

National Cheng Kung University

Department of Electrical Engineering

Introduction to VLSI CAD (Spring 2023)

Lab Session 4

Register Files, Manhattan Distance and LFSR

| Name | Student ID | |
|-------------------------------|------------|-------|
| 陳慕丞 | E94096097 | |
| Practical Sections | Points | Marks |
| Prob A | 25 | |
| Prob B | 25 | |
| Prob C | 25 | |
| Report | 15 | |
| File hierarchy, naming...etc. | 5 | |
| Superlint | 5 | |
| Notes: | | |

Due Date: 15:00, March 23, 2020 @ moodle

Deliverables

- 1) All Verilog codes including testbenches for each problem should be uploaded.
NOTE: Please **DO NOT** paste source code in the report!
- 2) **Noted! TA will use commands in Appendix A to check your design in SoC Lab. If TA can not compile your code with the commands, you will not get full credit.**
- 3) If you upload a dead body which we can't even compile, you will get **NO** credit!
- 4) All Verilog file should get at least **85%** SuperLint Coverage.
- 5) All homework requirements should be uploaded in this file hierarchy or you will not get full credit.

NOTE: Please **DO NOT** upload waveforms!

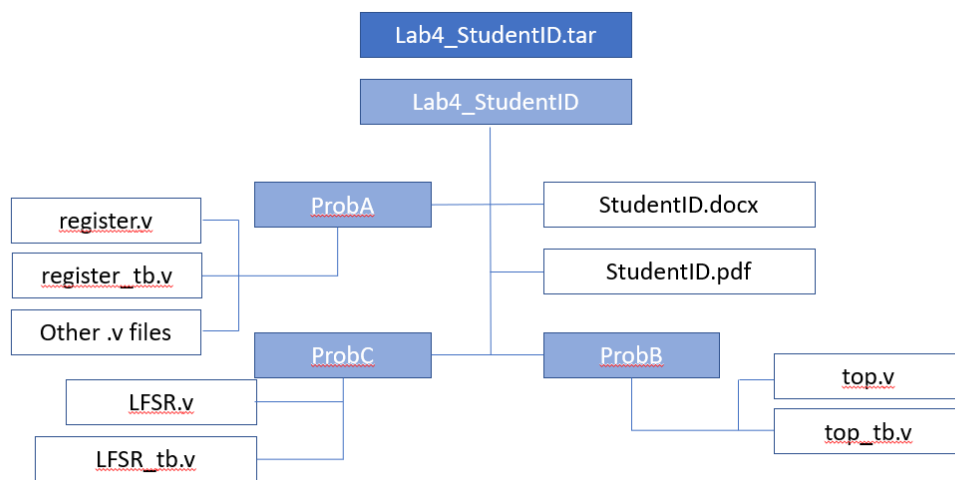
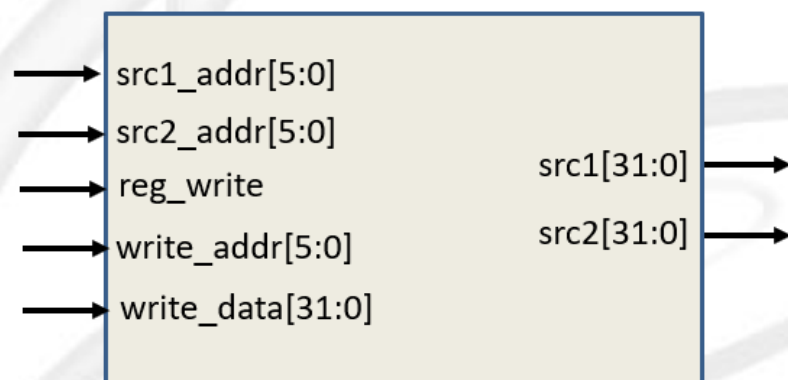


Fig.1 File hierarchy for Homework submission

Prob A: Register File



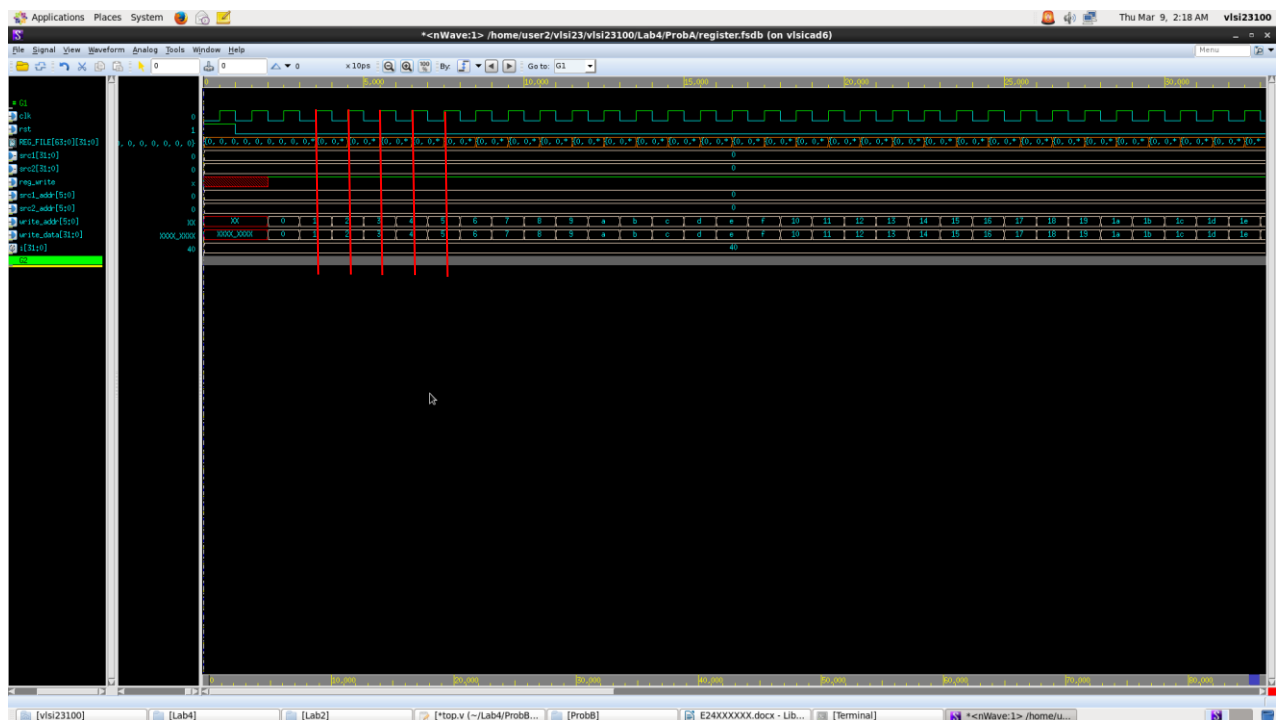
1. Based on the register file structure in LabA, please design a **64 x 32** register file by yourself.
2. Port list

| Signal | Type | Bits | Description |
|------------|--------|------|---------------------|
| clk | input | 1 | clock |
| rst | input | 1 | reset |
| reg_write | input | 1 | 0 → read, 1 → write |
| src1_addr | input | 6 | source1 address |
| src2_addr | input | 6 | source2 address |
| write_addr | input | 6 | write address |
| write_data | input | 32 | write data |
| src1 | output | 32 | read data source1 |
| src2 | output | 32 | read data source2 |

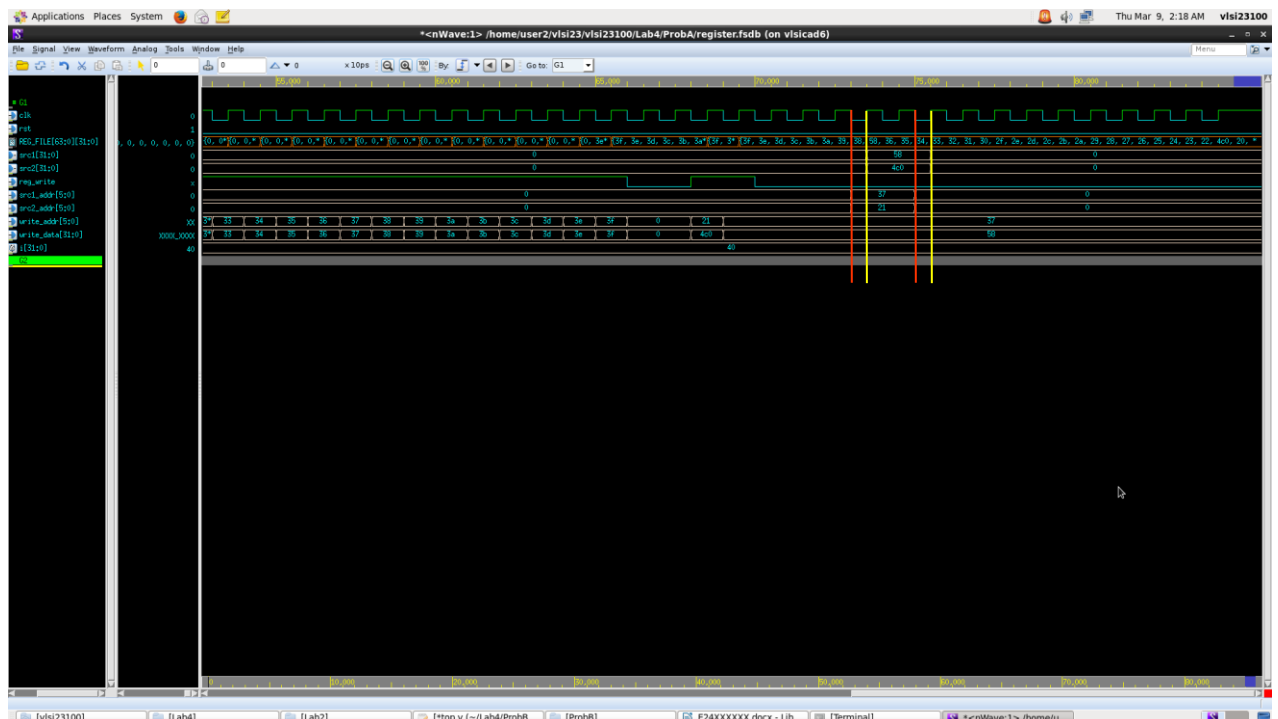
3. You should follow the file name rules as follow.

- Register file
 - File name: **register.v**
 - Module name: **register**
- Register file testbench
 - File name: **register_tb.v**

4. Show waveforms to explain that your register work correctly when **read** and **write**.

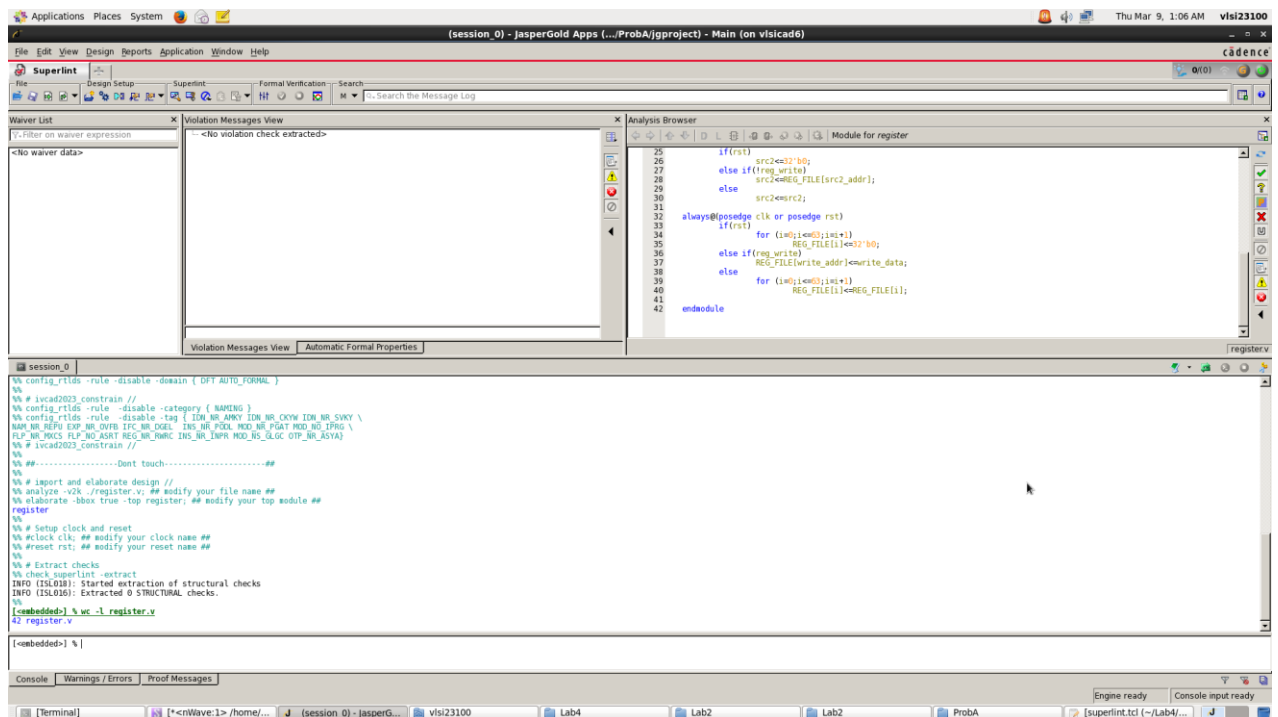


當 `reg_write=1`，clock falling edge 送入 `write_data` 時，下一個 cycle 的 clock rising edge 將 `write_data` 寫入 `REG_FILE[write_addr]`，如上圖所示。



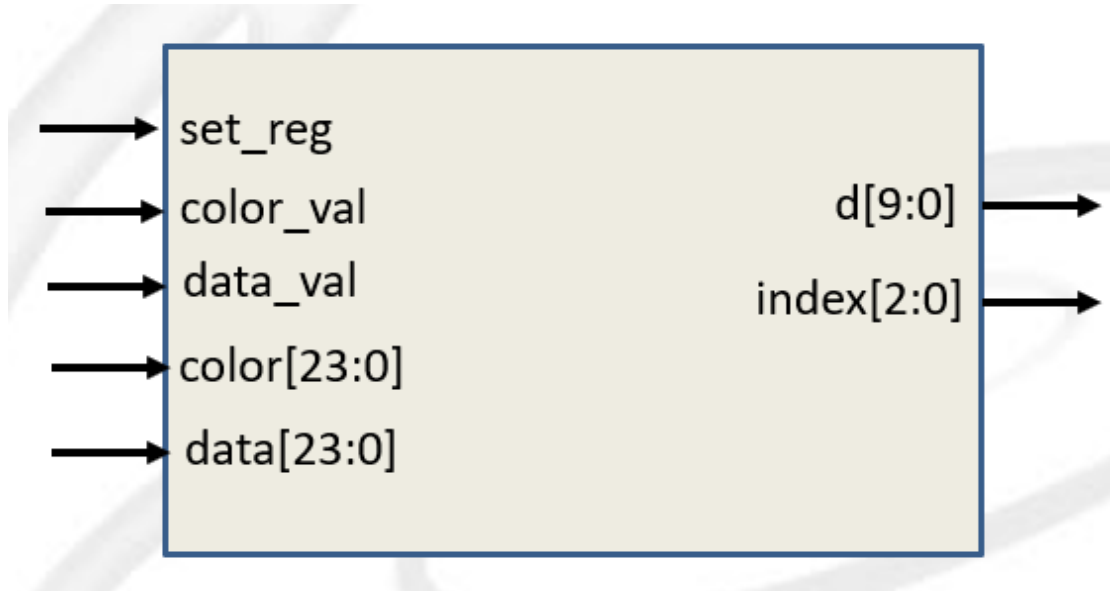
當 `reg_write=0`，clock falling edge 送入 `src1_addr` 和 `src2_addr` 時，下一個 cycle 的 clock rising edge 將 `REG_FILE[src1_addr]` 輸出到 `src1`，`REG_FILE[src2_addr]` 輸出到 `src2`，如上圖所示。

5. Show SuperLint coverage



Coverage:100%

Prob B: Finding Smallest Distance

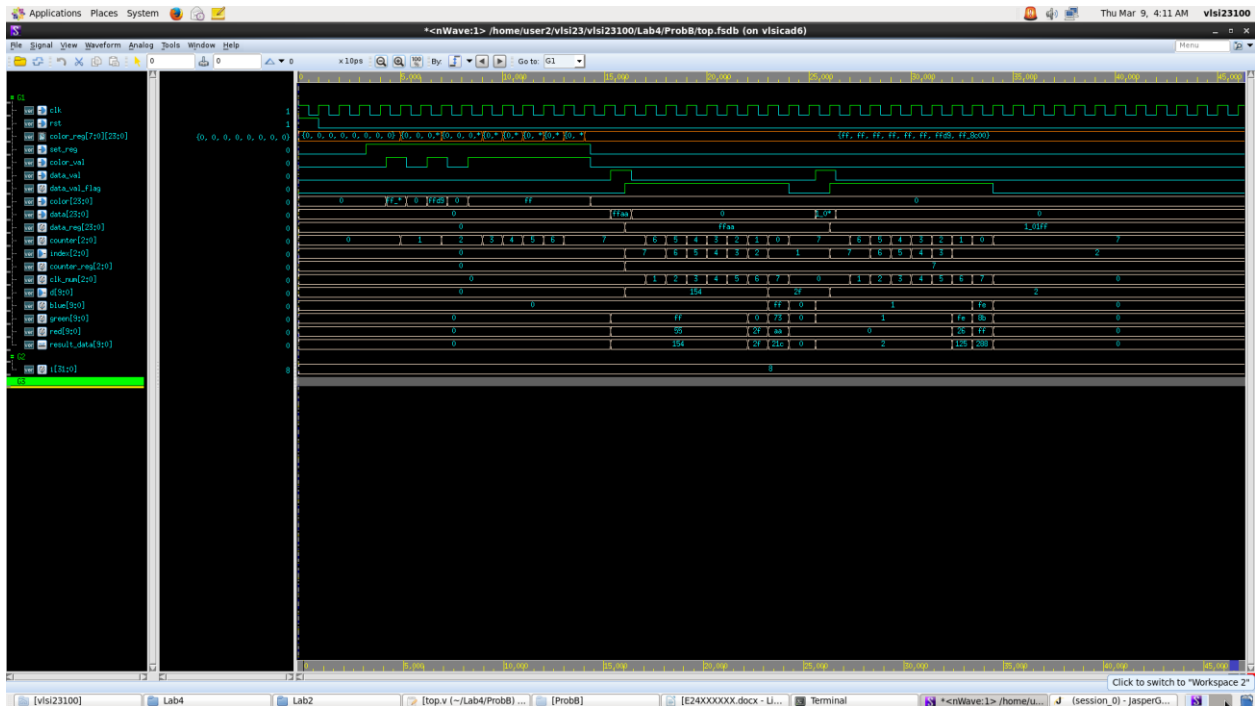


Please design a circuit that will find the smallest distance between the local registers and the input data, based on the structure given in the LAB4 slide.

Port list

| Signal | Type | Bits | Description |
|-----------|--------|------|---|
| clk | input | 1 | Clock pin. |
| rst | input | 1 | Reset pin. |
| set_reg | input | 1 | 1: write mode. In this mode, when the hardware reads color_val, it will fill register[0] with color, wait for the next color_val and fill register[1]... and so on, until set_reg=0. 0: keep the color registers the same. |
| color_val | input | 1 | 1: color data is <u>valid</u> , color shall be written into the local register if set_reg=1. 0: color data isn't available, keep the local registers the same. |
| data_val | input | 1 | 1: input data is <u>valid</u> , output shall be available after 8 cycles. 0: input data isn't available. |
| color | input | 24 | color data that will be stored in the local register. |
| data | input | 24 | Input data that will be compared among <u>all</u> of the color registers. |
| d | output | 10 | Output distance data. |
| index | output | 3 | Output index. If there are 2 or more colors that has the same smallest distance between input data, output the one that has the smallest index. |

1. Show waveforms to explain that your design works correctly.

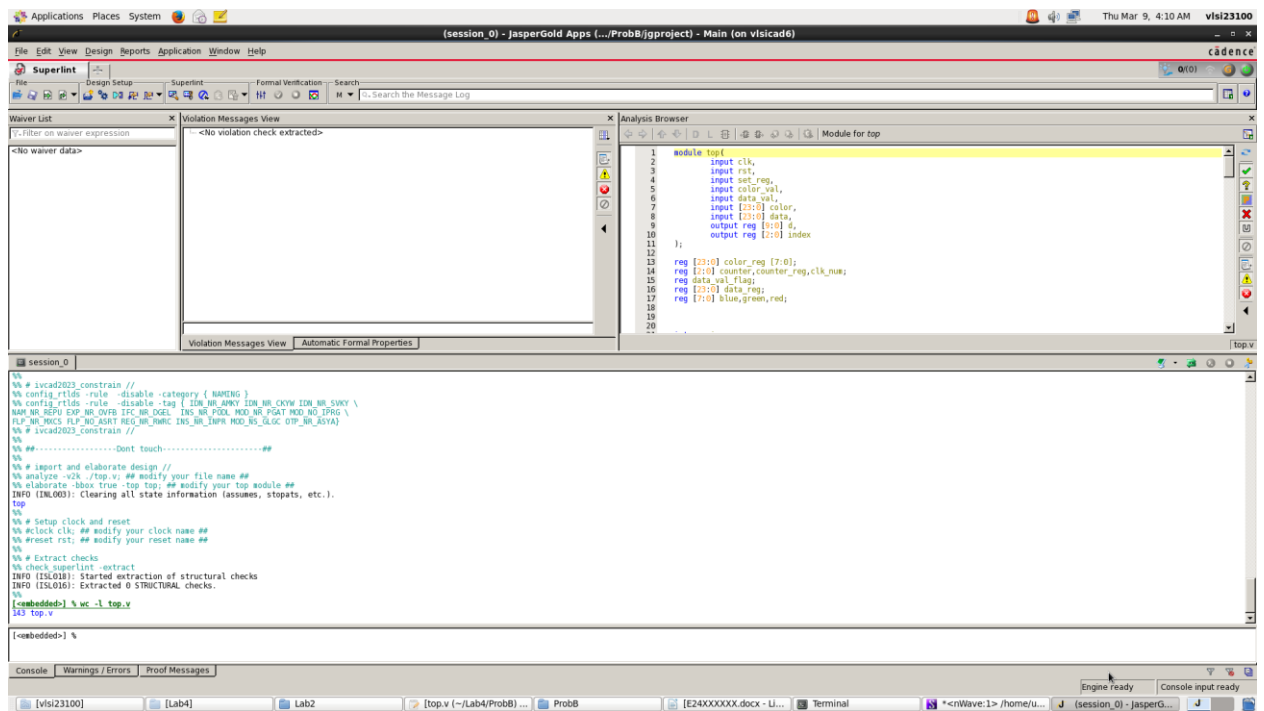


Input: clk, rst, set_reg, color_val, data_val, color, data

Output: d, index

當 set_reg 和 color_val 為 1 時，將 color 存進 color_reg [counter] 中，並將 counter 加 1；當 data_val 為 1 時，將 1 存進 data_val_flag，將 data 存進 data_reg，將 counter 的值存進 counter_reg(記住存到第幾個 color_reg)；當 data_val 或 data_val_flag 為 1 時，counter 每個 cycle 減 1(每個 cycle 用 color_reg [counter] 中的值比較)，clk_num 每個 cycle 加 1(負責數是否到第 8 個 cycle)；運算中 blue 為 $|B_i - B_j|$ ，green 為 $|G_i - G_j|$ ，red 為 $|R_i - R_j|$ ；若比較後距離等於或小於 d 中的值，則將距離的值存進 d 中，counter 的值存進 index 中；最後當 clk_num=7 時，重設 counter 為 counter_reg 中的值，data_val_flag 為 0。

2. Show SuperLint coverage



Coverage:100%

Prob C: LFSR

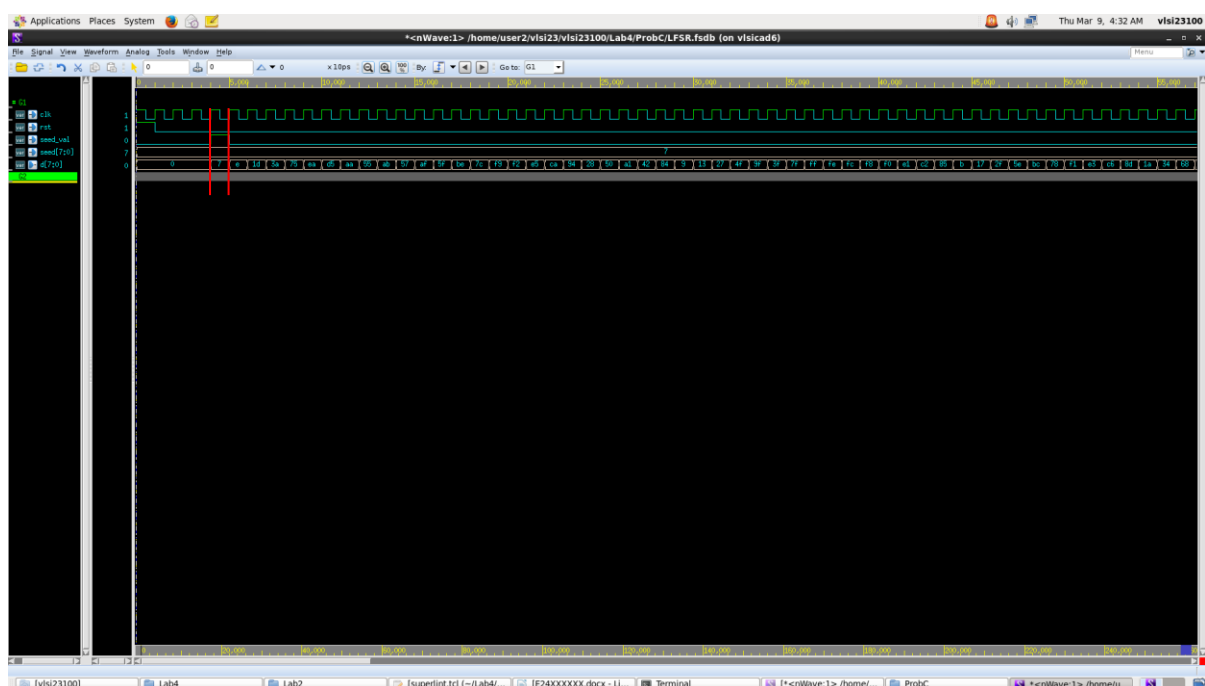
1. Please design an 8-bit-LFSR, with the given feedback function in the LAB4 slide.
2. Port list

| Signal | Type | Bits | Description |
|----------|--------|------|---|
| clk | input | 1 | Clock pin. |
| rst | input | 1 | Reset pin. Reset all of the flip flops to zeros. |
| seed_val | input | 1 | 1: the flip flops take seed as the initial state. 0: the flip flops works as linear feedback shift register. |
| seed | input | 8 | Initial state value of LFSR. |
| d | output | 8 | Output value of LFSR |

3. Feedback function

$$d[0] = (d[7] \wedge \sim d[5]) \wedge (d[4] \wedge \sim d[3])$$

4. Show waveforms to explain that your LFSR module works correctly.



當 seed_val=1 時，d 被 initial 成 seed 的值 7，如上圖所示，當 seed_val=0 時，d 為 $\{d'[6:0], ((d'[7] \sim d'[5]) \wedge (d'[4] \sim d'[3]))\}$ (上一個 cycle 的 d 值為 d')。

5. Show the simulation result on the terminal.

```

Applications  Places  System
File Edit View Search Terminal Help
2200 output = 118, expect = 118
2210 output = 236, expect = 236
2220 output = 217, expect = 217
2230 output = 179, expect = 179
2240 output = 103, expect = 103
2250 output = 207, expect = 207
2260 output = 158, expect = 158
2270 output = 61, expect = 61
2280 output = 123, expect = 123
2290 output = 247, expect = 247
2300 output = 239, expect = 239
2310 output = 223, expect = 223
2320 output = 191, expect = 191
2330 output = 126, expect = 126
2340 output = 253, expect = 253
2350 output = 250, expect = 250
2360 output = 244, expect = 244
2370 output = 233, expect = 233
2380 output = 211, expect = 211
2390 output = 166, expect = 166
2400 output = 76, expect = 76
2410 output = 153, expect = 153
2420 output = 51, expect = 51
2430 output = 102, expect = 102
2440 output = 205, expect = 205
2450 output = 154, expect = 154
2460 output = 53, expect = 53
2470 output = 106, expect = 106
2480 output = 212, expect = 212
2490 output = 168, expect = 168
2500 output = 81, expect = 81
2510 output = 163, expect = 163
2520 output = 70, expect = 70
2530 output = 140, expect = 140
2540 output = 24, expect = 24
2550 output = 48, expect = 48
2560 output = 96, expect = 96
2570 output = 193, expect = 193
2580 output = 131, expect = 131

*****
**                                     **
** Congratulations !!                 **
**                                     **
** Simulation PASS!!                  **
**                                     **
*****
                                     |__|
                                     / 0.0 |
                                     / ^ ^ ^ \
                                     | ^ ^ ^ ^ |w|
                                     \m _m _|

Simulation complete via $finish(1) at time 2590 NS + 0
./LFSR tb.v:54 $finish;
xcelium> exit
TOOL: xmverilog 20.09-s007: Exiting on Mar 08, 2023 at 21:43:26 CST (total: 00:00:02)
vlsicad6:/home/user2/vlsi23/vlsi23100/Lab4/ProbC %

[vlsi23100] Lab4 Lab2 LFSR.v (~Lab4/ProbC...
```

6. Show SuperLint coverage

Appendix A : Commands we will use to check your homework

| Prob | Syn | command |
|------|-----|--|
| A | pre | ncverilog register_tb.v +define+FSDB +access+r ncverilog register_tb.v +define+FSDB_ALL +access+r |
| B | pre | ncverilog top_tb.v +define+FSDB +access+r |
| C | pre | ncverilog LFSR_tb.v +define+FSDB +access+r |