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	
簡笠恩	E14102305	
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**.)

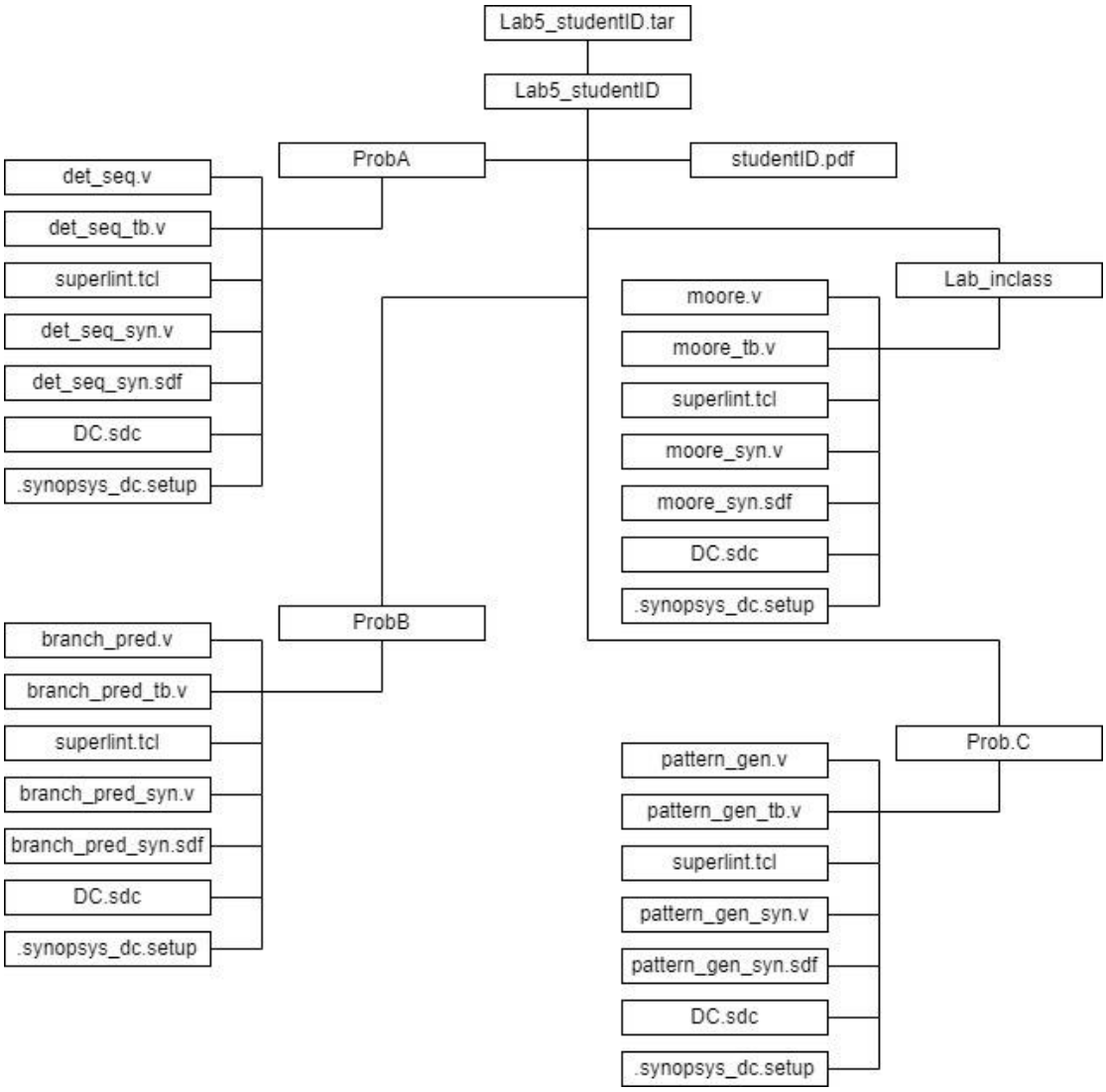
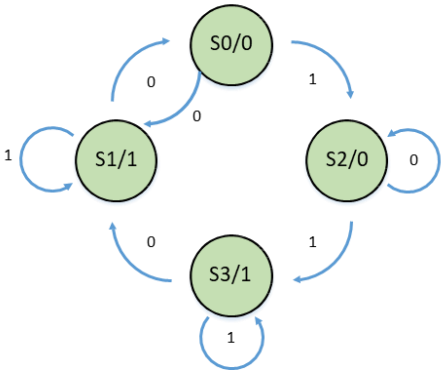


Fig.1 File hierarchy for Homework submission

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
我的狀態機為 4 個狀態:S0、S1、S2、S3，他們的輸出分別為 0、0、1、1，且在 S0 時，input 為 1 會傳至 S2，反之，S1。S1 時，1 時 S1，反之 S0。 S2 時，1 時 S3，反之 S2。 S3 時，1 時 S3，反之 S1。即為一完整狀態機。

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	**4.147**	**6.7439e-03 mW**

4) Please attach your design waveforms.

Your simulation result on the terminal.

RTL

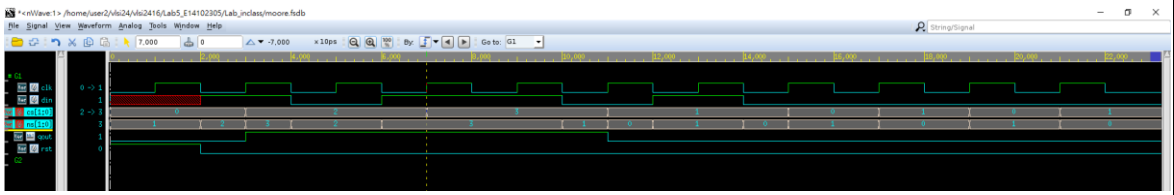
```
        ****************************
        **                        **       |\__||
        **  Congratulations !!    **      / O.O  |
        **                        **    /_____   |
        **  Simulation PASS!!     **   /^ ^ ^ \  |
        **                        **  |^ ^ ^ ^ |w|
        ****************************   \m___m__|_|


$finish called from file "moore_tb.v", line 98.
$finish at simulation time                23000
           V C S   S i m u l a t i o n   R e p o r t
Time: 230000 ps
CPU Time:      0.530 seconds;       Data structure size:   0.0Mb
Fri Mar 29 08:20:17 2024
CPU time: .358 seconds to compile + .444 seconds to elab + .291 seconds to link
+ .564 seconds in simulation
```

SYN

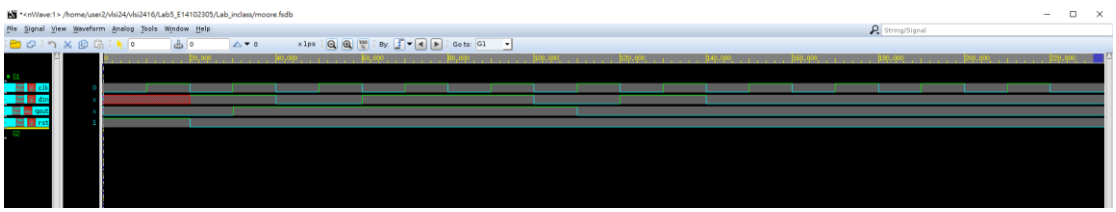
```
        ****************************
        **                        **       |\__||
        **  Congratulations !!    **      / O.O  |
        **                        **    /_____   |
        **  Simulation PASS!!     **   /^ ^ ^ \  |
        **                        **  |^ ^ ^ ^ |w|
        ****************************   \m___m__|_|


$finish called from file "moore_tb.v", line 98.
$finish at simulation time              230000
           V C S   S i m u l a t i o n   R e p o r t
Time: 230000 ps
CPU Time:      0.700 seconds;       Data structure size:   0.5Mb
Fri Mar 29 10:24:45 2024
CPU time: 10.824 seconds to compile + .544 seconds to elab + .694 seconds to lin
k + .738 seconds in simulation
```

Your waveform :

RTL



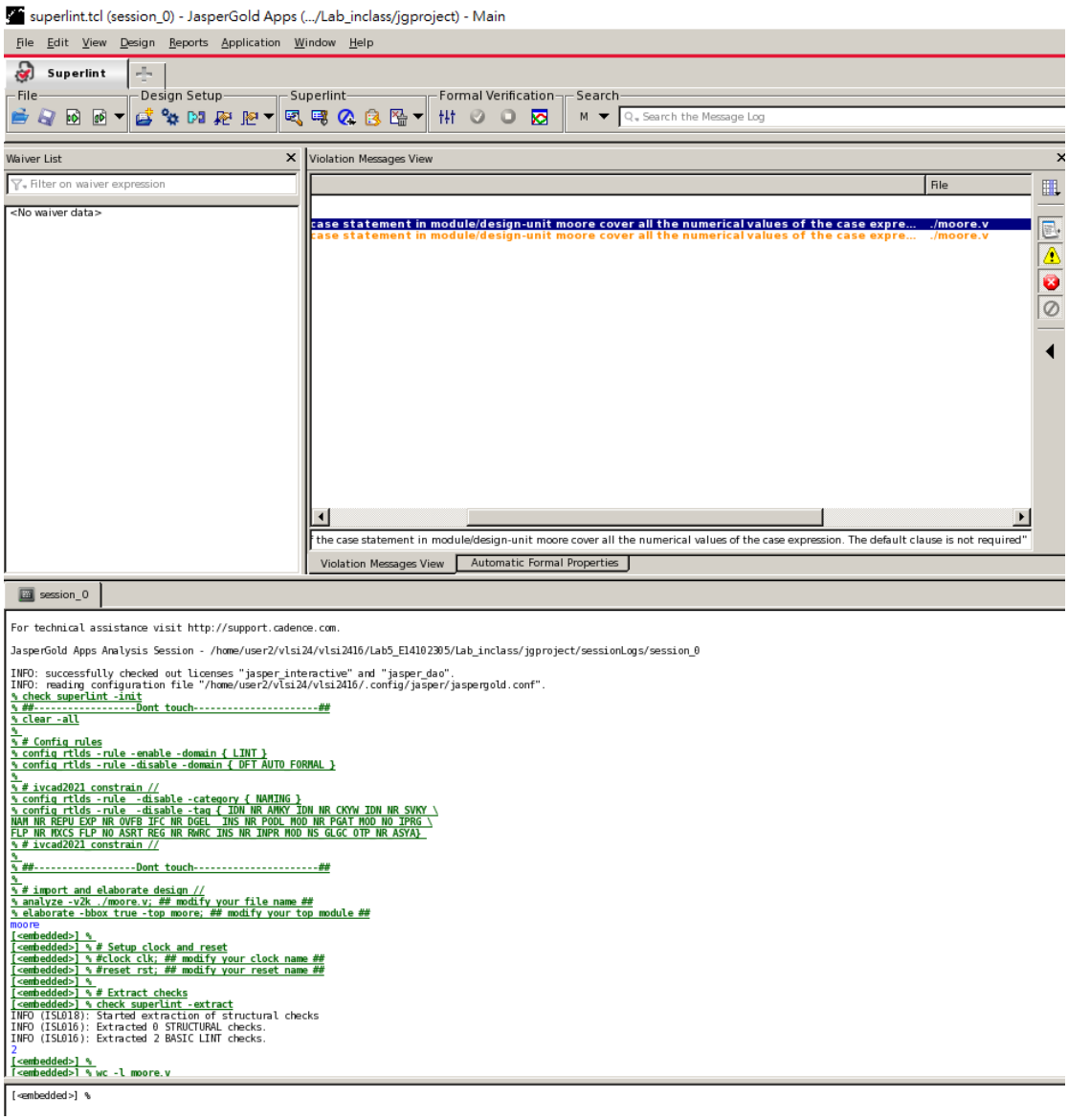
SYN



Explanation of your waveform :

在虛線這段附近，在 din 升起時，我的 ns 從 2 轉至 3，在虛線這個正源觸發時，cs 存入 ns 為 3，而 2、3 皆為輸出 1 故這段都是 1。

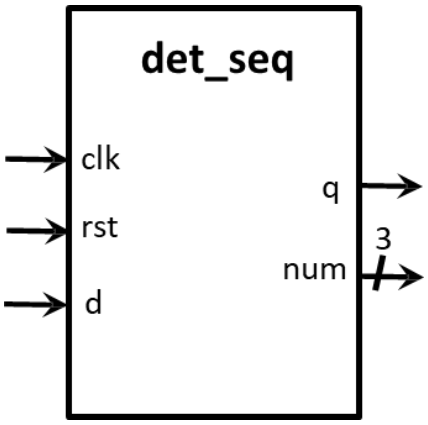
Superlint Coverage



1-2/72=97.222%

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
我的狀態機分成 7 個狀態 S0 為起始數字為 0，S1 為起始數字為 1，S2 為 10，S3 為 101，S4 為 1010，S5 為 10101，S6 為 101011 即為 q 可以升起 num 可以加一，沒意外的話會是從 S0 到 S6，有其他裝況會是以判斷式轉換狀態。

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

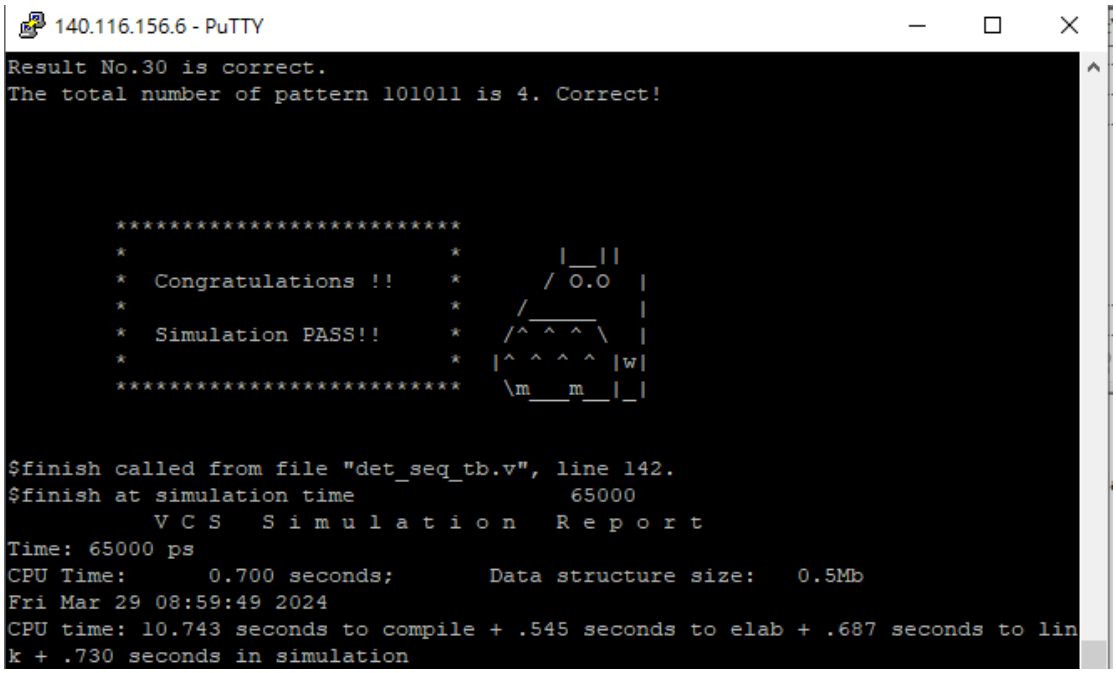
Timing (slack)	Area (total cell area)	Power (total)
0.32	**11.508**	**1.3884e-02 mW**

4) Please attach your design waveforms.

Your simulation result on the terminal.

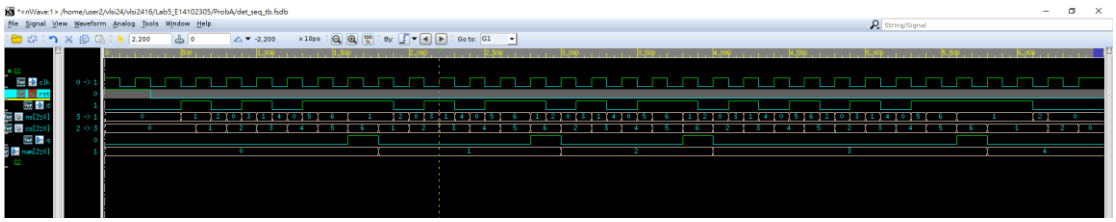
RTL

syn

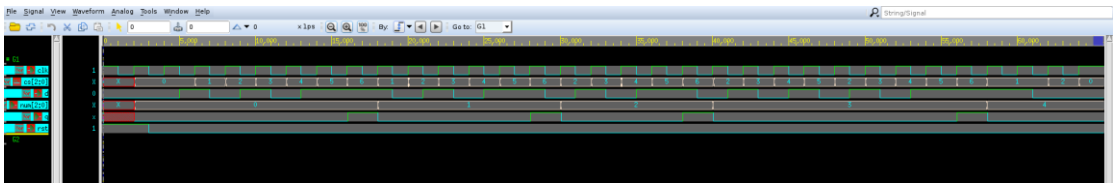


Your waveform :

RTL



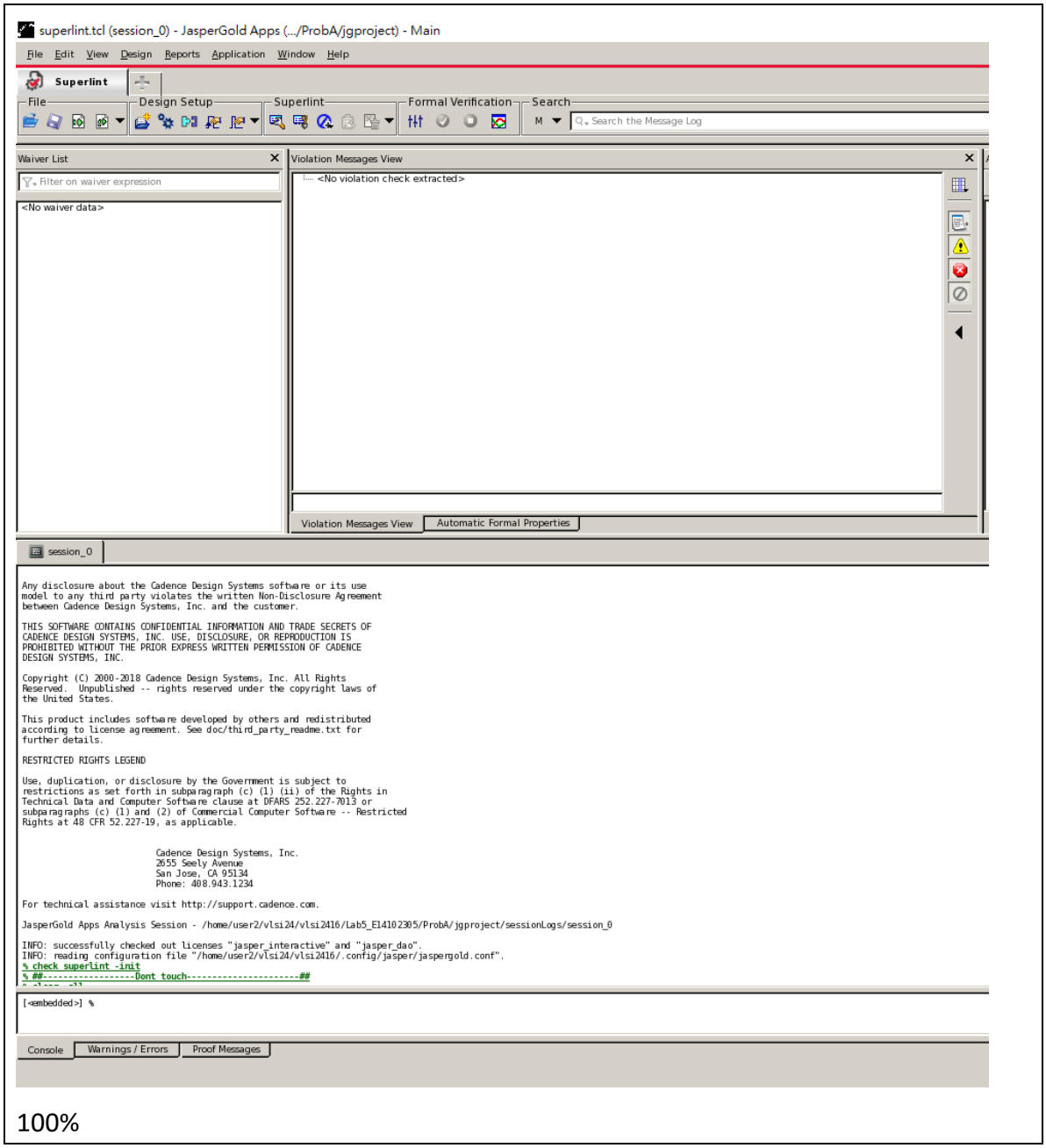
SYN



Explanation of your waveform :

我們看 1800~2800 這段，我們開始判斷第 1 位，再來下幾個 clk 升起時恰恰對到我們的答案 101011 所以在讀取到最後一個 1 時，我們會輸出 1 且在下一個 clk 時，num 會被加入 1。

Superlint Coverage

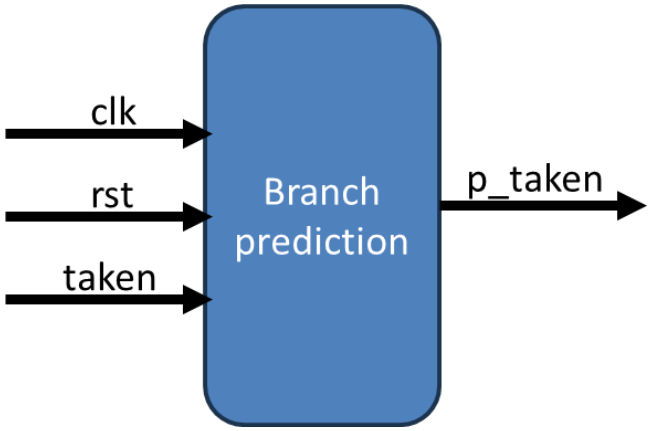


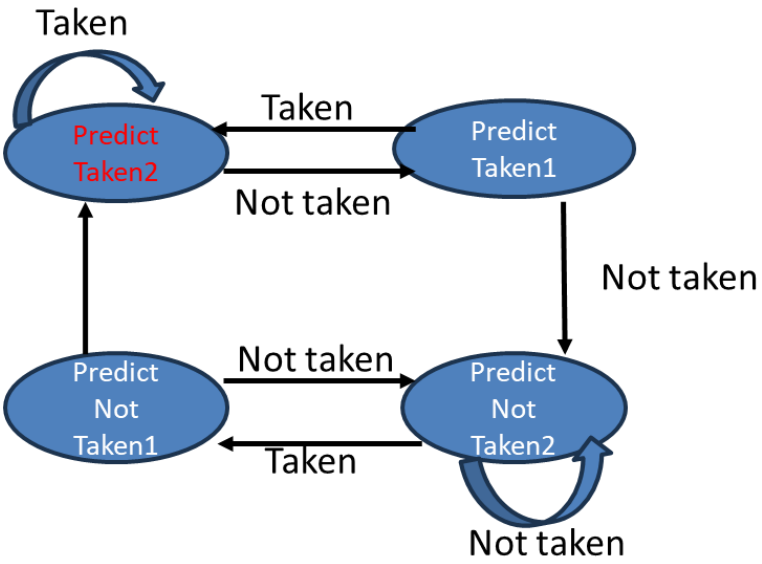
100%

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
我令了 4 個參數分別代表 s0:taken1、s1:taken2、s2:nottaken1、s3:nottaken2，而他們會隨著 taken 的改變去變動，output 的部分則是在 s0、s1 時會將 p_taken 拉起，其他就是維持 0。

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	6.791	8.5827e-03 mW

4) Please attach your design waveforms.

Your simulation result on the terminal.

RTL

```
        ****************************
        **                        **       |\__||
        **  Congratulations !!    **      / O.O  |
        **                        **    /_____   |
        **  Simulation PASS!!     **   /^ ^ ^ \  |
        **                        **  |^ ^ ^ ^ |w|
        ****************************   \m___m__|_|


$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:      0.520 seconds;       Data structure size:   0.0Mb
Fri Mar 29 09:05:40 2024
CPU time: .367 seconds to compile + .435 seconds to elab + .286 seconds to link
+ .554 seconds in simulation
```

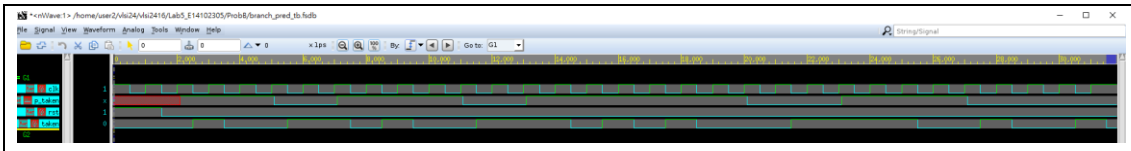
SYN

```
        ****************************
        **                        **       |\__||
        **  Congratulations !!    **      / O.O  |
        **                        **    /_____   |
        **  Simulation PASS!!     **   /^ ^ ^ \  |
        **                        **  |^ ^ ^ ^ |w|
        ****************************   \m___m__|_|


$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:      0.720 seconds;       Data structure size:   0.5Mb
Fri Mar 29 13:54:37 2024
CPU time: 11.173 seconds to compile + .550 seconds to elab + .766 seconds to lin
k + .753 seconds in simulation
```

Your waveform :

RTL

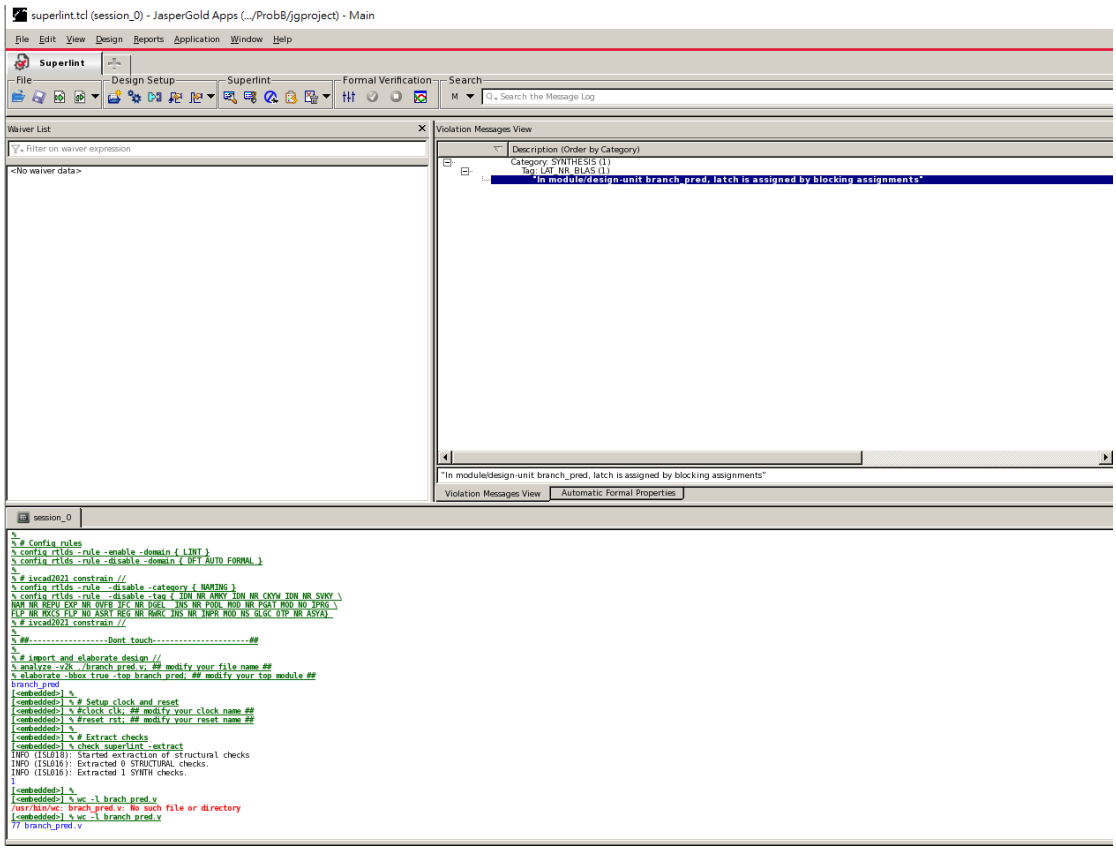
SYN



Explanation of your waveform :

我們拉 6000 時這個正源觸發來說，在 clk 升起，因為我們在 notaken2，且 taken==1，所以依照 state diagram 我們可以知道他下一個狀態是 nottaken1，所以 p_taken 不會升起。

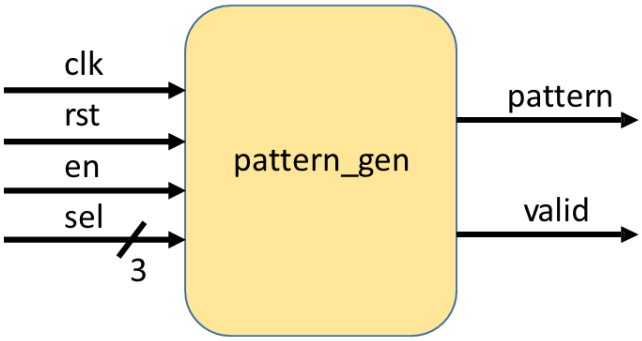
Superlint Coverage



1-1/77=98.7%

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 once. 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.

Explanation about your FSM

在這我用了 5 個狀態
s0:不存數字，s1:存第一個，s2:存第二個……
在 en 升起時會進 s1 模式循序下去到 en 降下，而在 s1~s4 時會將 valid 升起。
輸出則以 sel 輸入
S0:sel[2]
S1:sel[2]
S2:sel[1]
S3:sel[0]

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)
9.91	**6.324**	**9.3976e-04 mW**

4) Please attach your design waveforms.

Your simulation result on the terminal.

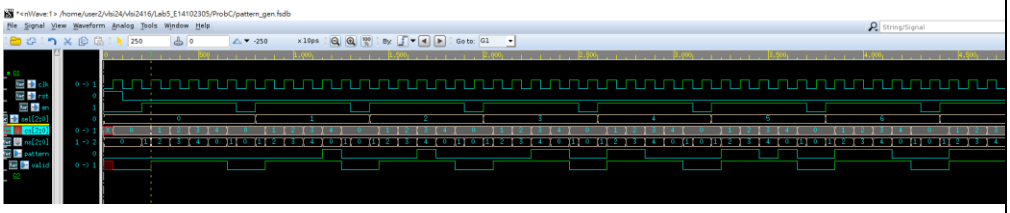
RTL

SYN

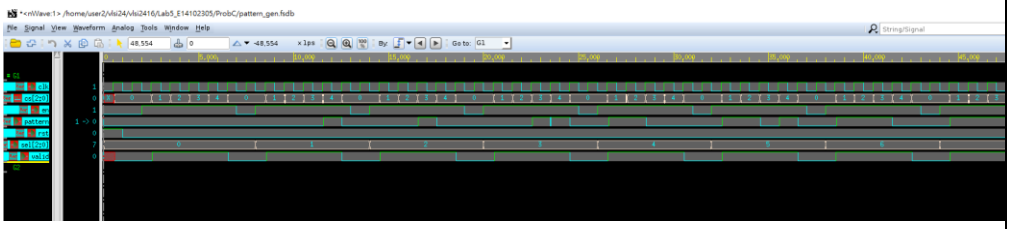
```
        ****************************
        **                        **       |__||
        **  Congratulations !!    **      / O.O  |
        **                        **    /_____   |
        **  Simulation PASS!!     **   /^ ^ ^ \  |
        **                        **  |^ ^ ^ ^ |w|
        ****************************   \m__m__|_|


$finish called from file "pattern_gen_tb.v", line 332.
$finish at simulation time                52000
           V C S   S i m u l a t i o n   R e p o r t
Time: 52000 ps
CPU Time:      0.690 seconds;       Data structure size:   0.5Mb
Fri Mar 29 10:10:25 2024
CPU time: 11.408 seconds to compile + .553 seconds to elab + .711 seconds
k + .730 seconds in simulation
```

Your waveform :

RTL



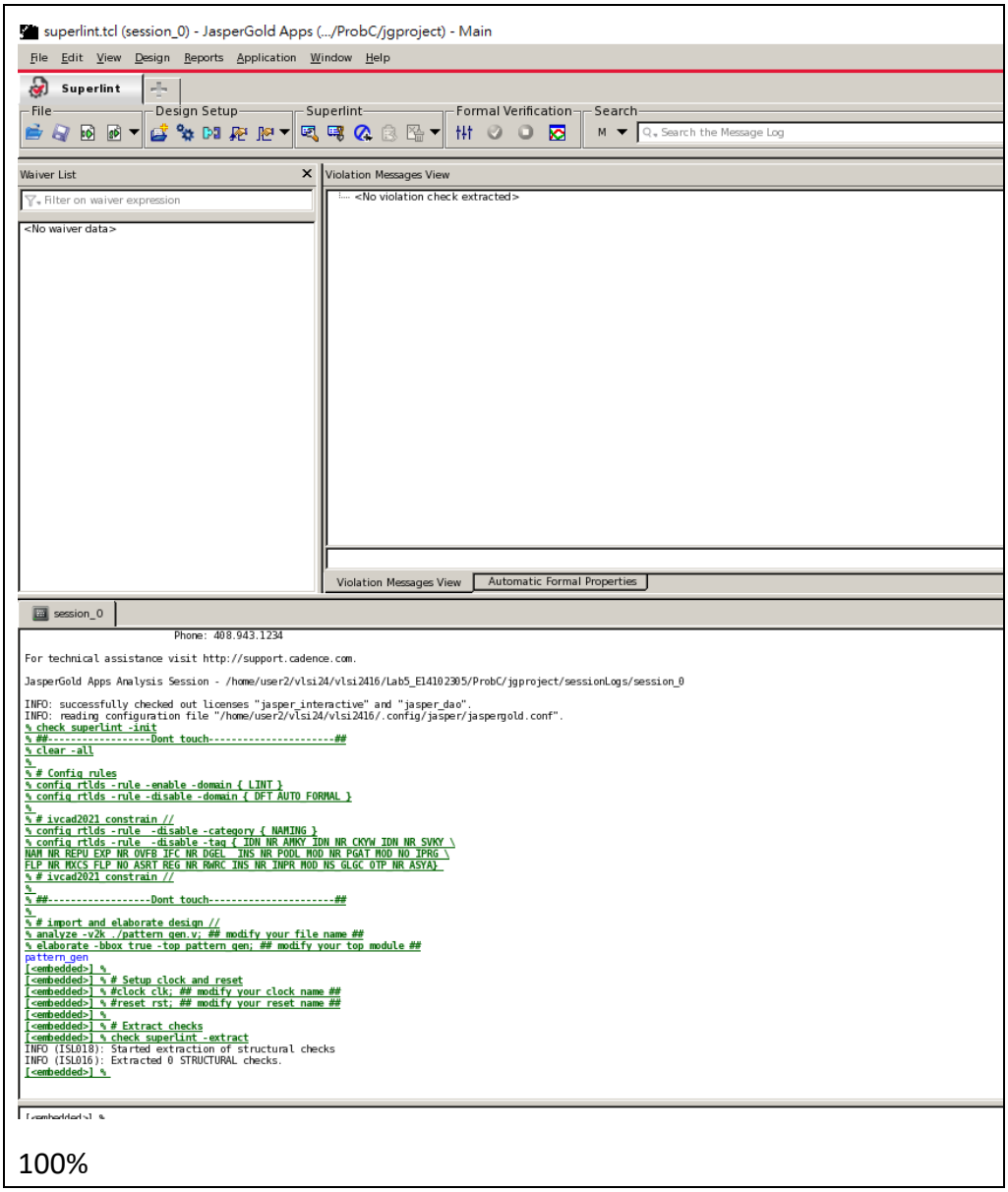
SYN



Explanation of your waveform :

我們看當 en 升起時 ns 會進入 1 即開始輸出模式，cs 進 1 吐第一個數值，2 吐第二個，以此類推所以當 sel = 0 時 會依序輸出 0 0 0 0，sel = 4(100)時會輸出 1 1 0 0，其他也是如此。

Superlint Coverage



100%

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

這次 **lab** 學到了關於 **sequential** 電路的合成，還有 **DC** 檔的使用，有別於之前繁瑣的步驟，這次的步驟略為簡單，也學到了 **state machine** 的用法，可喜可賀**!!!**

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