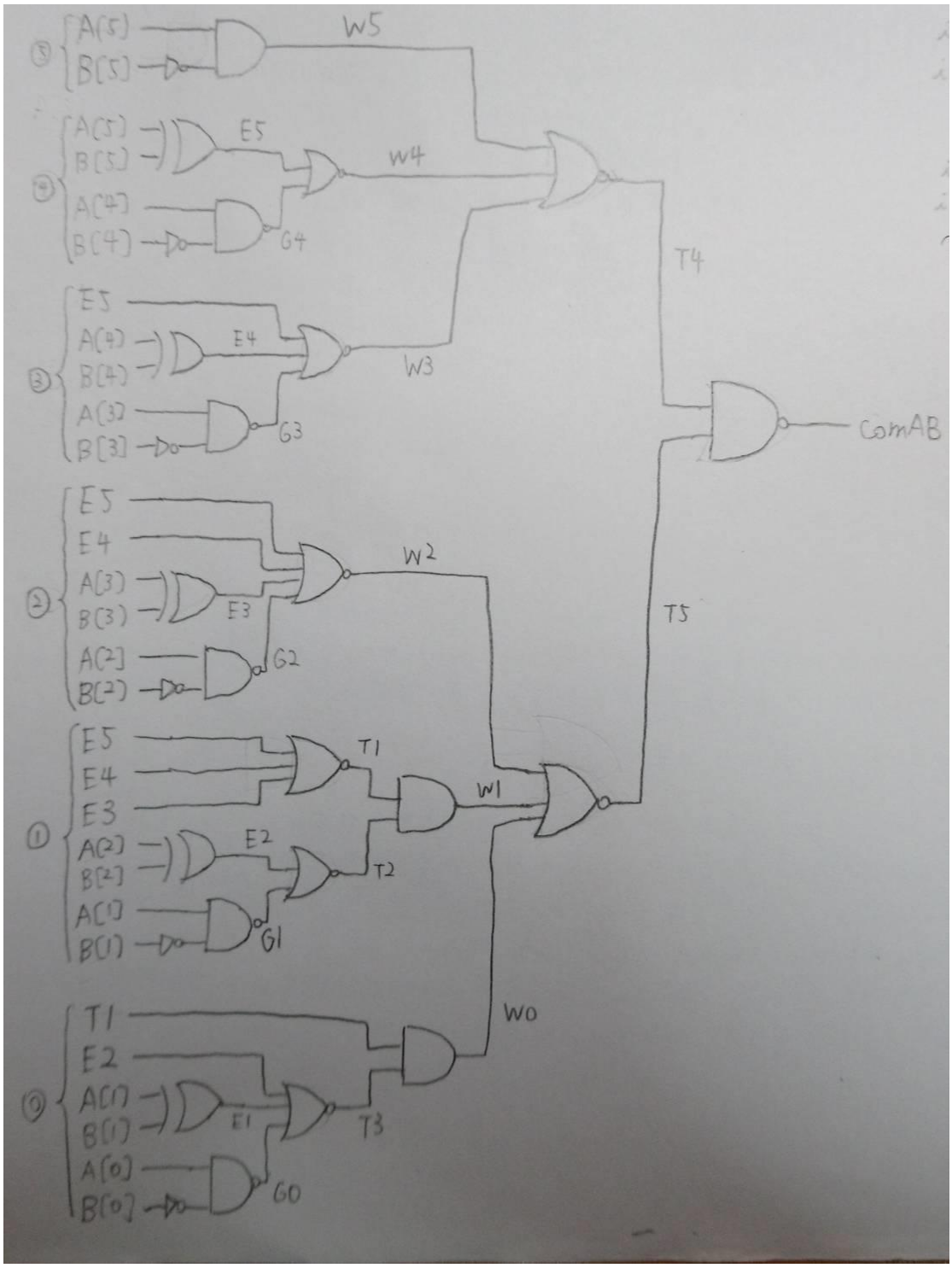
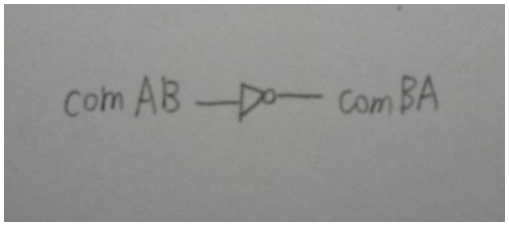
學號:B06902136
系級:資工四
姓名:賴冠毓

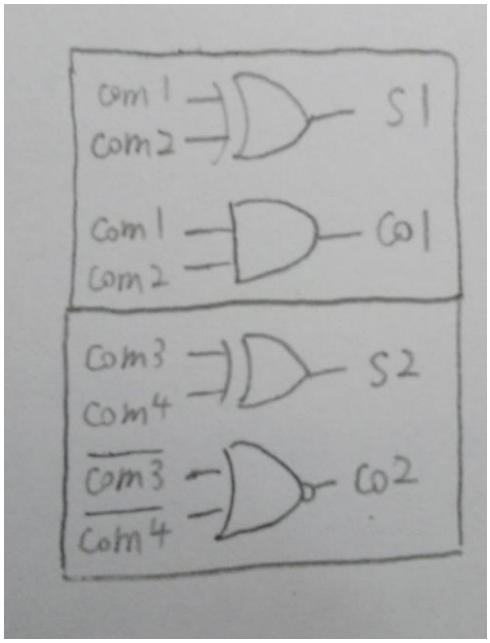
1. Circuit diagram
(有些電路功能相同，但由於 bubble pushing 後取最佳狀況，所以有不同樣貌)
6-bit unsigned comparator: 10 個



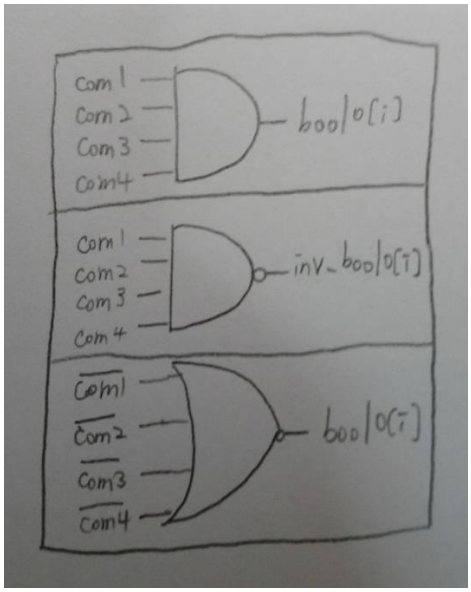
ivcom: com 前後相反的結果(也是 10 個)



half adder: 一組 2 個，共 5 組
如果至少有一個 comij 的 i<j，為上
如果兩個 comij 的 i>j，則為下

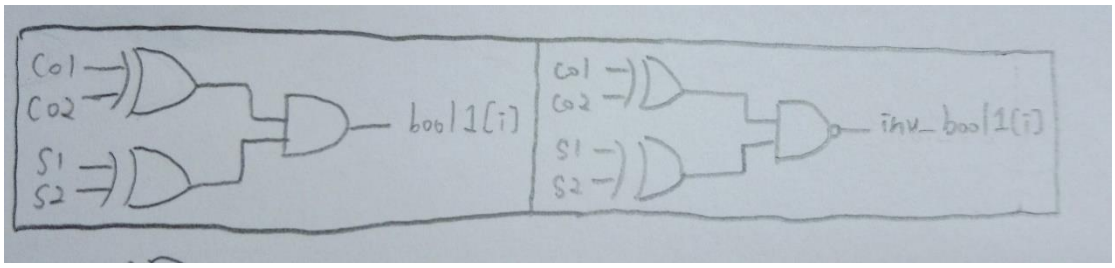


Choose rank0: 計算 bool0
i=0 為上，i=1、2、3 為中，i=4 為下



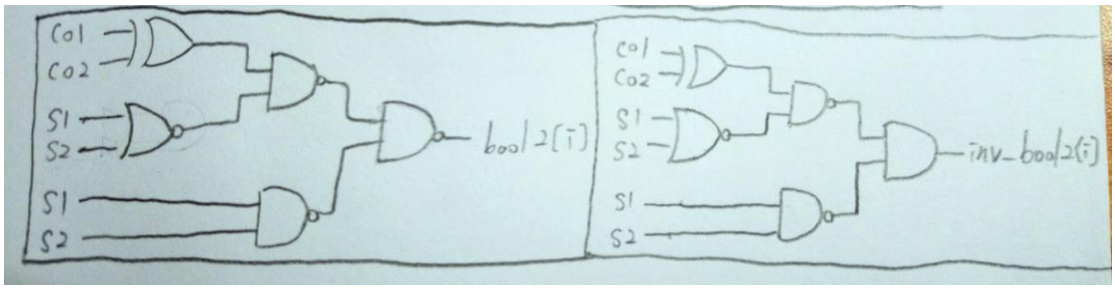
Choose rank1: 計算 bool1

i=0、4 為左，i=1、2、3 為右



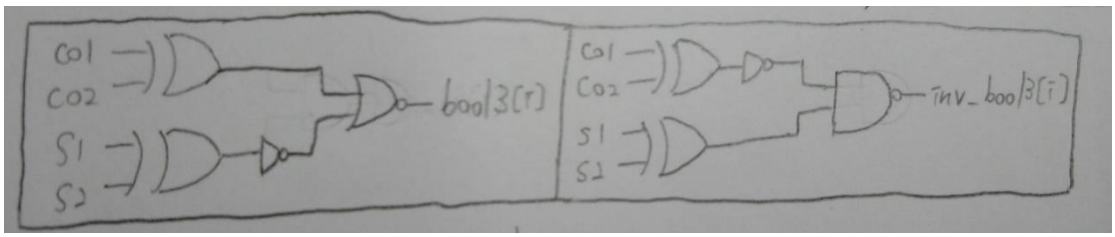
Choose rank2: 計算 bool2

i=0、4 為左，i=1、2、3 為右



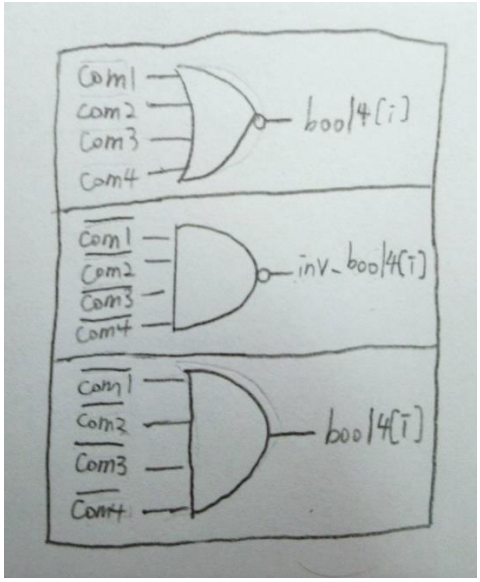
Choose rank3: 計算 bool3

i=0、4 為左，i=1、2、3 為右

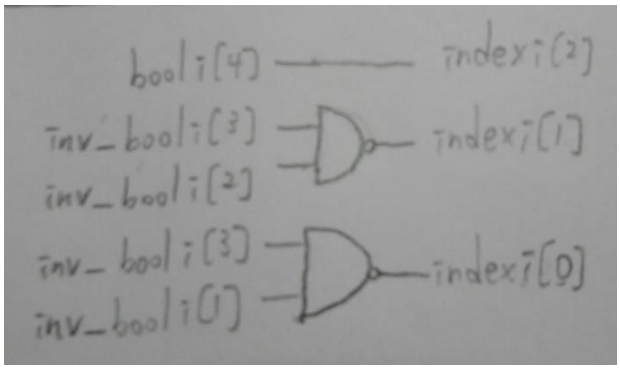


Choose rank4: 計算 bool4

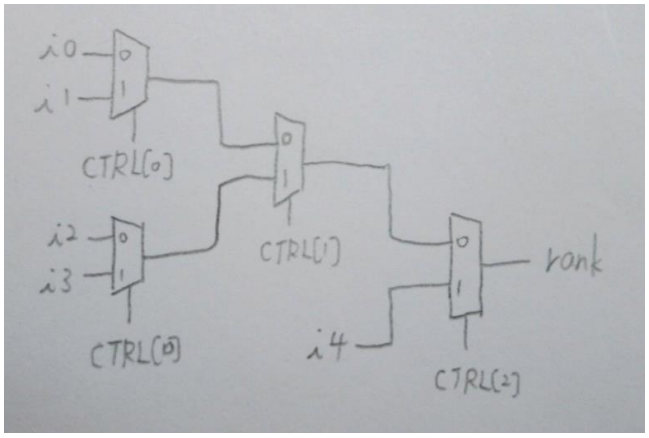
i=0 為上，i=1、2、3 為中，i=4 為下



把 bool 轉成 index: 5 個，一個 bool 配一個



用 index 選數字: 一組 6 個(1 bit 1 個)，共 5 組

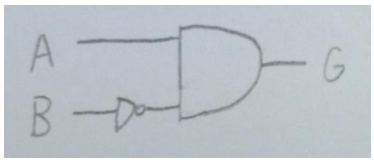


2. Discussion

(1) how I sort these numbers

先做出兩個數字的比較器，從 1-bit 開始:

A	B	G(A>B)
0	0	0
0	1	0
1	0	1
1	1	0



2-bit unsigned comparator: (2 位數:10)

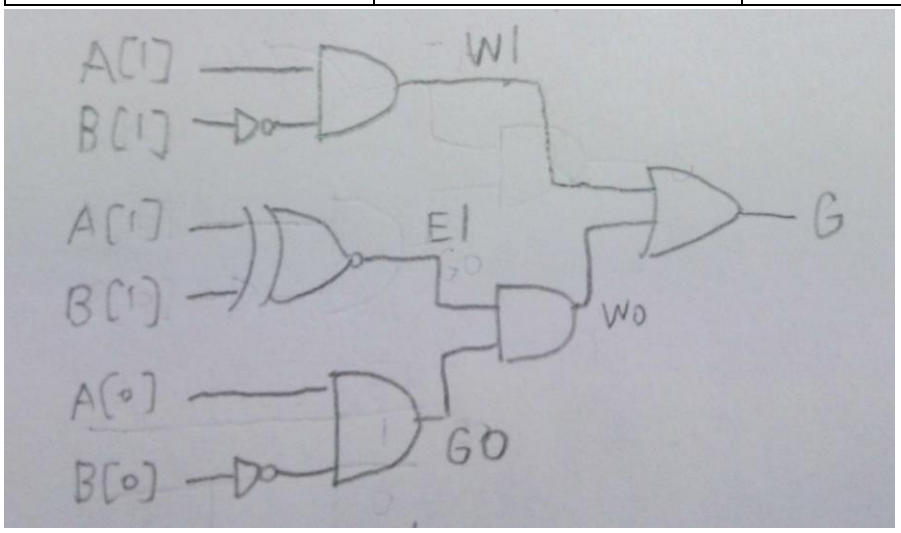
可分為兩種情況:

Case 1: 在首位數就比出大小

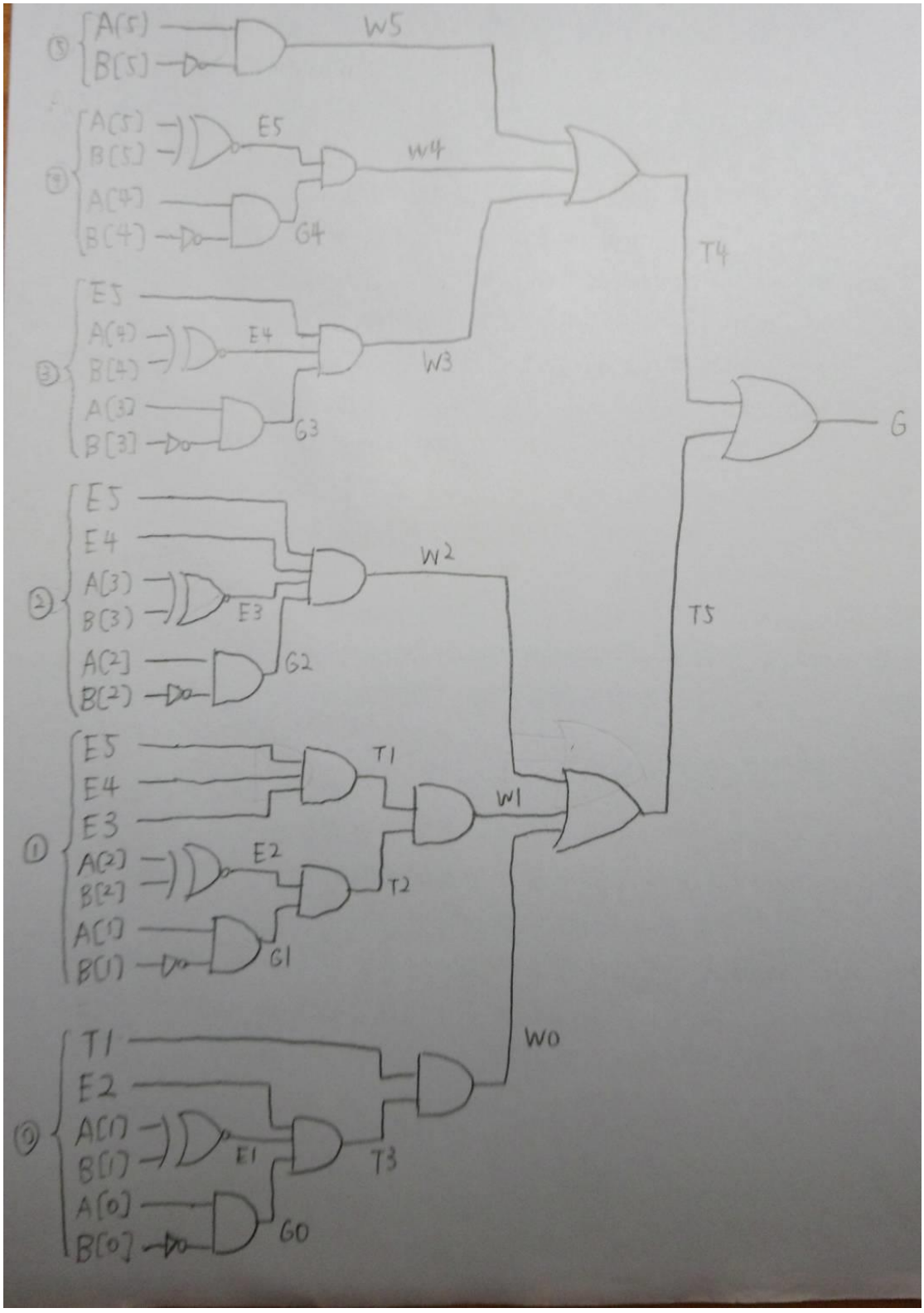
Case 2: 首位數相等，在末位數才比出大小

A[i]	B[i]	E[i](A[i]=B[i])
0	0	1
0	1	0

1	0	0
1	1	1

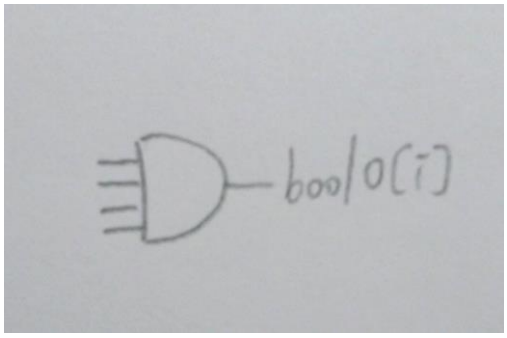


由上 2 個例子，可推導出 6-bit unsigned comparator: (6 位數:543210)
總共 6 個 case，用 MSD radix sort 看兩個數在第幾位判斷出大小。
(G 表示 A[i]>B[i]；E 表示 A[i]=B[i]；W 表示在第 i 位判斷出大小)

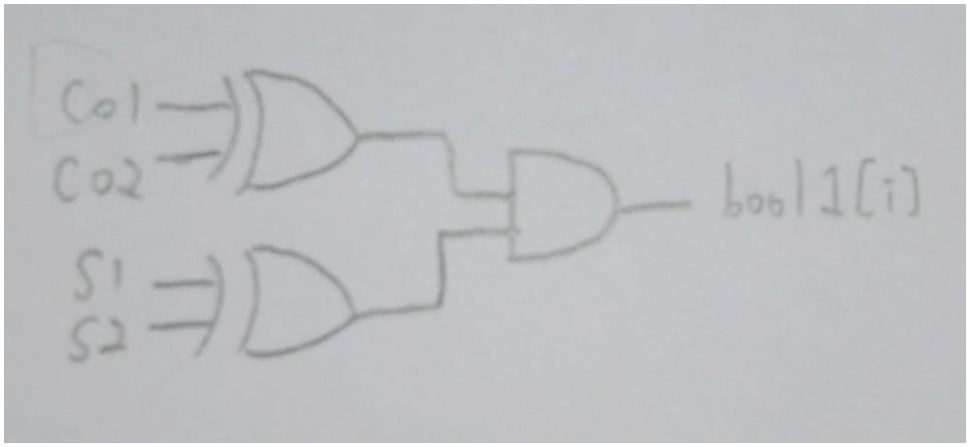


再來透過比較器比較出兩兩數字的結果，記錄在 com 裡。comij 表示第 i 個數跟第 j 個數的比較結果，若 comij=1，表示第 i 個數大於第 j 個數，反之則小於等於。而 comji 為 comij 相反的結果，透過這個方法可以處理兩個數字相等的情況，硬是假定其中一個大，如此一來巧妙避開了之後選數字可能重複選取的問題。

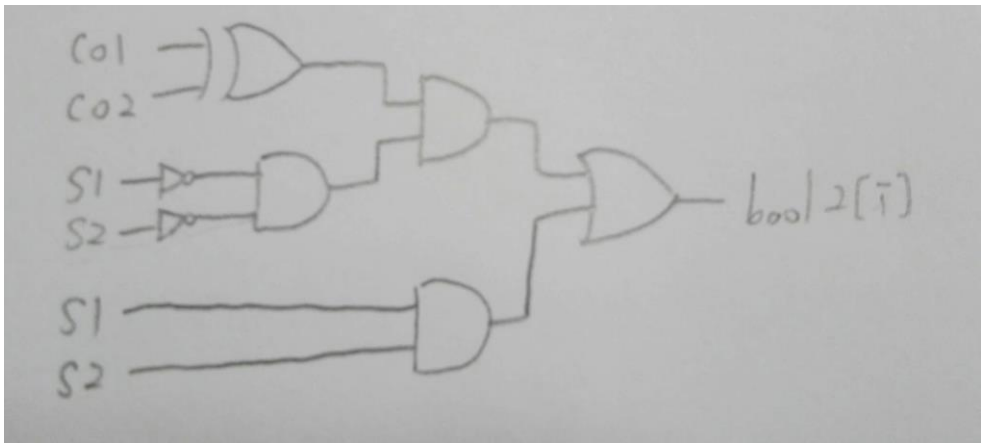
之後每個數總共會有 4 個比較結果，表示它跟其他數比較完的結果，這 4 個結果可以判斷該數字應該要在哪個 rank。我們先將這些數字平行化處理，把每個數字的 4 個結果兩兩用 half adder 相加，得到 co1 s1/co2 s2 這組數字(總共 5 組，每個數各有一組)。舉例來說如果第 i 個數的比較結果為 1100，那分解成 11/00，相加得到 co1 s1/co2 s2=10/00。
再來我們用 5 個 5 位數 bool，判斷每個 rank 我們要選哪個數字，每個 bool 只會有 1 個 bit 為 1(表示我們要選那個數)。舉例來說若 booli=01000，表示 rank i 的數字為 i3。inv_bool 則為 bool 相反的結果。
我們開始計算每個 rank 各自的 bool:
Choose rank0: 他的比較結果為 1111(大於全部的數)
第 i 個數所有比較結果 AND 起來就是 bool0[i]。



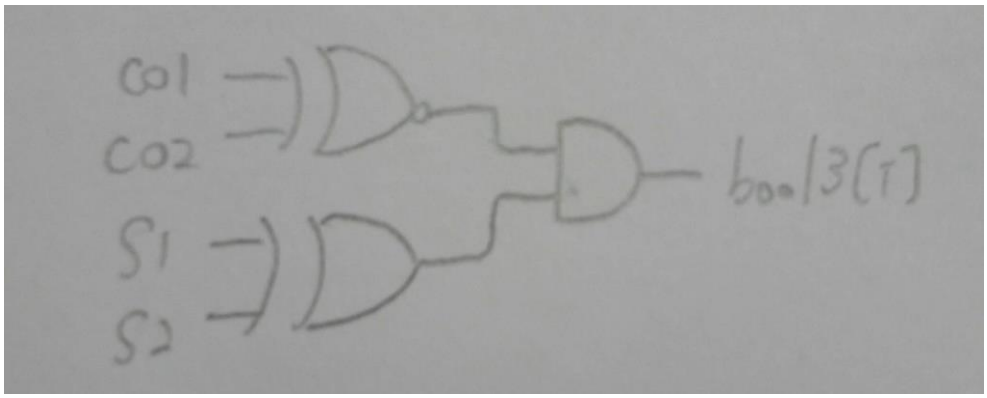
Choose rank1: co1 s1/co2 s2=10/01 or 01/10(大於 3 個數)
運用前面的算好的每組 co1 s1/co2 s2，如果符合 10/01 or 01/10 就 output1。



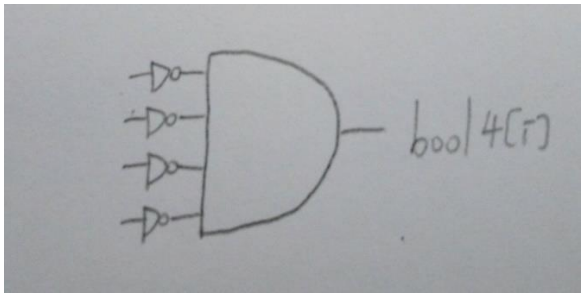
Choose rank2: co1 s1/co2 s2=10/00 or 01/01 or 00/10(中位數)
運用前面的算好的每組 co1 s1/co2 s2，如果符合 10/00 or 01/01 or 00/10 就 output1。



Choose rank3: co1 s1/co2 s2=01/00 or 00/01(大於 1 個數)
運用前面的算好的每組 co1 s1/co2 s2，如果符合 01/00 or 00/01 就 output1。



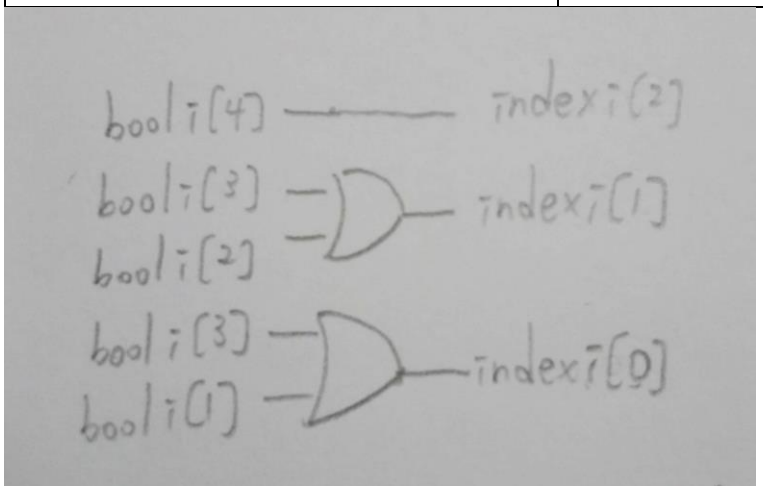
Choose rank4: 他的比較結果為 0000(沒有大於任何數)
第 i 個數所有比較結果相反 AND 起來就是 bool4[i]。



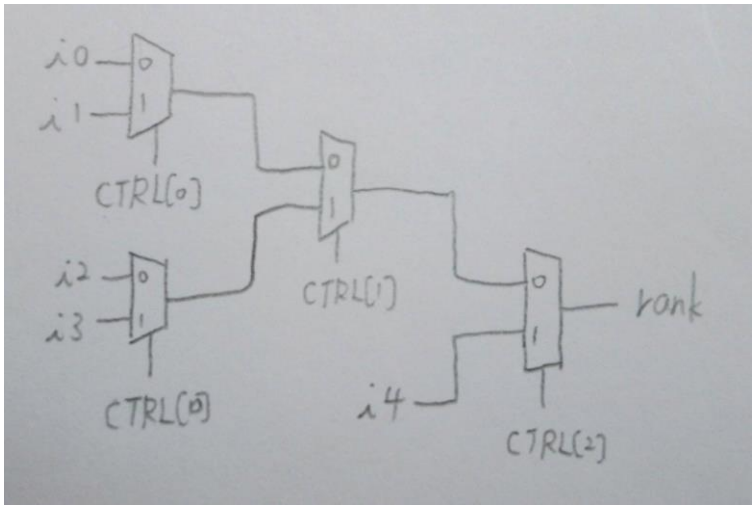
然後我們要將 5 個 5 位數 bool 轉成 5 個 3 位數 index，表示每個 rank 我們要選的數字的 index。舉例來說若 booli=01000，表示 rank i 的數字為 i3，那 indexi 為 011。

bool	index
10000	100
01000	011
00100	010

00010	001
00001	000



最後以 index 作為 CTRL，用 mux 選出正確 rank 的數字。



(2) how I improve my critical path

將整個電路的邏輯閘彼此搭配，盡量透過 bubble pushing 換成 NAND、NOR、INV 這種 delay 比較少的 gate，特別將 XNOR 這個 delay 超多的邏輯閘換成 XOR + INV 節省大量時間，同時也把 half adder 換成原始的 XOR 跟 AND 加速。

另外也盡量做平行處理，像最後選數字用 mux 不要一次選一個疊上去，而是根據 index 位數同時做選擇。而有多 input 的 gate 就用，不做多層邏輯閘的疊加。不過這樣還是只從 below 6 降到 below 4 而已 QQ