

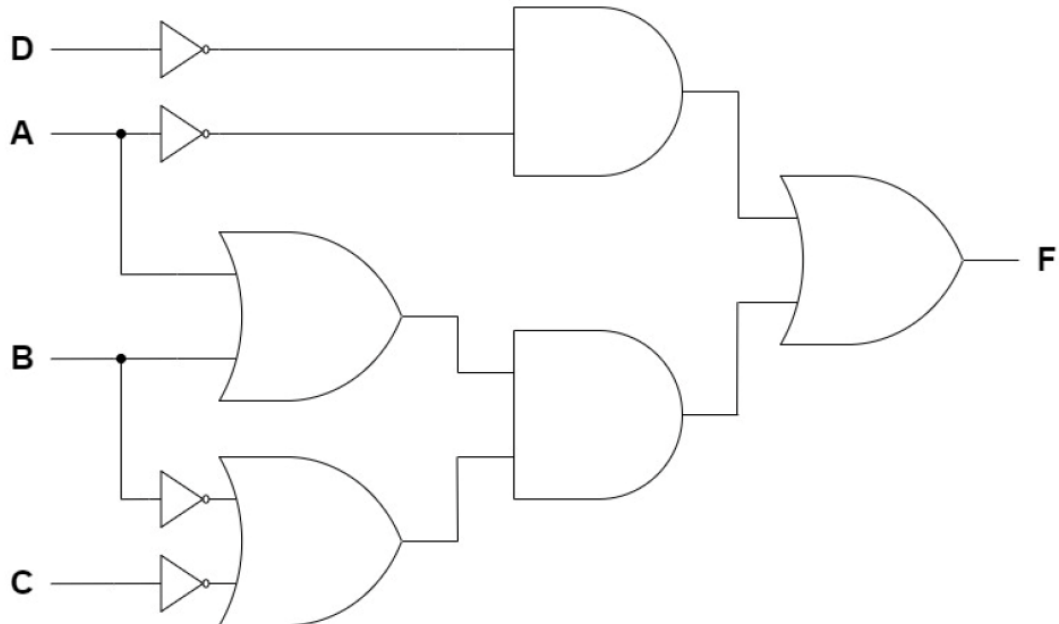
數位系統設計 HW1

學號: 0819823

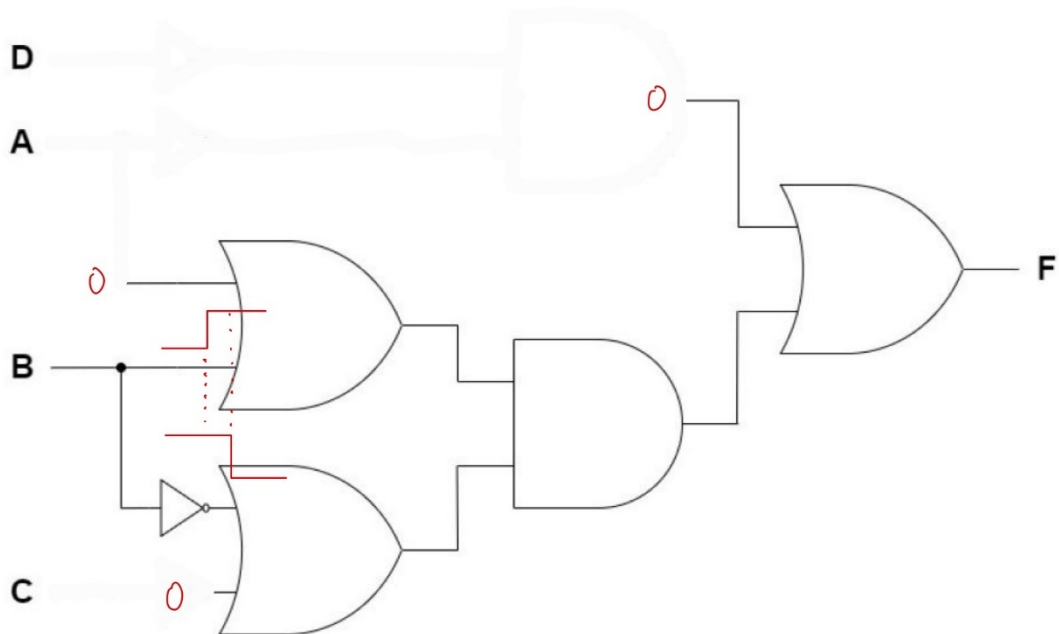
姓名: 陳子祈

A.

i. 此電路有 static 0-hazard 。

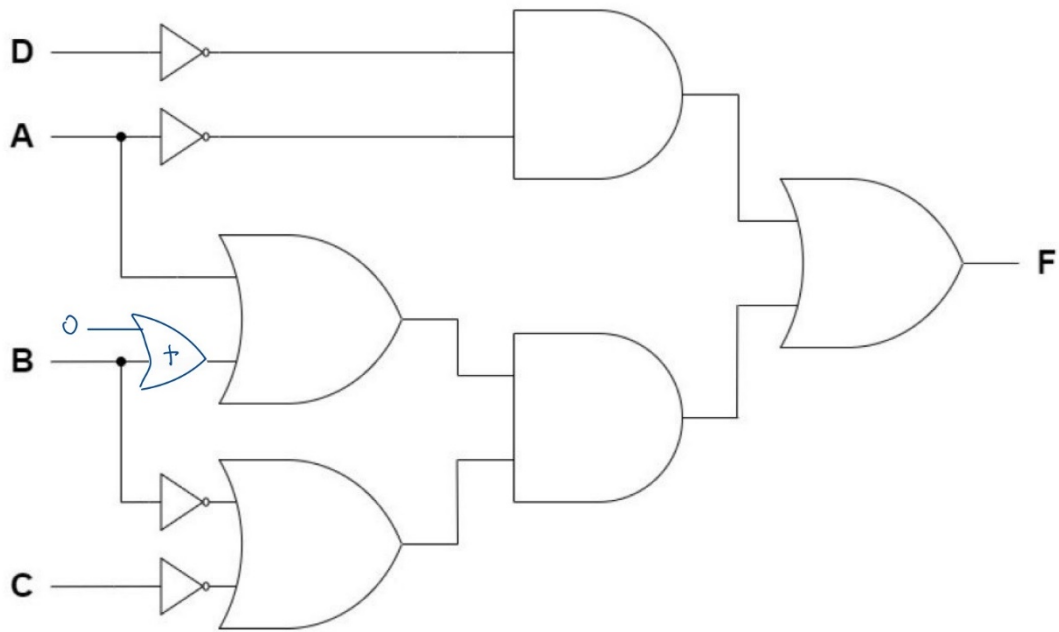


這是因為 B 與  $\sim B$  到輸出訊號有不同的 propagation delay，設定  $A=0$ 、 $C=1$ 、 $D=1$ ，則可把電路簡化為:



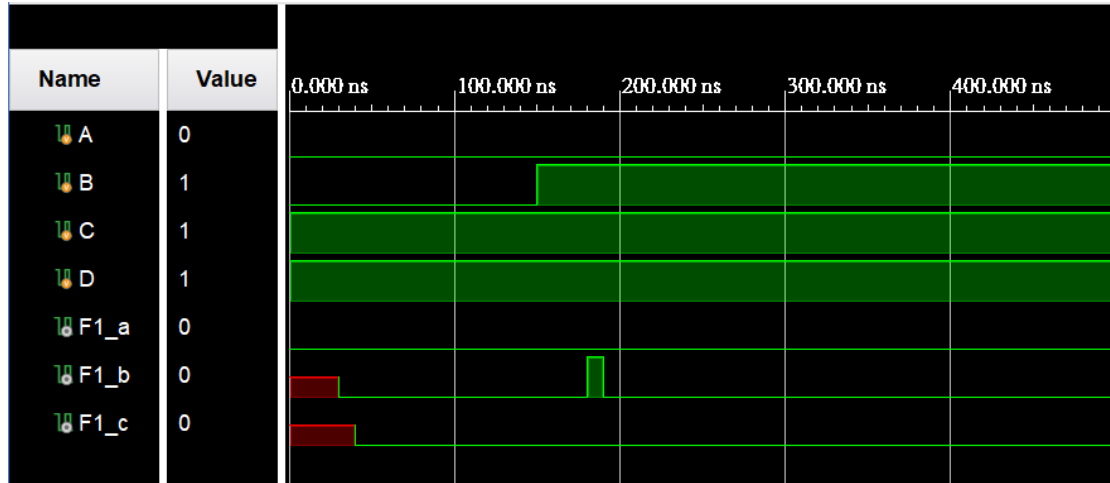
這樣可以更一目瞭然當  $B=0$  變成 1 的時候，會有 10ns 的時間兩邊的輸入訊號都是 1 則輸出訊號 F 是 1，其他時間只要一邊的輸入訊號有 0 則輸出訊號 F 就是

0，因此 0 hazard 的問題就會發生。



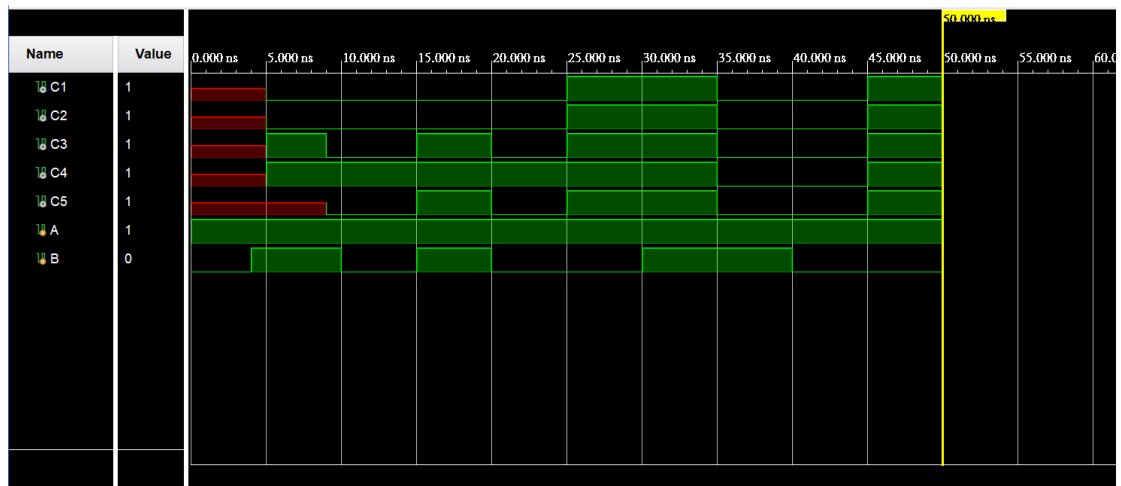
為了解決 0 hazard，可利用 OR gate 延長 B 輸入訊號的 propagation delay，如此一來就可消除 0 hazard 得到 hazard-free 電路，從下面所附的示波圖可觀察到。

ii. A(d)模擬結果之波形圖如下：



從示波圖可觀察到當 A、C、D 這些輸入訊號維持不變，B 這個輸入訊號從 0 變成 1 的時候，A(a)使用 dataflow 的寫法不會產生 hazard，但是 A(b)使用 gate level 的寫法就會產生 0 hazard，調整成 A(c)的 hazard-free 電路之後就沒有 0 hazard 了。

B.



Delay 分為 inertial delay 及 transport delay，C5 那行 statement 符合 inertial delay 的寫法，C3 那行 statement 符合 transport delay 的寫法，因此這兩個輸出都是正確的，因為 B 的 pulse 間隔都大於等於 5ns，因此 C3 及 C5 都是 B 反向經過 5ns 的 delay。C1、C2、C4 那幾行的 statement 都是不正確的寫法，因此這幾個輸出都是錯誤的。

C.

i. memory input equation 及 output equation 經過我的推算如下:

Present State	Next State		Output (DB)	
	X=0	X=1	X=0	X=1
S0	S1	S1	00	10
S1	S2	S3	10	00
S2	S4	S5	10	00
S3	S5	S5	00	10
S4	S0	S0	11	00
S5	S0	S0	00	10

Present state	Next state		Output DB	
	X=0	X=1	X=0	X=1
0 0 0	0 0 1	0 0 1	0 0	1 0
0 0 1	0 1 0	0 1 1	1 0	0 0
0 1 0	1 0 0	1 0 1	1 0	0 0
0 1 1	1 0 1	1 0 1	0 0	1 0
1 0 0	0 0 0	0 0 0	1 1	0 0
1 0 1	0 0 0	0 0 0	0 0	1 0
1 1 0	X X X	X X X	X X	X X
1 1 1	X X X	X X X	X X	X X

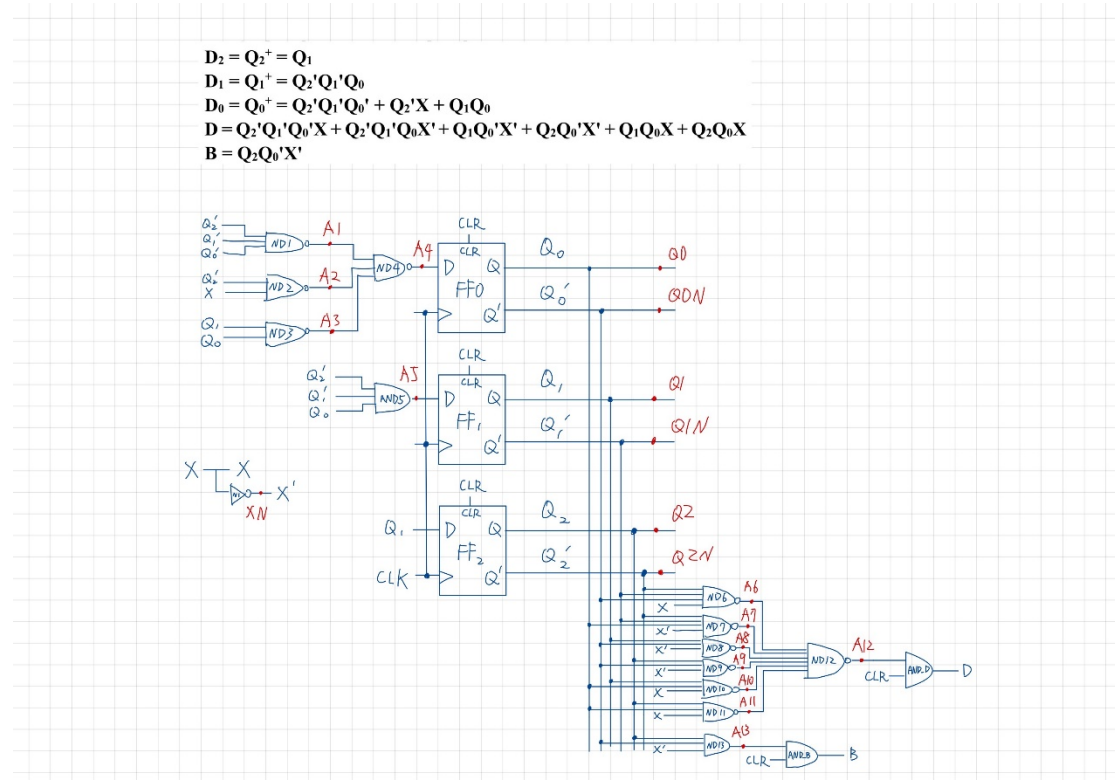
$D_2 = Q_2^+ = Q_1$   
 $D_1 = Q_1^+ = Q_2'Q_1'Q_0$   
 $D_0 = Q_0^+ = Q_2'Q_1'Q_0' + Q_2'X + Q_1Q_0$   
 $B = Q_2Q_0'X'$

$D = (X'Q_2Q_0' + XQ_2'Q_1'Q_0' + X'Q_2'Q_1'Q_0 + XQ_2Q_0) \& CLR$   
 $B = X'Q_2Q_0' \& CLR$

$D_2 = Q_2^+ = Q_1$   
 $D_1 = Q_1^+ = Q_2'Q_1'Q_0$   
 $D_0 = Q_0^+ = Q_2'Q_1'Q_0' + XQ_2' + Q_1Q_0$

我推算的結果與本題提供的 equation 一樣無誤，不過因為 CLR 若為 0，output 應該都是 0，因此 output equation 加上 and CLR。

ii. 此電路的 logic circuit diagram 如下(包含各 gate、D FF 的編號):



iii. C(d)的波形圖如下:



Behavioral(D1\_3\_a、B1\_3\_a)、Dataflow(D1\_3\_b、B1\_3\_b)、Structural(D1\_3\_c、B1\_3\_c)電路的波形圖都正確。

另外我試著把 CLR 每隔 35ns 反向，看看輸出會不會在 CLR=0 時清除。



在 CLR=0 時各 output 都有被清為 0。

#### D. 心得

第一次的作業量真的非常多，我第一次寫這麼多作業題數的 code 作業，而且還需要對電路的 hazard、verilog 的 delay 等熟悉才能知道一些模擬電路的細節。我上學期計組沒有學太多 verilog 的細節需要注意的地方，這門課教了非常多模擬電路需要遵守的法則及需要遵守的原因，讓我獲益良多。像是有問助教最後一題 CLR 的部分要怎麼寫，助教提醒我不要把 output equation 寫到 `always @(posedge CLK or negedge CLR)` 裡面，應該要把 combinational circuit 另外寫在 `always (*)` 裡面，才不會把 sequential circuit 跟 combinational circuit 混在一起，然後在 combinational circuit 的 `always` 裡面判斷 CLR 是否為 0，如果是 0 輸出就強制為 0，這樣寫就不會有問題了。我也問助教可不可以用 `assign` 來寫 combinational circuit，最後的輸出再 & CLR，助教說可以，如果是簡單的電路可以這樣做。