

# CSIE 2344: Homework 4

Due May 20 (Monday) at Noon

There are 90 points in total. Points will be deducted if no appropriate intermediate step is provided.

When you submit your homework on Gradescope, please select the corresponding page(s) of each question.

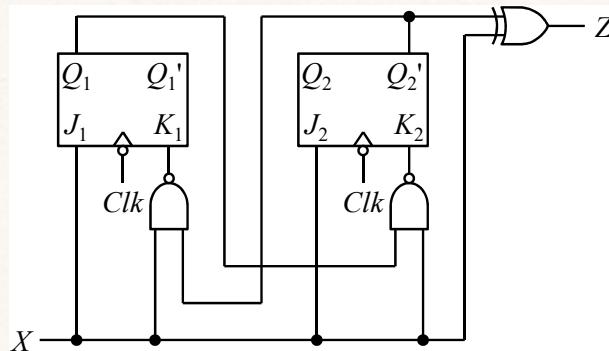
## 1 Counter Design (24pts)

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  $K_A$ .

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

Given the following circuit,



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

### **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. (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). (Four states are sufficient.)

### **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 register of the last stage refreshes)?

### **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 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  $K_A$ .

Ans:

①

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

$C$

$BA$	0	1
00	X	0
01	0	1
11	0	1
10	1	1

$C^+$

$C$

$BA$	0	1
00	X	0
01	1	0
11	1	0
10	1	1

$B^+$

$C$

$BA$	0	1
00	X	1
01	1	0
11	0	1
10	0	1

$A^+$

②

$BA$	0	1
00	X	0
01	0	1
11	0	1
10	1	1

$D_C$

$$D_C = A'C + A'B$$

③

$BA$	0	1
00	X	1
01	0	0
11	0	0
10	1	0

$T_C$

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

(4)

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

 $S_B$ 

$$S_B = C'$$

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

 $R_B$ 

$$R_B = AC$$

(5)

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

 $J_A$ 

$$J_A = C$$

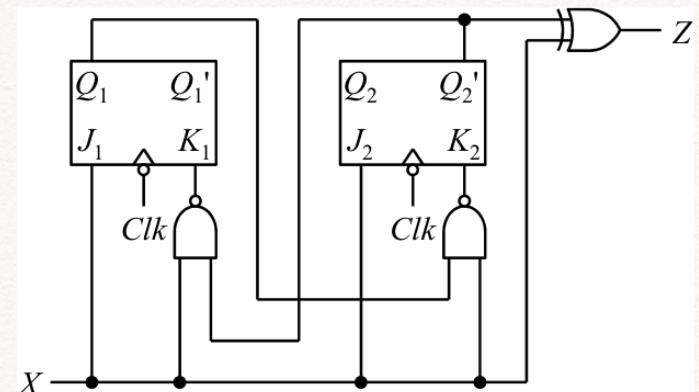
	C	0	1
BA		X	X
00			
01		0	1
11		1	0
10		X	X

 $K_A$ 

$$K_A = B'C + BC'$$

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

Given the following circuit,

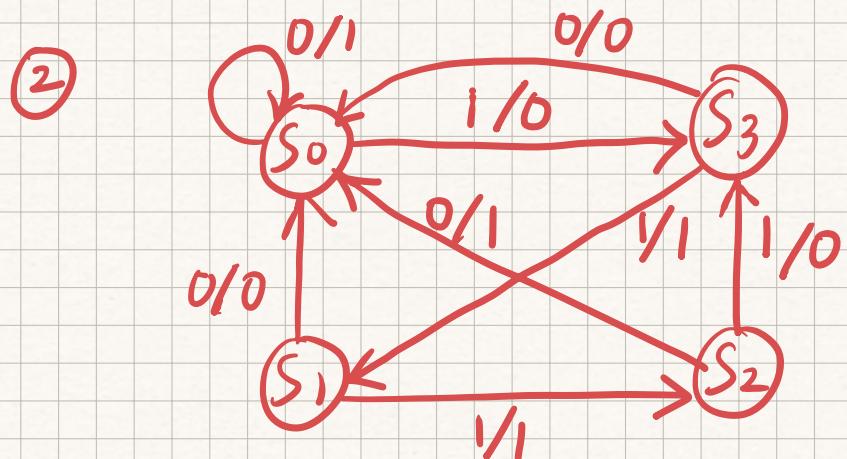


1. (6pts) Construct the state table.
2. (6pts) Construct the state graph.

**Ans:**

①

		$X=0$	$X=1$
		$Q_1^+ Q_2^+$	$Z$
$Q_1 Q_2$		$Q_1^+ Q_2^+$	$Z$
0 0 ( $S_0$ )	0 0 ( $S_0$ )	1	1 1 ( $S_3$ ) 0
0 1 ( $S_1$ )	0 0 ( $S_0$ )	0	1 0 ( $S_2$ ) 1
1 0 ( $S_2$ )	0 0 ( $S_0$ )	1	1 1 ( $S_3$ ) 0
1 1 ( $S_3$ )	0 0 ( $S_0$ )	0	0 1 ( $S_1$ ) 1



### 3 Derivation of State Tables (12pts)

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

Ans:

①

- **State 0 ( $S_0$ ):** This is the initial/reset state where no bits or all zero bits have been received so far. At this point, the input can still form a valid BCD code.
  - Includes the cases of no bits or 0000.
- **State 1 ( $S_1$ ):** This state indicates that the bits received so far can still form a valid BCD digit.
  - Includes the cases of 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001
- **State 2 ( $S_2$ ):** This state indicates that the bits received so far cannot form a valid BCD digit.
  - Includes the cases of 1010, 1011, 1100, 1101, 1110, 1111

Current State	Input X	Next State	Output Z
$S_0$	0	$S_0$	0
$S_0$	1	$S_1$	0
$S_1$	0	$S_1$	0
$S_1$	1	$S_1, S_2$	0
$S_2$	0	$S_1, S_2$	1
$S_2$	1	$S_1, S_2$	1

(2)

- State 0 (Valid): This state represents that the sequence received so far forms a valid BCD digit (0000 to 1001).
- State 1 (Invalid): This state represents that the sequence received so far does not form a valid BCD digit (1010 to 1111).
- State 2 and 3: These states are used to gather four bits of input before making a judgment on validity.

### Explanation of States and Transitions

#### 1. State 0 (Valid):

- Output  $Z=0$

-  $Z=0$ , indicating previous inputs were a valid BCD digit.

- Any input leads to  $S_2$ , where the machine starts collecting the next set of inputs.

#### 2. State 1 (Invalid):

- Output  $Z=1$

-  $Z=1$ , indicating previous inputs were not a valid BCD digit.

- Any input leads to  $S_3$ , which will continue collecting inputs but is oriented towards determining if the sequence stays invalid.

#### 3. State 2:

- A transitional state with  $Z=0$

-  $Z=0$  where the next inputs start being collected. After one input, it transitions to either  $S_0$  or  $S_1$  depending on whether the input continues to lead to a valid sequence.

- If input is 0, it moves to  $S_0$  (anticipating a valid sequence with next inputs).

- If input is 1, it moves to  $S_1$  (anticipating an invalid sequence).

#### 4. State 3:

- A transitional state with  $Z=1$

-  $Z=1$  used to handle sequences that are likely to remain invalid.

- Independently of the next input (0 or 1), it transitions to  $S_1$ , assuming the sequence remains invalid.

Current State	Input X	Next State	Output Z
---------------	---------	------------	----------

$S_0$	0	$S_2$	0
$S_0$	1	$S_2$	0
$S_1$	0	$S_3$	1
$S_1$	1	$S_3$	1
$S_2$	0	$S_0$	0
$S_2$	1	$S_1$	0
$S_3$	0	$S_1$	1
$S_3$	1	$S_1$	1

## 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 register of the last stage refreshes)?

Ans:

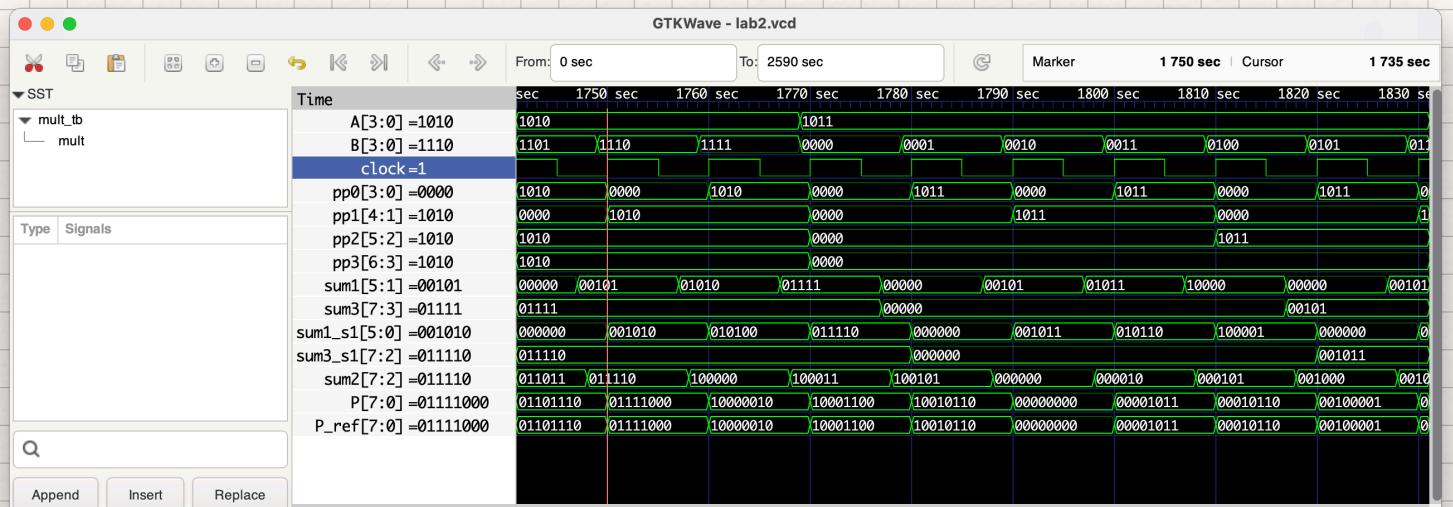
①

```
1 // pipelined fast multiplier
2 module mult_fast(
3   output reg[7:0] P, // product
4   input [3:0] A, B, // multiplicand and multiplier
5   input clk // clock (posedge)
6 );
7 // stage 0 (input)
8 reg[3:0] a_s0, b_s0;
9 always @(posedge clk) begin
10   a_s0 <= A;
11   b_s0 <= B;
12 end
13 // stage 1
14 wire[3:0] pp0 = a_s0 & {4{b_s0[0]}};
15 wire[4:1] pp1 = a_s0 & {4{b_s0[1]}};
16 wire[5:2] pp2 = a_s0 & {4{b_s0[2]}};
17 wire[6:3] pp3 = a_s0 & {4{b_s0[3]}};
18 reg[5:1] sum1;
19 always @(pp0, pp1)
20   sum1[5:1] <= #7 pp0[3:1] + pp1[4:1];
21 reg[7:3] sum3;
22 always @(pp2, pp3)
23   sum3[7:3] <= #7 pp2[5:3] + pp3[6:3];
24 reg[5:0] sum1_s1;
25 reg[7:2] sum3_s1;
26 always @(posedge clk) begin
27   sum1_s1 <= {sum1, pp0[0]};
28   sum3_s1 <= {sum3, pp2[2]};
29 end
30 // stage 2 (outout)
31 reg[7:2] sum2;
32 always @(sum1_s1, sum3_s1)
33   sum2[7:2] <= #8 sum1_s1[5:2] + sum3_s1[7:2];
34 always @(posedge clk) begin
35   P <= {sum2, sum1_s1[1:0]};
36 end
37 endmodule
```

③

30 seconds

②



## 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 seconds

②

