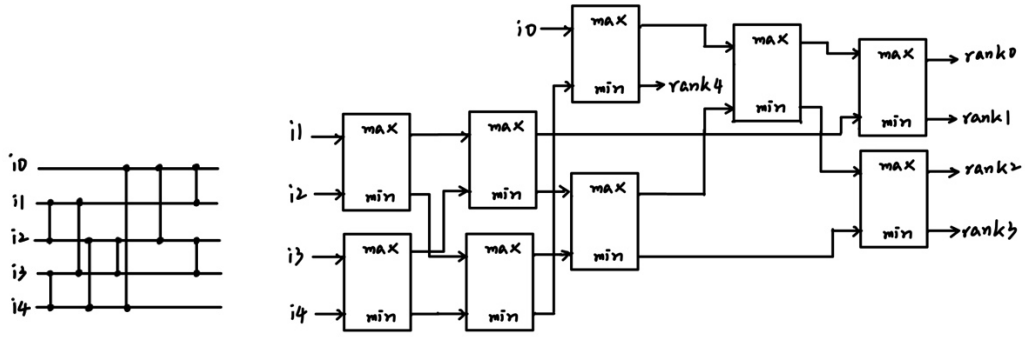
電機四 孟妍 b06901066

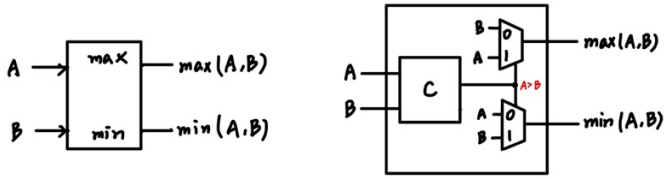
HW3 Report

1. Discussion of circuit design

本來用了類似 merge sort 的方法如下，以演算法角度來看的話應該是不要比較太多次比較好，但對於電路設計，因為有些比較不能平行的做，因此 critical path 只能壓在 6ns 以下。



maxmin circuit 設計如下：



剛剛的做法問題主要是兩個數字比較使用的 comparator 的 critical path 較長，因此後來換了一種想法，在最一開始就比較完所有數字兩兩的大小，雖然需要十個比較但可以平行的做，之後利用兩兩的大小關係排出 rank0~rank4，舉例來說，A, B 通過 comparator 的 output 是 A>B 的 bool，對於每個數字 (i0, i1, i2, i3, i4)，可以得到 4 個和其他數字比較關係：

i0	bool(i0>i1)	i3	bool(i3>i0)
	bool(i0>i2)		bool(i3>i1)
	bool(i0>i3)		bool(i3>i2)
	bool(i0>i4)		bool(i3>i4)
i1	bool(i1>i0)	i4	bool(i4>i0)
	bool(i1>i2)		bool(i4>i1)
	bool(i1>i3)		bool(i4>i2)
	bool(i1>i4)		bool(i4>i3)
i2	bool(i2>i0)		
	bool(i2>i1)		
	bool(i2>i3)		
	bool(i2>i4)		

若 i0 是 rank0(biggest)，則 i0 與其他數字比較的 4 個 output 都會是 1。若 i0 為 rank1，會有 3 個 output 是 1。若 i0 為 rank2，會有 2 個 output 是 1，若 i0 為 rank3，只會有 1 個 output 是 1。若 i0 為 rank4(smallest)，則 4 個 output 皆為 0。

其他數字以此類推，利用上述條件就可以平行的排出 rank0~rank4，只需要加上一些 mux 去選擇我們要的 rank 對應的 input，中間不需要再用 comparator 就可以快很多，用這個架構 critical path 就成功壓在 3ns 以內。

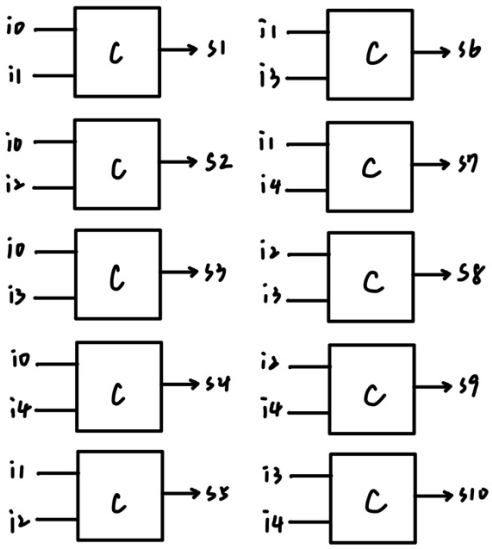
```
*Verdi* : Enable +mda and +packedmda dumping.
*Verdi* : End of traversing the MDAs.
Congratulations! Your critical path is below 3!

Simulation complete via $finish(1) at time 210 US + 0
./tb_sorting.v:137                      $finish;
ncsim> exit
[b06066@cad29 vlsi_hw3]$
```

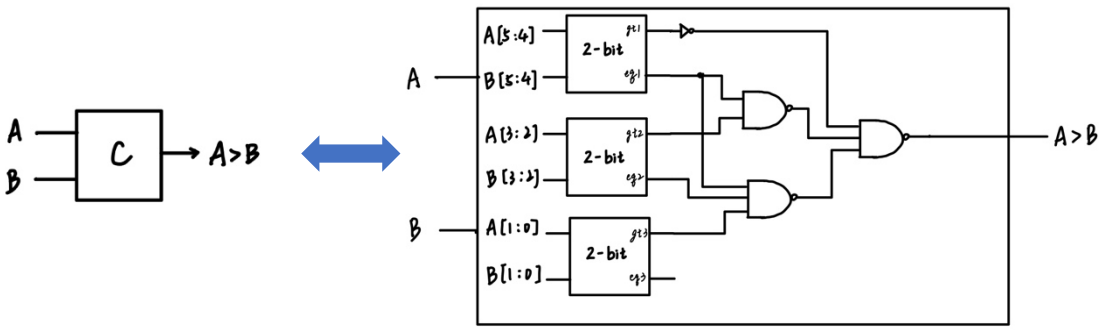
2. Circuit Design

part1. comparator

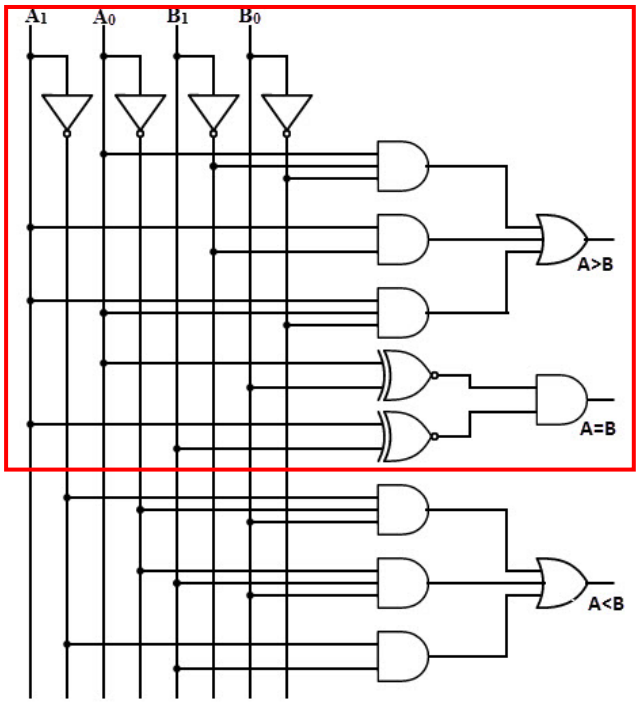
整個 circuit 的大架構就是在一開始使用 10 個得出兩兩數字的大小關係，令為 s1.s2,⋯,s10，再用三層 mux 選出每個 rank 分別是哪個 input。



一個 comparator(6-bit) C 是用 3 個 2-bit comparator 連接成 6-bit。



1 個 2-bit comparator 用的是這個架構：

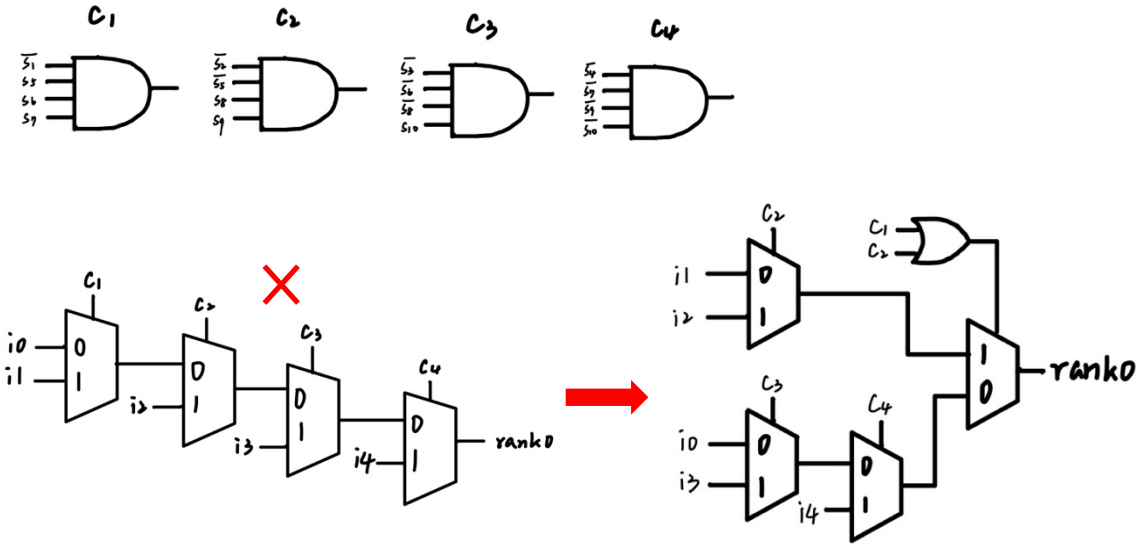


假如三個 2-bit comparator 由上至下分別輸出(gt1,eq1), (gt2,eq2), (gt3,eq3)，則變成 6-bit 的 comparator 的條件為 gt1 or (eq1 and gt2) or (eq1 and eq2 and gt3)。

part2. conditions for selecting each rank

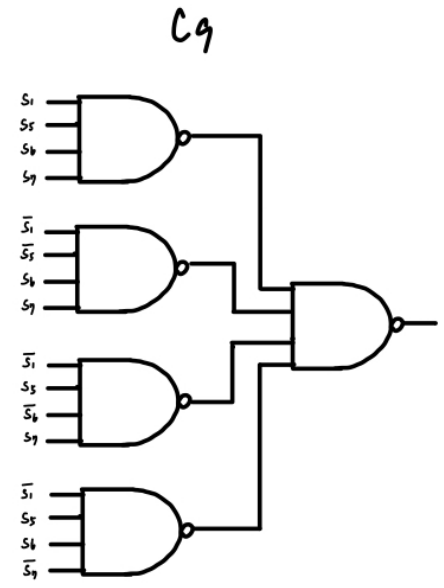
(1) rank0

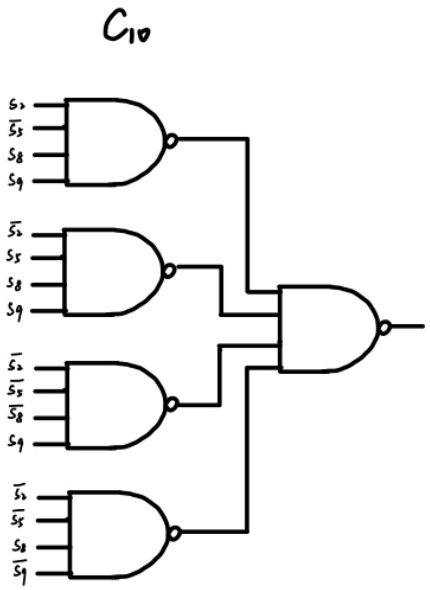
假如 c1=1，代表 i1 是 rank0 的數字，若 c2=1，則 i2 是 rank0 的數字，以此類推

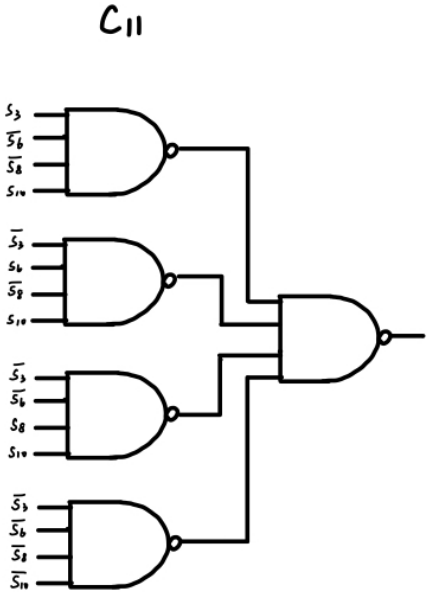


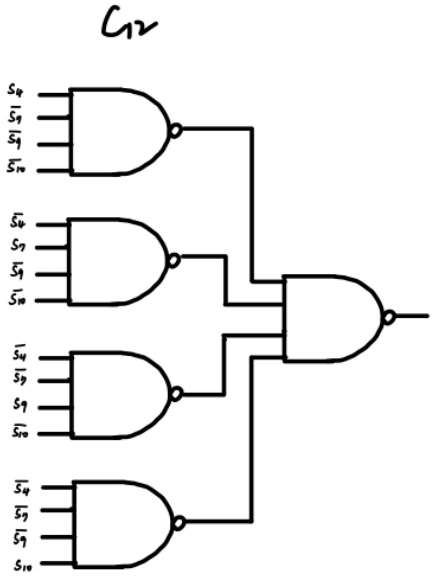
原本在 select 的時候用了四層 mux，但這樣的 critical path 會稍微超過 3ns，因此後來想到改成像上圖這樣三層，時間就可以壓到 3ns 以內。

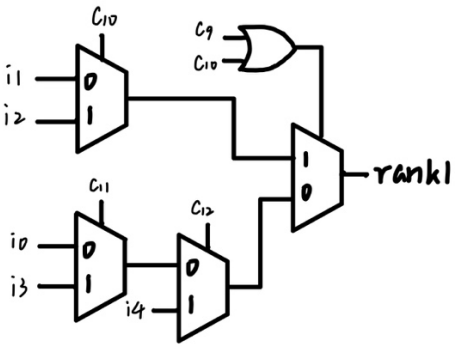
(2) rank1



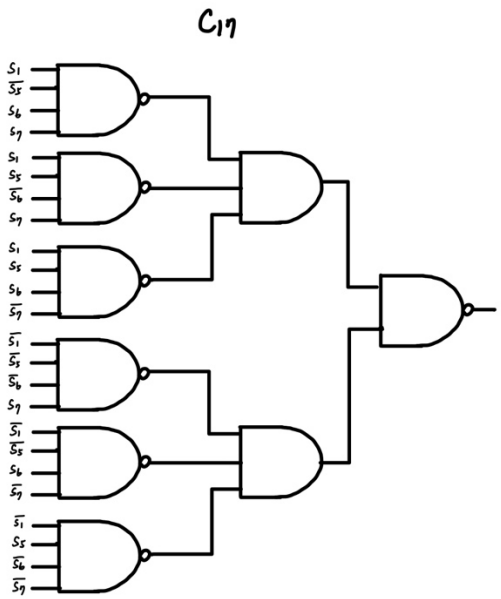


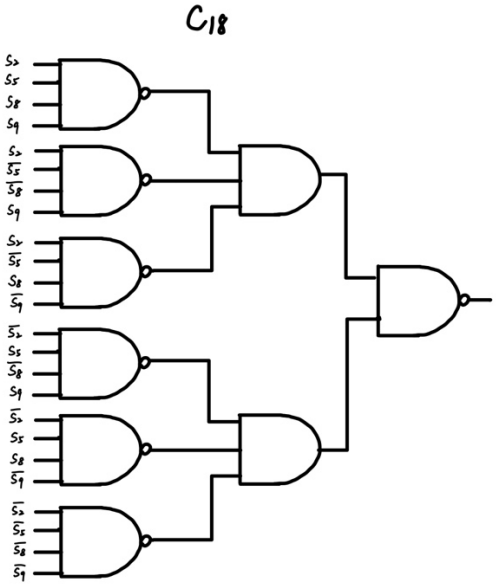


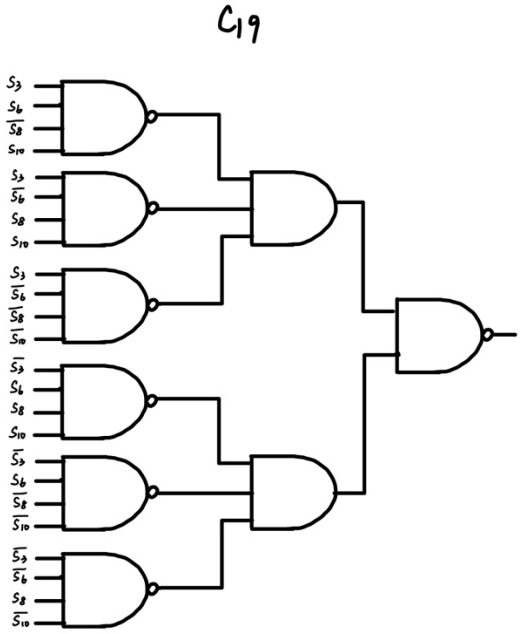


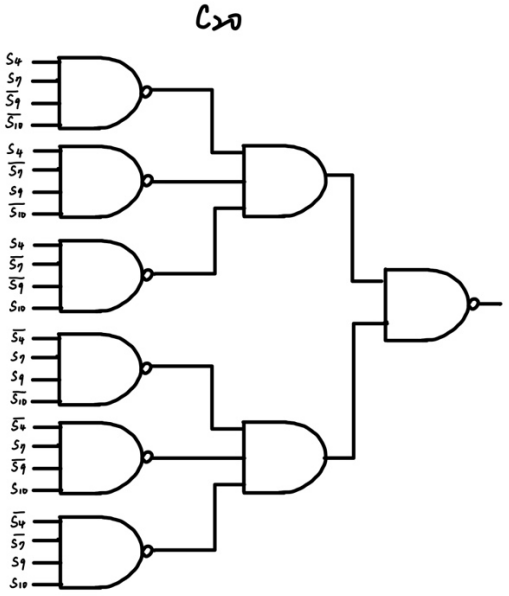


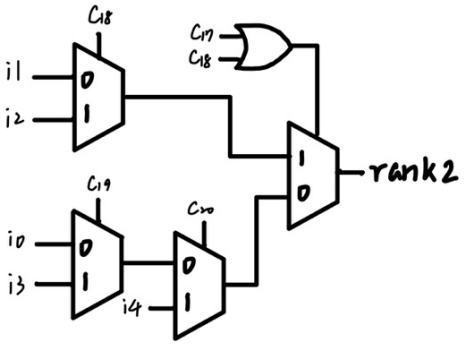
(3) rank2



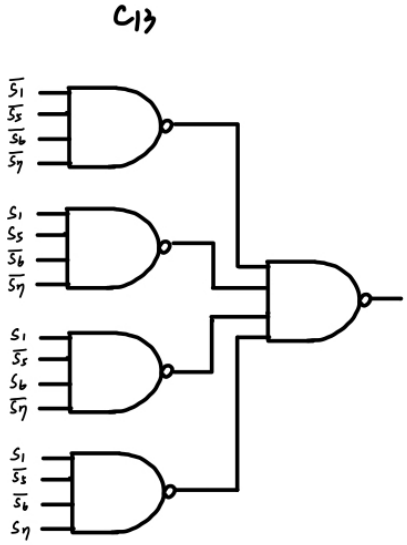


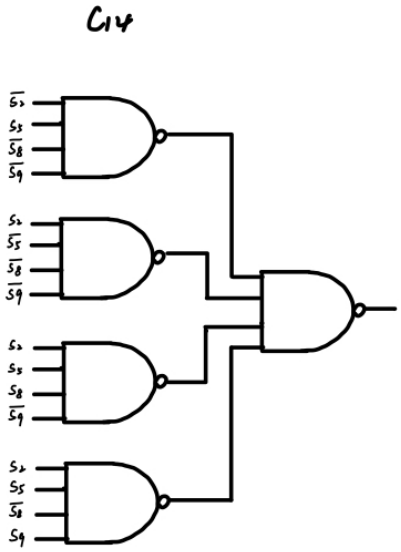


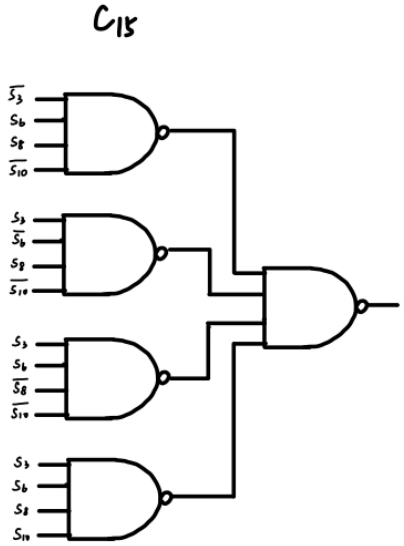


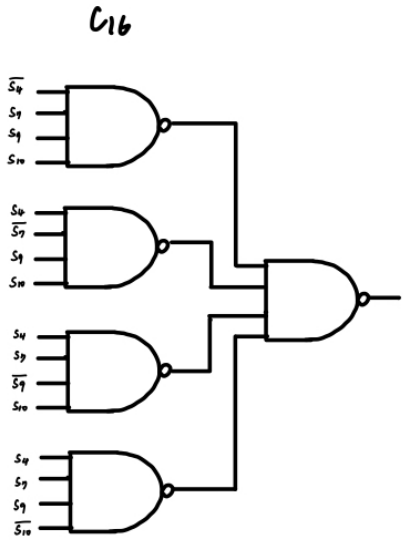


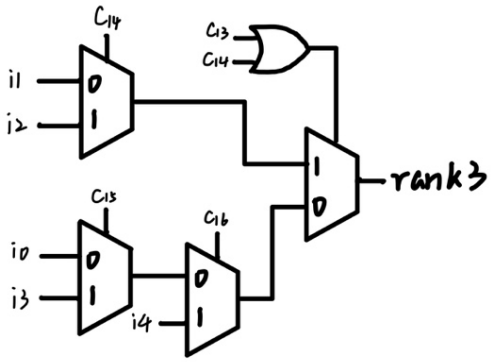
(4) rank3



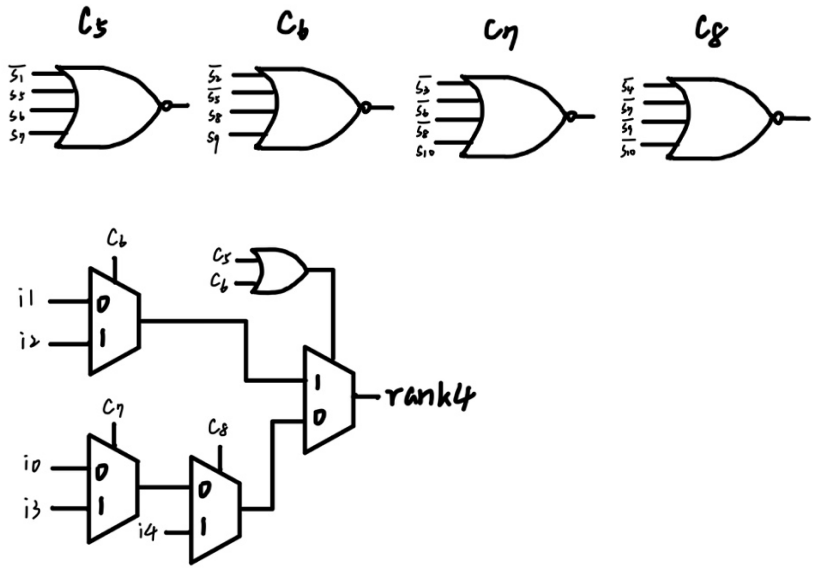








(5) rank4



3. Conclusion

最長的 path 為選出 rank2 的 path，使用以上架構

critical path = 1.031(comparator) + 0.127(invert the output of the comparator) + 0.747(path of getting the condition of rank2) + 0.347*3(mux selecting) = **2.946 ns**