EE 316 Lab 5.1 Cover Sheet

Name: Joshua Dong

EID: jid295

Section: MW 11-12A

Design Problem: 16.4

Attachment checklist:

1. Coversheet for Lab 5.1
2. Preparatory Questions & Answers
3. State graph and table, K-maps, and equations (by hand)
4. Printout of state table and verification window (*Print All* command in LogicAid)

5-1. A state assignment (by hand)

5-2. Printout of two different state assignments (5-1 and another) in LogicAid

6. Transition table determined from the SimUaid simulation (by hand)
7. Output sequences for Z under two different input sequences
8. Printout of waveforms in landscape mode (scaled to 20ns/div)
9. Printout of SimUAid circuit (with minimum no. of gates)

(Before submission, you should save/remember all your Lab 5.1 work for Lab 6.2)

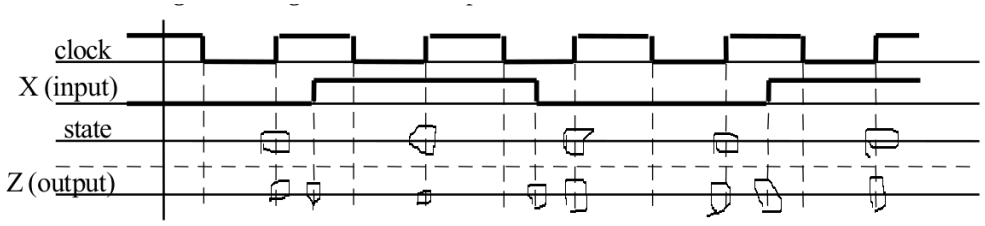
Signatures of the TA:

Problem Statement

Design a sequential circuit which adds two to a binary number in the range 0000 through 1001. The input and output should be serial with the least significant bit first. Find a state table with a minimum number of states. Design the circuit using NAND gates, NOR gates, and three D flip-flops. Any solution which is minimal for your state assignment and uses 10 or fewer gates and inverters is acceptable.

Preparatory Questions

a.



b. 0 to 1, rising edge

c. Yes, for that moment. On the rising edge, the value gets updated to the latest version. The output may be incomplete, but it will be correct for the values when the rising edge is triggered.

d. No.

e. Before. The future states of the machine may be determined by this.

f. Before. The future states of the machine may be determined by this.

g. Reading the output before the clock is triggered is a flawed approach. A is correct.

State Table Derivations

We start with a table of the desired states:

X				Z			
t_3	t_2	t_1	t_0	t_3	t_2	t_1	t_0
0	0	0	0	0	0	1	0
0	0	0	1	0	0	1	1
0	0	1	0	0	1	0	0
0	0	1	1	0	1	0	1
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	1	0	0	0
0	1	1	1	1	0	0	1
1	0	0	0	1	0	1	0
1	0	0	1	1	0	1	1

We create a truth table for the state transitions:

Time	Aggregate Input (little endian)	Q	Q^+		Z	
			$X=0$	$X=1$	$X=0$	$X=1$
t_0	reset	A	B	C	0	1
t_1	0	B	D	F	1	0
	1	C	E	G	1	0
t_2	00	D	H	L	0	1
	01	E	I	M	0	1
	10	F	J	N	1	0
	11	G	K	P	1	0
t_3	000	H	A	A	0	1
	001	I	A	A	0	1
	010	J	A	-	0	-
	011	K	A	-	0	-
	100	L	A	-	0	-
	101	M	A	-	0	-
	110	N	A	-	1	-
	111	P	A	-	1	-

We now have the table for state relationships. Simplfication results in:

Q	Q^+		Z	
	$X=0$	$X=1$	$X=0$	$X=1$
A	B	B	0	1
B	D	F	1	0
D	H	H	0	1
F	H	N	1	0
H	A	A	0	1
N	A	-	1	-

This table can get Boolean expressions for logic.

State Machine Design

Initially, we will start the circuit in a reset state designated A ($A000$ in the graph). The first input bit will be echoed back out and A will always transition to B (depicted as $B001$).

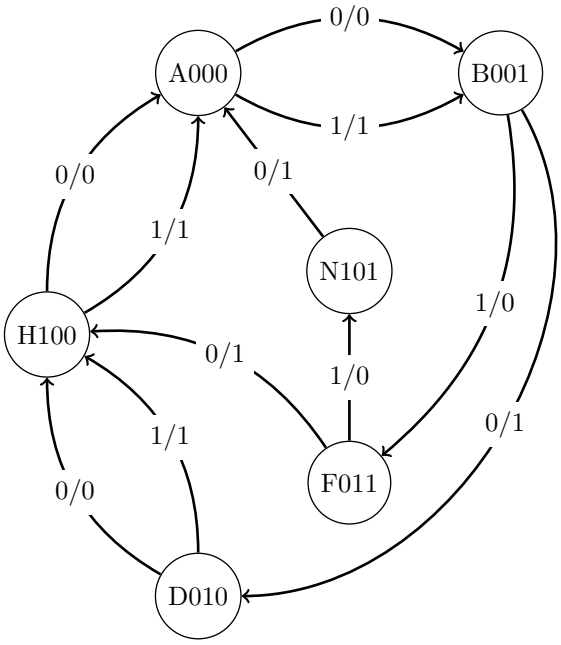
When in state B, the next bit given will be flipped on output. If a 1 is received, then B transitions to F (depicted $F011$), else D (depicted $D010$).

After this, D will always transition to H (depicted $H011$) since D represents the state where 0 is in the 2's place. The input is simply echoed in this state.

If the current state is F, then there may be a carry to deal with. If the next input bit is 0, then F transitions to H while echoing the input value. However, if the input bit while in state F is 1, then F transitions to N (depicted below as $N101$), outputting 0. N represents the state where the three least significant bits were $X11$ (little endian) and a carry bit is needed in the most significant output digit.

Finally, H and N always go back to the reset state A. H has no carry, so input is echoed to output. N has a carry, so the bit is flipped. Note that N never receives a 1, by constraint of our problem. However, it would be reasonable to let it still transition to A while outputting a 0 if 1 was received (not implemented here).

State Graph



LogicAid

The desired data is entered into LogicAid:

State_Table1

PS	NS		OUTPUTs*		INPUT-VARs
	0	1	0	1	X
A	B	B	0	1	
B	D	F	1	0	
D	H	H	0	1	
F	H	N	1	0	
H	A	A	0	1	
N	A	-	1	-	

State Assignment

After verification, we perform a state assignment - specifically we select the initial reset state A to be encoded as flip-flop state 000, B to be 100, D to be 101, F to be 111, H to be 110, and N to be 001. Z represents output, X an input bit, Q_n the n^{th} bit, and Q_n^+ the next n^{th} bit. The following is the transition table after the state assignment:

	$Q_1Q_2Q_3$	$Q_1^+Q_2^+Q_3^+$		Z	
		$X=0$	$X=1$	$X=0$	$X=1$
A	000	001	001	0	1
B	001	010	011	1	0
D	010	100	100	0	1
F	011	100	101	1	0
H	100	000	000	0	1
N	101	000	-	1	-

The state assignment in LogicAid is shown below:

State_Table1

ASSIGN	PS	NS		OUTPUTs*		INPUT-VARs
		0	1	0	1	X
000	A	B	B	0	1	
001	B	D	F	1	0	
010	D	H	H	0	1	
011	F	H	N	1	0	
100	H	A	A	0	1	
101	N	A	-	1	-	

Equations

From the transition table, we plot the next-state maps for the flip-flops and the map for the output function Z and derive the equations for Q_1^+, Q_2^+, Q_3^+, and Z:

$Q_1^+ = Q_2$

Q_2Q_3 \ XQ_1	00	01	11	10
00	0	0	0	0
01	0	0	X	0
11	1	X	X	1
10	1	X	X	1

$Q_2^+ = Q_1'Q_2'Q_3 = (Q_1 + Q_2 + Q_3')' = \downarrow (Q_1, Q_2, Q_3')$

Q_2Q_3 \ XQ_1	00	01	11	10
00	0	0	0	0
01	1	0	X	1
11	0	X	X	0
10	0	X	X	0

$$Q_3^+ = Q_1'Q_2'Q_3' + XQ_3 = ((Q_1'Q_2'Q_3')'(XQ_3)')' = \uparrow(\uparrow(Q_1', Q_2', Q_3'), \uparrow(X, Q_3))$$

Q_2Q_3 \ XQ_1	00	01	11	10
00	1	0	0	1
01	0	0	X	1
11	0	X	X	1
10	0	X	X	0

$$Z = X'Q_3 + XQ_3' = ((X'Q_3)'(XQ_3')')' = \uparrow(\uparrow(X', Q_3), \uparrow(X, Q_3'))$$

Q_2Q_3 \ XQ_1	00	01	11	10
00	0	0	1	1
01	1	1	X	0
11	1	X	X	0
10	0	X	X	1

Circuit

Below is a schematic of the circuit in SimUaid: