National Cheng Kung University

Department of Electrical Engineering

Introduction to VLSI CAD (Spring 2024)

Lab Session 4

Register Files, Manhattan Distance and LFSR

Name	Student ID	
游宗謀	E94106151	
Practical Sections	**Points**	**Marks**
Prob A	30	
Prob B	30	
Prob C	20	
Report	15	
File hierarchy, naming…etc.	5	
Notes:		

Due Date: 15:00, March 27, 2024 @ moodle

Deliverables

1) All Verilog codes including testbenches for each problem should be uploaded.
   NOTE: Please **DO NOT** include source code in the paper report!
2) All homework requirements should be uploaded in this file hierarchy or you will not get the full credit.
   NOTE: Please **DO NOT** upload waveforms!
3) Important! TA will use the command in Appendix A to check your design under SoC Lab environment, if your code can not be recompiled by TA successfully using the commands, you will not get the full credit.
4) If you upload a dead body which we can't even compile you will get **NO** credit!
5) All Verilog file should get at least 90% superLint Coverage.
6) File hierarchy should not be changed; it may cause your code can not be recompiled by TA successfully using the autograding commands

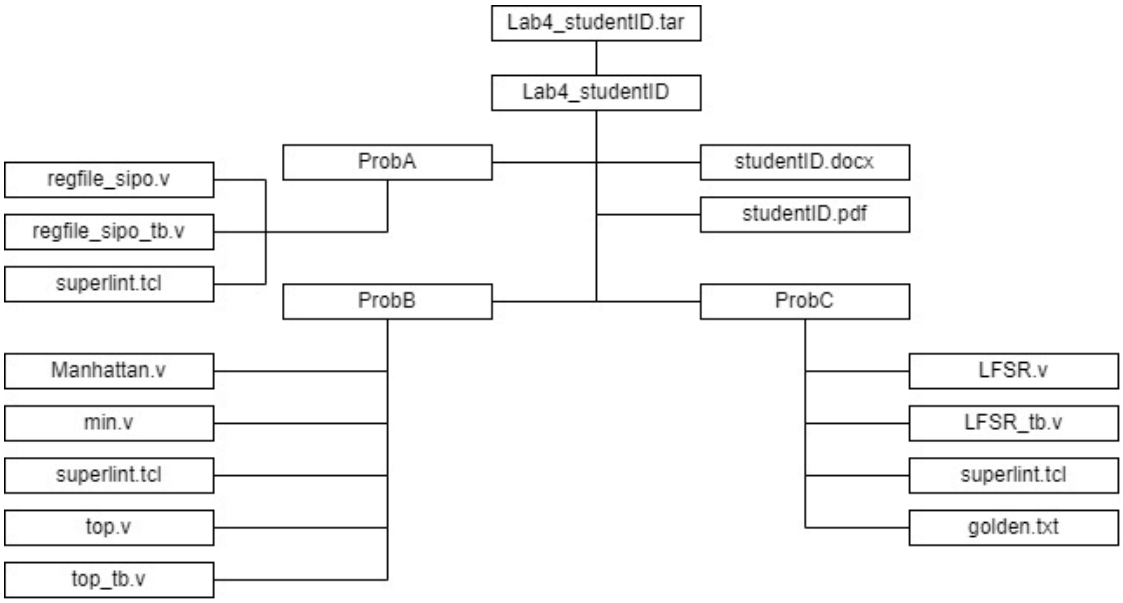
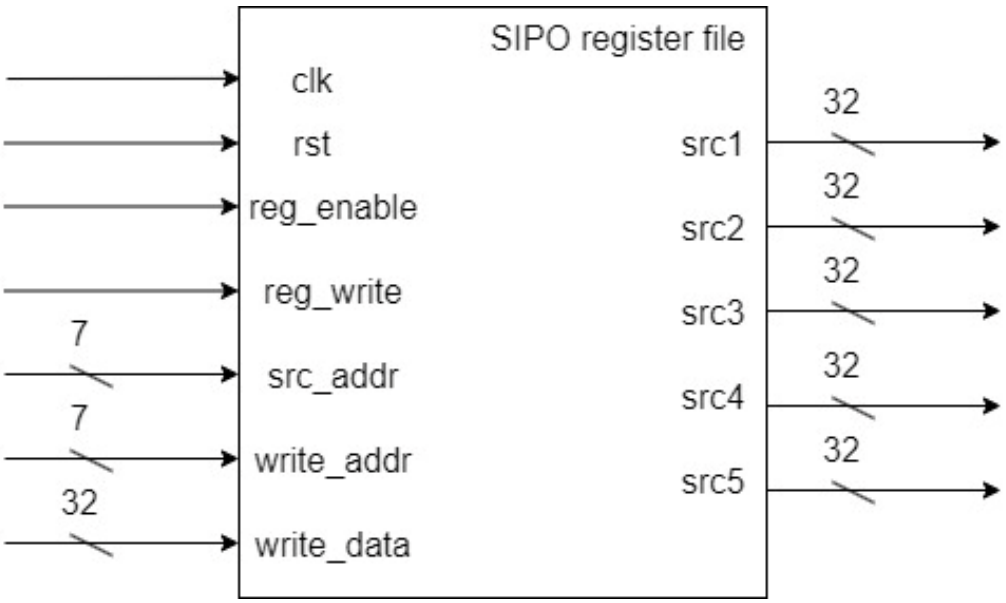


Fig.1 File hierarchy for Homework submission

Prob A: SIPO Register File

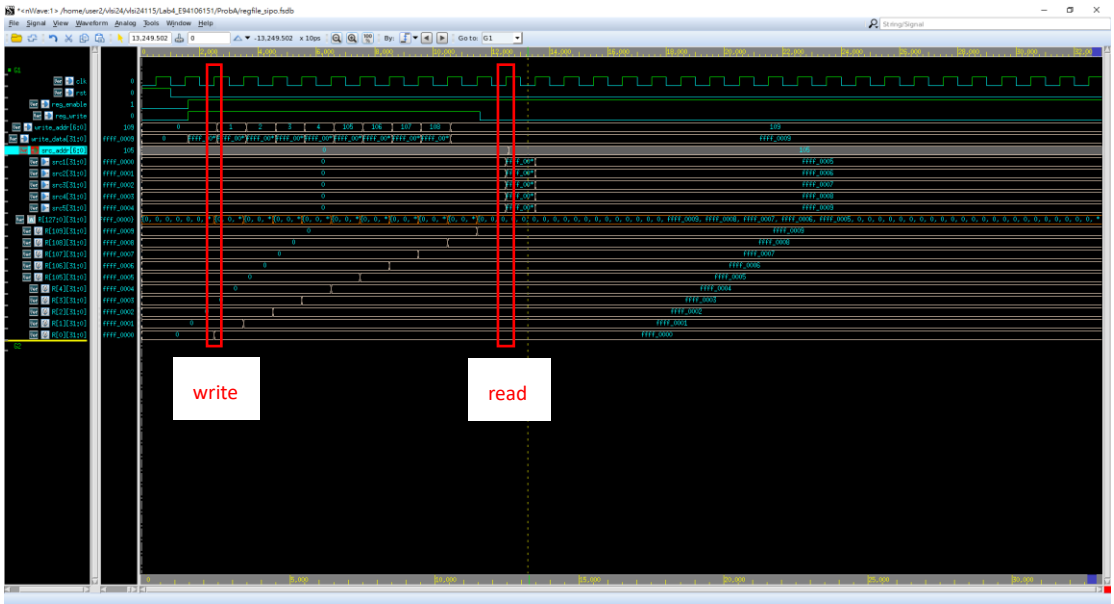


1. Based on the SIPO register file structure in LabA, please design a 128 x 32 SIPO register file with 5 output ports.
2. Port list

Signal	Type	Bits	Description
clk	input	1	clock
rst	input	1	reset
reg_enable	input	1	register file enable
reg_write	input	1	0 → read 1 → write
src_addr	input	7	source address
write_addr	input	7	write address
write_data	input	32	write data
src1	output	32	read data source1
src2	output	32	read data source2
src3	output	32	read data source3
src4	output	32	read data source4
src5	output	32	read data source5

3. Show the simulation result on the terminal.

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

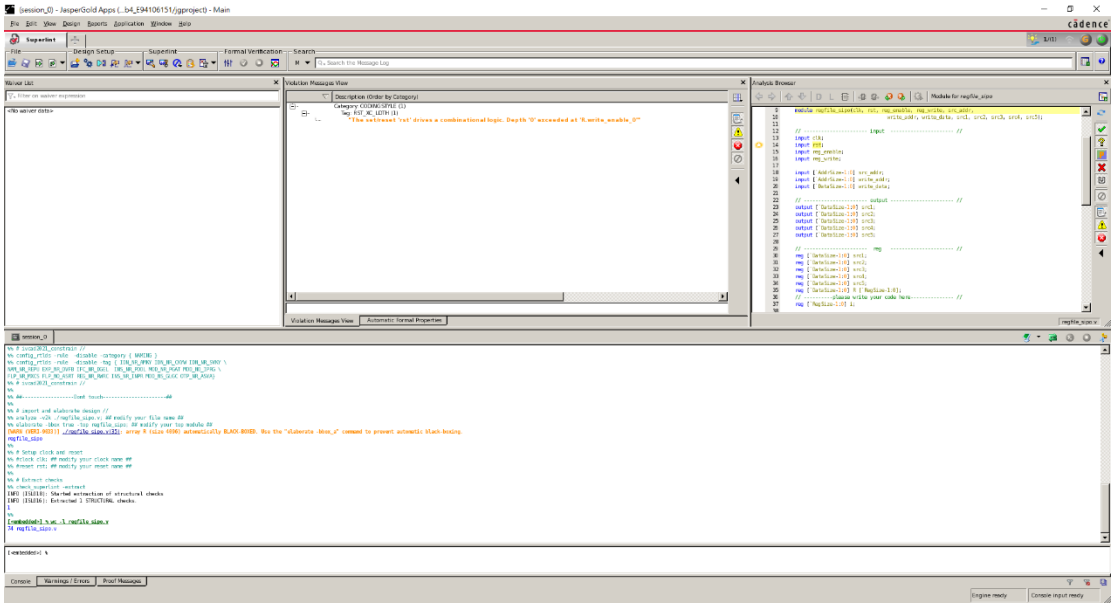


Read: 從標示出的位置可以看到要讀取的位址(src_addr)為 0 到 0+4=4，根據 register(R)可以看出 R[0] = 32’hffff0000, R[1] = 32’hffff0001, R[2] = 32’hffff0002, R[3] = 32’hffff0003, R[4] = 32’hffff0004，對應到輸出 src1 = R[0] = 32’hffff0000, src2 = R[1] = 32’hffff0001, src3 = R[2] = 32’hffff0002, src4 = R[3] = 32’hffff0003, src5 = R[4] = 32’hffff0004。

Write: 從標示出的位置可以看到要寫入的位址(write_addr)為 0、要寫入的值(write_data)為 32’hffff0000，根據 register(R)可以看出輸入後 R[0] = 32’hffff0000。

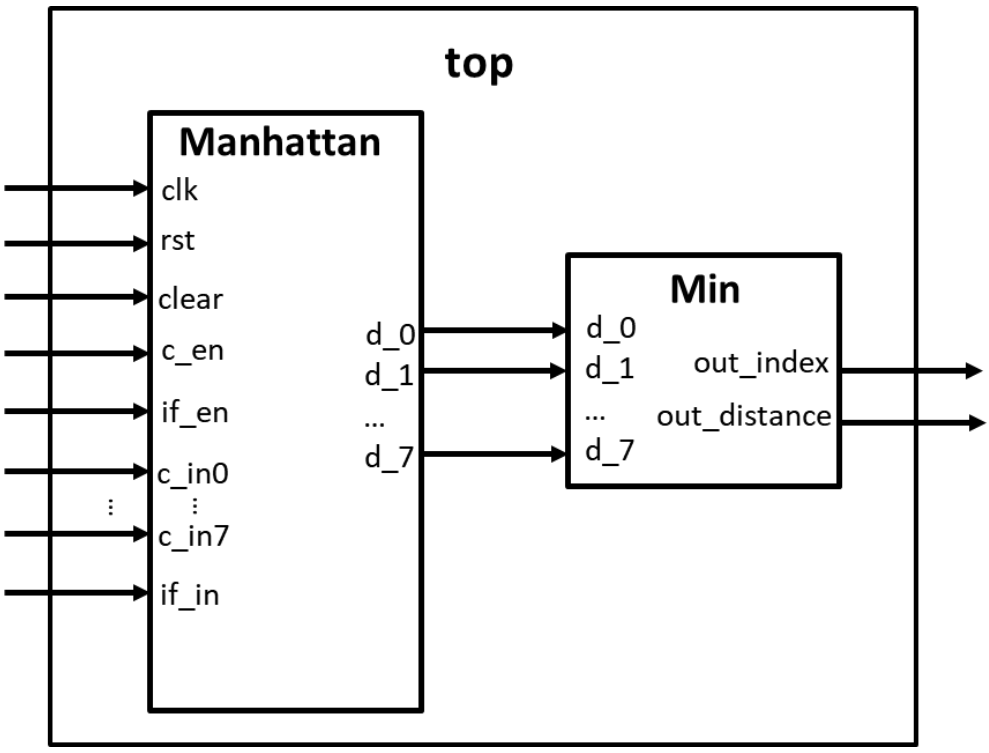
由此可知 register 有正常運作。

5. Show SuperLint coverage



Coverage = (1-(1/74))*100% = 98.65%

Prob B: Finding Smallest Distance



1. Please design a circuit that will find the smallest distance between the input feature and input colors, based on the structure given in the LAB4 slide.
2. Port list

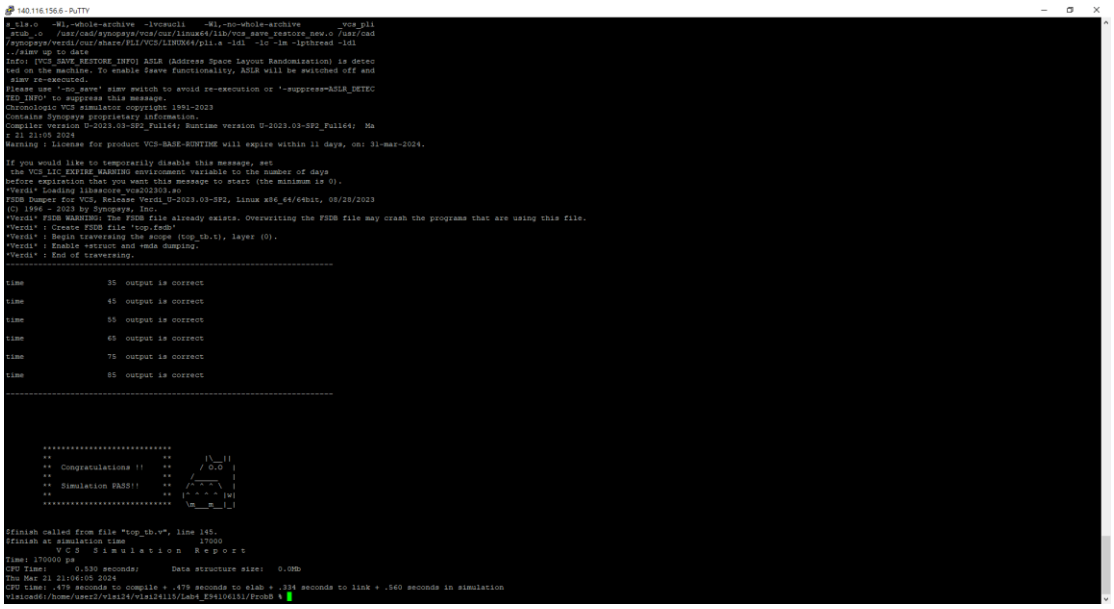
Manhattan:

Signal	Type	Bits	Description
clk	input	1	Clock pin.
rst	input	1	Reset pin.
clear	input	1	Set all registers to 0.
c_en	input	1	Write compared colors enable. When c_en is high, then c_in0~7 is available.
if_en	input	1	Write input pixel enable. When if_en is high, then if_in is available.
c_in0~7	input	24 each	Input color data.
if_in	input	24	Input input feature data.
d_0~7	output	10 each	Output distance data.

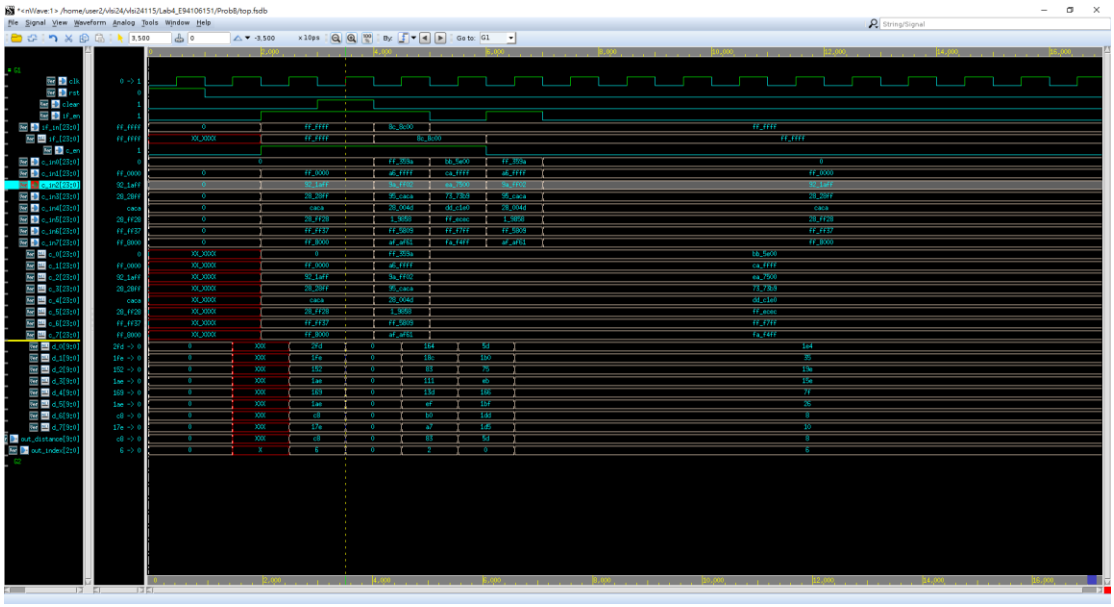
Min:

Signal	Type	Bits	Description
d_0~7	input	10 each	Input data.
out_index	output	3	Output index.
out_distance	output	10	Output minimum distance.

3. Show the simulation result on the terminal.

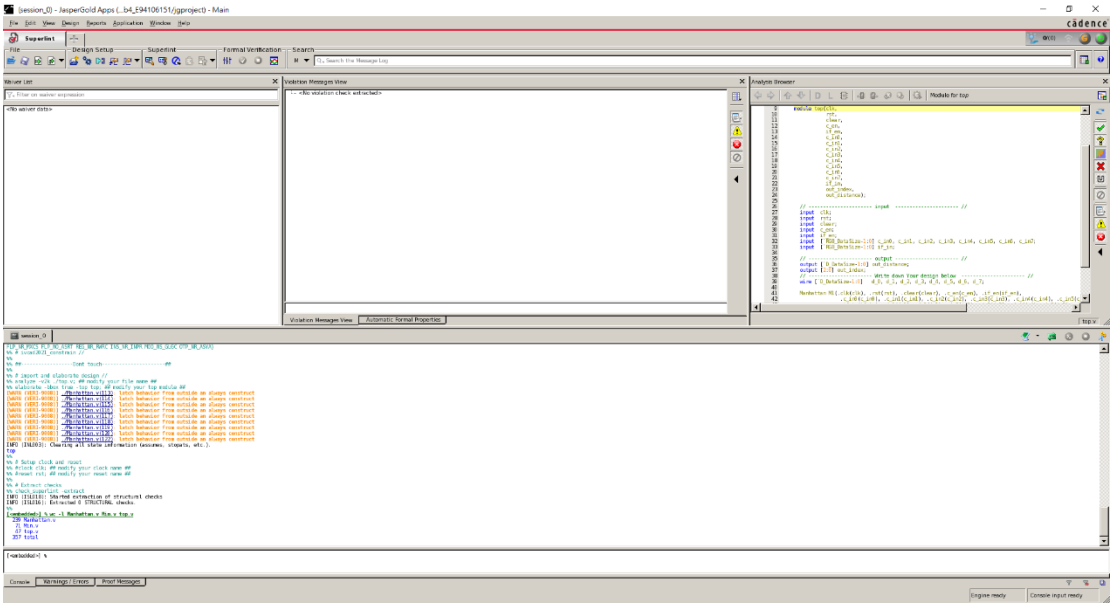


4. Show waveforms to explain that your design works correctly.



根據所標示的位置，Input feature data 為 if_en = 24’hffffff(255, 255, 255)，Input color data 6 為 c_in6 = 24’hffff37(255, 255, 55)，相差 distance data 為 d_6 = 10’h0c8(200)是相比其他輸入差最少的，所以 output minimum distance 為 out_distance = 10’h0c8，output index 為 out_index = 3’h6。

5. Show SuperLint coverage



Coverage = (1-(0/357))*100% = 100%

Prob C: LFSR

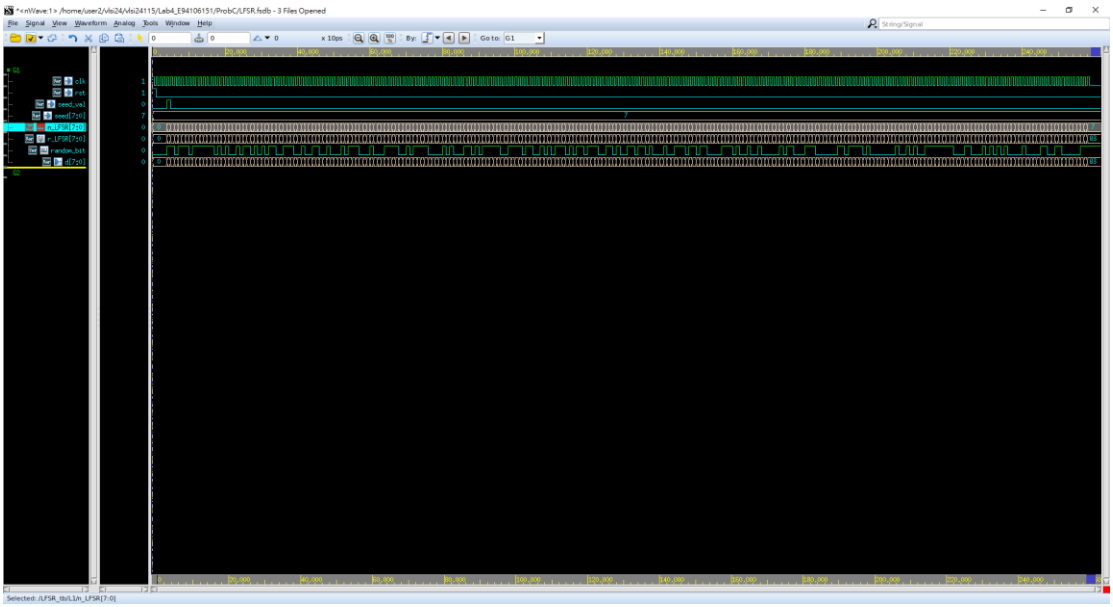
1. Please design an 8-bit-LFSR, with the given feedback function in the LAB4 slide.
2. Port list

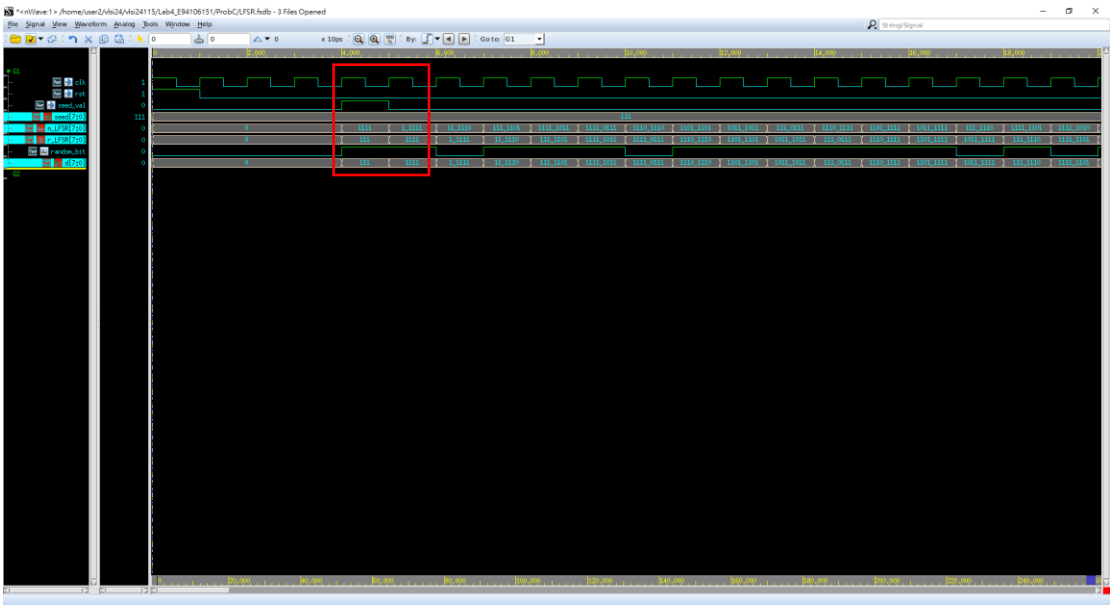
Signal	Type	Bits	Description
clk	input	1	Clock pin.
rst	input	1	Reset pin. Reset all of the flip flops to zeros.
seed_val	input	1	1: the flip flops take seed as the initial state. 0: the flip flops works as linear feedback shift register.
seed	input	8	Initial state value of LFSR.
d	output	8	Output value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

3. Feedback function

$$d[0] = (d[7] \wedge d[5]) \wedge (d[4] \wedge d[2])$$

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

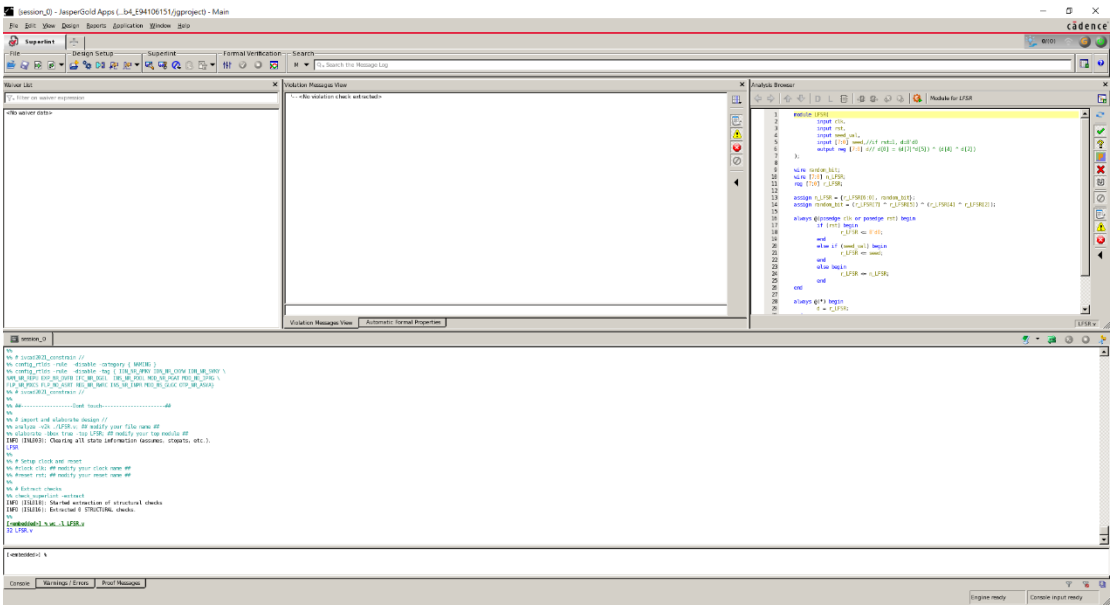




從標示的位置可以看到在 seed_val 為 1 時，seed = 8’b00000111 有確實輸入進電路當中(d = 8’b00000111)，而在下一個 clk posdege 時，根據 Feedback function 對應 d[0] = (d[7] ^ d[5]) ^ (d[4] ^ d[2]) = (0 ^ 0) ^ (0 ^ 1) = 0 ^ 1 = 1，可得 d = 8’b00001111。

5. Show the simulation result on the terminal.

6. Show SuperLint coverage



Coverage = (1-(0/32))*100% = 100%

At last, please write the lesson you learned from Lab4

在這次的實驗課中我學到我們在計算機組織中上到的 register file 是是如何透過 verilog 實作出來的，讓我更了解計算機的架構。此外還學到了 Manhattan 和 Euclidean 兩種不同計算相對距離的方法，其中 Manhattan 是這次第二題所用到的方法，我想如果要用 verilog 實作出影像辨識，這應該會是其中一小部分吧？最後是我第一次接觸到的 LFSR，我從來沒想過原來我們可以用這種方法用電路實作出接近隨機的數字產生器，沒想過原來 D type Flip-Flop 也有如此功用，讓我大開眼界。

Appendix A : Commands we will use to check your homework

Problem		Command
Prob A	Compile	% vcs -R regfile_sipo.v -full64
	Simulate	% vcs -R regfile_sipo_tb.v -debug_access+all -full64 +define+FSDB
Prob B	Compile	% vcs -R top.v -full64
	Simulate	% vcs -R top_tb.v -debug_access+all -full64 +define+FSDB
Prob C	Compile	% vcs -R LFSR.v -full64
	Simulate	% vcs -R LFSR_tb.v -debug_access+all -full64 +define+FSDB