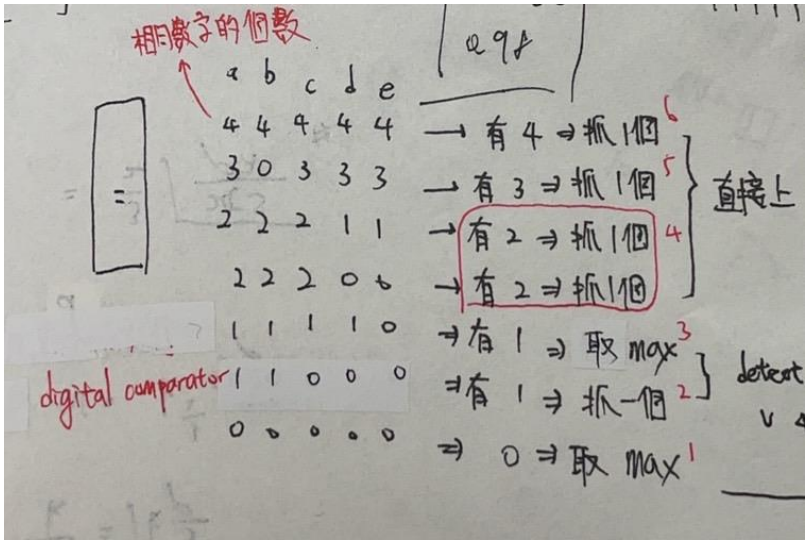
積體電路設計 HW3

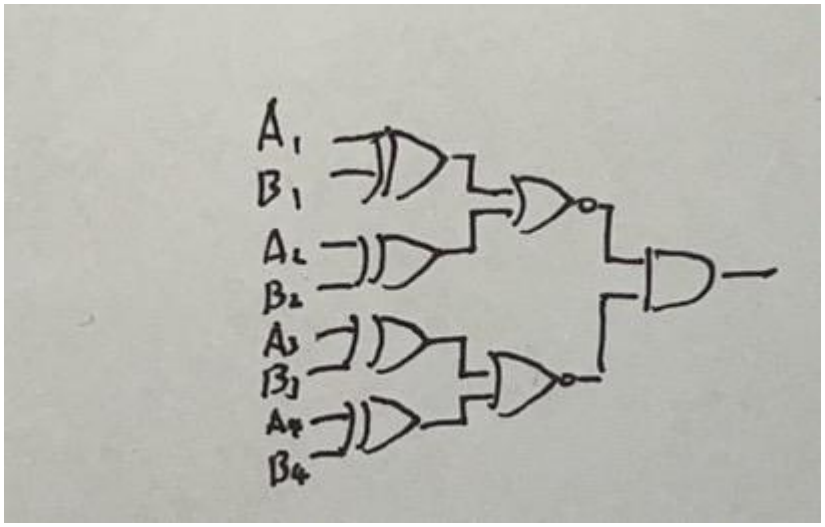
B07901087 電機四 蔡沛洹

我將電路分做 6 cases，我接下來將對每一個 case 做解釋與附上 circuit

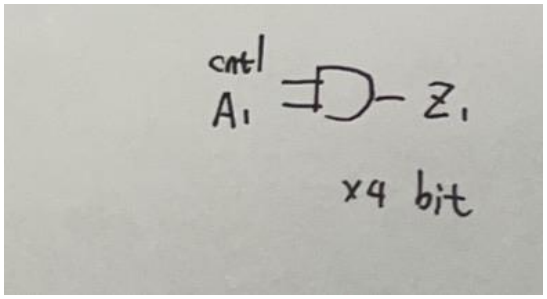


主要有兩個電路：

首先我們必須要有 equaler(equaler)，也就是比較兩數是否相同的電路。



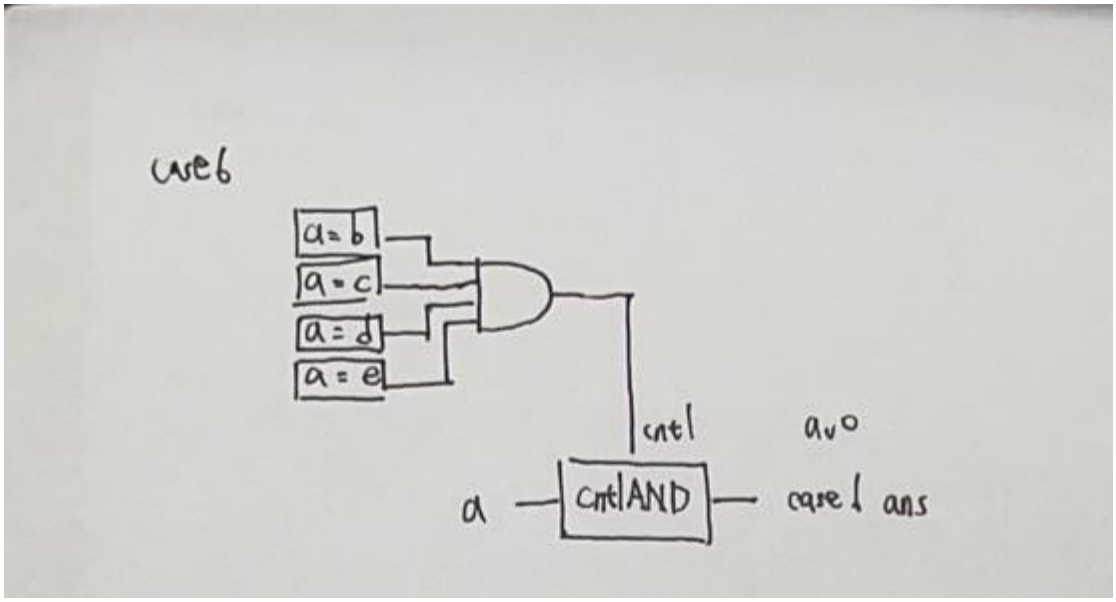
然後我們必須要有過濾器(cntlAND)，也就是當 case 的條件符合再將答案送出，否則就輸出 0



題目給 5 個數(我令為 a, b, c, d, e)，Equaler 共要用 c5 取 2 次，也就是 10 個，這樣我們就有所有數字的相等關係，接下來就用這 10 個結果去做以下的 case 的電路，其中一些較基本的 gate 我就忽略了！剩下的小 block 我放在 appendix。

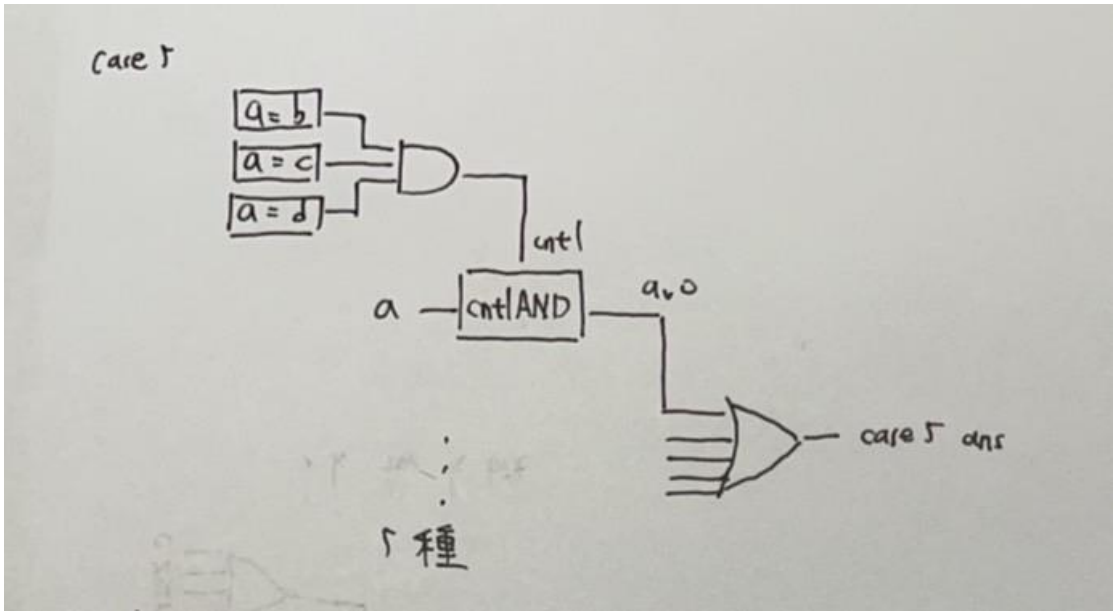
Case6 全部的數字都相同

這裡是五個數字都相同的 case，所以取 a=b, a=c, a=d, a=e 做 AND，即可判斷是否為 case6，然後再用 cntlAND 送出 case6 這邊的答案即可(若不成立就送 0，剩下的 case 也都是如此)



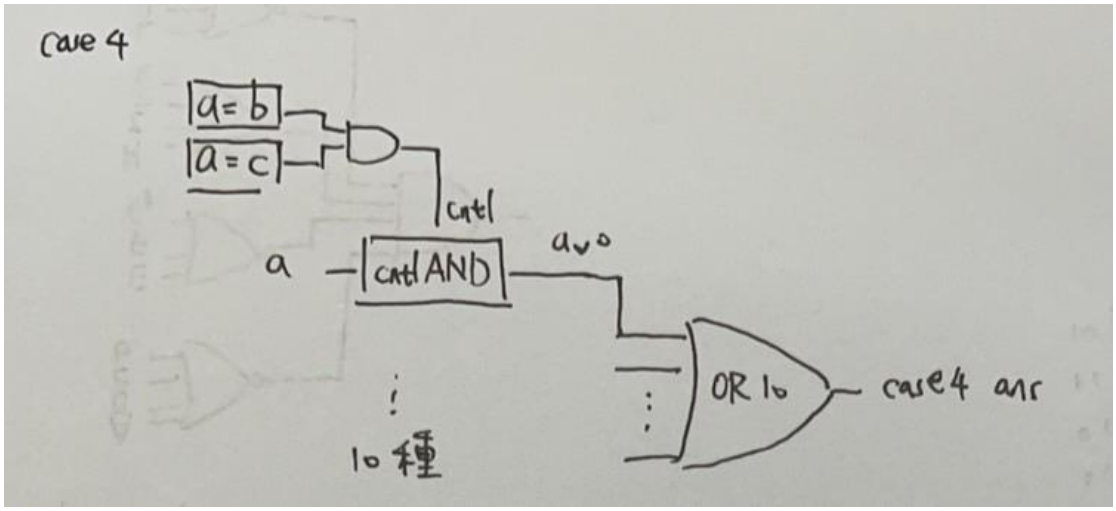
Case5 4 個數字相同

這邊就考慮五種情況，看是哪一個數字不同，以(k1, k1, k1, k1, k2)為例，就取 a=b, a=c, a=d 做 AND，以此 control signal 去和 a 做 cntlAND，輸出一個 subanswer，在這個 case 的最後，再用一個 OR5 把所有 subanswer 包起來

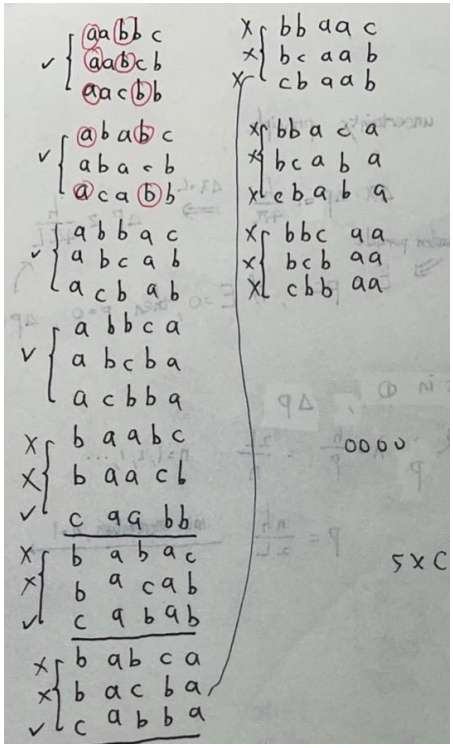


Case4　3 個數字相同

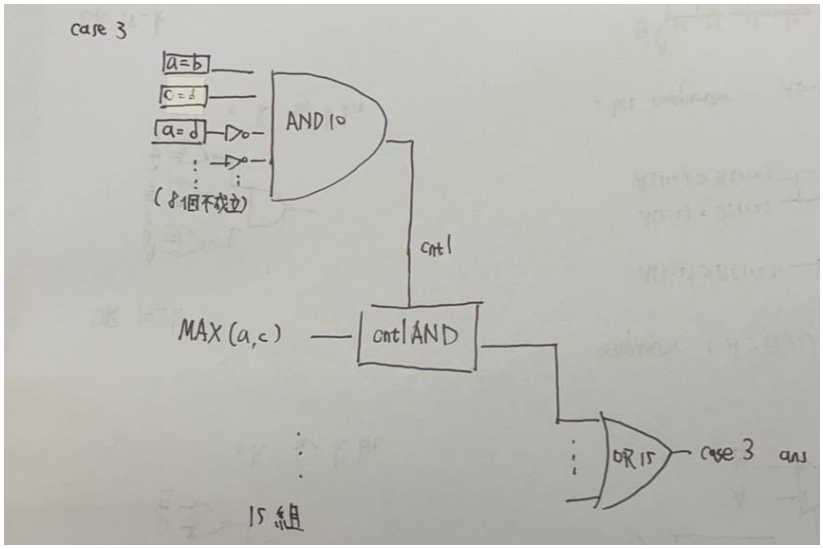
這邊需要考慮 10 個情況，也就是 c5 取 3，看哪三個數字相同。舉其中一個例子: (k1, k1, k1, k2, k3)，這邊我們就拿 a=b, a=c 做 AND，然後以此為 control signal 和 a 做 cntlAND，輸出 subanswer。在最後，我們再用一個 OR10 把所有 subanswer 包起來



Case3　2 個數字相同，且有另兩個數字也相同(也就是有兩組)

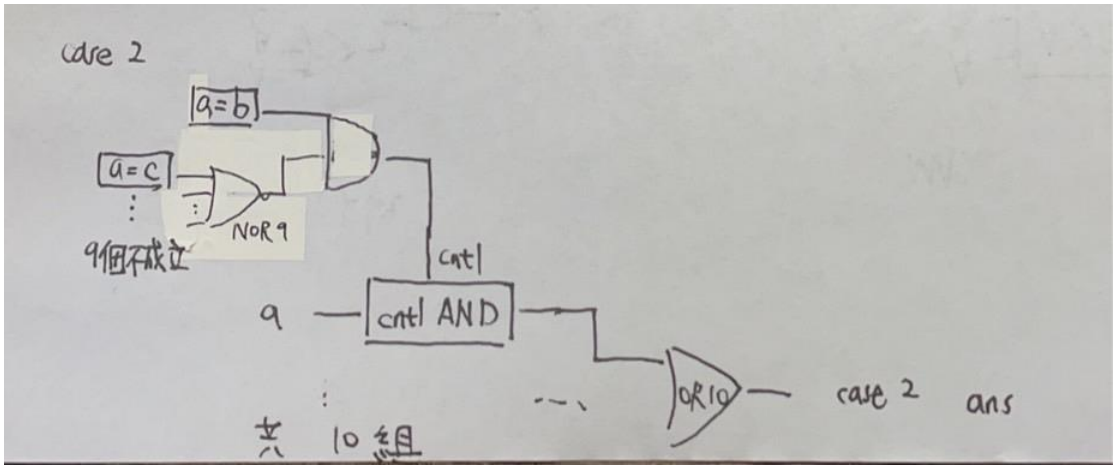


在這個情況下有 15 個 cases，也舉其中一個(k1, k1, k2, k2, k3)為例，我們需要拿 a=b, c=d 和剩下的所有 equaler 的結果的 bar 去做 AND10 (也就是在十個等號中只有兩個等號成立)，以此 control signal 去和 max(a, c)做 cntlAND，在最後用 OR15 把所有 subanswer 合起來



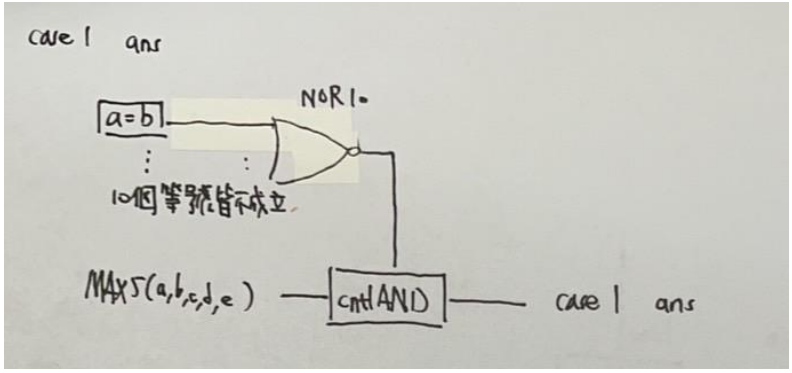
Case2　2 個數字相同

這個情況下有 10 個 cases，也就是 c10 取 1，十個等號中只有一個等號成立，我在這裡的作法是先對不成立的 9 個等號去做 NOR9，再拿此結果和等號成立的 signal 去做 AND2，以此做為 control signal 去和等號成立的數字做 cntlAND。最後用 OR10 把所有 subanswer 合起來。

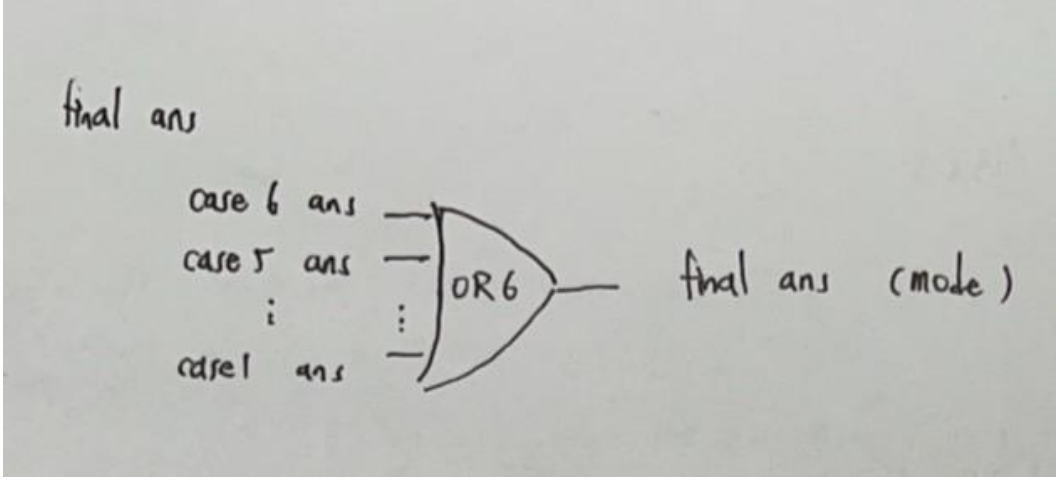


Case1　沒有數字相同

做一個 NOR10 確定所有等號皆不成立，以此 control signal 去控制 MAX5 (也就是 max(a, b, c, d, e))是否要輸出。



在最後使用 OR6 把所有 case 的答案 OR 起來，得到電路的最終答案！



Discussion: critical path 的尋找

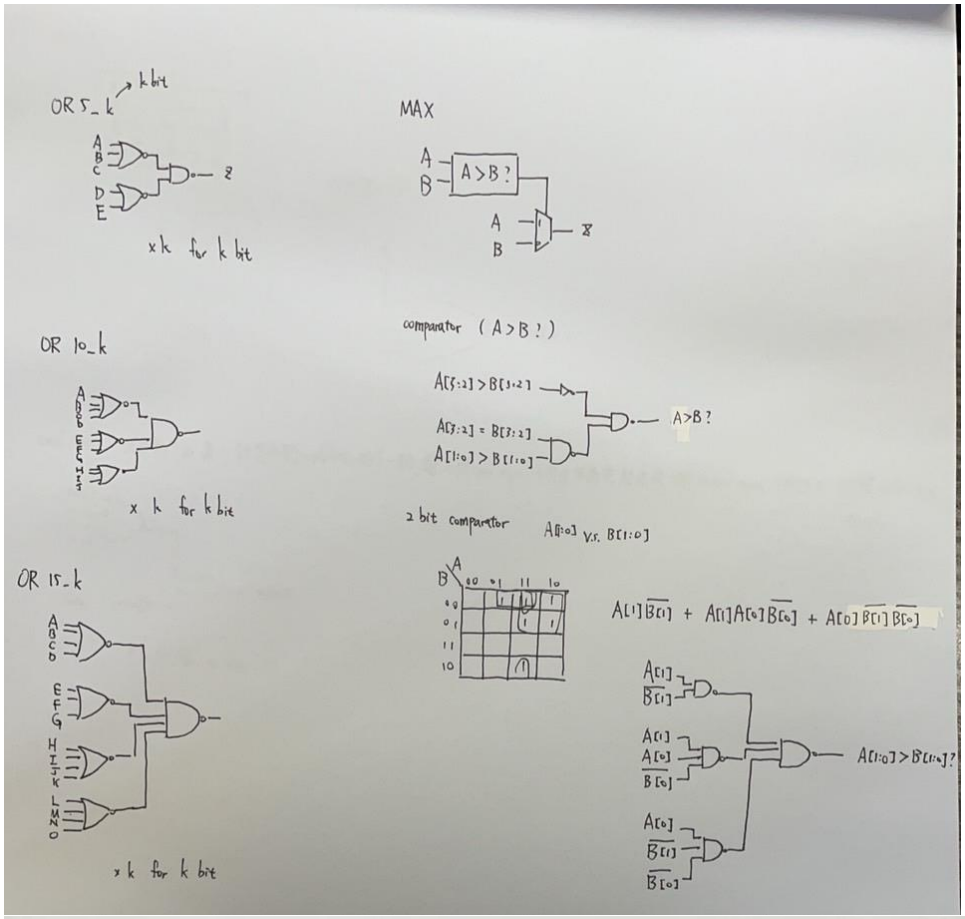
我做出的第一版電路是 4ns，離 3 差了一點，我先去看 tb_mode.v

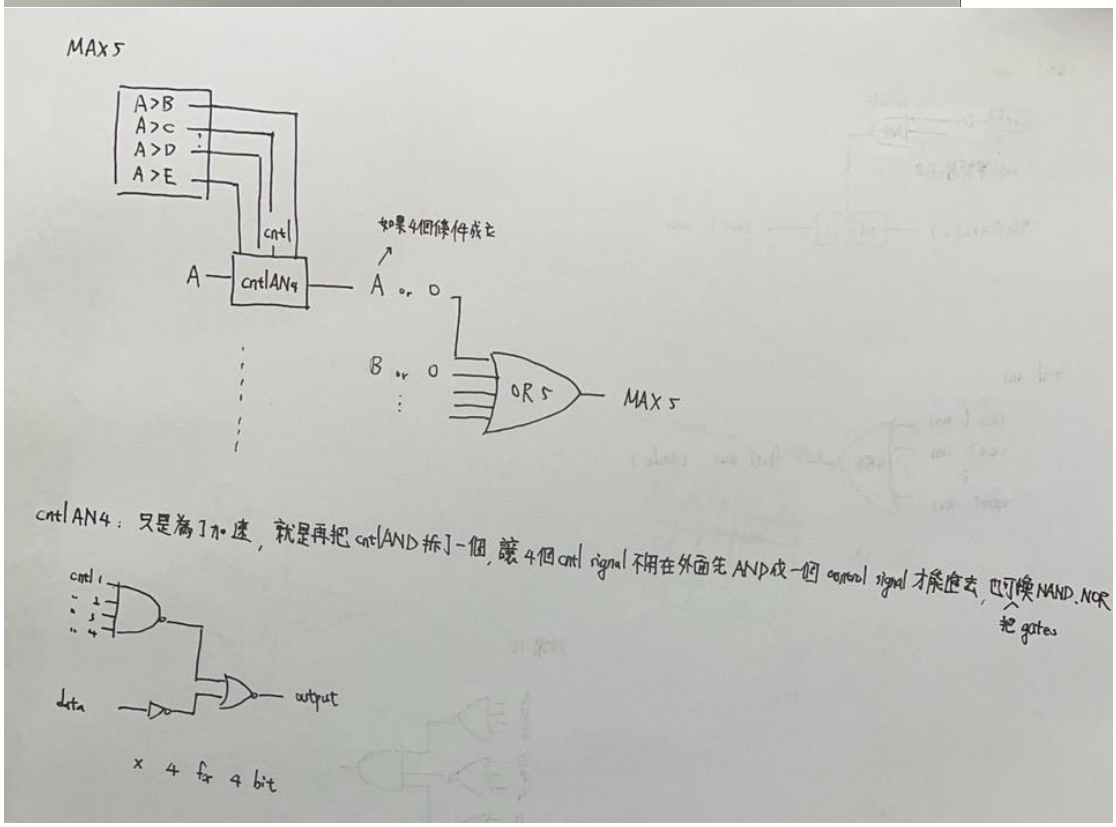
```
70      always begin
71          #3
72          if(ans!==mode)
73              err3 = err3 + 1;
74          #1
75          if(ans!==mode)
76              err4 = err4 + 1;
77          #1
78          if(ans!==mode)
79              err5 = err5 + 1;
80          #1
81          if(ans!==mode)
82              err6 = err6 + 1;
83          #1
84          if(ans!==mode)
85              err7 = err7 + 1;
86          #1
87          if(ans!==mode)
88              err10 = err10 + 1;
89          #12
90          if(ans!==mode)
91              err20 = err20 + 1;
92          #1
93          i = i + 1;
94      end
```

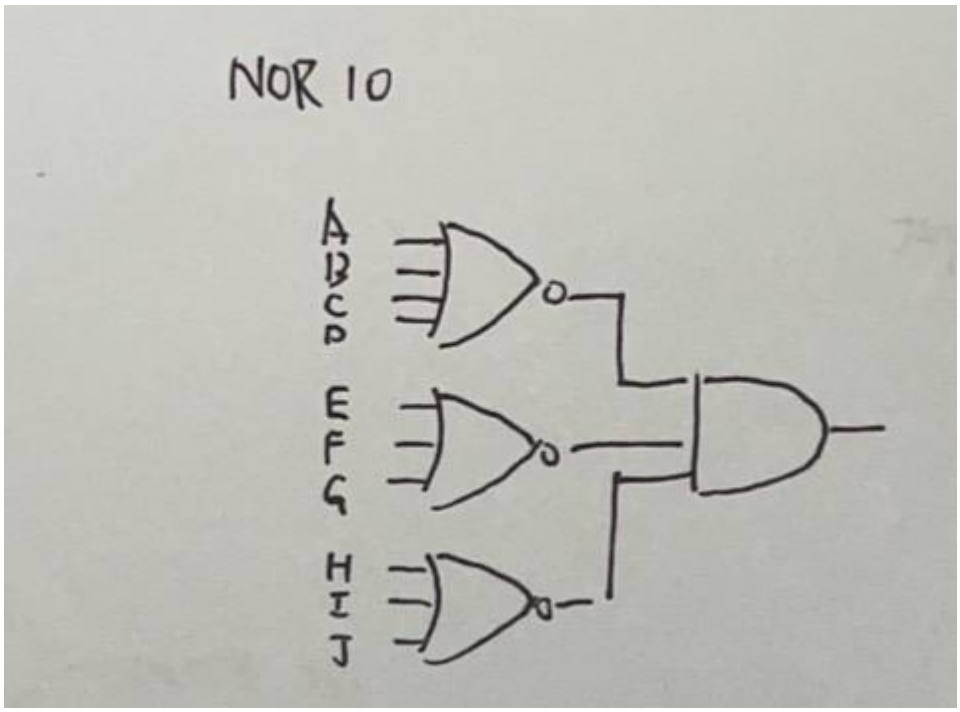
發現這邊似乎就是輸出 error 的地方，所以我在 err3 裡面加一個 display 把 i 印出來，發現 i 都在 4000~6000，沒記錯的話應該都是 case3，而我 case3 原本是會先把 15 個 case 做 grouping 再輸出，這種做法似乎會讓 gate 的深度增加，再加上我覺得增加 input 對 delay 的影響是小於 gate 的，也就是說為了用 OR2、OR3 而先把電路做 subgroup，不如用 4-input 的 gate，而且只要兩層就能支援 16 input，所以我稍微改了一下，直接在 case3 輸出 15 個 subanswer 做 OR15，就讓 critical path < 3 了！

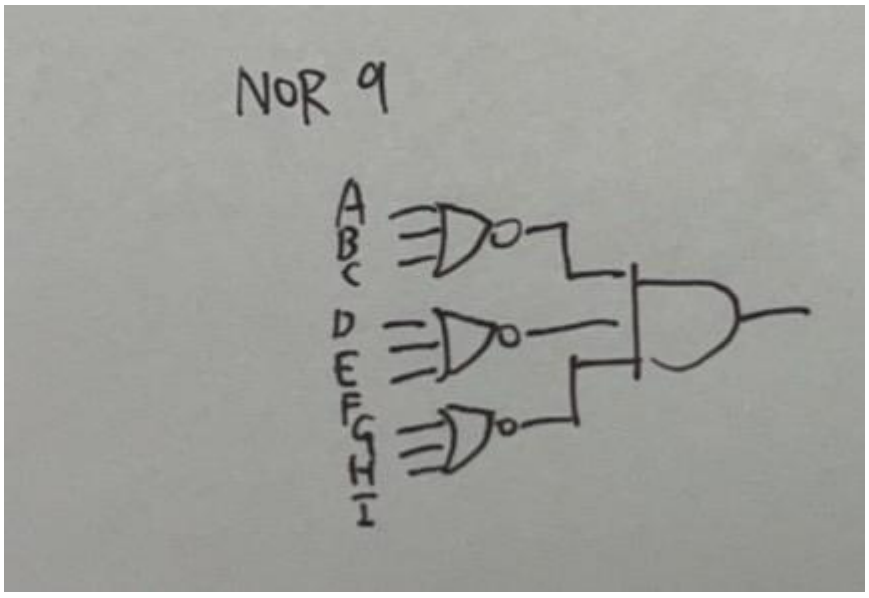
APPENDIX :

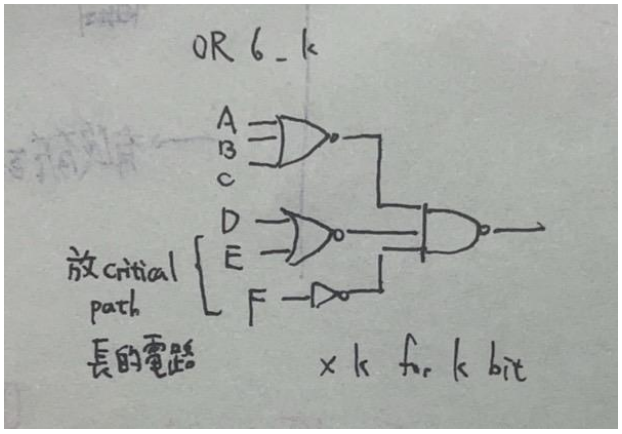
基本電路圖






NOR 10
A
B
C
D
E
F
G
H
I
J


NOR 9
A
B
C
D
E
F
G
H
I


OR 6_k
A
B
C
D
E
F
放 critical path 長的電路
x k for k bit