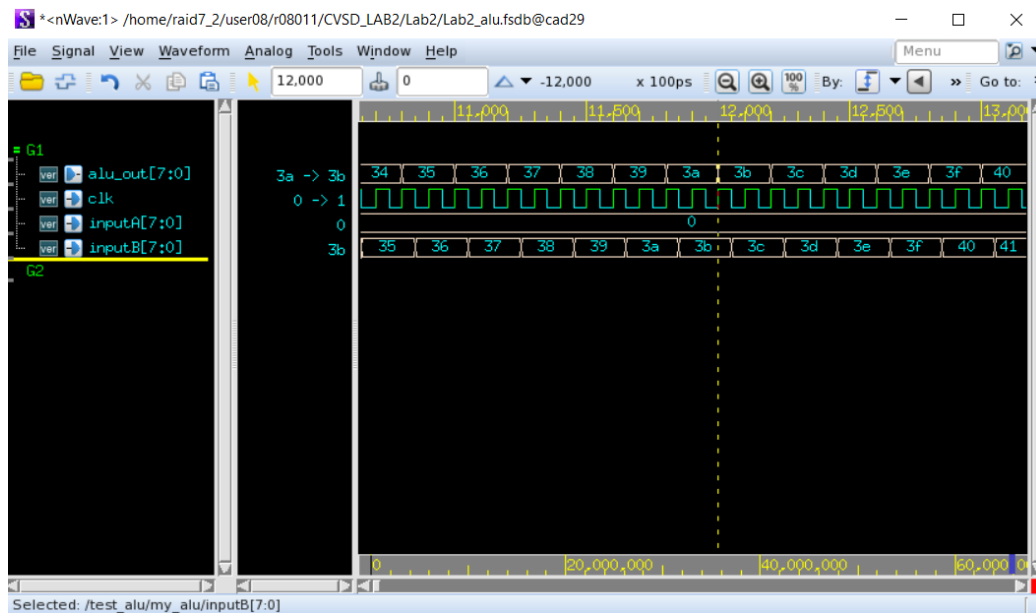
2. Open the VCD file we obtained. VCD file will be translated to FSDB file, you can either open the VCD file using GUI or run:

```
nWave Lab2_alu.vcd &
```

3. Choose the signals we are interested in.

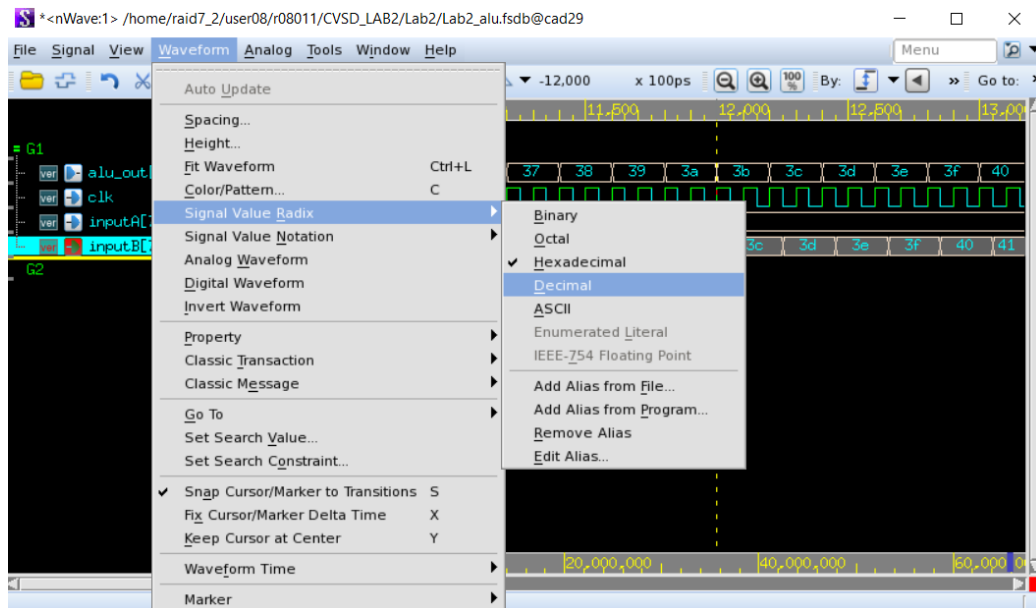


4. View the waveform:



5. Change the radix and sign representation:

- "Waveform" – "Signal Value Radix" – "Decimal"
- "Waveform" – "Signal Value Notation" – "Signed 2's Complement"



Lab 2.3. Gate-level Simulation

1. After synthesizing Lab2_alu.v, we can derive the gate-level netlist file "Lab2_alu_s.v", which is already provided in this lab.

Try this command for gate-level simulation:

```
ncverilog +access+rw Lab2_test_alu.v Lab2_alu_s.v  
/home/raid7_2/course/cvtd/CBDK_IC_Constest_v2.5/Verilog/tsmc13_neg.v
```

or

```
ncverilog +access+rw -f Lab2_alu_run_s.f
```

2. You will see a lot of warnings about "**timing violation**." That's because the **simulator does not know about the propagation delay of each gate**. So the simulator cannot combine the delay information with the "tsmc13_neg.v" to check if your design can run at the clock frequency defined in the testbench.

Please open the testbench Lab2_test_alu.v, and uncomment this line:

```
$sdf_annotate("Lab2_alu_s.sdf", my_alu);
```

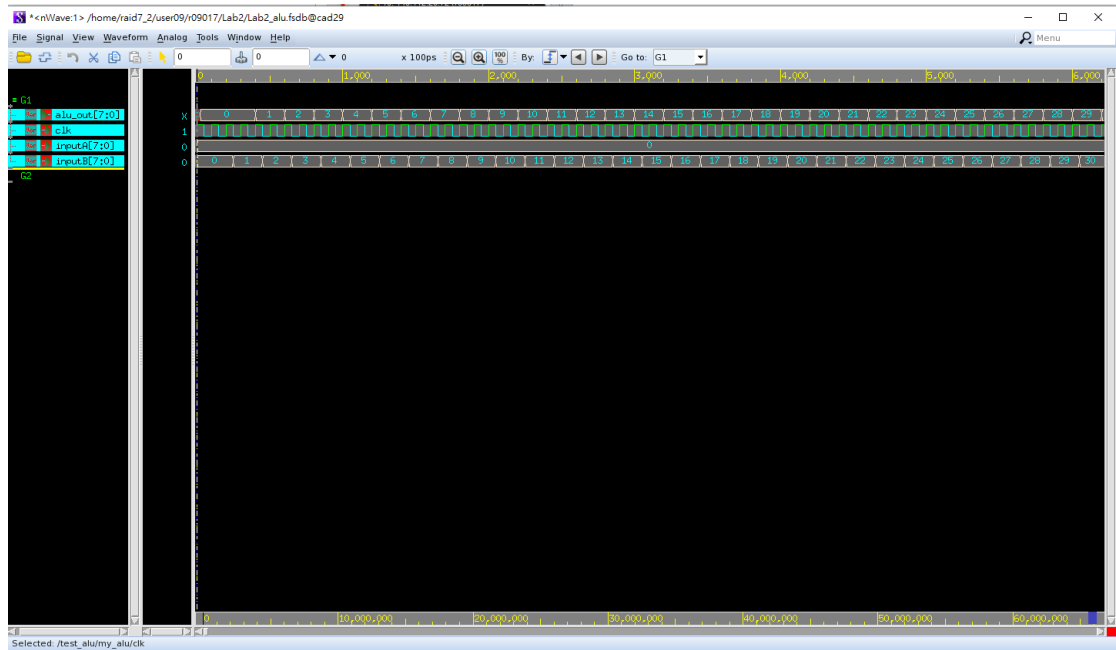
This line annotates the timing information (.sdf) of the netlist.

3. Save the modified testbench and re-run step 1.
4. During the simulation, there is no timing violation anymore.
5. Open the waveform file to see how different it is between RTL waveform and gate-level waveform.

Checkpoints

Please check with TAs before leaving this lab to make sure the following goals are accomplished and to get credits.

1. Please generate the FSDB waveform.
2. Open nWave and take the snapshots of alu_out, clk, inputA and inputB.



3. Fix the timing violations in gate-level simulation.
4. Take a snapshot after passing gate-level simulation.

```

Reading SDF file from location "Lab2_alu.sdf"
Annotating SDF timing data:
  Compiled SDF file:    Lab2_alu.sdf.X
  Log file:
  Backannotation scope: test_alu.my_alu
  Configuration file:
  MTM control:
  Scale factors:
  Scale type:
Annotation completed successfully...
SDF Statistics: NO. of PathDelays = 3284  Annotated = 13.28% -- No. of Tchecks = 1368  Annotated = 12.28%
  Path Delays      Total      Annotated      Percentage
    $hold          64          0             0.00
    $width         451         84             18.63
    $setuphold     853         84             9.85
Building instance overlay tables: ..... Done
Generating native compiled code:
  worklib.DFFRX1:v <0x6002b0e3>
    streams: 0, words: 0
  worklib.DFFRX1:v <0x62e7bc7b>
    streams: 0, words: 0
  worklib.test_alu:v <0x0d19f72a>
    streams: 7, words: 17985
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
  Modules:          628      530
  UDPs:             289      12
  Primitives:      2074     10
  Timing outputs:   794     569
  Registers:        194     174
  Scalar wires:     3186      -
  Expanded wires:    20        3
  Always blocks:     1        1
  Initial blocks:    1        1
  Pseudo assignments: 6        6
  Timing checks:    2221     939
  Interconnect:      285      -
  Delayed tcheck signals: 665    616
  Simulation timescale: 1ps
Writing initial simulation snapshot: worklib.test_alu:v
Loading snapshot worklib.test_alu:v ..... Done
*Verdi* Loading libsscore_ius152.so
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run
FSDB Dumper for IUS, Release Verdi_R-2020.12, Linux, 11/19/2020
(C) 1996 - 2020 by Synopsys, Inc.
*Verdi* FSDB WARNING: The FSDB file already exists. Overwriting the FSDB file may crash the programs that are using this file.
*Verdi* : Create FSDB file 'Lab2_alu.fsdb'
*Verdi* : Begin traversing the scopes, layer (0).
*Verdi* : End of traversing.

Congratulations!! Your Verilog Code is correct!!
Simulation complete via $finish(1) at time 6558735 NS + 0
./Lab2_test_alu.v:127  #10 $finish;
ncsim> exit
[r09017@cad29 ~:/Lab2]$

```

Submission

1. Due Tuesday, Mar. 21, 19:00

- Take the snapshot of the result shown in previous section and submit to NTU COOL in pdf format.