1 Counter Design (24pts)

Design a 3-bit counter which counts CBA in the sequence: 001, 011, 010, 110, 111, 101, 100, and repeats.

1. (4pts) Show the truth table and the Karnaugh map of C^+, B^+, and A^+.
2. (4pts) Use D flip-flops. Derive a minimum sum-of-products expression for D_C.
3. (4pts) Use T flip-flops. Derive a minimum sum-of-products expression for T_C.
4. (6pts) Use S-R flip-flops. Derive a minimum sum-of-products expression for S_B and R_B.
5. (6pts) Use J-K flip-flops. Derive a minimum sum-of-products expression for J_A and K_A.

1.

C	B	A	C^+	B^+	A^+
0	0	0	X	X	X
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	1

BA \ C	0	1
00	X	0
01	0	1
11	0	1
10	1	1

C^+

BA \ C	0	1
00	X	0
01	1	0
11	1	0
10	1	1

B^+

BA \ C	0	1
00	X	1
01	1	0
11	0	1
10	0	1

A^+

2.

$C^+ = D_C = CA + BA'$ #

3.

BA \ C	0	1
00	x	0
01	0	1
11	0	1
10	1	1

C^+

BA \ C	0	1
00	x	1
01	0	0
11	0	0
10	1	0

T_C

$T_C = C'A' + B'A'$

4. (B的 0.1看清楚)

BA \ C	0	1
00	x	0
01	1	0
11	1	0
10	1	1

B^+

BA \ C	0	1
00	x	0
01	1	0
11	x	0
10	x	x

S_B

BA \ C	0	1
00	x	x
01	0	x
11	0	1
10	0	0

R_B

$S_B = C'$ $R_B = CA$

5.

BA \ C	0	1
0 0	x	1
0 1	1	0
1 1	0	1
1 0	0	1

A^+

BA \ C	0	1
0 0	x	1
0 1	x	x
1 1	x	x
1 0	0	1

J_A

BA \ C	0	1
0 0	x	x
0 1	0	1
1 1	1	0
1 0	x	x

K_A

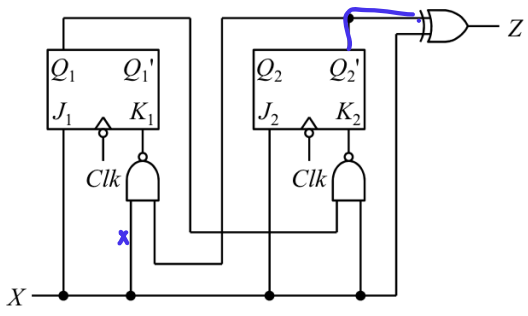
$J_A = C$

$K_A = CB' + C'B$

2 Construction of State Table and State Graph (12pts)

Given the following circuit,

2.



1. (6pts) Construct the state table.
2. (6pts) Construct the state graph.

step 1: $J_1 = X$, $K_1 = (X \cdot Q_2')' = X' + Q_2$

$J_2 = X$, $K_2 = (X Q_1)' = X' + Q_1'$

$Z = X \oplus Q_2'$

step 2: $Q_1^+ = J_1 Q_1' + K_1' Q_1$

$= X Q_1' + X Q_2' Q_1$

$Q_2^+ = J_2 Q_2' + K_2' Q_2$

$= X Q_2' + X Q_1 Q_2$

step 3: next-state map

Q_1Q_2 \ X	0	1
00	0	1
01	0	1
11	0	0
10	0	1

Q_1^+

Q_1Q_2 \ X	0	1
00	0	1
01	0	0
11	0	1
10	0	1

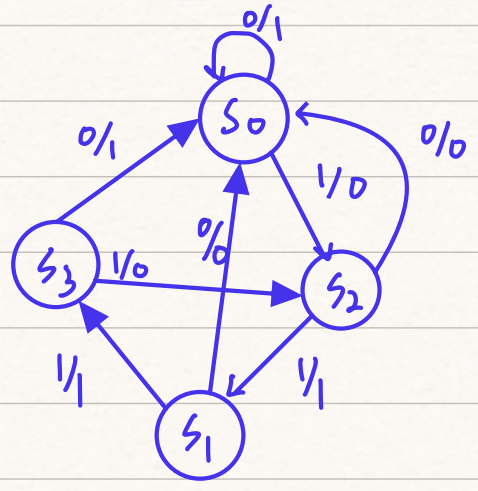
Q_2^+

state 4: next state table

	Q_1Q_2	$Q_1^+ Q_2^+$ x=0	$Q_1^+ Q_2^+$ x=1	Z x=0	Z x=1
S_0	0 0	0 0	1 1	1	0
S_1	0 1	0 0	1 0	0	1
S_2	1 1	0 0	0 1	0	1
S_3	1 0	0 0	1 1	1	0

#1

state 5: state graph (mealy machine



3 Derivation of State Tables (12pts)

1. (6pts) A Mealy sequential circuit has one input (X) and one output (Z). $Z = 1$ if and only if the most recent input, combined with the preceding three inputs, was not a valid BCD encoding of a decimal digit; otherwise, $Z = 0$. Assume the BCD digits are received least significant bit first. Derive a state table for the circuit and explain the meaning of each state. Assume that in the reset state all previous inputs were 0. Minimize the numbers of states (three states are sufficient).

2. (6pts) Repeat for a Moore circuit ($Z = 1$ if and only if the previous four inputs were not a valid BCD digit). Minimize the numbers of states (four states are sufficient).

Z=1 _ _ _ _

不合法 ↑先

不合法 case:

1010
1011
1100
1101
1110
1111

狀態 state map

		next state		Z	
		x=0	x=1	x=0	x=1
S_0	000	000	100	0	0
	001	000	100	0	0
S_1	010	001 S_0	101 S_2	0	1
	011	001	101	0	1
S_2	100	010 S_1	110 S_3	0	1
	101	010	110	0	1
S_3	110	011 S_1	111 S_3	0	1
	111	011	111	0	1

S_1 x

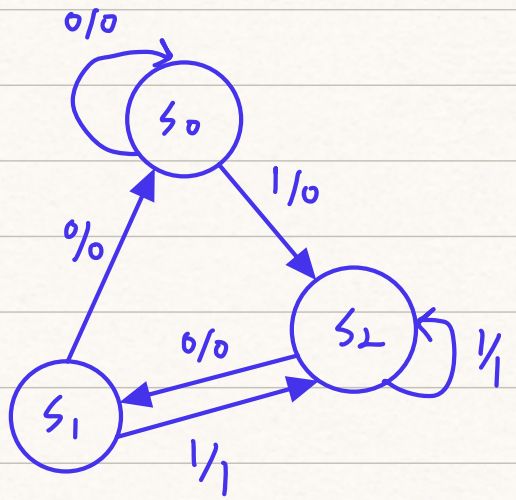
S_2 x $S_0 = S_1$ X $S_2 = S_3$

S_3 x $S_0 = S_1$ X $S_2 = S_2$

S_0 S_1 S_2

故 $S_2 \equiv S_3$

S_0	000	000 S_0	100 S_2	0	0
	001	000	100	0	0
S_1	010	001 S_0	101 S_2	0	1
	011	001	101	0	1
S_2	100	010 S_1	110 S_2	0	1
	101	010	110	0	1
S_3	110	011 S_1	111 S_3	0	1
	111	011	111	0	1



present	state	next state x=0	next state x=1	z x=0	z x=1
(00-)	S_0	S_0	S_2	0	0
(01-)	S_1	S_0	S_2	0	1
(1--)	S_2	S_1	S_2	0	1

LSB at right

2. moore machine

不合法

1010
1011
1100
1101
1110
1111

present	next x=0	next x=1	z	
(000-) S_0	S_0	S_4	0	$S_0 \equiv S_1$
(001-) S_1	S_0	S_4	0	
(010-) S_2	S_1	S_5	0	$S_2 \equiv S_3$
(011-) S_3	S_1	S_5	0	
(100-) S_4	S_2	S_6	0	
(101-) S_5	S_2	S_6	1	
(110-) S_6	S_3	S_7	1	
(111-) S_7	S_3	S_7	1	$S_6 \equiv S_7$

S_2	4 ~~5~~			
S_4	0 ~~2~~ 4 ~~6~~	5 ✓ 6		
S_5	x	x	x	
S_6	x	x	x	✓
	S_0	S_2	S_4	S_5

故 $S_5 \equiv S_6$

重新 assign state number

present	next x=0	next x=1	z
(00--) S_0	S_0	S_2	0
(01--) S_1	S_0	S_3	0
(100-) S_2	S_1	S_3	0
(11--) (101-) S_3	S_1	S_3	1

#

LSB at right

4 Lab 2: Part 1 (30pts)

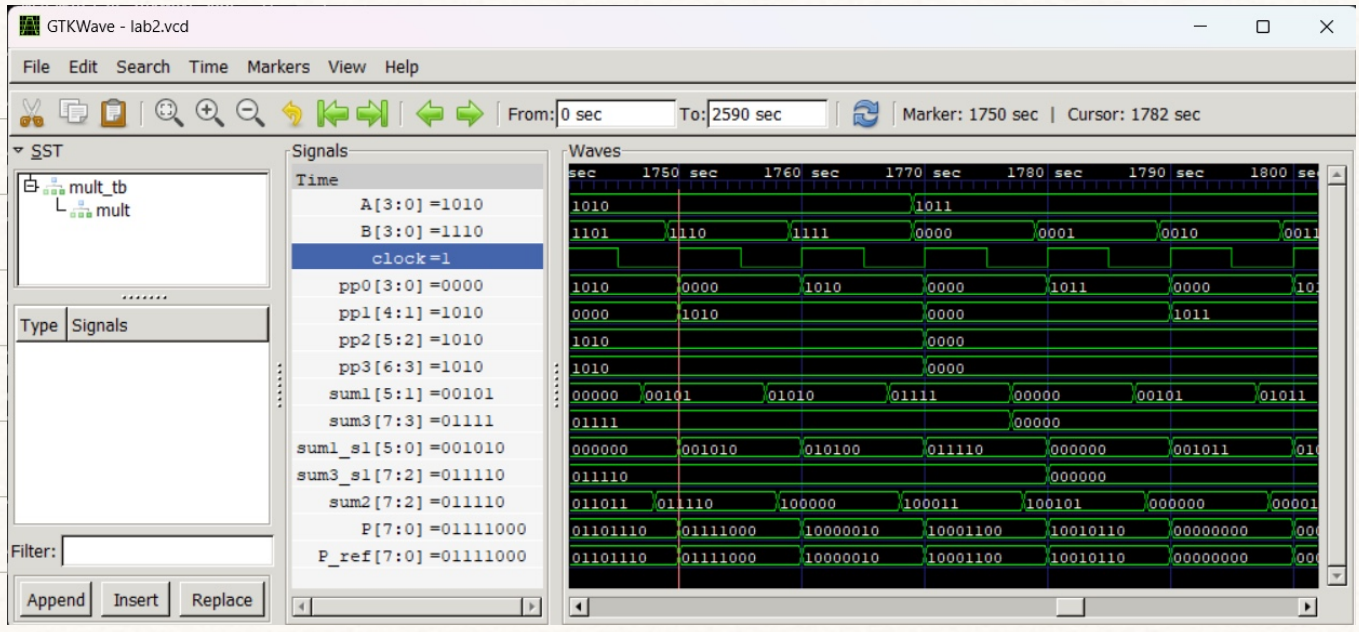
Replace <index> in lab2.v with proper indexes, but leave the rest of the code unchanged.

1. (12pts) Print out the module mult_fast.
2. (12pts) Show the waveform with the settings in lab2.sav (should include 1750 to 1800 sec).
3. (6pts) What is the latency (worst case waiting time from the input becomes steady to the registe of the last stage refreshes)?

1.

```
// pipelined fast multiplier
module mult_fast(
        output reg[7:0] P,  // product
        input[3:0] A, B,    // multiplicand and multiplier
        input clk                   // clock (posedge)
        );
        // stage 0 (input)
        reg[3:0] a_s0, b_s0;
        always @(posedge clk) begin
                a_s0 <= A;
                b_s0 <= B;
        end
        // stage 1
        wire[3:0] pp0 = a_s0 & {4{b_s0[0]}}; // ignore the delays of AND gates
        wire[4:1] pp1 = a_s0 & {4{b_s0[1]}}; // ignore the delays of AND gates
        wire[5:2] pp2 = a_s0 & {4{b_s0[2]}}; // ignore the delays of AND gates
        wire[6:3] pp3 = a_s0 & {4{b_s0[3]}}; // ignore the delays of AND gates
        reg[5:1] sum1;
        always @(pp0, pp1)
                sum1[5:1] <= #7 pp0[3:1] + pp1[4:1]; // delay of the 4-bit adder
        reg[7:3] sum3;
        always @(pp2, pp3)
                sum3[7:3] <= #7 pp2[5:3] + pp3[6:3]; // delay of the 4-bit adder
        reg[5:0] sum1_s1;
        reg[7:2] sum3_s1;
        always @(posedge clk) begin
                sum1_s1 <= {sum1, pp0[0]};
                sum3_s1 <= {sum3, pp2[2]};
        end
        // stage 2 (outout)
        reg[7:2] sum2;
        always @(sum1_s1, sum3_s1)
                sum2[7:2] <= #8 sum1_s1[5:2] + sum3_s1[7:2]; // delay of the 6-bit adder
        always @(posedge clk) begin
                P <= {sum2, sum1_s1[1:0]};
        end
endmodule
```

2.


GTKWave - lab2.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 2590 sec
Marker: 1750 sec | Cursor: 1782 sec
SST
mult_tb
mult
Type Signals
Filter:
Append Insert Replace
Signals
Time
A[3:0] =1010
B[3:0] =1110
clock=1
pp0[3:0] =0000
pp1[4:1] =1010
pp2[5:2] =1010
pp3[6:3] =1010
sum1[5:1] =00101
sum3[7:3] =01111
sum1_sl[5:0] =001010
sum3_sl[7:2] =011110
sum2[7:2] =011110
P[7:0] =01111000
P_ref[7:0] =01111000
Waves
1750 sec
1760 sec
1770 sec
1780 sec
1790 sec
1800 sec

3. 30 ticks

5 Lab 2: Part 2 (12pts)

Minimize the clock cycle by changing the delays in the module mult_tb.

1. (6pts) What is the minimum clock cycle?
2. (6pts) Show the waveform with the settings in lab2.sav (should include 1750 to 1800 sec).

1. 8 ticks

2.