National Cheng Kung University

Department of Electrical Engineering

Introduction to VLSI CAD (Spring 2022)

Lab Session 5

Synthesis of Sequential Logic and Some Tips

Yu-Chi Chu

Objectives:

To make you be familiar with some designs of sequential logic and Design Complier. **You can follow this document to practice. Please show your best.**

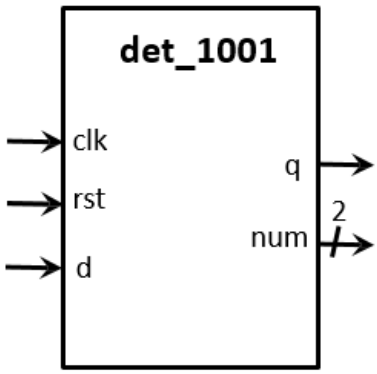
Note that you can extend the spacing if it is not enough for you to answer.

All labs should be synthesized, and clock period should not over 10 ns.

Lab5_1: Design a circuit "detecting pattern 1001"

---

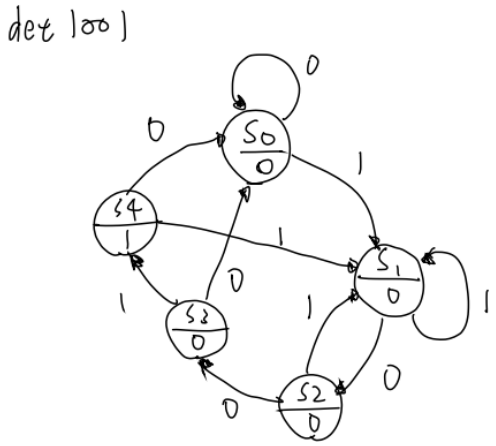
1) **Design a pattern 1001-detecting circuit that can be synthesized with moore machine. The following is det_1001 module's specification. (Do NOT add or delete I/O ports, but you can change their behavior.)**



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 1001, q pulls to high at next clock posedge. Otherwise, q is low.
num	output	2	Count the number of pattern 1001
st	reg	3	Current state register, st_ns is next state

2) Please describe your FSM in detail

Explanation about your FSM



FSM 1: Sequence Detector

- **Initial State (s1):** When the input is 1, the FSM enters state s1. The current state is represented as 1.
- **Transition to s2:** When the input is then 0, the FSM transitions to state s2. The state becomes 10, and the output is 0.
- **Transition to s3:** Another input of 0 causes a transition to state s3. The state is 100, and the output remains 0.
- **Transition to s4 and Output:** Finally, an input of 1 leads to state s4. The state is 1001, and the output becomes 1.
- **Loop in s1:** An input of 1 in s1 keeps the FSM in s1. The state becomes 11, which is equivalent to 1, effectively looping back to s1.
- **Return to s0 from s3:** An input of 0 in s3 causes the FSM to return to the initial state s0. The state becomes 1000, which is treated as starting from 0.

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

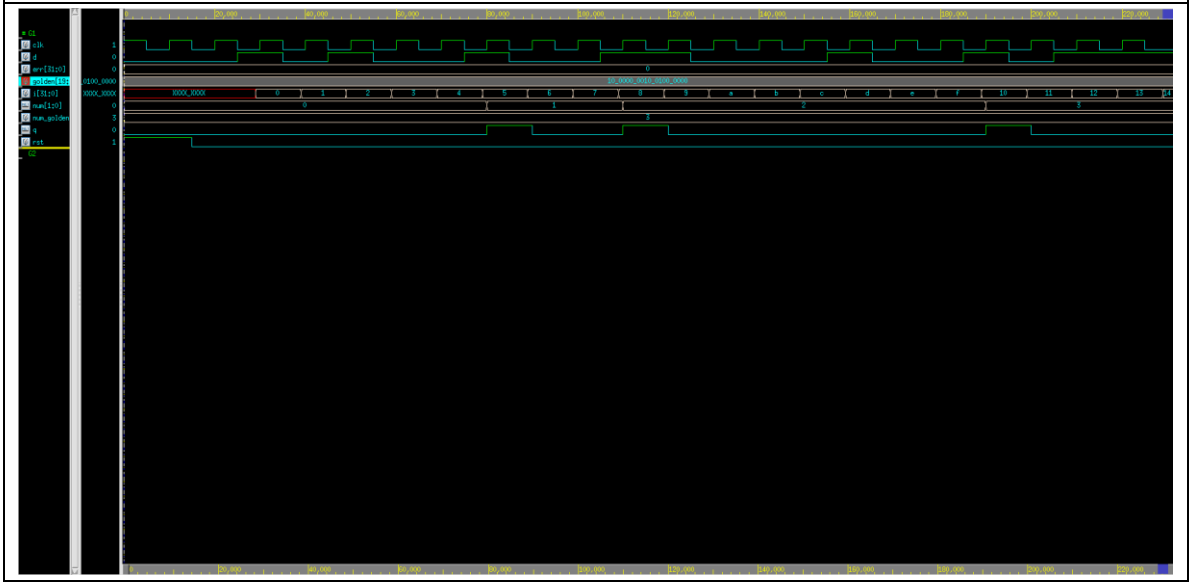
Timing (slack)	**Area (total cell area)**	**Power (total)**
3.53Met	**9.9900um^2**	**0.5682uW**

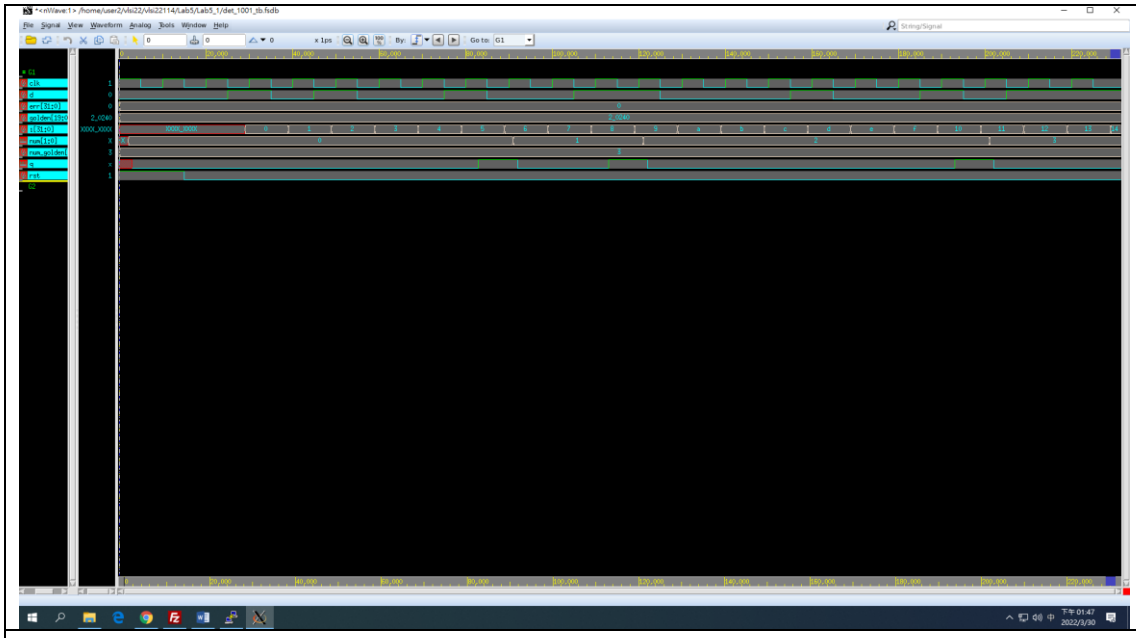
4) Please attach your design waveforms.

Your simulation result on the terminal.

```
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.
The totoal number of pattern 1001 is 3. Correct!

     ************************************************
     *                                              *
     *  Congrats! All results are correct. (^o^)b   *
     *                                              *
     ************************************************

Simulation complete via $finish(1) at time 229 NS + 0
./det_1001_tb.v:123     $finish;
```

Your waveform :





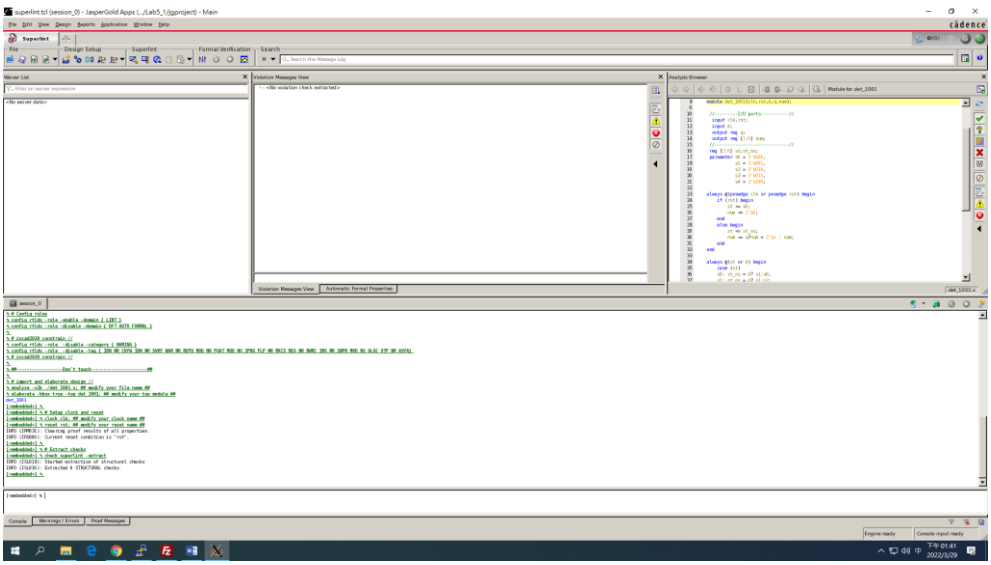
Explanation of your waveform :

Reset (rst = 0): The reset signal resets the state to s0.

Input Sequence and Output: Starting from the third clock cycle, the FSM evaluates the input. From the fifth to eighth clock cycles, the input is 1001, resulting in an output of 1 on the eighth clock cycle. Similarly, the input from the eighth to eleventh clock cycles is also 1001, producing an output of 1 on the eleventh clock cycle.

Overlapping Detection: This FSM can detect overlapping sequences. The input 1001001 is recognized as two occurrences of 1001, demonstrating that the FSM doesn't need a complete reset between sequence detections.

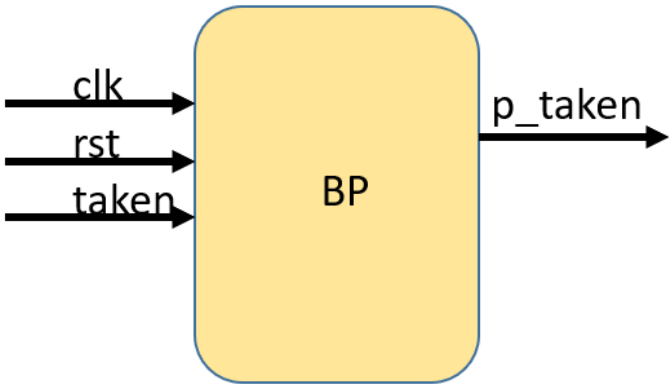
Superlint Coverage



Coverage = 1

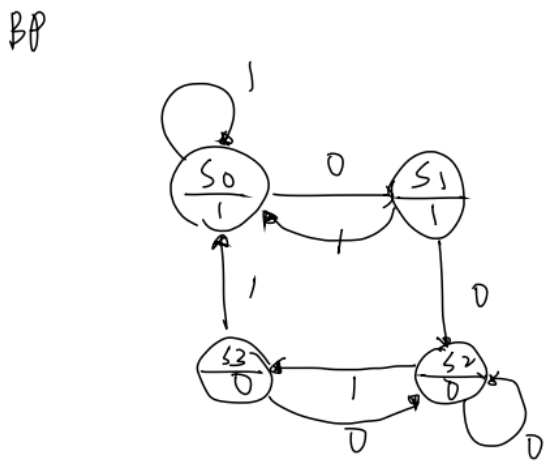
Lab5_2: Design a "2-bit branch prediction" unit

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



Signal	Type	Bits	Description
clk	input	1	clock
rst	input	1	reset, active high
taken	input	1	Actual branch taken signal
p_taken	output	1	Predicted branch taken signal
st	reg	2	Current state register, st_ns is next state

2) Please describe your FSM in detail.

Explanation about your FSM

Initial State (s0): The initial output is 1. **Incorrect Guess (Input 0):** An input of 0 (incorrect guess) transitions the FSM to state s1. The output remains 1. **Second Incorrect Guess (Input 0):** Another input of 0 (two consecutive incorrect guesses) transitions the FSM to state s2. The output changes to 0. **Pattern Reversal:** The subsequent transitions and outputs follow a similar logic, but with the outputs inverted (0s become 1s and 1s become 0s).

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

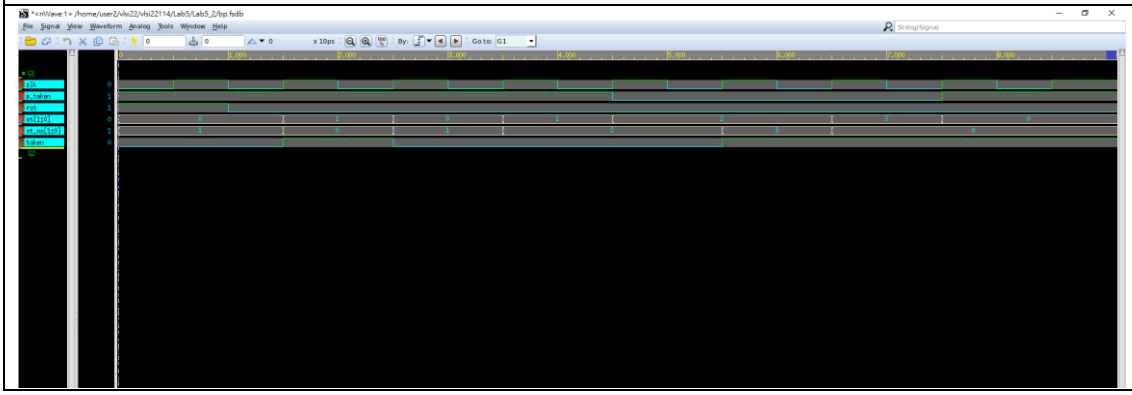
Timing (slack)	Area (total cell area)	Power (total)
3.29	**3.6408um^2**	**0.2599uW**

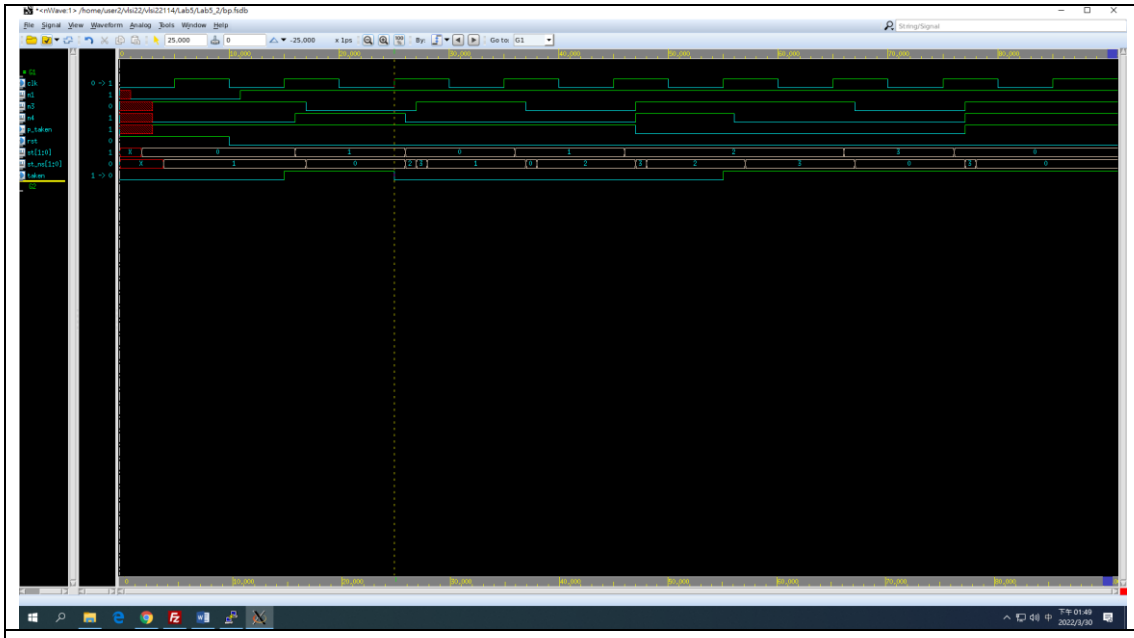
4) Please attach your design waveforms.

Your simulation result on the terminal.

```
Behavior of BP is correct!!, value of p_taken is 1, value of taken is 1
Behavior of BP is correct!!, value of p_taken is 1, value of taken is 0
Behavior of BP is correct!!, value of p_taken is 1, value of taken is 0
Behavior of BP is correct!!, value of p_taken is 0, value of taken is 0
Behavior of BP is correct!!, value of p_taken is 0, value of taken is 1
Behavior of BP is correct!!, value of p_taken is 0, value of taken is 1
Behavior of BP is correct!!, value of p_taken is 1, value of taken is 1



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


Simulation complete via $finish(1) at time 90 NS + 0
./BP tb.v:76            $finish;
```

Your waveform :





Explanation of your waveform :

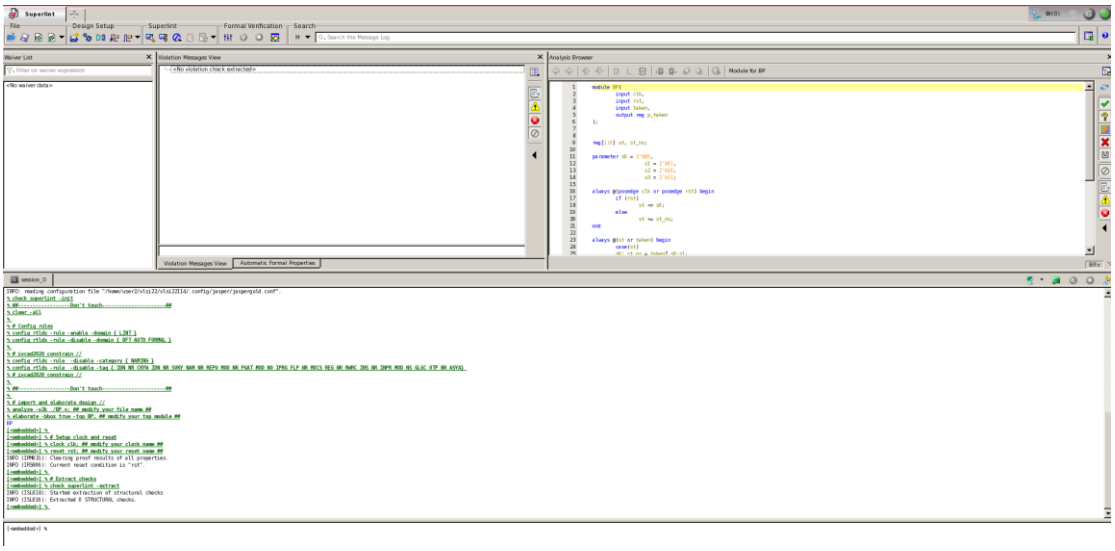
Reset (rst = 1): The reset signal initializes the state (st) to 0. The output p_taken is initialized to 1.

Evaluation (clk 2 onwards): Starting from the second clock cycle, the FSM evaluates the input.

Output Change: On the third and fourth clock cycles, taken is 0, while p_taken is 1. Since these are different for two consecutive cycles, p_taken becomes 0 on the fifth clock cycle.

Another Output Change: On the sixth and seventh clock cycles, taken is 1, and p_taken is 0. Again, two consecutive different values cause p_taken to become 1 on the eighth clock cycle.

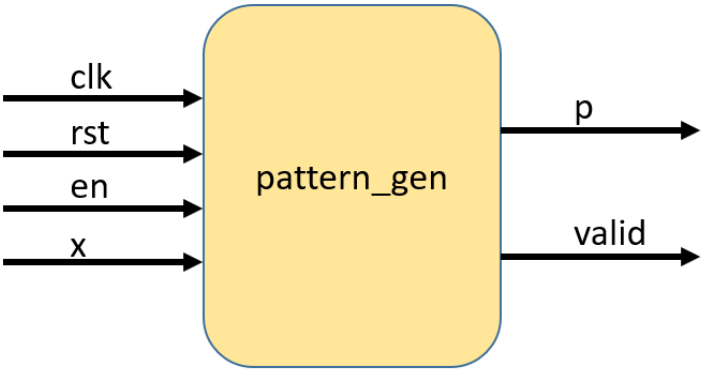
Superlint Coverage



Coverage = 1

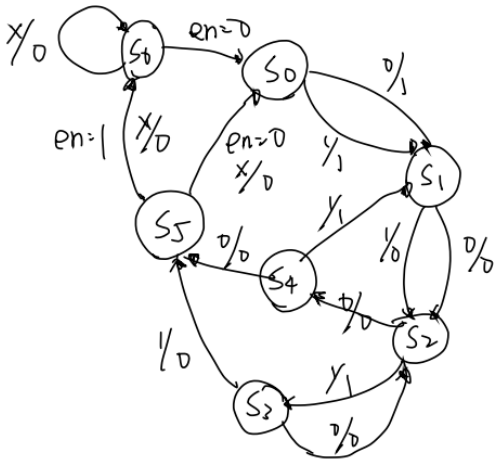
Prob 5_3: Design a pattern generator

1) Design a pattern generator which can create pattern 1010 and 1000, and use mealy machine. The following is pattern generator specification.



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 1000 or 1010 . 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.
x	1	input	If x is 1, system will make pattern 1010; if x is 0, system will make pattern 1000
p	1	output	Pattern output
valid	1	output	When valid is 1, p's value is valid.
st	3	reg	Current state register, st_ns is next state

2) Please describe your FSM in detail.

Explanation about your FSM
 **Initial State (s0):** The FSM starts in state s0. **Transition to s1:** Regardless of the input, the FSM transitions to state s1. The state is 1. **Transition to s2:** Regardless of the input, the FSM transitions to state s2. The state becomes 10. **Branching and Convergence:** From s2, the path depends on whether the third bit of the input pattern is 0 or 1. These two paths eventually converge at state s5. This indicates the completion of one pattern detection. **Loop or Reset:** If the enable signal (en) is not reset at this point, the FSM loops back to state s6 indefinitely. If en is 0, the FSM returns to s0 to start the process again.

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

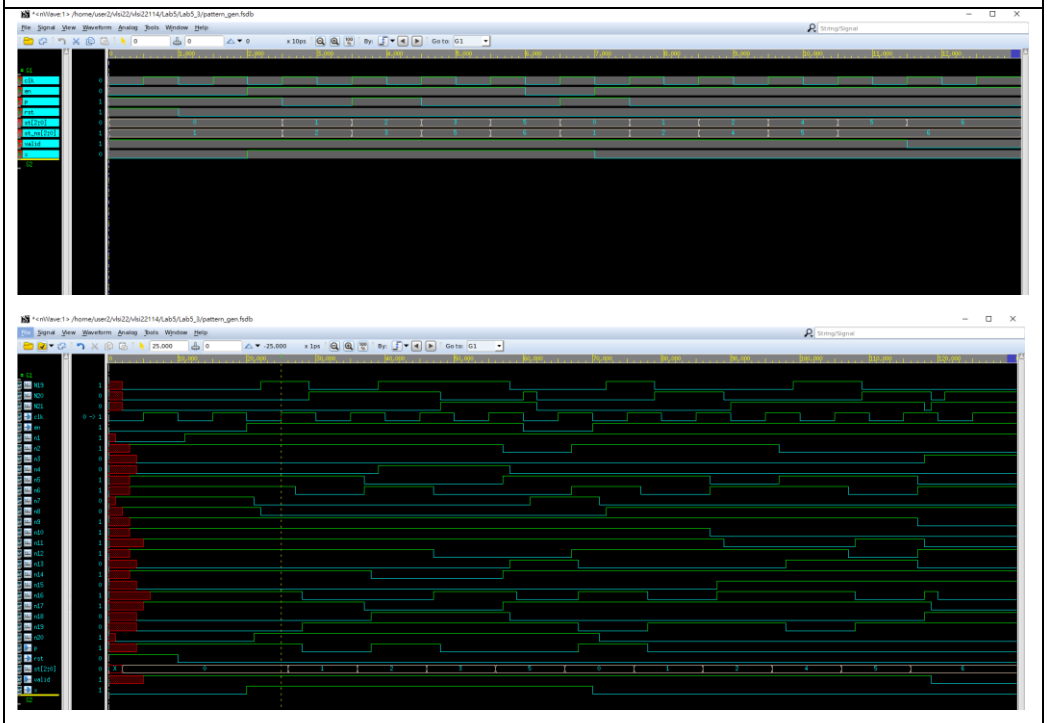
Timing (slack)	Area (total cell area)	Power (total)
3.67	**9.2352**	**0.4581**

4) Please attach your design waveforms.

Your simulation result on the terminal.

```
//////////////////x is 1//////////////////
p is 1, pass!!!
p is 0, pass!!!
p is 1, pass!!!
p is 0, pass!!!
//////////////////x is 0//////////////////
p is 1, pass!!!
p is 0, pass!!!
p is 0, pass!!!
p is 0, pass!!!



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


Simulation complete via $finish(1) at time 130 NS + 0
```

Your waveform :



Explanation of your waveform :

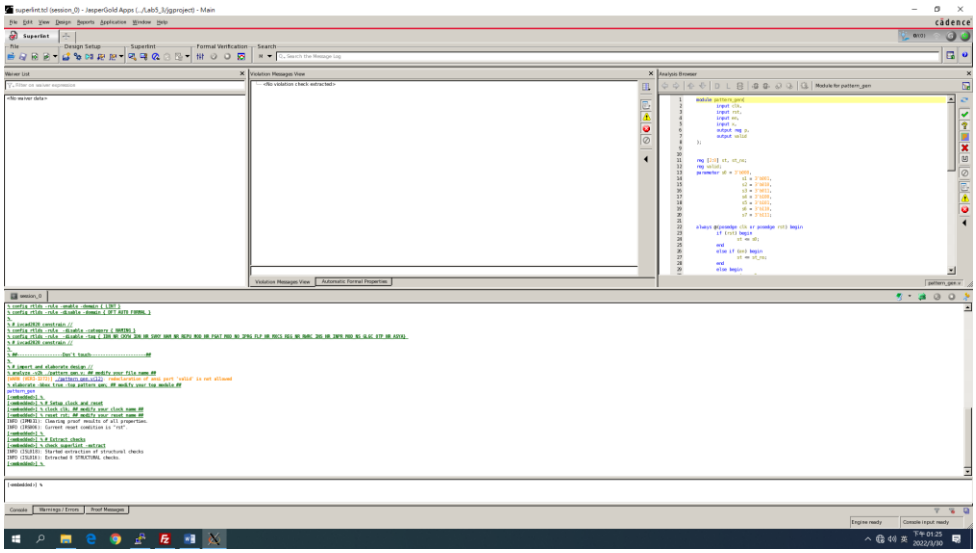
Initial State (rst = 1, en = 0): During the first and second clock cycles, rst is 1 and en is 0. The output p is 0.

Pattern Output (en = 1, x = 1): On the third clock cycle, en is 1 and x is 1. This initiates the output of the pattern 1010. The output 1010 is generated from the fourth to seventh clock cycles.

Another Pattern Output (x changes, en resets): On the seventh clock cycle, x changes, and en is reset. This triggers the output of another pattern, 1000, which is generated and the FSM enters state s2.

Completion and Valid Signal: Because en was reset, the valid signal becomes 0 on the twelfth clock cycle, indicating the completion of the pattern output.

Superlint Coverage



Coverage = 1

Appendix A : Commands we will use to check your homework

Problem	syn	Command
Lab5_1	pre	% ncverilog det_1001_tb.v +access+r +define+FSDB
	post	% ncverilog det_1001_tb.v +access+r +define+FSDB+syn
Lab5_2	pre	% ncverilog BP_tb.v +access+r +define+FSDB
	post	% ncverilog BP_tb.v +access+r +define+FSDB+syn
Lab5_3	pre	% ncverilog tb.v +access+r +define+FSDB
	post	% ncverilog tb.v +access+r +define+FSDB+syn