**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 | |
| 鄭智宇 | E94096110 | |
| **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!

1. 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.
2. **If you upload a dead body which we can’t even compile, you will get NO credit!**
3. **All Verilog file should get at least 85% SuperLint Coverage.**
4. All homework requirements should be uploaded in this file hierarchy or you will not get full credit.

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

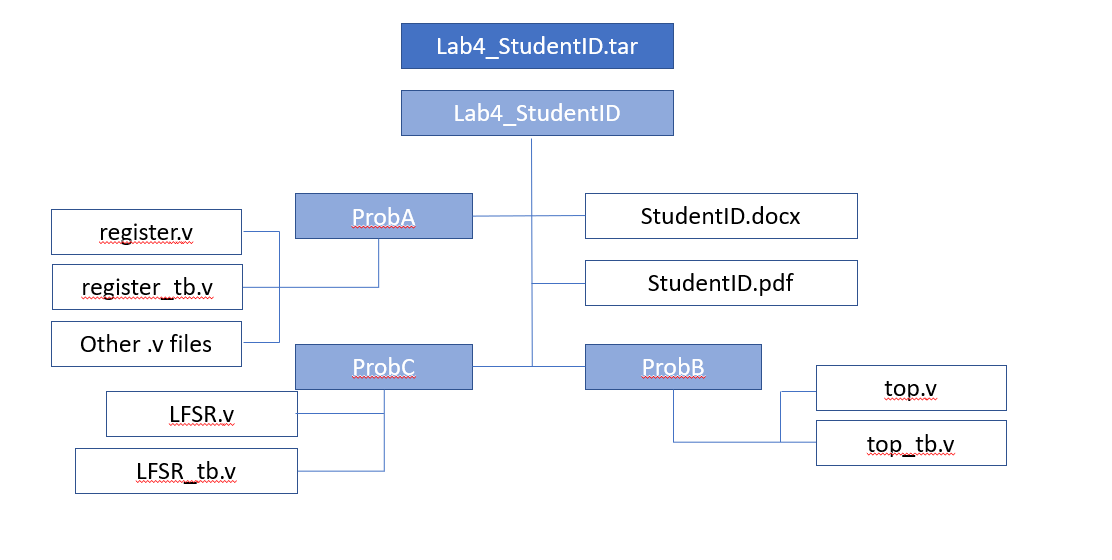
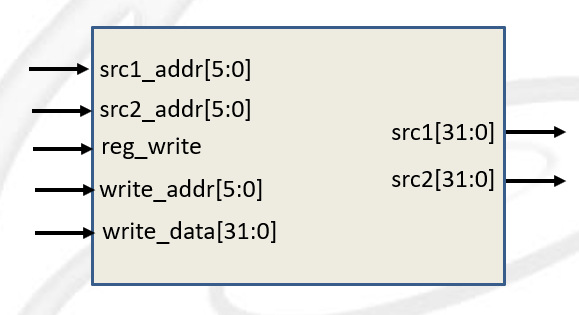
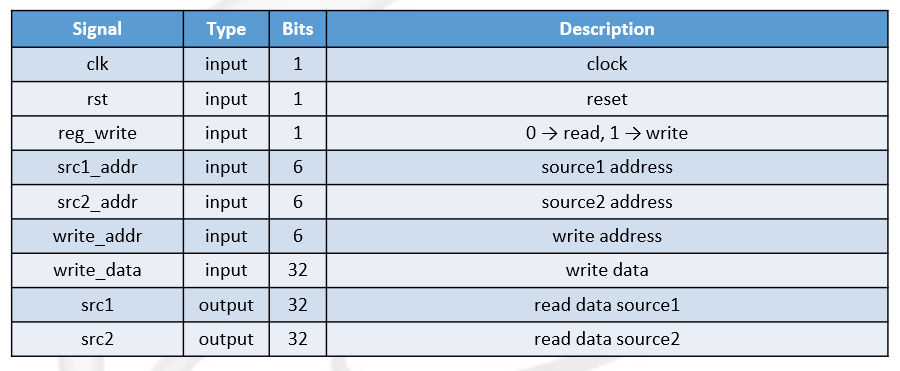


Fig.1 File hierarchy for Homework submission

Prob A: Register File



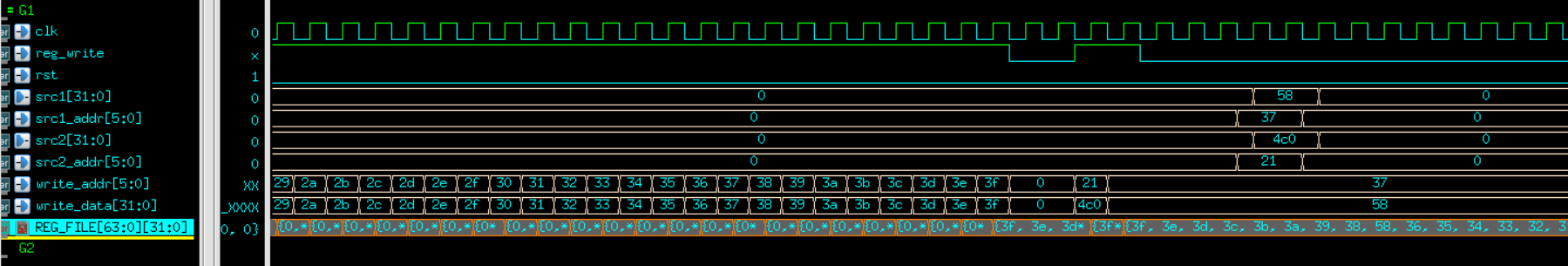
1. Based on the register file structure in LabA, please design a 64 x 32 register file by yourself.
2. Port list



1. 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**

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



**For REG\_FILE**

當reg\_write為1時，將write\_data寫入地址為write\_addr的**REG\_FILE**中

當reg\_write為0時，所有地址為的**REG\_FILE**的值保持不變

**For src1, src2**

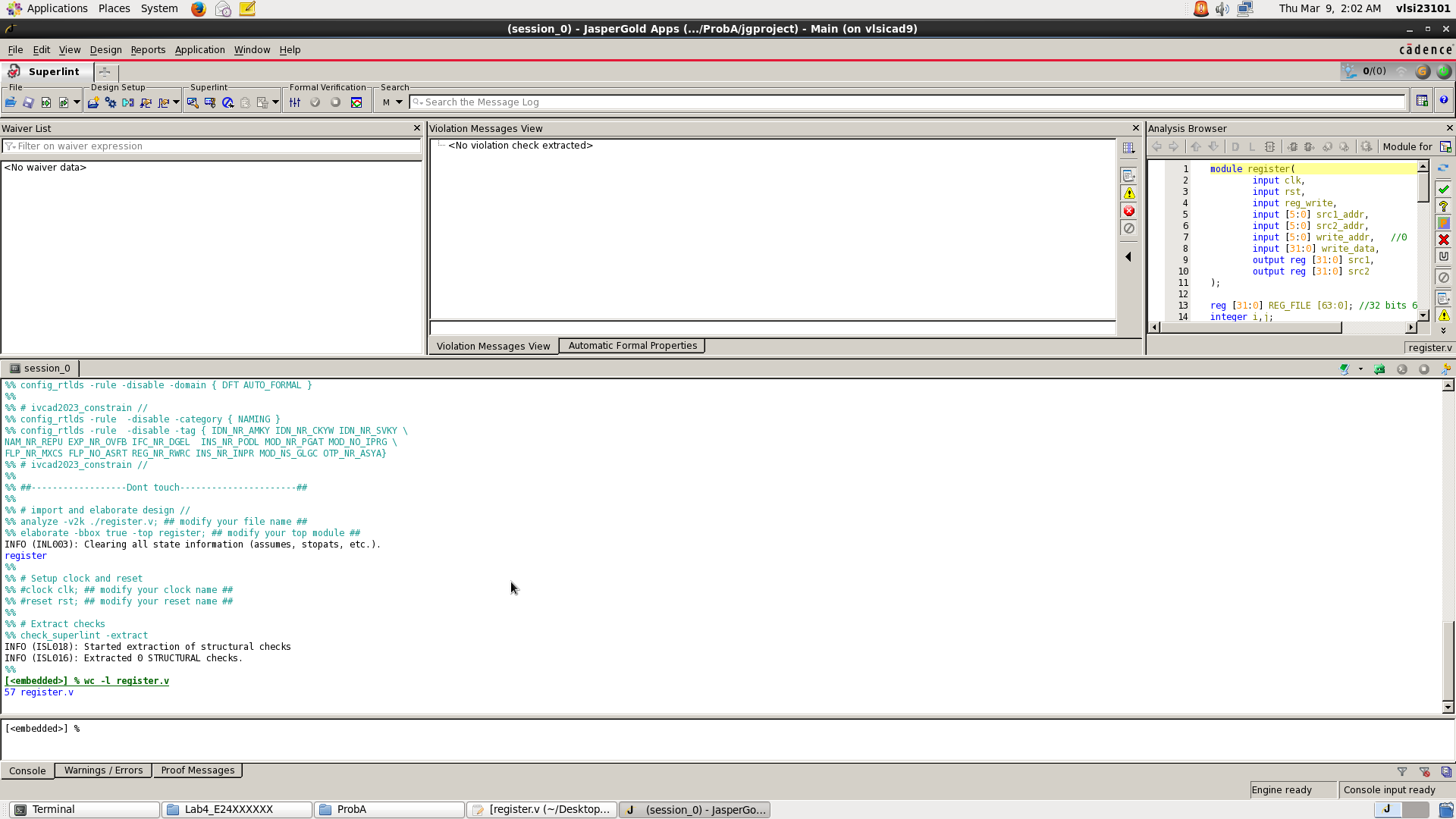
當reg\_write為0時，src1讀取src1\_addr指定的暫存器的值

當reg\_write為1時，src1的值保持不變

當reg\_write為0時，src2讀取src2\_addr指定的暫存器的值

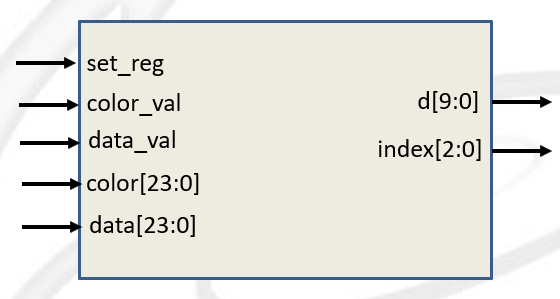
當reg\_write為1時，src2的值保持不變

1. Show SuperLint coverage



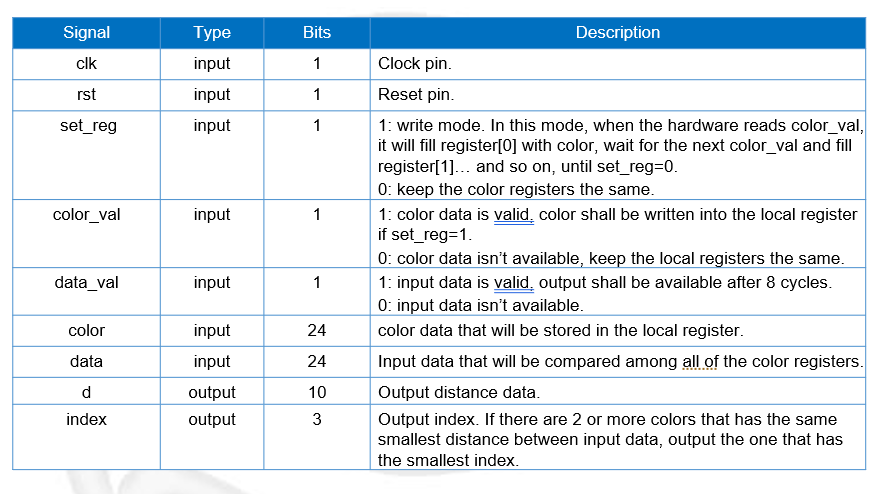
Coverage : 100 %

Prob B: Finding Smallest Distance

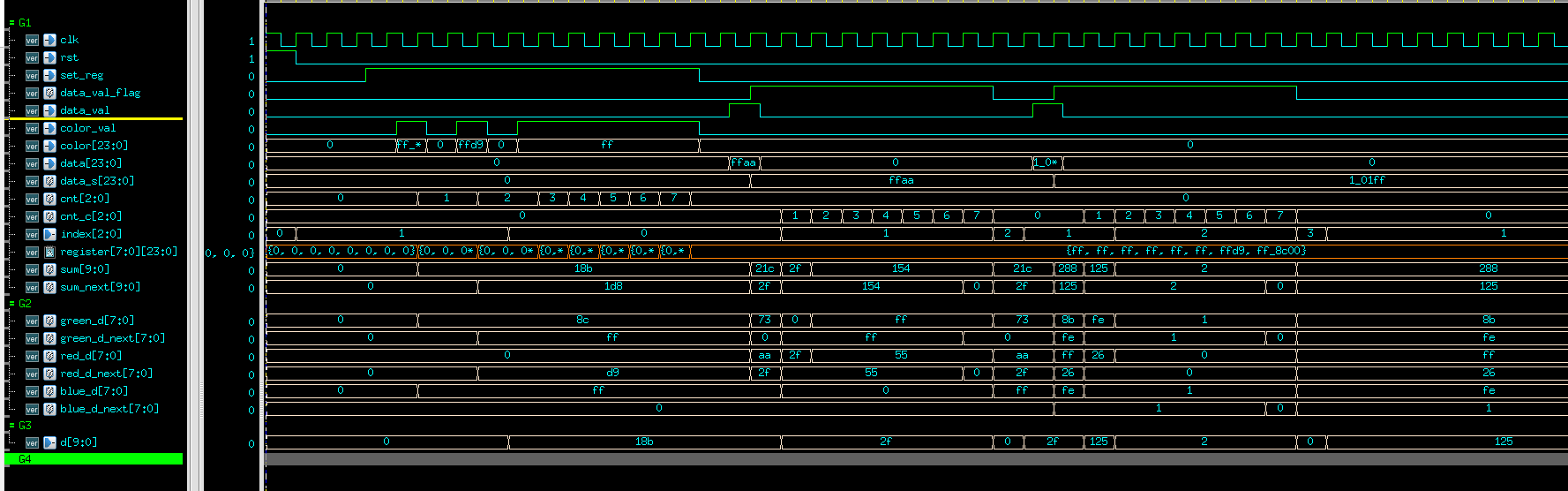


Please design a circuit that will find the smallest distance between the local registers and the input data, based on the structure given in the LAB4 slide.

Port list



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



1. register:

* 8個元素的 register 陣列，每個元素是一個 24-bit 寬的 reg。
* 這個陣列用於存儲顏色數據

1. cnt\_c:

* 3-bit 寬的 reg，
* 用於計算最近的顏色向量與輸入數據之間的距離。

1. cnt:

* 3-bit 寬的 reg，用於計數要更新哪個顏色向量。
* cnt 用於指示下一個要更新的顏色向量的索引。

1. data\_val\_flag:
   * + 用於標識輸入數據的有效性。
     + 如果 data\_val\_flag 是 1，則表示當前輸入數據有效，可以更新 register 中的數據。否則，將忽略輸入數據。
2. blue\_d, green\_d, red\_d:

這些都是 8-bit 寬的 reg，用於計算輸入數據與 register 中每個顏色向量之間的距離。

1. blue\_d\_next, green\_d\_next, red\_d\_next:

這些都是 8-bit 寬的 reg，用於計算輸入數據與下一個 register 中的每個顏色向量之間的距離。

1. sum, sum\_next:

這些都是 10-bit 寬的 reg，用於計算輸入數據與 register 中每個顏色向量之間的總距離，用於決定d與index。

1. data\_s:

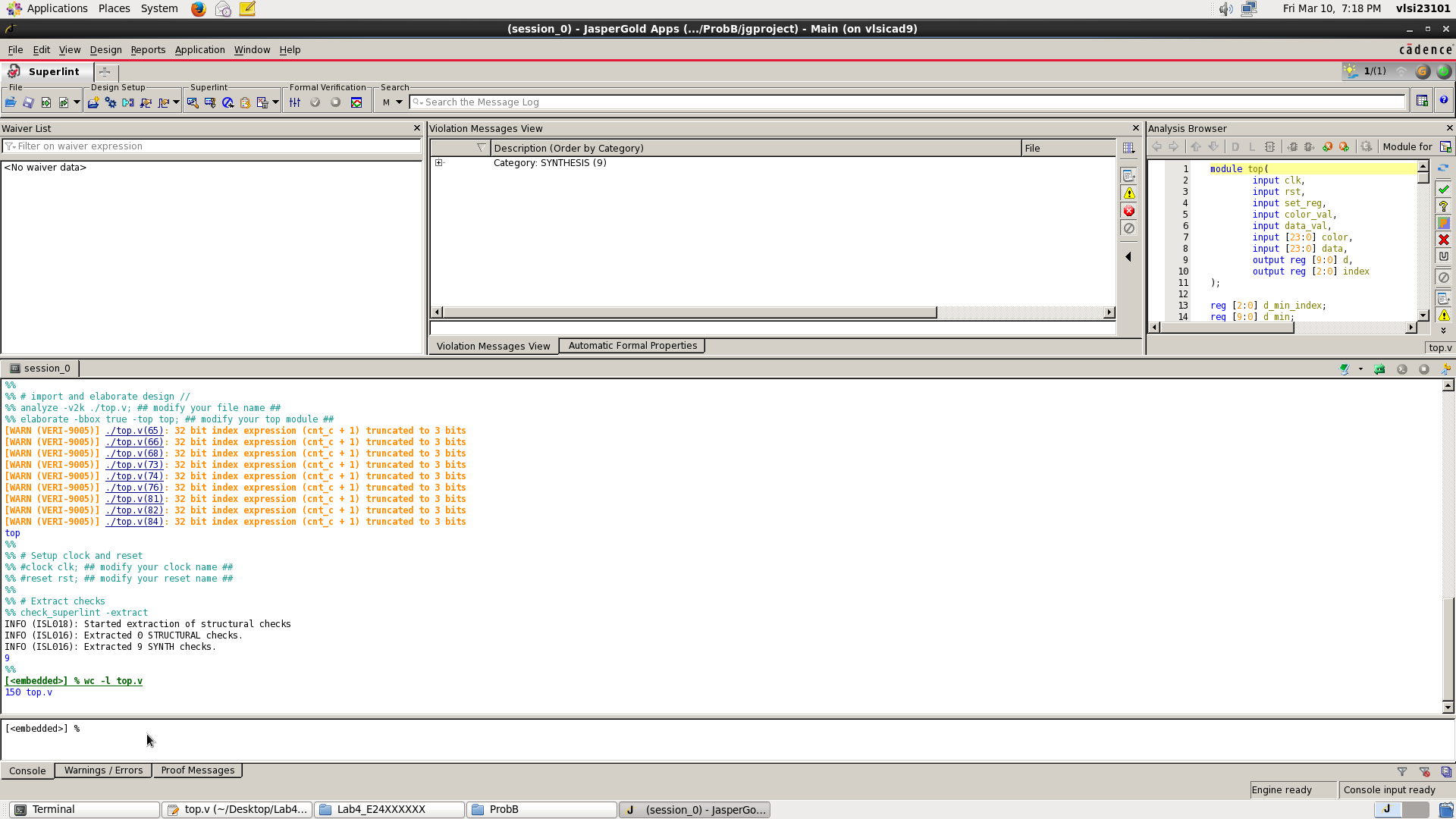
這是一個 24-bit 寬的 reg，用於存儲最近的輸入數據。當 data\_val 信號有效時，data\_s 將更新為輸入數據。

1. d, index:

這些都是輸出信號，分別是 10-bit 寬和 3-bit 寬的 reg。它們用於記錄最近的顏色向量和其索引。當 cnt\_c = 0 時，d 和 index 將被更新。

運行步驟:

1. 宣告了一個大小為8的暫存器陣列register，其中每個元素都是24bits reg。
2. 設置了一個data\_s變數，其中存儲了從data輸入獲取的24位數據。
3. 計算與記錄了data\_s與register中指定的寄存器之間的距離，分別是藍色分量、綠色分量和紅色分量，以及與下一個寄存器之間的距離。
4. 計算並記錄了sum和sum\_next，分別表示data\_s與register[cnt\_c]和register[cnt\_c+1]之間的總距離。
5. 根據sum和sum\_next的值，更新d和index變數的值，以選擇最接近的顏色寄存器。
6. 計算和更新了cnt\_c變數的值，以指示目前正在檢查哪個寄存器。
7. 計算和更新了cnt變數的值，以指示目前正在設置哪個寄存器。
8. 初始化了register陣列，並在set\_reg和color\_val信號接收到時，將顏色值存儲在選定的寄存器中。
9. Show SuperLint coverage

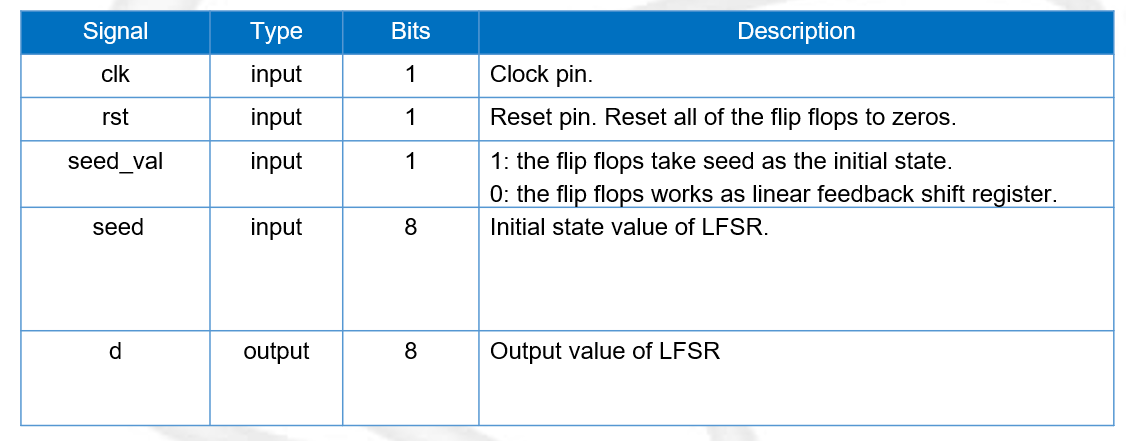


9 warning

Coverage : 94%

Prob C: LFSR

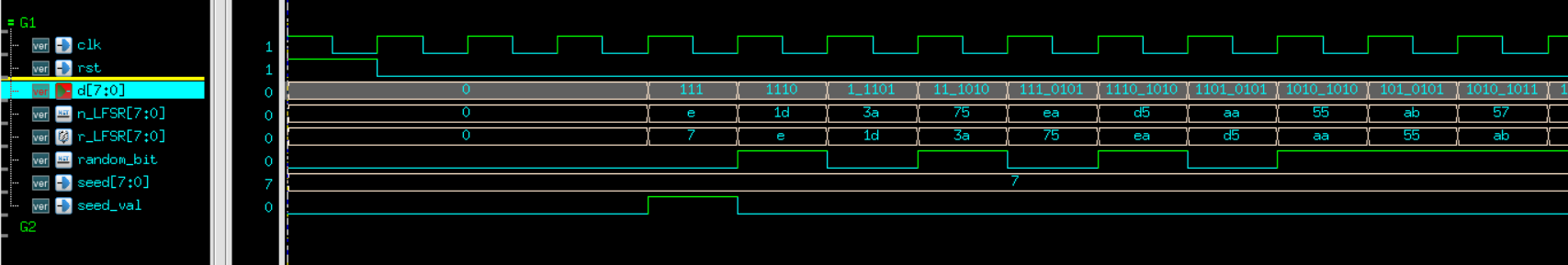
1. Please design an 8-bit-LFSR, with the given feedback function in the LAB4 slide.
2. Port list



1. Feedback function



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



Step1：初始化 d 和 r\_LFSR 為 0。

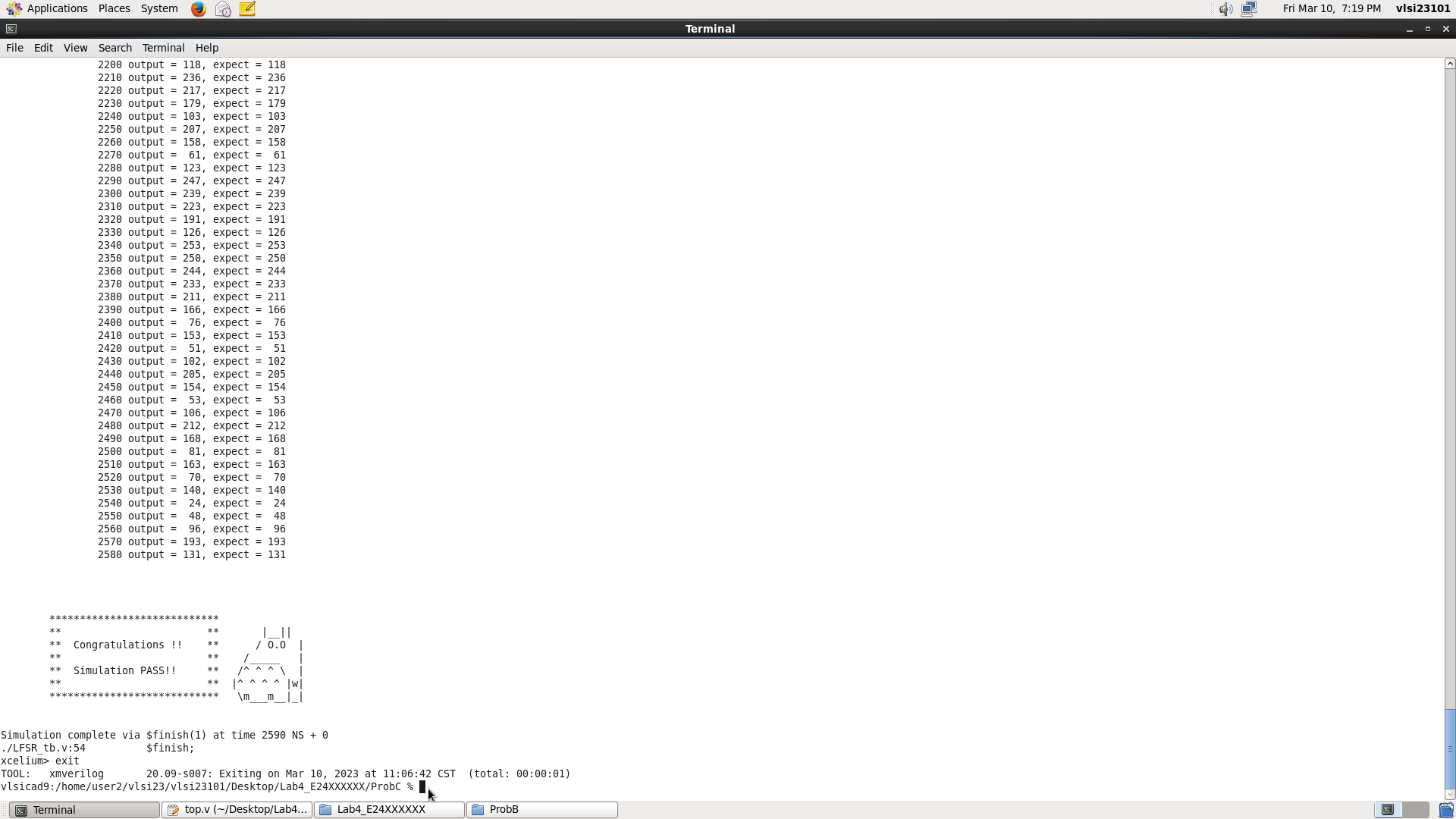
Step2 : 當時鐘節拍到達時，檢查 rst 是否為 1，是則將 r\_LFSR 設為 0。

Step3 : 若種子值有效 seed\_val 為 1，則使用 seed 初始化 r\_LFSR。

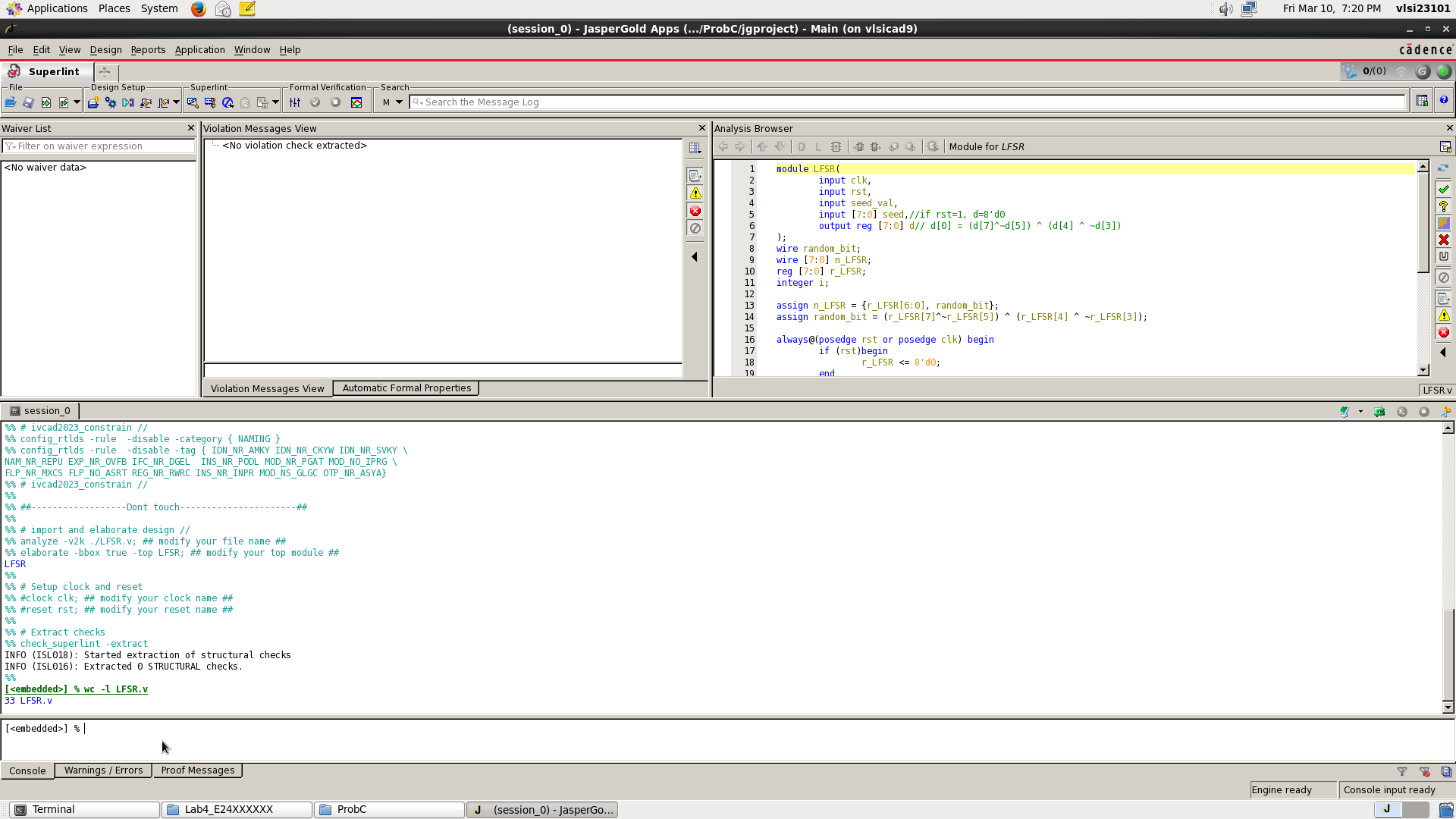
Step4 : 計算 random\_bit 和 n\_LFSR，並將 n\_LFSR 賦值給 r\_LFSR。

Step5 : 輸出 d 為 r\_LFSR 值，該值的前 8 位是當前的隨機序列值。

1. Show the simulation result on the terminal.



1. Show SuperLint coverage



coverage : 100%

1. At last, please write the lesson you learned from Lab4

組合電路與時序電路得差別與使用時機

Appendix A : Commands we will use to check your homework

