CSC258

Lab 6

Morgan Chang

1005127113

Part I

1.

- Given the starter circuit, is the Reset signal a synchronous or asynchronous reset?

  Asynchronous

- Is it active high, or active low signal?

  Active low, since the value of z is reset to 0 when reset value is high.

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

  The reset signal should be low throughout the tests on the FSM. It should be high only when we want to reset the FSM, then set to low when we want to start another sequence of tests.

2.

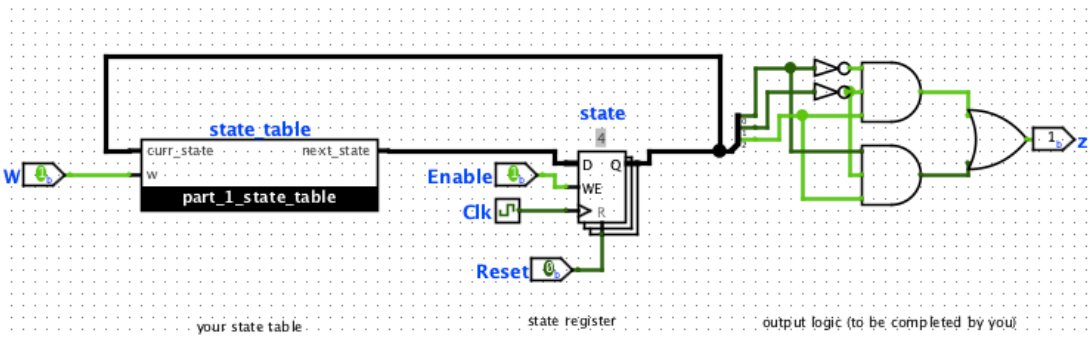
Flip-Flop Assignments

State	Flip Flop Values
A	000
B	001
C	011
D	111
E	110
F	100
G	101

State Table

F_2	F_1	F_0	w	F_2	F_1	F_0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	0
0	0	1	1	0	1	1
0	1	1	0	1	1	0
0	1	1	1	1	1	1
1	1	1	0	1	1	0
1	1	1	1	1	0	0
1	1	0	0	0	0	0
1	1	0	1	1	0	1
1	0	0	0	1	1	0
1	0	0	1	1	0	0
1	0	1	0	0	0	0
1	0	1	1	0	1	1

4. Output logic: $z = 1$ when the output state is at F or G.



5. Each test case is consecutive with the previous one.

All test cases are set with Enable = 1, reset = 0.

w	cycle	z
1	1	0
1	2	0
1	3	0
1	4	1
1	5	1
0	6	0
1	7	1

These test cases verify the correctness of the circuit because it tests if both the sequence 1111 and 1101 works. It also verifies if overlapping sequences are allowed.

Part II

1.

Load data_in value into R_A, R_B, R_C, R_X by setting Id_a, Id_b, Id_c, Id_x on, respectively.

Overwrite R_A or R_B by setting Id_a or Id_b on, along with Id_alu_out = 1.

Load value into input_1 or input_2 of part_2_ALU by setting value on alu_select_a or alu_select_b.

When select value = 00, the value stored in R_A is loaded into the input(s).

When select value = 01, the value stored in R_B is loaded into the input(s).

When select value = 10, the value stored in R_C is loaded into the input(s).

When select value = 11, the value stored in R_X is loaded into the input(s).

Store final value into R_R by setting Id_r = 1.

When alu_op = 0, the ALU performs addition on the two inputs. When alu_op = 1, it performs multiplication instead.

2.

data_in	Id_a	Id_b	Id_c	Id_x	alu_select_a	alu_select_b	Id_alu_out	alu_op	Id_r	R_A_val	R_B_val	R_C_val	R_X_val	data_result
0011	1	0	0	0	0	0	0	0	0	0011	0000	0000	0000	0000
0010	0	1	0	0	0	0	0	0	0	0011	0010	0000	0000	0000
0110	0	0	1	0	0	0	0	0	0	0011	0010	0110	0000	0000
0100	0	0	0	1	0	0	0	0	0	0011	0010	0110	0100	0000
0000	0	1	0	0	11	01	1	1	0	0011	1000	0110	0100	0000
0000	0	1	0	0	00	01	1	0	0	0011	1011	0110	0100	0000
0000	1	0	0	0	11	11	1	1	0	0001 0000	1011	0110	0100	0000
0111	1	0	0	0	00	10	1	1	0	0110 0000	1011	0110	0100	0000
1000	0	0	0	0	00	01	0	0	1	0110 0000	1011	0110	0100	0110 1011

3.

state	Id_a	Id_b	Id_c	Id_x	alu_select_a	alu_select_b	Id_alu_out	alu_op	Id_r	next_state
0000	1	0	0	0	0	0	0	0	0	0001
0001	0	1	0	0	0	0	0	0	0	0010
0010	0	0	1	0	0	0	0	0	0	0011
0011	0	0	0	1	0	0	0	0	0	0100
0100	0	1	0	0	11	01	1	1	0	0101
0101	0	1	0	0	00	01	1	0	0	0110
0110	1	0	0	0	11	11	1	1	0	0111
0111	1	0	0	0	00	10	1	1	0	1000
1000	0	0	0	0	00	01	0	0	1	1001

5. Verification to the correctness of the FSM with test cases of Part 2 Q2.

input values: $a = 3, b = 2, c = 6, x = 4$

output: $6b$, which equals to 107 in decimal number.