Course: CSC258F

Professor: Steve Engels

Experimenter: Yuhao Yang

Student ID: 1005808057

Lab6

Pre-Lab Report

Part1:

1. Answer following questions:
--- Given the starter circuit, is the Reset signal a synchronous or asynchronous reset?

It is an asynchronous reset in stater circuit.

--- Is it active high, or active low signal?

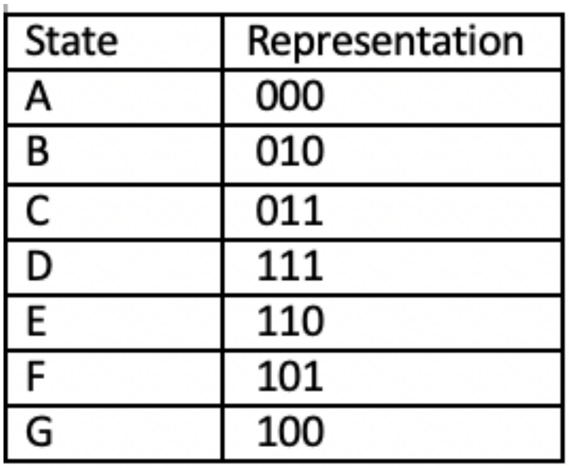
It is active high.

--- How should the Reset signal feature in the tests that you run on your FSM?

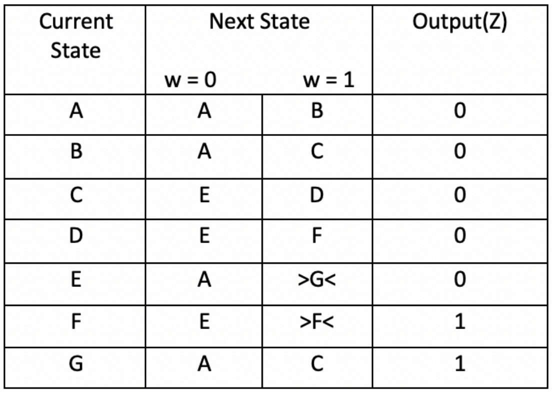
In the test, if I activate reset (to 1) and then the FSM should go back to state A.

2. Assign flip-flop values to each of the states create a state table that illustrates the state transitions in response to the input signal w.

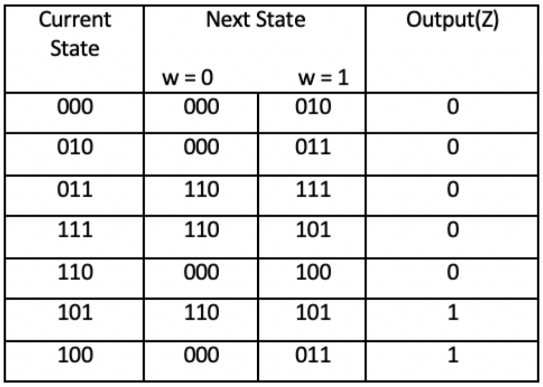
Since there are 7 states totally which are A, B, C, D, E, F and G, then we only need celling(log$_2$7) = 3 flip flops to cover all states.



State	Representation
A	000
B	010
C	011
D	111
E	110
F	101
G	100

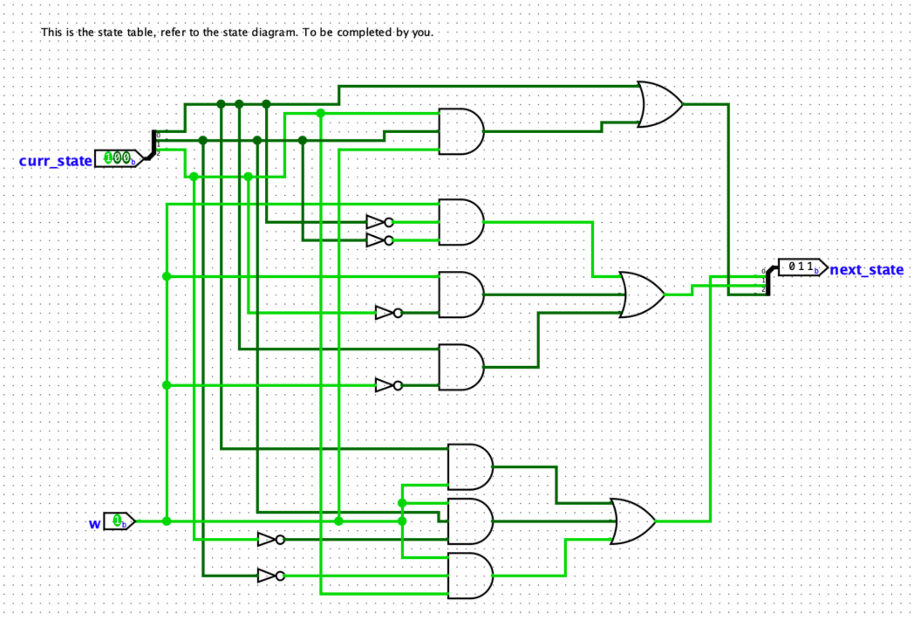


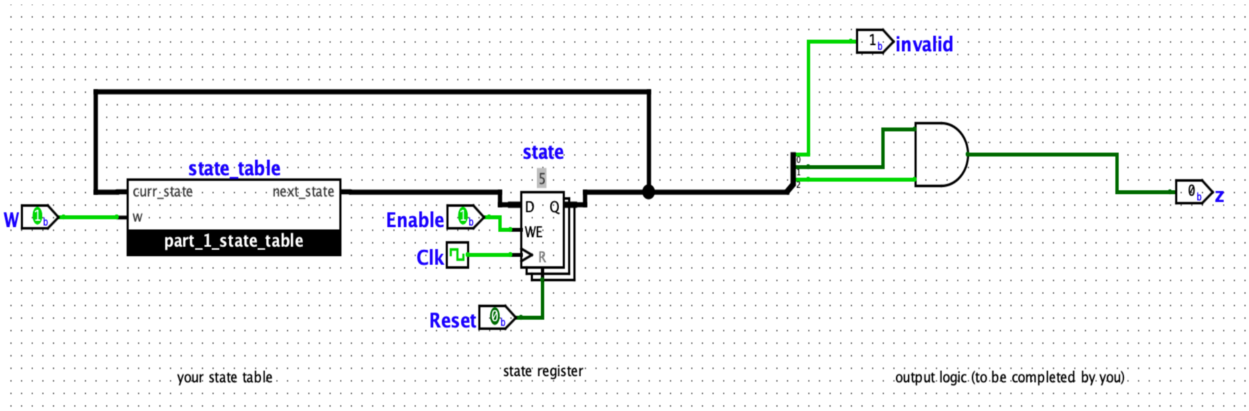
Current State	Next State		Output(Z)
	w = 0	w = 1	
A	A	B	0
B	A	C	0
C	E	D	0
D	E	F	0
E	A	>G<	0
F	E	>F<	1
G	A	C	1



Current State	Next State		Output(Z)
	w = 0	w = 1	
000	000	010	0
010	000	011	0
011	110	111	0
111	110	101	0
110	000	100	0
101	110	101	1
100	000	011	1

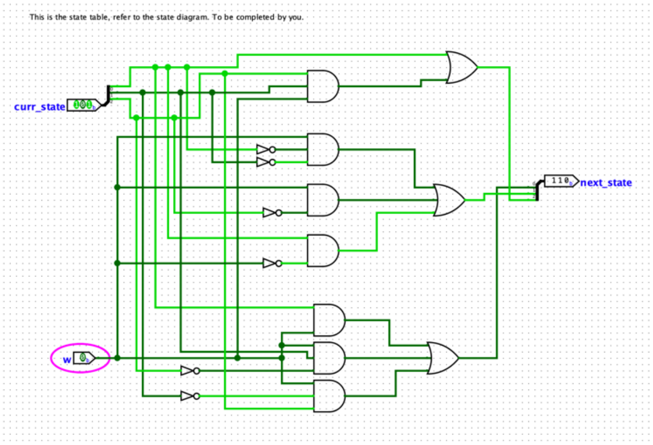
4. Implement the output value circuit for z in part1_FSM.

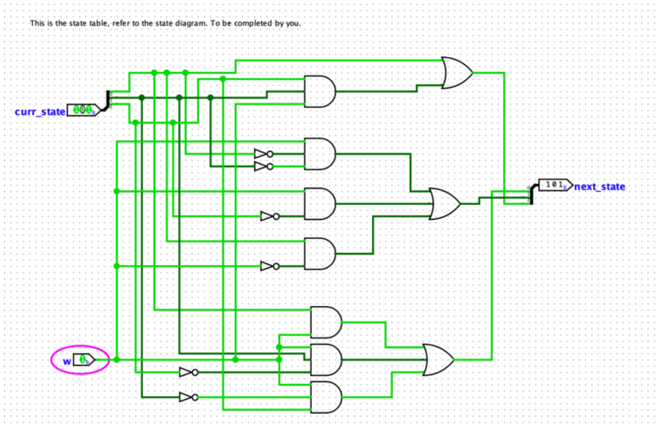

This is the state table, refer to the state diagram. To be completed by you.
curr_state
100b
011b
next_state
w
1b

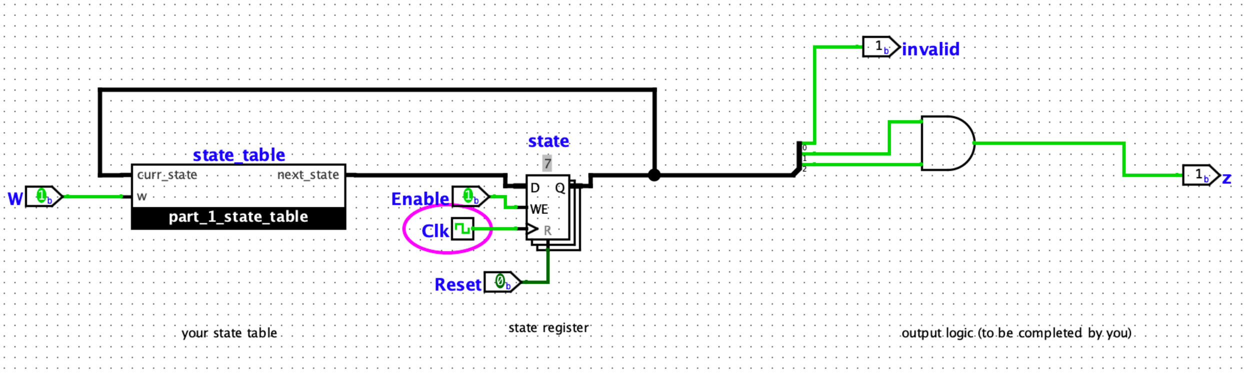


5. Outline the test plan for your circuit in your prelab report and why these test cases verify the correctness of your circuit.

We have to understand how FSM and state table work, I list a way to get output 1, it starts from State A when w = 1 then goes to State B, then when w = 1 goes to State C, when w = 1 goes to State D, then when w = 1 goes to state F and get 1 here. Actually, there are more than one way to get output of 1, but I just provided one way to test the correctness of the circuit. I also showed some screenshots of both FSM and state table.


This is the state table, refer to the state diagram. To be completed by you.
curr_state
000
w
0
110
next_state


This is the state table, refer to the state diagram. To be completed by you.
curr_state
000
w
0
101
next_state



Part2:

1. I fully understand the circuit already.

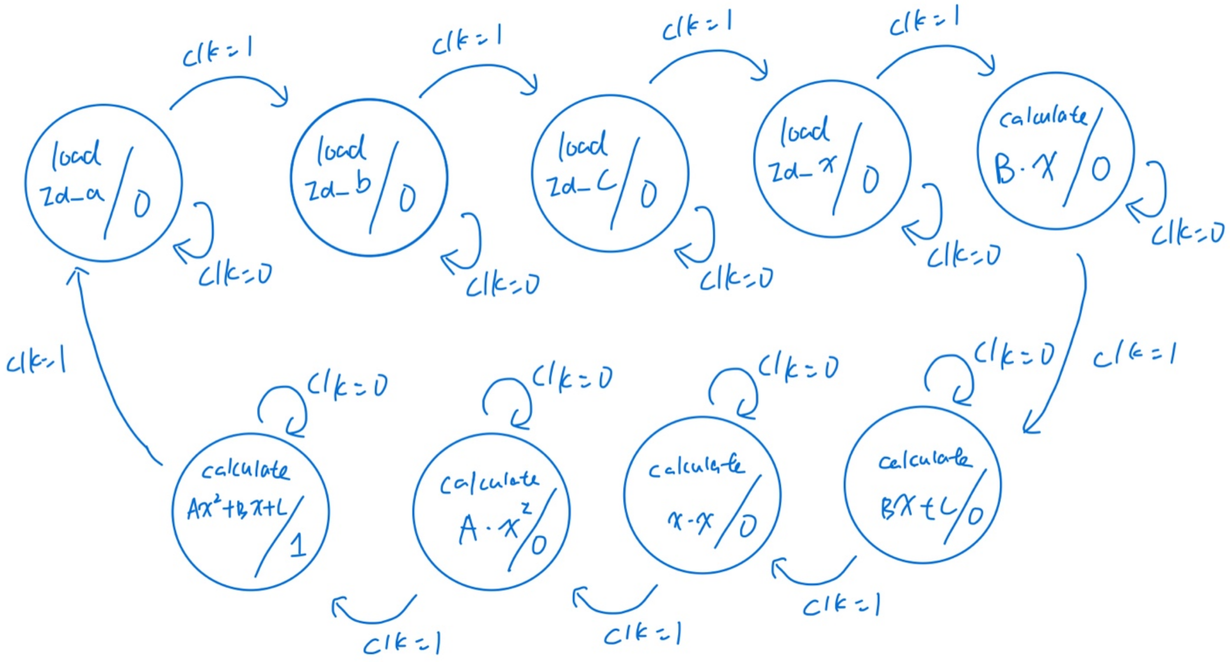
2. Determine a sequence of steps similar to the datapath example shown in lecture to control the datapath to perform the required computation. You should draw a table that shows the contents of the registers and the control signal values for each cycle of your computation. Include this table in your prelab.

To do the calculation of $AX^2 + BX + C$, we need 9 steps in total. The first 4 circles are loading $R_{A,}$ R_B, R_C and R_X prospectively. Then we calculate X multiple B from register B and register X then store the result into register B. Next, we calculate BX add C from register B and register C then store the result into the register A. We next calculate X multiple x from register x and register X then store the result into register B. After that, we calculate C multiple X^2 from register C and

register B then store the result into register B. Last, we calculate CX^2 add BX + A from register B and register A then store into the result into register R and so far, we finish the whole calculation of AX^2 + BX + C.

Cycle	Id _c	Id_x	Id _a	Id_b	Id_r	Id_alu_out	alu_select_a	alu_select_b	alu_op	State
1	0	0	1	0	0					0000
2	0	0	0	1	0					0001
3	1	0	0	0	0					0010
4	0	1	0	0	0					0011
5	0	0	0	1	0	1	01	11	1	0100
6	0	0	1	0	0	1	00	01	0	0101
7	0	0	0	1	0	1	11	11	1	0110
8	0	0	0	1	0	1	10	01	1	0111
9	0	0	0	0	1	0	01	00	0	1000

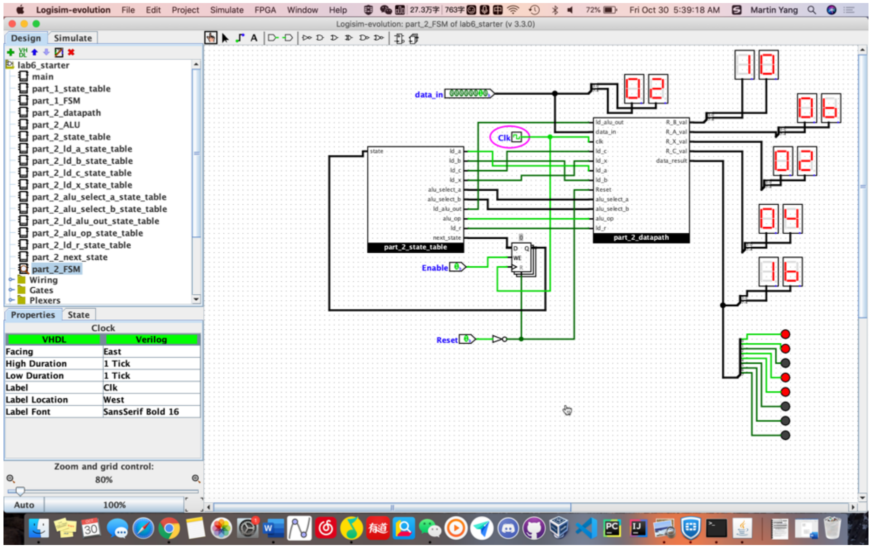
3. Draw a state diagram to correspond the FSM and the table shows.



5. Test the circuit with poke to verify the correctness.

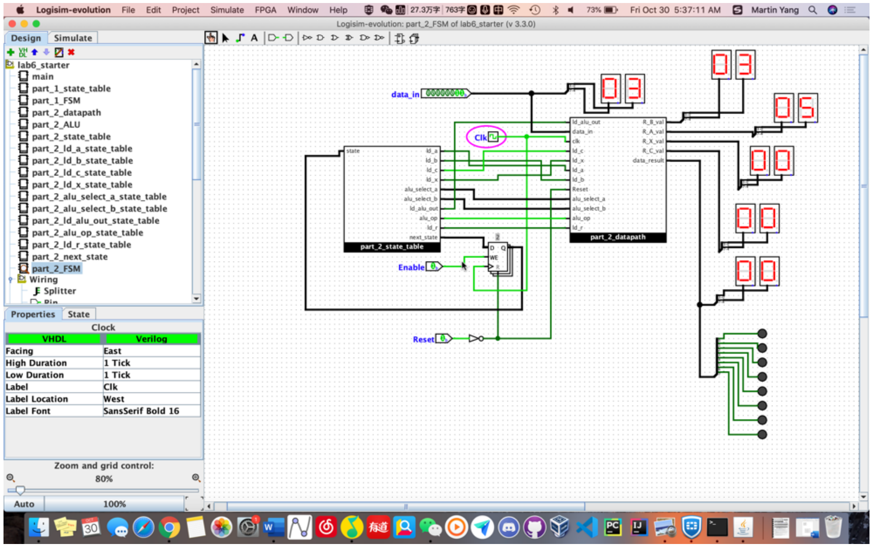
When can have a follow processes to verify the correctness. (There is more than one way to do.)

1. Load A to 5
2. Load B to 3
3. Load C to 4
4. Load X to 2
5. calculate BX, then will get 6 for R_B val
6. calculate BX + A, then will get 11 for R_Aval
7. calculate X^2, then will get 4 for R_B val
8. calculate CX^2, then will get 16 for R_B val
9. calculate CX^2 + BX + A, then will get 27 for data register



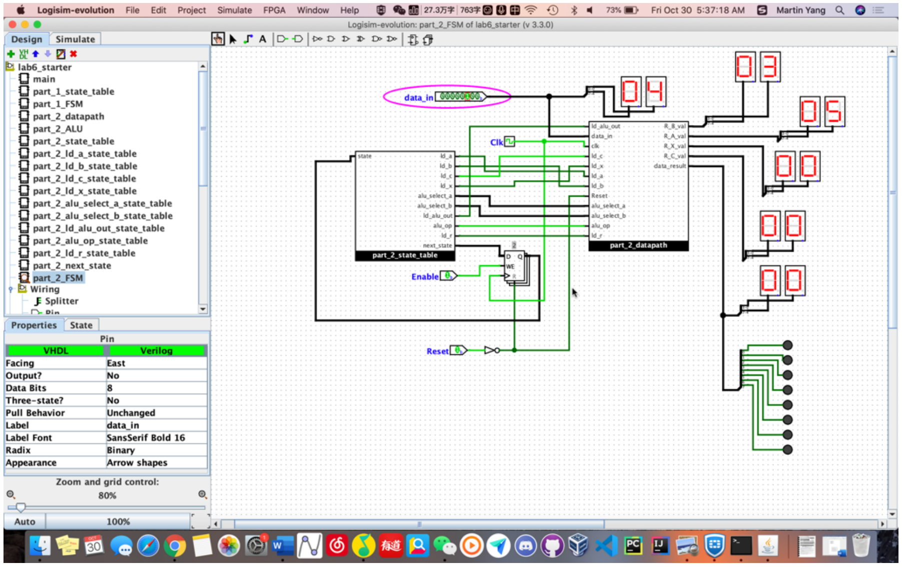
Clock

VHDL	Verilog
Facing	East
High Duration	1 Tick
Low Duration	1 Tick
Label	Clk
Label Location	West
Label Font	SansSerif Bold 16

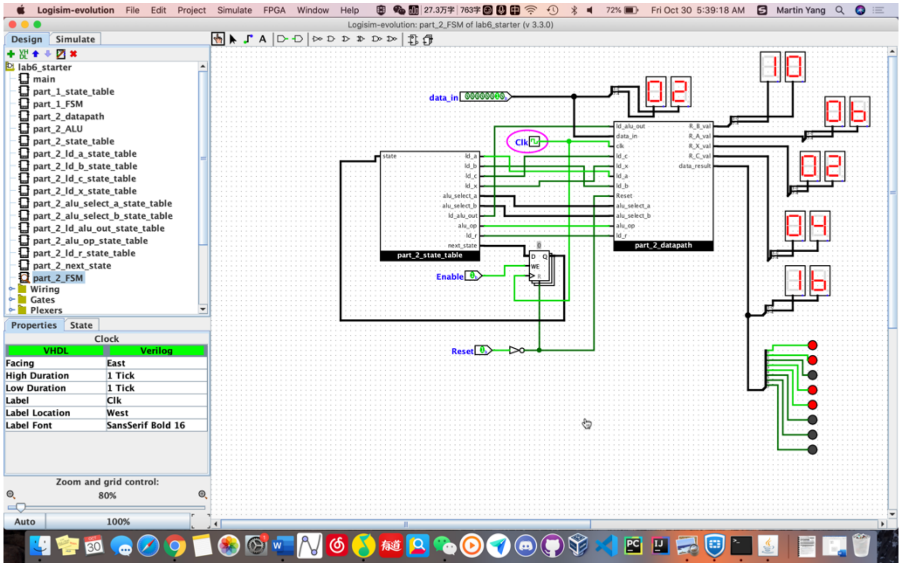


Clock

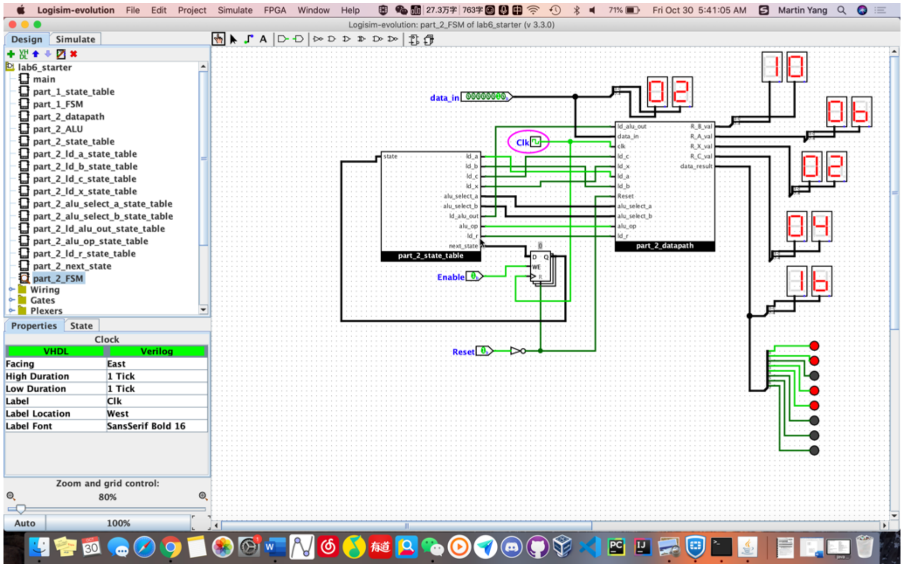
VHDL	Verilog
Facing	East
High Duration	1 Tick
Low Duration	1 Tick
Label	Clk
Label Location	West
Label Font	SansSerif Bold 16



Pin	
VHDL	Verilog
Facing	East
Output?	No
Data Bits	8
Three-state?	No
Pull Behavior	Unchanged
Label	data_in
Label Font	SansSerif Bold 16
Radix	Binary
Appearance	Arrow shapes



Clock	
VHDL	Verilog
Facing	East
High Duration	1 Tick
Low Duration	1 Tick
Label	Clk
Label Location	West
Label Font	SansSerif Bold 16



Clock	
VHDL	Verilog
Facing	East
High Duration	1 Tick
Low Duration	1 Tick
Label	Clk
Label Location	West
Label Font	SansSerif Bold 16