國立陽明交通大學考試試卷

Exam Paper of National Yang Ming Chiao Tung University

課程名稱 Course Name: Digital system design

姓名 Name: 陳子矜

學號 Student ID: 0819823

班別 Department: 資111

日期 Date: 11/3

※ 考試作弊者將受記大過以上處分

※If you cheat in exam, you will be punished.

題號 NO	分數 Score
1	8
2	12
3	9
4	4.5
5	10.5
6	6
7	4
8	6
9	0
10	0
11	4
12	
13	
14	
15	
總分 Total	64

3. input to output = ?

(a) i. $X \to FF1$: $9+8 = 17$ ns

$FF1 \to FF2$: $20+4+8 = 32$ ns

$FF2 \to FF1$: $20+9+8 = 37$ ns

$FF2 \to Z$: $20+8 = 28$ ns

$t_{clk\ min} = 37$ ns

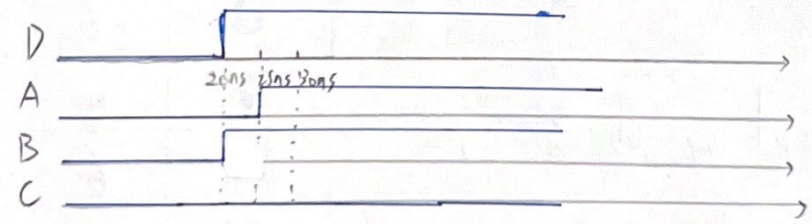
ii. $t_x \geq t_{cx,max} + t_{su} = 9+8 = 17$ ns

$t_y + t_{cx,min} \geq t_h \Rightarrow t_y \geq 3-1 = 2$ ns

(b) $t_{clk\ min} = 37$ ns，因為 t_{skew} 可讓 FF1 → FF2 這條 path 的 delay time 減少，對 FF2 → FF1 沒有影響 ✗

4.

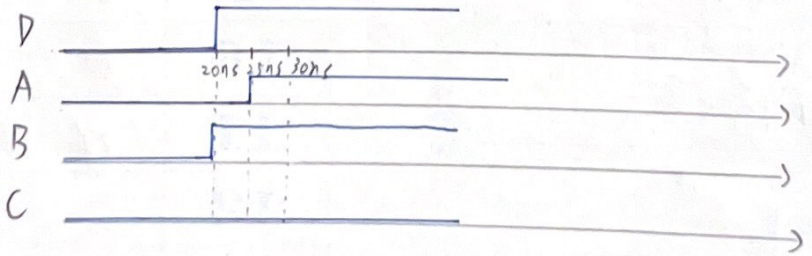
(a)



B: 0 → 1 @ 25 ns

C: 0 → 1 @ 35 ns

(b)



B: 0 → 1 @ (20+D) ns

C 還是 B 在 20ns 的值經過 10ns delay

5.

(a)

```
module p4 (X, CLK, Z);
input X, CLK;
output Z;  reg Z;
reg State [1:0]; reg NextState [1:0];
initial begin
State = 0;
Next State = 0;
end
alway (X or State) begin
   case (State)
      0: begin
         if (X==0) begin NextState <= 0; end
         else begin NextState <= 1; end
         end
      1: begin
         if (X==0) begin Next State <=2; end
         else begin NextState <= 1; end
         end
      2: begin
         if (X==0) begin NextState <= 0; end
         else begin NextState <= 1; end
         end
   endcase
end
always (X or State) Z <= (State[0] && (~X)) || State [1];
always (posedge CLK) State <= NextState;
endmodule
```

we're not writing Java.

use "=" in combinational block.

combine

Current State	Z		Next State	
	X=0	X=1	X=0	X=1
00	0	0	0	1
01	1	0	2	1
10	X	1	0	1
11	X	X	X	X

$Z = State[0]\,X' + State[1]$

(b)

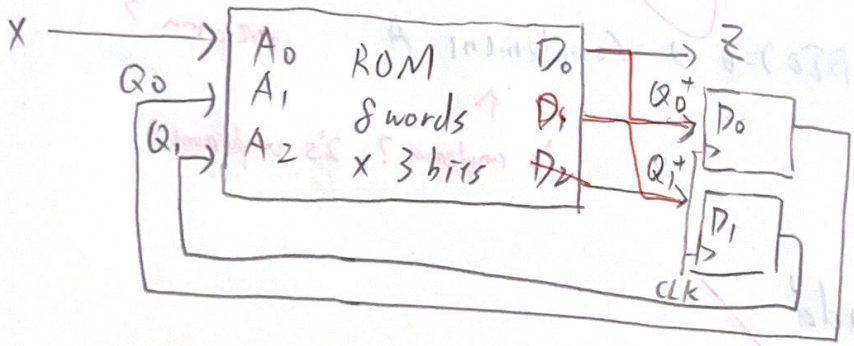
```
module P4 (X, CLK, Z);
input X, CLK;
output Z;
reg State [1:0]; reg Next State [1:0];
always (X or State) begin
   case (State)
      0: if (X==0) NextState = 0;
         else NextState = 1;
      1: if (X==0) NextState = 2;
         else NextState = 1;
      2: if (X==0) NextState = 0;
         else NextState = 1;
   endcase
end
always (posedge CLK) State <= NextState;
assign Z = (State[0] && (~X)) || State[1];
endmodule
```

6.

(a)

Q_1	Q_0	X	D_1	D_0	Z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	1	1
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	X	X	X
1	1	1	X	X	X

(b)



9. $I_1 = A'$

$I_2 = AC$

$I_3 = B$

$I_4 = C'$

$I_5 = B$

$I_6 = B'C$

$I_7 = B'$

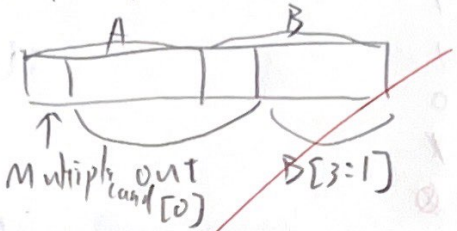
$I_8 = C'$

$I_1 + I_2 + \cdots + I_8 \neq 0$

11.

(a) 1. 於 $S_t = 1$ 則 Load multiplier into B 且 clear A

2. if B[0] = 0 A, B Shr
   else out = A + Multiplicand

⇒ 

3. repeat 2

4. if B[0] = 0 → Complement A

exception ?

↑ is complement? 2's complement?

(b)

(c) 6 add
0 sub

姓名：陳子玲
學號：0819823

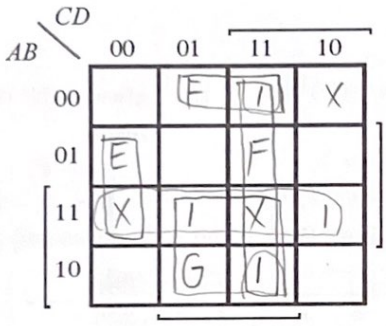
Digital System Design

Exam #1 Nov. 2021

- 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 107。(This is a close-book examination. Please describe your answers as detailed as possible, and draw the block diagram or state diagram 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:** 107)

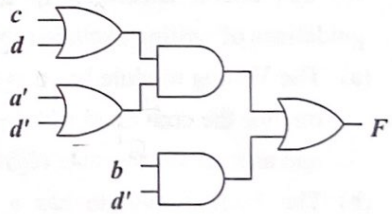
1. *(8%) For the following function, find the minimum sum-of-products (SoP) form using 4-variable maps with map-entered variables. Determine MS_0, MS_E, MS_F, MS_G, and the minimum SoP form of the function Z. Circle each AND term derived on K-map. In the following function, m_i represents a minterm of variables A, B, C, and D.

$Z(A, B, C, D, E, F, G) = \Sigma m(3, 11, 13, 14) + \Sigma d(2, 12, 15) + E(m_4) + F(m_1 + m_7) + G(m_9)$



$Z = AB + B'CD + EBC'D' + FA'B'D + FCD + GAD$ *

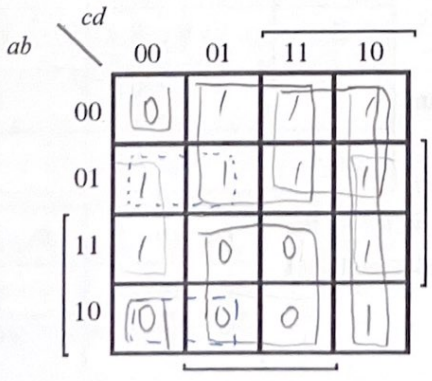
2. *(12%) For the circuit given below,



(a) 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.

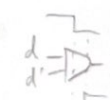
(b) Derive the product-of-sums expression of F, show the OR terms in the following Karnaugh map, and find all of the static 0-hazards in the circuit. State the condition under which each hazard can occur.

(c) Redesign a hazard-free sum-of-products circuit according to (a). Indicate which changes are necessary to eliminate the hazards, and derive the revised sum-of-products equation of *F*.



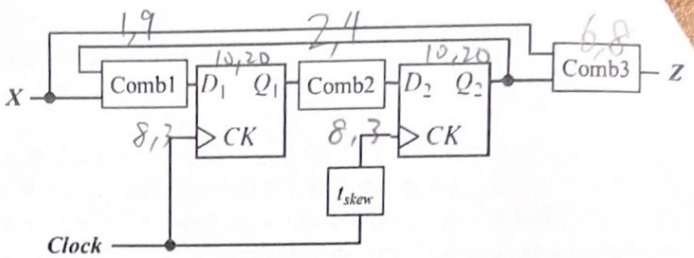
(a) $F = bd' + (c+d)(a'+d')$
$= bd' + a'c + a'd + cd'$
If $a=0$ $b=1$ $c=0$, there is a static 1-hazard when $d = 0$ to 1

(b) $F = (b+c+d)(a'+d')$
If $a=1, b=0, c=0$, there is a static 0-hazard when $d = 1$ to 0



(c) $F = bd' + a'c + a'd + cd' + a'bc'$ *

3. (12%) 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 = 8 ns, FF hold time = 3 ns, FF propagation delay = 10 to 20 ns

Tcomb1 = 1 ns to 9 ns, Tcomb2 = 2 ns to 4 s, Tcomb3 = 6 ns to 8 ns

(a) 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) Assume that t_{skew} = 5 ns. Determine the minimum clock period at which this circuit can be safely clocked and explain why.

4. (6%) In the following Verilog code, A, B, C, and D are 0 at time 10ns. If D changes to 1 at 20ns, specify the (simulation) times at which A, B, and C will change and the values they will take.

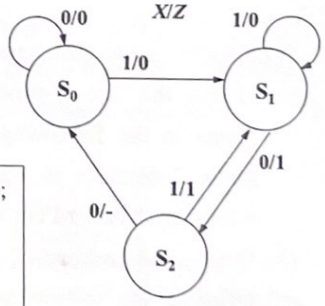
(a)

```
always @(D)
begin
   #5 A <= 1;
   B <= A + 1;
   #10 C <= B;
end
```

(b)

```
always @(D)
begin
   A <= #5 1;
   B <= A + 1;
   C <= #10 B;
end
```

5. (16%) Given the state diagram of a Mealy-type synchronous sequential circuit with one input (X) and one outputs (Z). Write the Verilog code for this circuit according to the state diagram, and do follow the guidelines of writing synthesizable code.



(a) The Verilog module has two **always** blocks, one for the combinational part of the circuit and another for the state register.

(b) The Verilog module has a single **always** block and use continuous assignment statement "**assign**" to generate the output Z.

```
module P4 (X, CLK, Z);
input X, CLK;
output Z;
...
endmodule
```

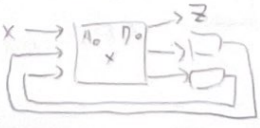
6. (7%) Implement the following state table using a ROM and D flip-flops. Use a straight binary state assignment, i.e., $S_0 = 00$, $S_1 = 01$, $S_2 = 11$ (Q_1Q_0).

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	0
S_1	S_2	S_1	0	1
S_2	S_2	S_0	1	0

(a) (4%) Show the ROM truth table. Truth table column headings should be in the order Q_1 Q_0 X D_1 D_0 Z.

Q_1	Q_0	X	D_1	D_0	Z
0	0	0			
...					

(b) (3%) Draw 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***. Make the ROM as small as possible without using any external gate.



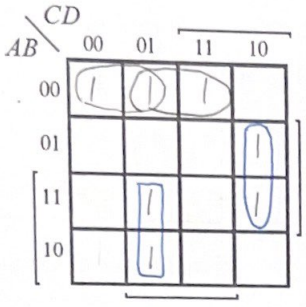
*(11%) Find a minimum-row PLA to implement the following three functions:

$F(A, B, C, D) = \Sigma m(0, 1, 3, 6, 9, 13, 14)$,

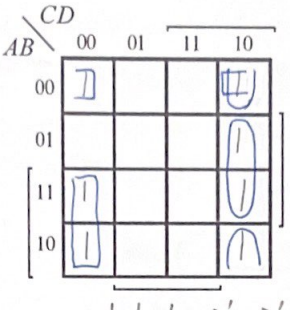
$G(A, B, C, D) = \Sigma m(0, 2, 6, 8, 10, 12, 14)$

$H(A, B, C, D) = \Sigma m(0, 2, 8, 9, 10, 12, 13)$

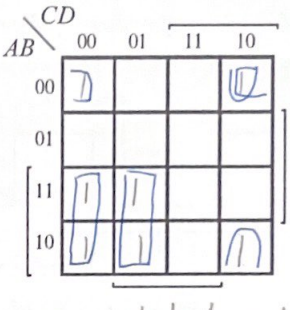
(a) (6%) Use Karnaugh maps to find common terms. Derive the logic equations with common terms underlined and circle each AND term derived on K-map.



$F = A'B'C' + A'B'D + BCD' + AC'D$



$G = A'B'D' + B'CD' + BCD' + AC'D'$



$H = A'B'D' + B'CD' + AC'D' + AC'D$

(b) (5%) List the distinct terms, specify the size of the PLA as ***the number of inputs* × *the number of distinct product terms* × *the number of outputs***, and derive the PLA table.

	Product Term (e.g., *AB'C*)	Inputs *A*	*B*	*C*	*D*	Outputs f_1	f_2	f_3
1	A'B'C'	0	0	0		1		
2	A'B'D	0	0		1	1		
3	BCD'		1	1	0	1	1	
4	AC'D	1		0	1	1		1
5	A'B'D'	0	0		0		1	1
6	B'CD'		0	1	0		1	1
7	AC'D'	1		0	0		1	1

size of PLA ?

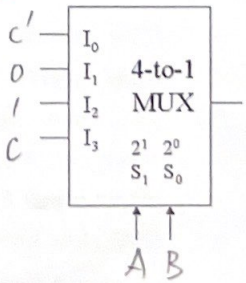
8. *(6%) Implement the following function using an FPGA with programmable logic blocks consisting of 4-to-1 multiplexers:

$F = AB' + A'B'C' + ABC$

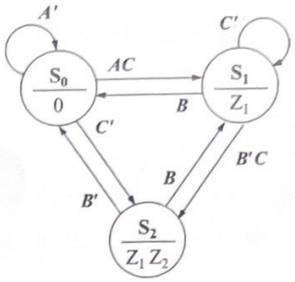
Show the truth table for deriving the inputs of the MUX and draw the block diagram of the function. Assume inputs and their complements are available.

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

C'
D
1
C



9. (10%) Show that the following state graph is a completely specified or an incompletely specified proper state graph or neither according to the two constraints on the input labels for every state. If it is incompletely specified, what conditions should be followed to make it a proper state graph?

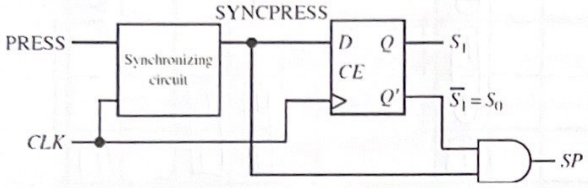


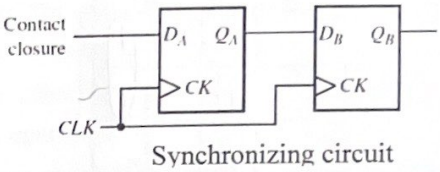
$I_1 I_2 = 0$ $I_1 + \cdots + I_n \neq 0$

$I_2 I_3 = 0$

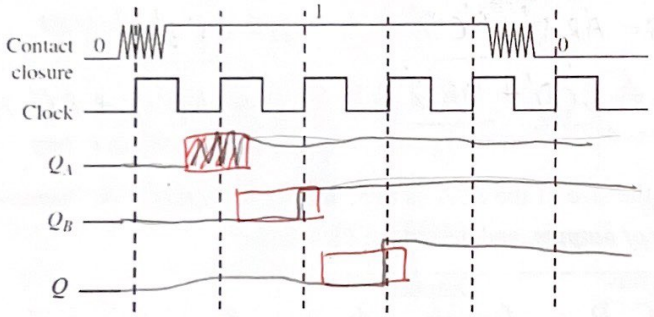
$\vdots$

10. *(6%) Given the single pulser with synchronizing circuit, complete the following timing waveform and explain the reason why two flip-flops are required for the synchronizing circuit.



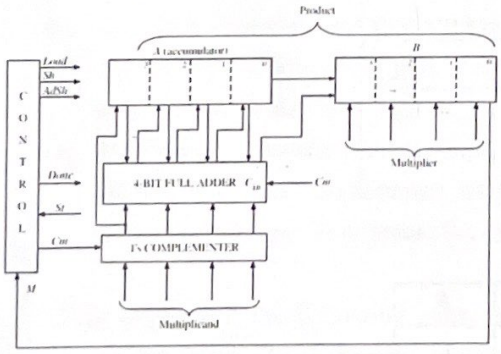


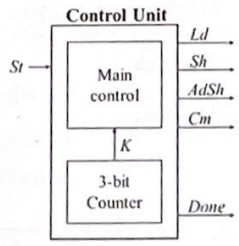
Synchronizing circuit



因為輸入不穩定的 debouncing time 若小於 2 clock period 則可使 output 穩定

11. (13%) 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 16-bit *signed* binary integers and generate a 32-bit product. Assume that the control unit of the circuit consists of a main control and a 4-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 15 (1111_2)
Done: completion signal

(a) (5%) Describe the procedure of the faster multiplication for signed numbers. What exception cases should be detected for this multiplier?

(b) (6%) Draw a state graph for the **Main control** of the 16×16 multiplier including the detection of exception. When a multiplication is completed, return to the start state. When an exception occurs, output signal ***Exception*** should be set to 1. Use as least states as possible.

(c) (2%) How many additions and subtractions are required to complete the following multiplication by using a similar 8×8 multiplier? Why?

i. 10110100 × 01011111 ii. 01111010 × 10110101