Digital System Design

Exam #1 Nov. 2019

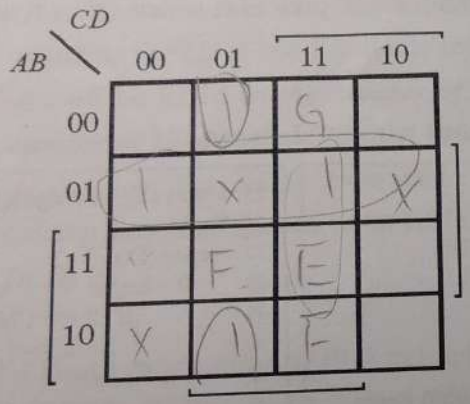
- 不可看書及任何參考資料，請詳述設計或演算過程，並依題號順序寫入答冊；否則高題號答案出現後，之後的低題號答案將不予計分。題目要求畫電路方塊圖或狀態圖時，請務必繪製清晰。題號前標「*」者，請直接在考卷上作答。總分 114。

(This is a close-book examination. Please describe your answers as detailed as possible, and draw the block diagram or state diagrm clearly when it is required. Write your answers into the answer sheets in the order of the question number. No score will be given to the answer of a lower number question if it is written behind that of a higher number question. Write the answer of a question marked with "*" on the examination paper, and hand in the examination paper together with the answer sheets. **Total score: 114**)

1. *(8%) For the following function, find the minimum sum of products using 4-variable maps with map-entered variables. In the following function, m_i represents a minterm of variables A, B, C, and D.

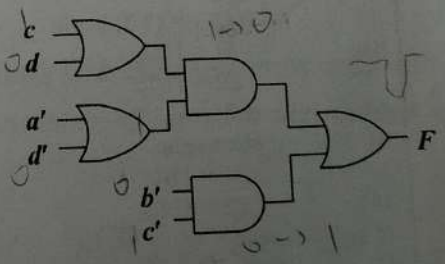
$Z(A, B, C, D, E, F, G) = \Sigma m(1, 4, 7, 9) + \Sigma d(5, 6, 8) + E(m_{15}) + F(m_{11} + m_{13}) + G(m_3)$

8



$Z = A'B + B'C'D + BCDE + C'DF + AB'DF + A'DG$

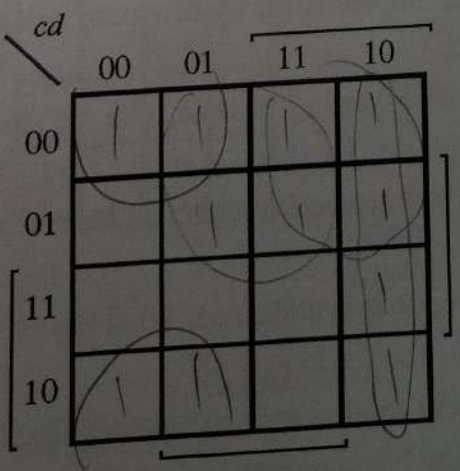
2. *(10%) For the circuit given below,



(a) (8%) Derive the sum-of-products expression of F, show the AND terms in the following Karnaugh map, and find all of the static 1-hazards in the circuit. State the condition under which each hazard can occur.

(b) (2%) Indicate which changes are necessary to eliminate the hazards, and derive the revised sum-of-products equation of *F*.

9



(a) $(c+d)(c'+d') + b'c'$

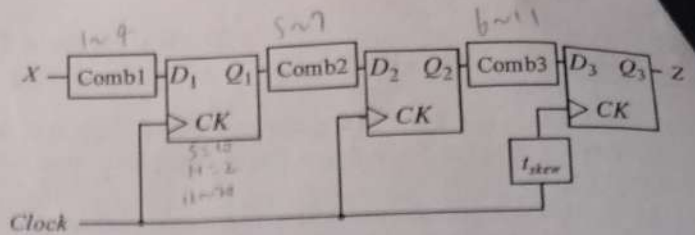
$= a'c + a'd + cd' + dd' + b'c'$

when $d=0, c=1, b=0$, $a=1$ or 0, and c from 1 to 0, output will have static 1 hazard

(b) add $a'b'$, make $F =$

$a'c + a'd + cd' + b'c' + a'b'$

3. (11%) Consider the following circuit where the combinational circuit is represented by COMB and clock skew is represented by t_{skew}.



Given the following parameters:

FF setup time = 10 ns, FF hold time = 2 ns, FF propagation delay = 12 to 20 ns

Tcomb1 = 1 ns to 4 ns, Tcomb2 = 5 ns to 7 ns, Tcomb3 = 6 ns to 11 ns

(a) (9%) Assume that $t_{skew} = 0$.

i. Indicate all the possible paths in the circuit, calculate the delay for each of the paths, and determine the minimum clock period at which this circuit can be safely clocked.

ii. Compute the latest time (t_x) before and earliest time (t_y) after the rising clock edge that input X is allowed to change and still have proper synchronous operation.

(b) (2%) Assume that t_{skew} = 3 ns. Determine the minimum clock period at which this circuit can be safely clocked.

4. (22%) Examine the following Verilog code of a synchronous sequential circuit.

```
1  module P6 (X, CLK, Z1, Z2);
2  input X, CLK;
3  output Z1, Z2;
4  reg  Z1, Z2;
5  reg [1:0] State;
6  always @(State, X)
7  begin
8    case(State)
9      0:  begin
10           if (X == 1'b0) begin
11             Z1 = 1'b0;
12             Z2 = 1'b1;
13           end
14           else begin
15             Z1 = 1'b1;
16             Z2 = 1'b0;
17           end
18         end
19     1:  begin
20           if (X == 1'b1) begin
21             Z1 = 1'b0;
22             Z2 = 1'b1;
23           end
24         end
25     2:  begin
26           if (X == 1'b0) begin
27             Z1 = 1'b1;
28           end
29           else begin
30             Z2 = 1'b1;
31           end
32         end
33   endcase
34 end
35 always @(posedge CLK)
36 begin
37   case (State)
38     0 : begin
39           if (X == 1'b0)
40             State <= 1;
41           else
42             State <= 2;
43         end
44     1 : begin
45           if (X == 1'b1)
46             State <= 2;
47         end
48     2 :
49         State <= 0;
50   endcase
51 end
52 endmodule
```

(a) (2%) Draw a block diagram of the sequential circuit implemented by this code using D flip-flops and a combinational block.

(b) (8%) Draw the state table that is implemented by this Verilog module. Use "?" to represent an uncertain state or output value.

(b) (4%) Rewrite the code for the outputs of the circuit by concurrent assignment statements.

(c) (8%) For the Verilog code given above,

i. Explain why latches would be created when the code is synthesized. What signal would appear at the latch output?

ii. Make the necessary changes in the **case** statement to eliminate the latches. Explain where to make changes or insert statements by indicating the line numbers in the code.

iii. Make the necessary changes outside the **case** statement to eliminate the latches. Explain where to make changes or insert statements by indicating the line numbers in the code.

5. (6%) Given the template of a Verilog code, draw the hardware obtained if each statement sequence is synthesized:

(a)
```
Z4 = Z3;
Z3 = Z2;
Z2 = Z1;
Z1 = X;
```

(b)
```
Z1 = X;
Z2 = Z1;
Z3 = Z2;
Z4 = Z3;
```

(c)
```
Z1 <= X;
Z2 <= Z1;
Z3 <= Z2;
Z4 <= Z3;
```

```
module P4 (Z1, Z2, Z3, Z4, X, CLK);
input     X, CLK;
output    Z1, Z2, Z3, Z4;
reg       Z1, Z2, Z3, Z4;

always @(posedge CLK)
begin

   ... ...

end

endmodule
```

6. (9%) Implement the following state table using a ROM and D flip-flops. Use a straight binary state assignment: $S_0 = 00$, $S_1 = 01$, $S_2 = 10$, $S_3 = 11$ (Q_1Q_0)

(a) (3%) Show the block diagram of the circuit including the ROM and flip-flops, and indicate the size of the ROM as ***the number of words × the number of bits per word***.

(b) (6%) Show the ROM truth table. Truth table column headings should be in the order Q_1 Q_0 X D_1 D_0 Z.

Present State	Next State X = 0	Next State X = 1	Output (Z) X = 0	Output (Z) X = 1
S_0	S_0	S_1	0	1
S_1	S_2	S_1	1	0
S_2	S_1	S_3	1	1
S_3	S_3	S_2	0	1

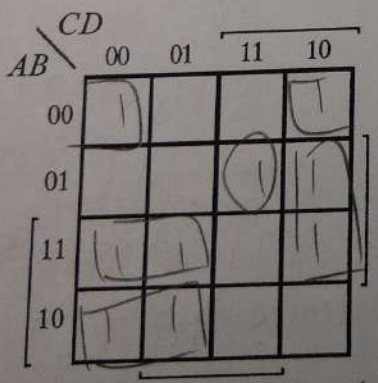
7. *(8%) Find a minimum-row PLA to implement the following three functions:

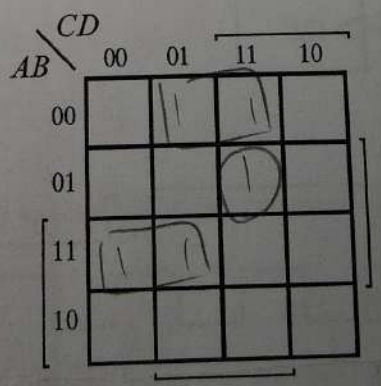
$f_1(A, B, C, D) = \Sigma m(0, 2, 6, 7, 8, 9, 12, 13, 14)$

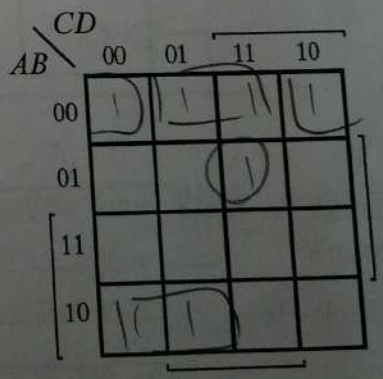
$f_2(A, B, C, D) = \Sigma m(1, 3, 7, 12, 13)$

$f_3(A, B, C, D) = \Sigma m(0, 1, 2, 3, 7, 8, 9)$

Use Karnaugh maps to find common terms. Give the logic equations with common terms underlined, list the distinct terms, and specify the size of the PLA as ***the number of inputs × the number of distinct product terms × the number of outputs***.







$f_1 = AB'C' + AB'C' + A'B'D' + BCD' + A'BCD$

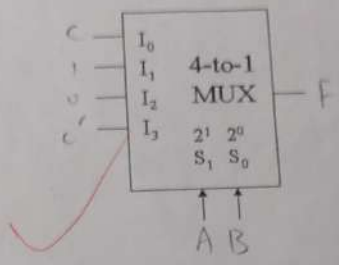
$f_2 = A'B'D + ABC' + A'BCD$

$f_3 = A'B'D' + A'B'D + AB'C' + A'BC$

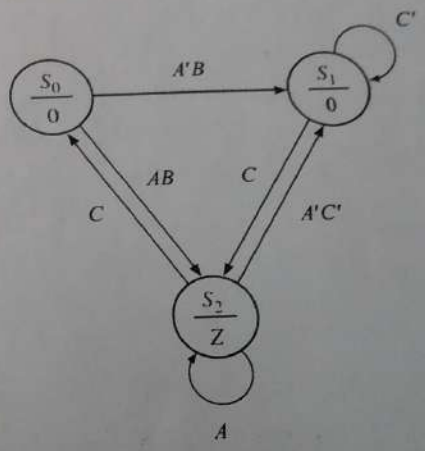
size: 4 × 6 × 3

8. *(8%) Implement the function $F = A'B'C + A'BC + BC'$ using an FPGA with programmable log blocks consisting of 4-to-1 multiplexers. Show the truth table for deriving the inputs of the MUX and draw the block diagram of the function. Assume inputs and their complements are available.

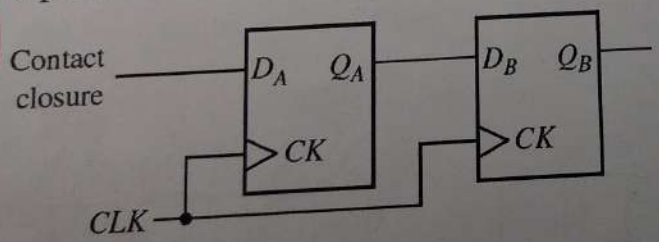
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

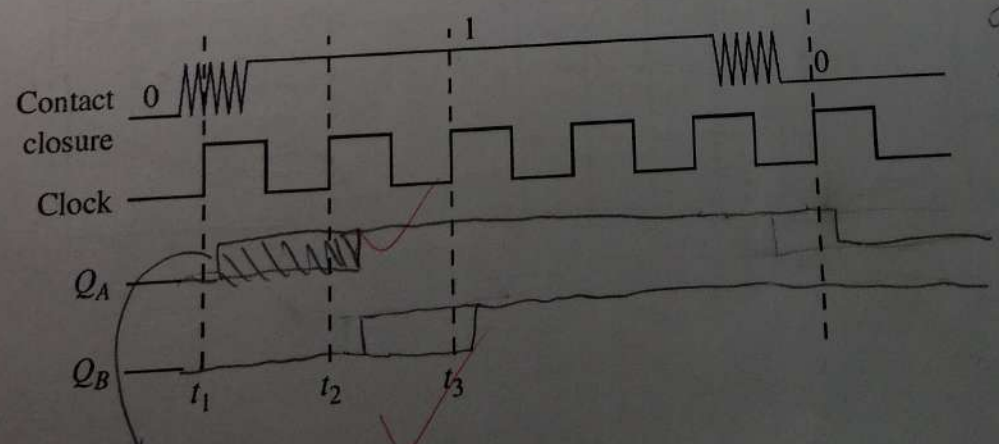


9. (10%) Show that whether the following state graph is a completely specified state graph or not according to the two constraints on the input labels for every state. If not, make the necessary modifications to the graph to make it completely specified and demonstrate that your answer is correct.

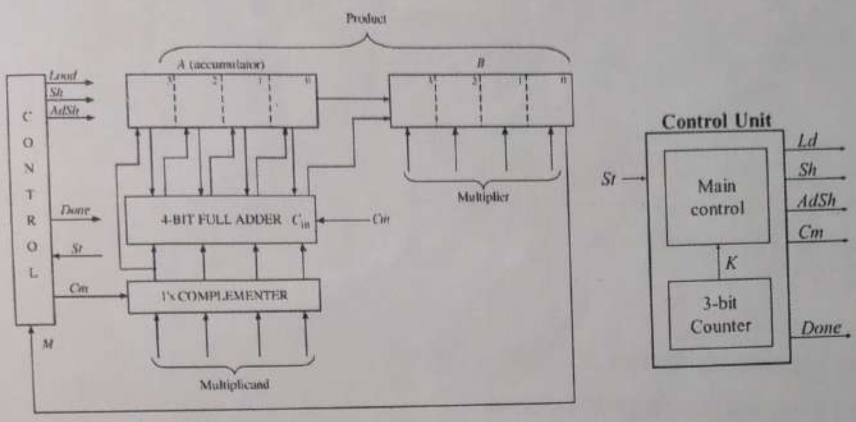


10. *(6%) Given the debouncing and synchronizing circuit, complete the following timing waveform and explain the reason why two flip-flops are required in this circuit.





1. *(16%) Give the block diagram of the faster 2's complement multiplier described in Section 4-10 of the textbook for reference. Extend the circuit to multiply two 8-bit signed binary integers and generate a 16-bit product. Assume that the control unit of the circuit consists of a main control and a 3-bit counter.



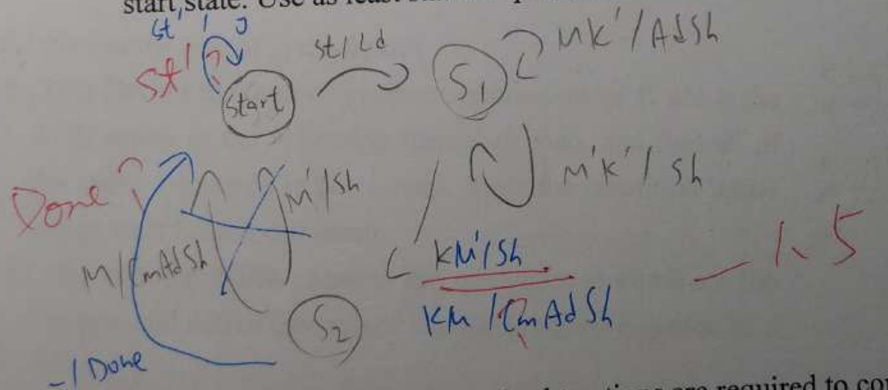
St: start signal
M: current multiplier bit
Load: load multiplier into B and clear A
Sh: shift signal, shift ACC right one place with sign extension
AdSh: add and shift signal
Cm: complement signal
K: = 1 when the counter counts to 7 (111_2)
Done: completion signal

(a) (4%) Describe the procedure of the faster multiplication for signed numbers.

1. Control Unit 傳送 Ld=1, load multiplier into B and clear A
2. 若 B[0] 為 1，將 A 加上被乘數。
3. 將 B[7:1] 存到 B[6:0]，A[0] 存到 B[7]，A[7:1] 存到 A[6:0]，~~A[6] 存到 A[7]~~。每執行一次，Counter 便加 1。
4. 重複 2、3，直到 Counter 數到 7
5. 若 B[0] 為 1，則 A 加上被乘數的 2's complement。
6. 執行 3。

−1

(b) (6%) Draw a state graph for the **Main control**. When a multiplication is completed, return to the start state. Use as least states as possible.



−1.5

(c) (2%) How many additions and subtractions are required to complete the following multiplication? Why?

i. 10010110 × 01101111

~~0 substractions, 6 additions.~~ pos multiplier with 6 ones −1

ii. 01111010 × 11010101

4 additions, 1 substraction

why? −0.5

(d) (4%) Repeat (c) if Booth's algorithm is applied for the multiplication

i. 10010110 × 01101111 0

2 substractions, 2 additions why? −1

ii. 01111010 × 11010101 0

~~4~~ substractions, ~~3~~ additions −2