

# National Cheng Kung University

## Department of Electrical Engineering

### *Introduction to VLSI CAD (Spring 2024)*

#### Lab Session 5

#### FSM&Synthesis of Sequential Logic

Name	Student ID	
鄭喆嚴	E14096724	
Practical Sections	Points	Marks
Lab in class	15	
Prob A	20	
Prob B	10	
Prob C	15	
Report	35	
File hierarchy, naming...etc.	5	
Notes:		

**Due Date: 14:59, April 4, 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 follow the naming rule 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 cannot even compile, you will get **NO** credit!
- 5) All Verilog file before synthesizing should get at least **95%** Superlint Coverage.
- 6) Lab5\_Student\_ID.tar (English alphabet of Student\_ID should be **capital**.)

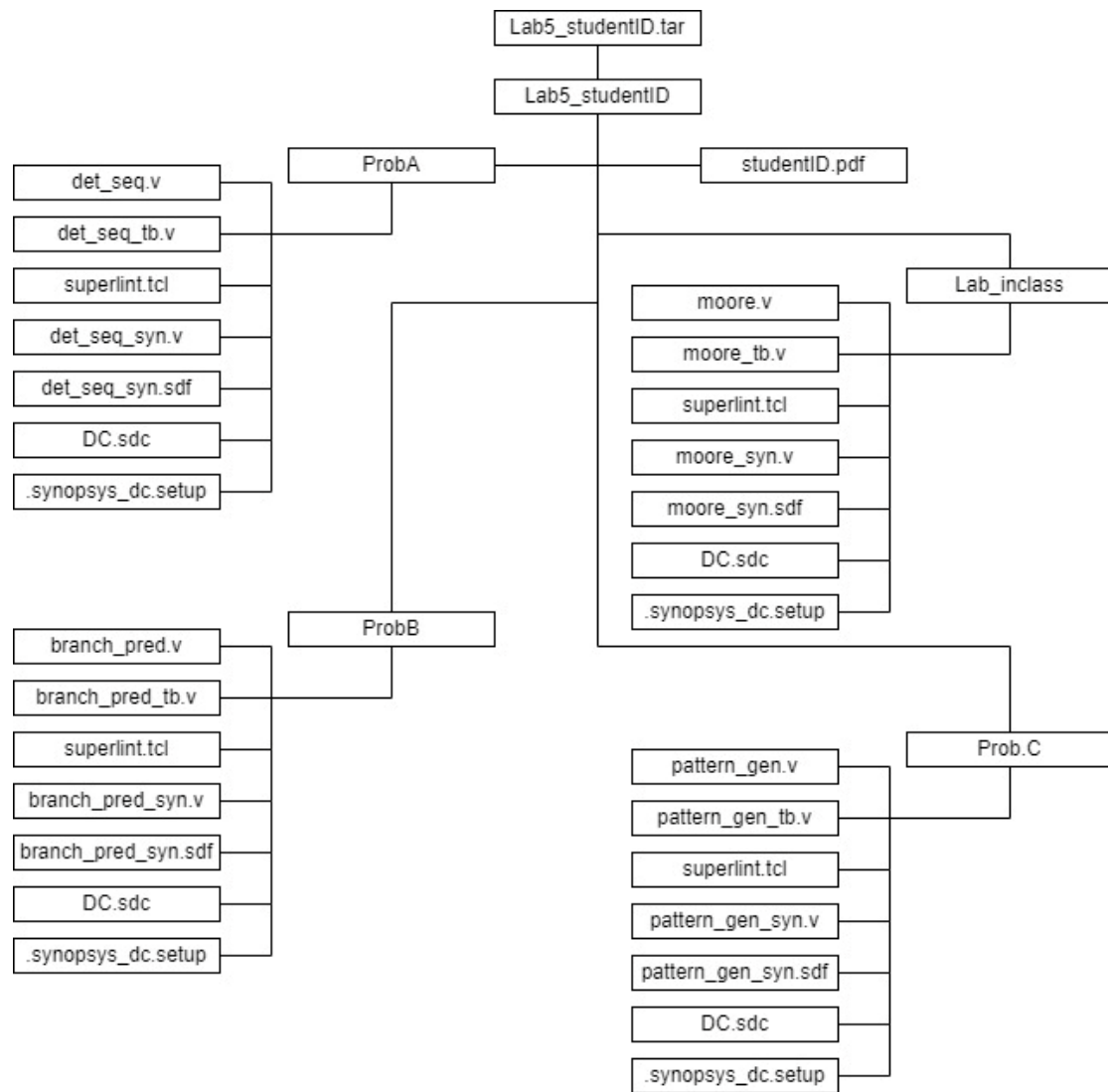
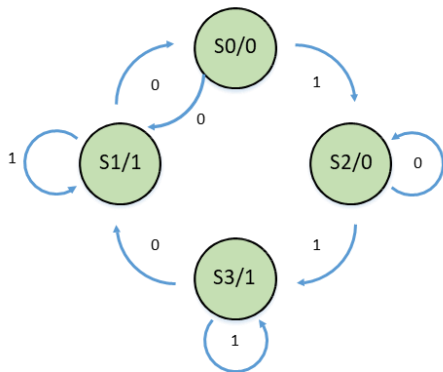


Fig.1 File hierarchy for Homework submission

*Lab in class: Design a circuit “Moore machine”*

- 1) Design a Moore machine circuit that can be synthesized. The following is Moore machine module’s specification. (Do **NOT** add or delete I/O ports, but you can change their behavior.)



Current State	Next State		qout
	din=0	din=1	
S0=00	S1	S2	0
S1=01	S0	S1	0
S2=10	S2	S3	1
S3=11	S1	S3	1

Signal	Type	Bits	Description
clk	input	1	Clock pin.
rst	input	1	Reset pin. Reset all of the flip flops to zeros.
din	input	1	Control signal for fsm.
qout	output	1	1:when current state==S1 or current state==S3 0: when current state==S0 or current state==S2

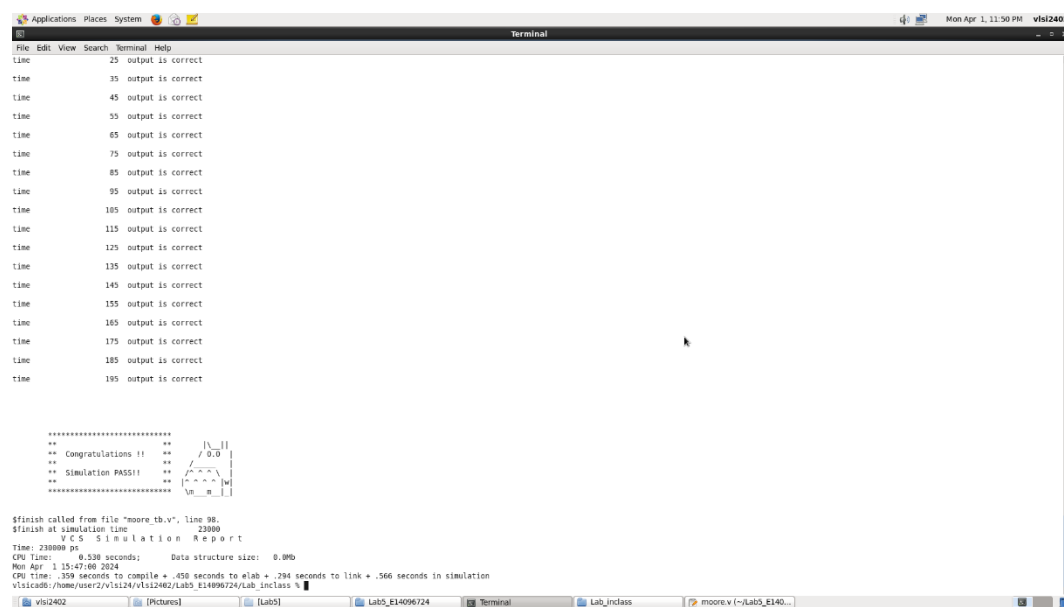
- 2) Please describe your FSM in detail

Explanation about your FSM				
	state	next_state		output
		din = 0	din = 1	
	S0	S1	S2	q = 0
	S1	S0	S1	q = 0
	S2	S2	S3	q = 1
	S3	S1	S3	q = 1

- 3) After synthesizing your design, you may have some information about the circuit. Please fill in the following form.

Timing (slack)	Area (total cell area)	Power (total)
0.32	4.147200	6.7439e-03mW

Your simulation result on the terminal.

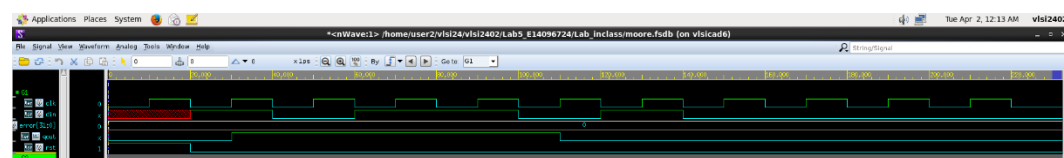


Your waveform :

RTL:



Synthesis:



Explanation of your waveform :

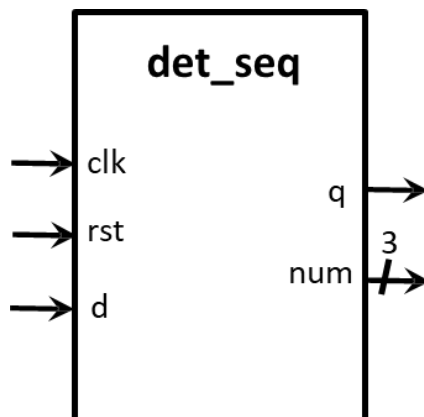
波形圖中 error 一直都為 0，代表全部的測資皆正確，波行中還可以看到這是一個 moore machine，state 隨著輸出改變，而 output 只隨著 state 改變，可以看到 state 和 output 的結果皆與 state diagram 和 state table 相符。

The screenshot shows the IntelliJ IDEA IDE with the 'JasperGold App' project open. The main editor displays the 'superlint' class file, which is a Java class for linting SuperLint rules. The class contains a main method that runs a series of checks on a set of rules. The IDE interface includes a top toolbar, a menu bar, and several tool windows: 'Superlint' (showing the current file), 'Violations Messages View' (showing linting results), 'Analysis Browser' (showing a list of rules), and 'Console' (showing the output of the linting process). The console output shows that the linting process was successful and that no violations were found.

moore.v :  $(1 - 0/52) * 100\% = 100\%$

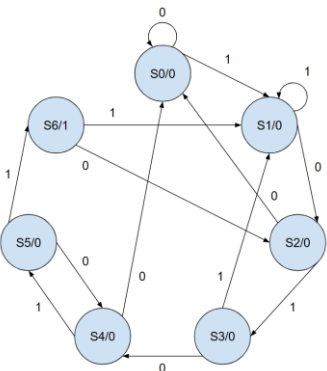
*ProbA: Design a circuit “detecting pattern 101011”*

- 1) Design a pattern seq-detecting circuit that can be synthesized with **moore machine**. The following is det\_seq module's specification. (Do **NOT** add or delete I/O ports, but you can change their behavior.)



Signal	Type	Bits	Description
clk	input	1	clock
rst	input	1	reset, active high
d	input	1	pattern bit
q	output	1	When detect pattern 101011, q pulls to high at next clock posedge. Otherwise, q is low.
num	output	3	Count the number of pattern 101011

**2) Please describe your FSM in detail**

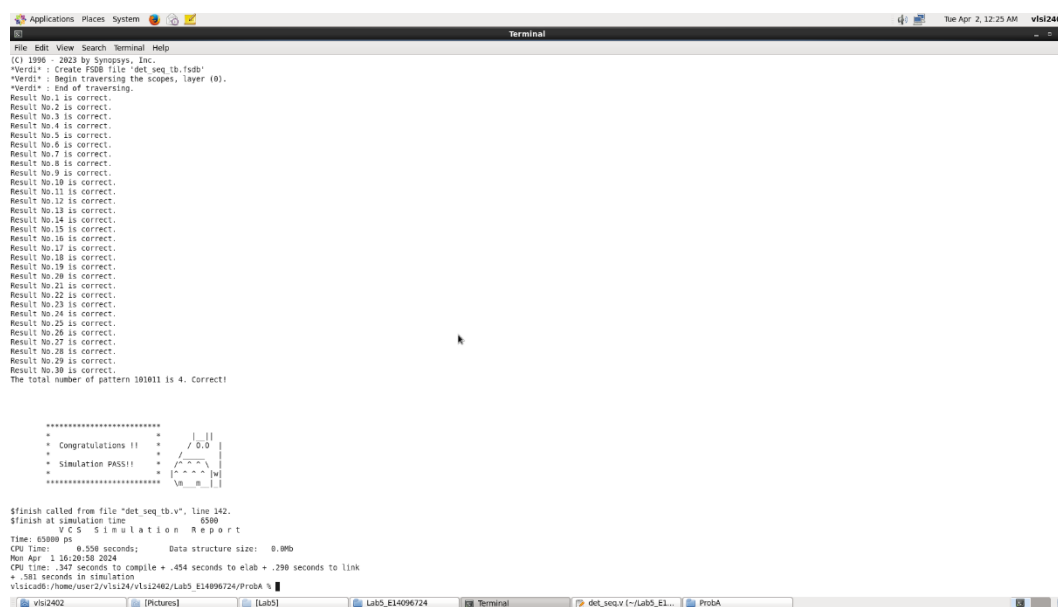
Explanation about your FSM					
	state	next_state		output	state information
		d = 0	d = 1		
	S0	S0	S1	q = 0	initial
	S1	S2	S1	q = 0	receive 1
	S2	S0	S3	q = 0	receive 10
	S3	S4	S1	q = 0	receive 101
	S4	S0	S5	q = 0	receive 1010
	S5	S4	S6	q = 0	receive 10101
	S6	S2	S1	q = 1	receive 101011

**3) After synthesizing your design, you may have some information about the circuit. Please fill in the following form.**

Timing (slack)	Area (total cell area)	Power (total)
0.30	12.286080	1.4213e-02mW

#### 4) Please attach your design waveforms.

Your simulation result on the terminal.



```
(C) 1990 - 2023 by Synopsys, Inc.
*Verdi*: Create FSDB file 'det_seq.tb.fsdb'
*Verdi*: Begin traversing the scopes, layer (0).
*Verdi*: End of traversing.
Result No.1 is correct.
Result No.2 is correct.
Result No.3 is correct.
Result No.4 is correct.
Result No.5 is correct.
Result No.6 is correct.
Result No.7 is correct.
Result No.8 is correct.
Result No.9 is correct.
Result No.10 is correct.
Result No.11 is correct.
Result No.12 is correct.
Result No.13 is correct.
Result No.14 is correct.
Result No.15 is correct.
Result No.16 is correct.
Result No.17 is correct.
Result No.18 is correct.
Result No.19 is correct.
Result No.20 is correct.
Result No.21 is correct.
Result No.22 is correct.
Result No.23 is correct.
Result No.24 is correct.
Result No.25 is correct.
Result No.26 is correct.
Result No.27 is correct.
Result No.28 is correct.
Result No.29 is correct.
Result No.30 is correct.
The total number of pattern 101011 is 4. Correct!

*****
* Congratulation !!
*
* Simulation PASS!!
*
*****

$finish called from file "det_seq.tb.v", line 342.
$finish at simulation time 65900
VCS Simulation Report
Time: 65900 ps
CPU Time: 0.358 seconds; Data structure size: 0.0Mb
Mon Apr 1 16:20:50 2024
CPU time: 347 seconds to compile + .454 seconds to elab + .290 seconds to link
+ .581 seconds in simulation
visi2402/home/user2/visi2402/Lab5_E14096724/ProbA/det_seq.tb.fsdb (on visicad6)
```

Your waveform :

RTL:



Synthesis

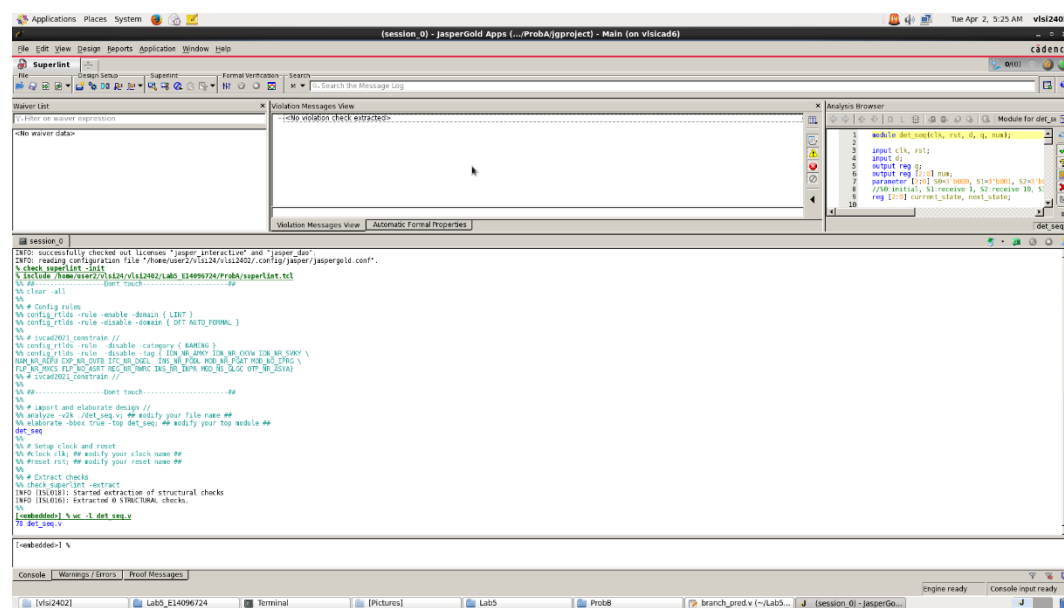


Explanation of your waveform :

波形圖中 error 一直都為 0，代表全部的測資皆正確，波行中還可以看到這是一個 moore machine，state 隨著輸出改變，而 output 只隨著 state 改變，可以看到 state 和 output 的結果皆與 state diagram 和 state table 相符。



## Superlint Coverage

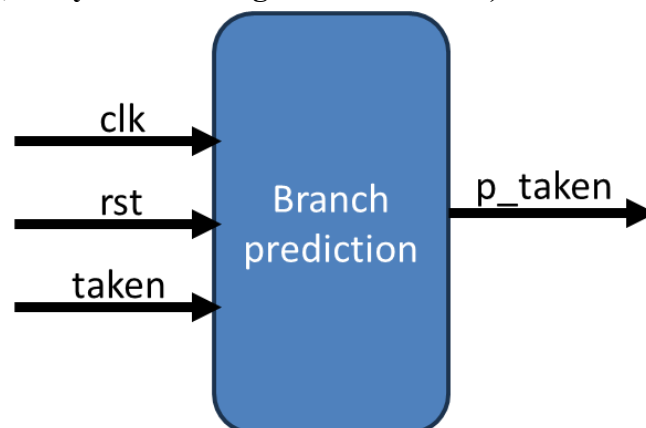


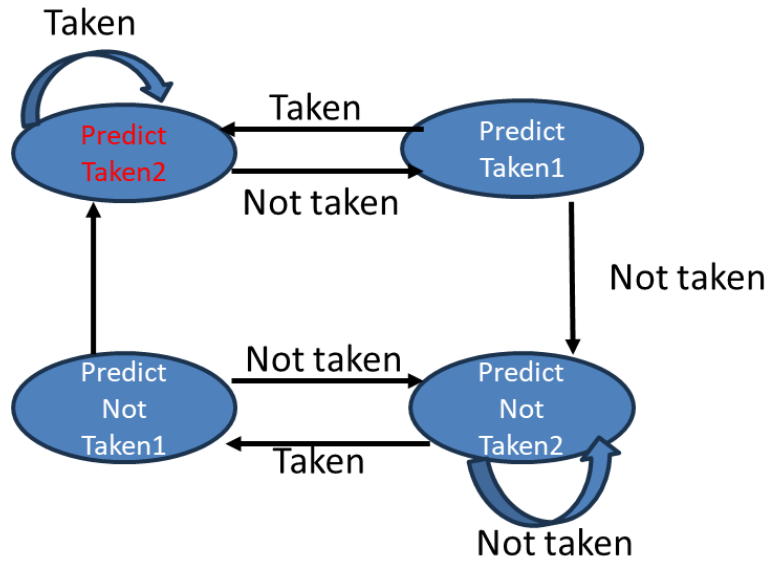
Coverage :

Det\_seq.v :  $(1 - 0/70) * 100\% = 100\%$

## ProbB: Design a 2-bit branch prediction

- 1) Design a 2-bit branch prediction with **moore machine**. The following is 2-bit branch prediction module's specification. (Do **NOT** add or delete any I/O ports, but you can change their behavior.)





2) Please describe your FSM in detail.

# Explanation about your FSM

```

graph LR
    S0_1((S0/1)) -- Taken --> S0_1
    S0_1 -- Taken --> S1_1((S1/1))
    S1_1 -- Not taken --> S0_1
    S1_1 -- Not taken --> S2_0((S2/0))
    S2_0 -- Taken --> S3_0((S3/0))
    S3_0 -- Not taken --> S2_0
    S2_0 -- Not taken --> S2_0
  
```

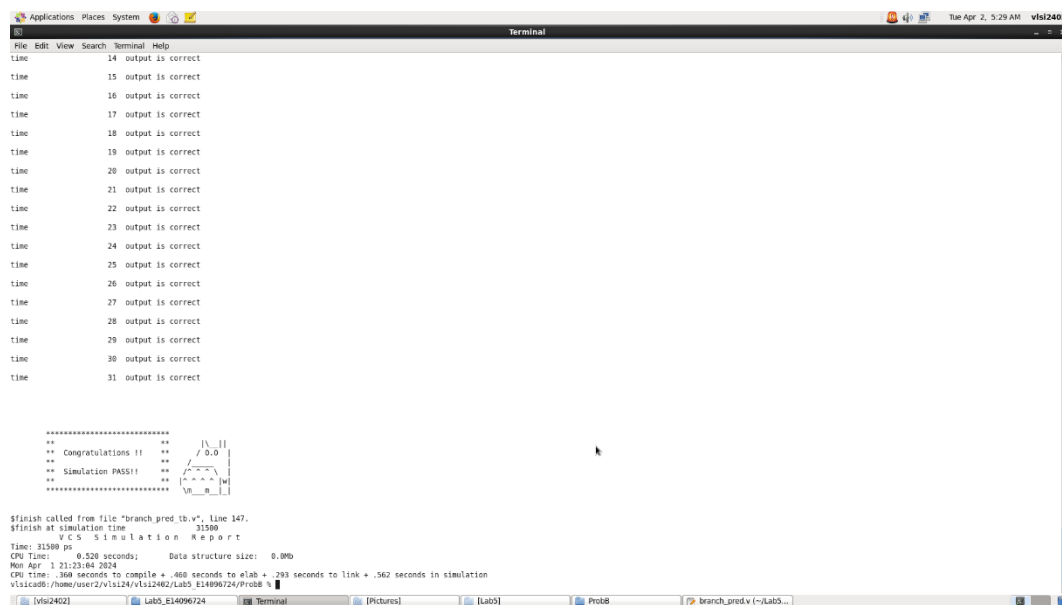
state	next_state		output	state information
	taken = 0	taken = 1		
S0	S1	S0	p_taken = 1	Predict Taken 2
S1	S2	S0	p_taken = 1	Predict Taken 1
S2	S2	S3	p_taken = 0	Predict Not Taken 2
S3	S2	S0	p_taken = 0	Predict Not Taken 1

3) After synthesizing your design, you may have some information about the circuit. Please fill in the following form.

Timing (slack)	Area (total cell area)	Power (total)
0.33	3.784320	5.6621e-03mW

#### 4) Please attach your design waveforms.

Your simulation result on the terminal.



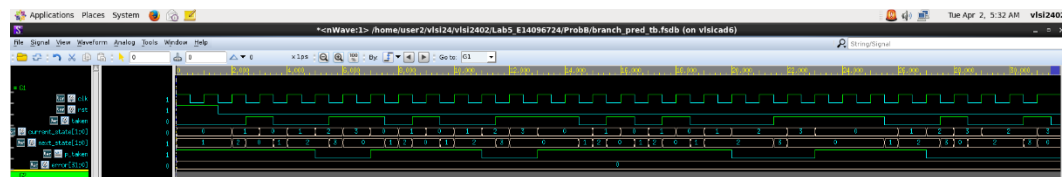
```
time 14 output is correct
time 15 output is correct
time 16 output is correct
time 17 output is correct
time 18 output is correct
time 19 output is correct
time 20 output is correct
time 21 output is correct
time 22 output is correct
time 23 output is correct
time 24 output is correct
time 25 output is correct
time 26 output is correct
time 27 output is correct
time 28 output is correct
time 29 output is correct
time 30 output is correct
time 31 output is correct

*****
**      Congratulations !!      **
**                               **
**      Simulation PASSED!!      **
**                               **
*****

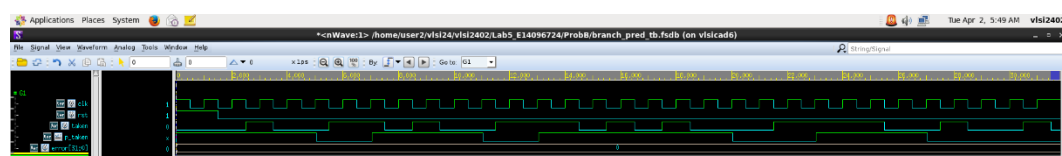
$finish called from file "branch_pred_tb.v", line 147.
$finish at simulation time 31500
V C S S i m u l a t i o n   R e p o r t
Time: 31500 ps
CPU Time: 0.528 seconds; Data structure size: 0.000
Mon Apr 1 21:23:04 2024
CPU time: .360 seconds to compile + .488 seconds to elab + .293 seconds to link + .562 seconds in simulation
visicad8:/home/user2/vlsi24/vlsi2402/Lab5_E14096724/Prob8 %
```

Your waveform :

RTL:



Synthesis



Explanation of your waveform :

波形圖中 error 一直都為 0，代表全部的測資皆正確，波行中還可以看到這是一個 moore machine，state 隨著輸出改變，而 output 只隨著 state 改變，可以看到 state 和 output 的結果皆與 state diagram 和 state table 相符。

## Superlint Coverage

```

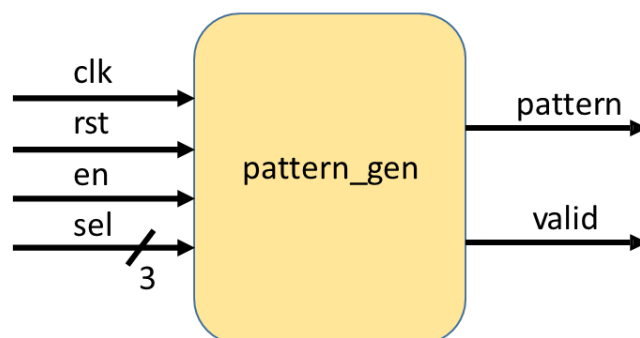
INFO: reading configuration file "/home/user2/visi2402/conf/jasper/jaspergold.conf".
% check superlint -lcl
% include /home/user2/visi2402/Lab5_E14096724/Prob5/jasperlint.tcl
% #-----DONT Touch-----#
% clear -all
% # Config rules
% config_rtl2s -rule -enable -domain ( LINT )
% config_rtl2s -rule -disable -domain ( DFT Auto_FORMAL )
% # Local2021 contrast //
% config_rtl2s -rule -disable -category ( RANDOM )
% config_rtl2s -rule -disable -log ( Lint_Warning_Lint_Warning_Lint_Warning_Lint_Warning )
% config_rtl2s -rule -disable -log ( Lint_Warning_Lint_Warning_Lint_Warning_Lint_Warning )
% config_rtl2s -rule -disable -log ( Lint_Warning_Lint_Warning_Lint_Warning_Lint_Warning )
% config_rtl2s -rule -disable -log ( Lint_Warning_Lint_Warning_Lint_Warning_Lint_Warning )
% # Local2021 contrast //
% #-----DONT Touch-----#
% # Import and elaborate design //
% analyze -vls /branch_pred.v ## modify your file name ##
% elaborate -block true -top branch_pred ## modify your top module ##
% # Setup clock and reset
% #clock clk; ## modify your clock name ##
% #reset rst; ## modify your reset name ##
% # Extract checks
% check superlint -extract
INFO: [135.036] Started extraction of structural checks.
INFO: [135.036] Extracted 0 STRUCTURAL checks.
% [embedded] % vcs -l branch_pred.v
% # branch_pred.v
%
[embedded-1] %
  
```

Coverage :

$$\text{branch\_pred.v} : (1 - 0/45) * 100\% = 100\%$$

## ProbC: Design a pattern generator

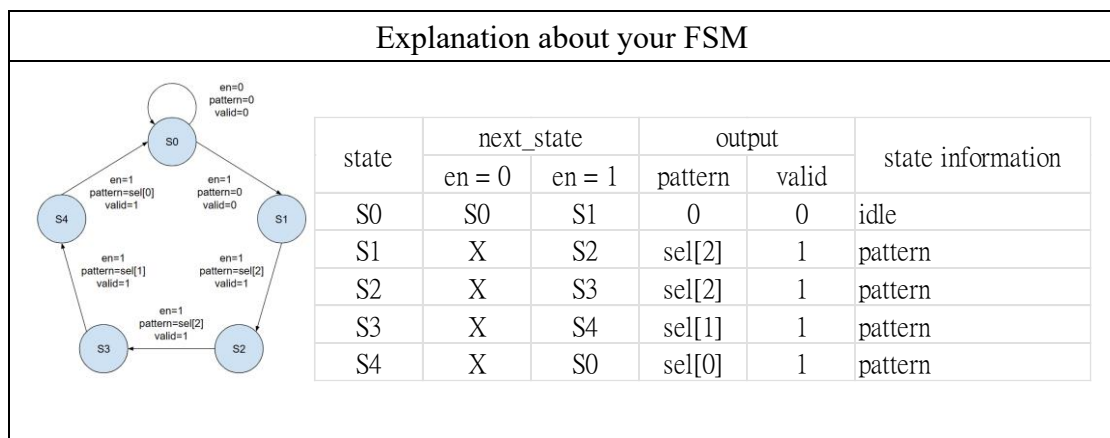
- 1) Design a pattern generator which can create the following pattern and use **mealy machine**. The following is pattern generator specification.



sel [2:0]	pattern
000	0000
001	0001
010	0010
011	0011
100	1100
101	1101
110	1110
111	1111

Signal	Bits	Type	Description
clk	1	input	clock
rst	1	input	reset, active high
en	1	input	When en is high, the system will start to make pattern. The pattern will be created <b>once</b> . If the host want to create the next pattern, it should pull down the en to 0, then restart en.
sel	3	input	According different sel signal to make different pattern
pattern	1	output	Pattern output
valid	1	output	When valid is 1, pattern's value is valid.

## 2) Please describe your FSM in detail.



## 3) After synthesizing your design, you may have some information about the circuit. Please fill in the following form.

Timing (slack)	Area (total cell area)	Power (total)
0.91	6.428160	9.2515e-03mW

單純使用 Clock Specification 為 1ns，未使用 DC.sdc

#### 4) Please attach your design waveforms.

Your simulation result on the terminal.

```
Applications Places System
Terminal
File Edit View Search Terminal Help

#####//sel is 011#####
pattern is 0, pass!!!
pattern is 0, pass!!!
pattern is 1, pass!!!
pattern is 1, pass!!!

#####//sel is 100#####
pattern is 1, pass!!!
pattern is 1, pass!!!
pattern is 0, pass!!!
pattern is 0, pass!!!

#####//sel is 101#####
pattern is 1, pass!!!
pattern is 1, pass!!!
pattern is 0, pass!!!
pattern is 1, pass!!!

#####//sel is 110#####
pattern is 1, pass!!!
pattern is 1, pass!!!
pattern is 0, pass!!!
pattern is 0, pass!!!

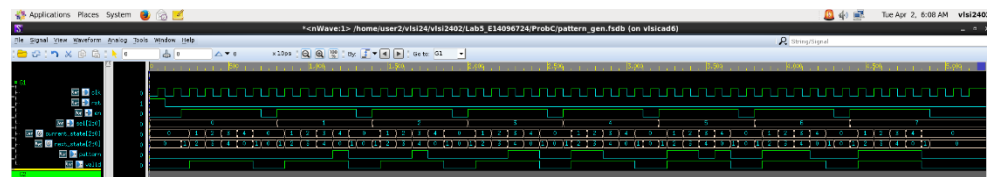
#####//sel is 111#####
pattern is 1, pass!!!
pattern is 1, pass!!!
pattern is 1, pass!!!
pattern is 1, pass!!!

*****
**                                     | |
** Congratulations !!                / 0.0 |
**                                     | 0.0 |
** simulation PASS!!                 | ^ ^ ^ |w|
**                                     | _ _ _ |
*****

$finish called from file 'pattern_gen.tb.v', line 132.
$finish at simulation time
VCS Simulation Report
Time: 52980 ps
CPU time: 0.539 seconds Data structure size: 0.00B
Mon Apr 1 21:49:53 2024
CPU time: 487 seconds to compile + .492 seconds to elab + .310 seconds to link
+ .359 seconds in simulation
vlsicade/home/user2/vlsi24/vlsi2402/Lab5_E14096724/ProbC %
```

Your waveform :

RTL:



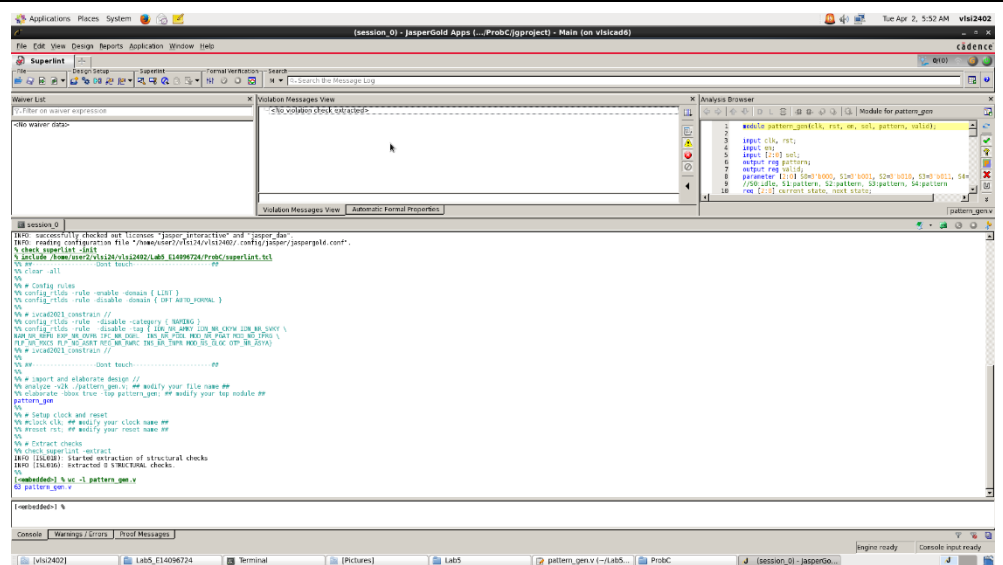
Synthesis



Explanation of your waveform :

波形圖中 error 一直都為 0，代表全部的測資皆正確，波行中還可以看到這是一個 mealy machine，state 隨著輸出改變，而 output 隨著 state 和 input 改變，可以看到 state 和 output 的結果皆與 state diagram 和 state table 相符。

## Superlint Coverage



Coverage :

pattern\_gen.v :  $(1 - 0/63) * 100\% = 100\%$

- 5) At last, please write the lessons learned from this lab session, or some suggestions for this lab session. Thank you.

這次的實驗讓我學習到了如何實作一個 moore machine 和 mealy machine，moore machine 和 mealy machine 在我之前都有在邏輯系統課程中學過，在實作過後我更加了解了 moore machine 和 mealy machine 的輸入和輸出在波型上的差別，還有更加了解了 moore machine 和 mealy machine 在 verilog 中的細節，在用完 superlint 後把每個 warning 都看過並修正，這讓我學習了很多也更熟悉如何寫 verilog。

*Appendix A : Commands we will use to check your homework*

Problem		Command
Lab	Compile	% vcs -R moore.v -full64
	RTL-sim	% vcs -R moore_tb.v -debug_access+all -full64 +define+FSDB
	Gate-sim	% vcs -R moore_tb.v -debug_access+all -full64 +define+FSDB+syn

Problem		Command
ProbA	Compile	% vcs -R det_seq.v -full64
	RTL-sim	% vcs -R det_seq_tb.v -debug_access+all -full64 +define+FSDB
	Gate-sim	% vcs -R det_seq_tb.v -debug_access+all -full64 +define+FSDB+syn
ProbB	Compile	% vcs -R branch_pred.v -full64
	RTL-sim	% vcs -R branch_pred_tb.v -debug_access+all -full64 +define+FSDB
	Gate-sim	% vcs -R branch_pred_tb.v -debug_access+all -full64 +define+FSDB+syn
ProbC	Compile	% vcs -R pattern_gen.v -full64
	RTL-sim	% vcs -R pattern_gen_tb.v -debug_access+all -full64 +define+FSDB
	Gate-sim	% vcs -R pattern_gen_tb.v -debug_access+all -full64 +define+FSDB+syn