

以下為主程式中的架構及對應之用途列表：

mode:

- lg\_lt\_than -> 取得 e01, e02...e34; lg01,lg02...lg34; lt01, lt02...lt34(兩兩之間的相等關係)
- IV e01 = ne01 ->取得兩數之間的不等於關係

case1 (全不同)

- NOR\_S10(NOR10) -> 全不同的情況（把所有兩兩等於的情況 NOR 起來）
- AN\_S5 (AN5) -> 把 NOR10 出來的結果，和特定的大小關係做 AND5(Ex. 假設 i0 是最大的 input，則此時的可能為 AN5(NOR10 結果, lt01, lt02, lt03, lt04))，此步驟可以得到在這個 case 中，mode=i0 的訊號、mode=i1 的訊號等...

case2(2 同,其他不同)

- AN\_S7(AN7) -> 列舉所有 2 同,其他不同的情況
- OR2 -> 把 mode=i0 的情況 OR 起來、把 mode=i1 的情況 OR 起來...，此步驟可以得到在這個 case 中，mode=i0 的訊號、mode=i1 的訊號等...

case3(3 以上同)

- AN3 -> 考慮至少 3 個 input 相同的情況
- OR2 -> 把 mode=i0 的情況 OR 起來、把 mode=i1 的情況 OR 起來...，此步驟可以得到在這個 case 中，mode=i0 的訊號、mode=i1 的訊號等...

case4(2 同,2 同,1 不同)

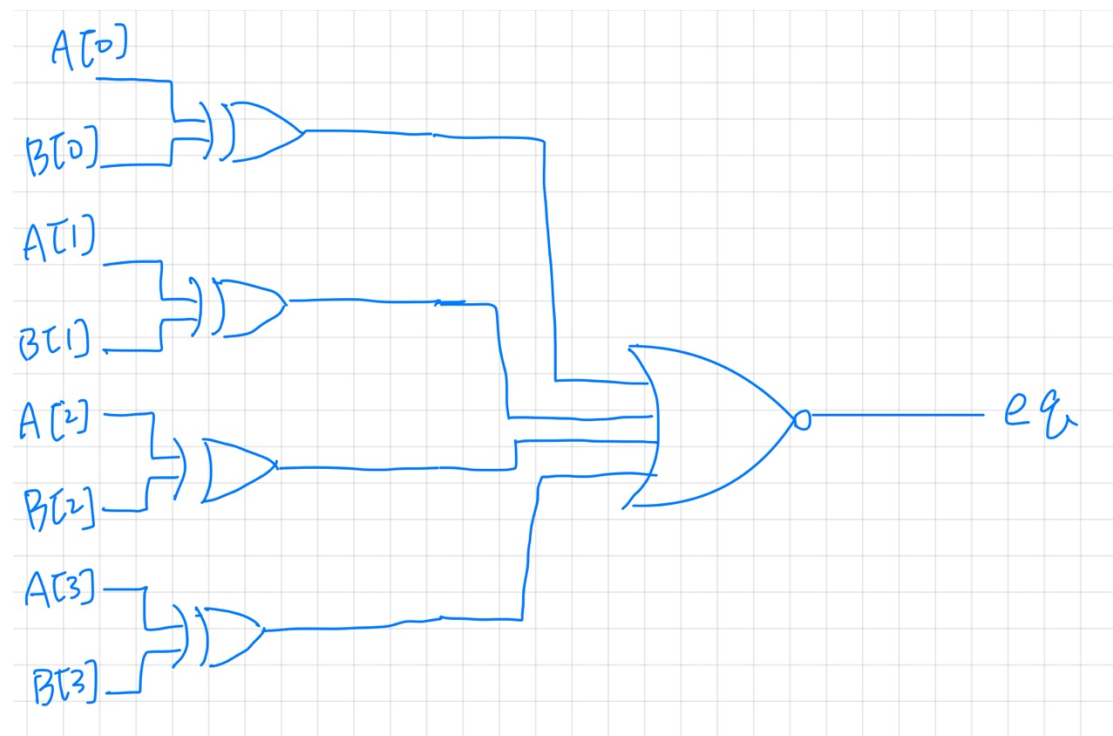
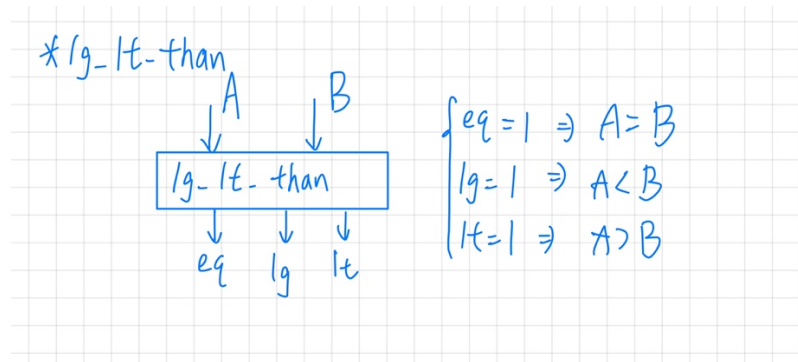
- AN\_S5(AN5) -> 列舉所有 2 同,2 同,1 不同的情況
- OR\_S6(OR6) -> 把 mode=i0 的情況 OR 起來、把 mode=i1 的情況 OR 起來...，此步驟可以得到在這個 case 中，mode=i0 的訊號、mode=i1 的訊號等...

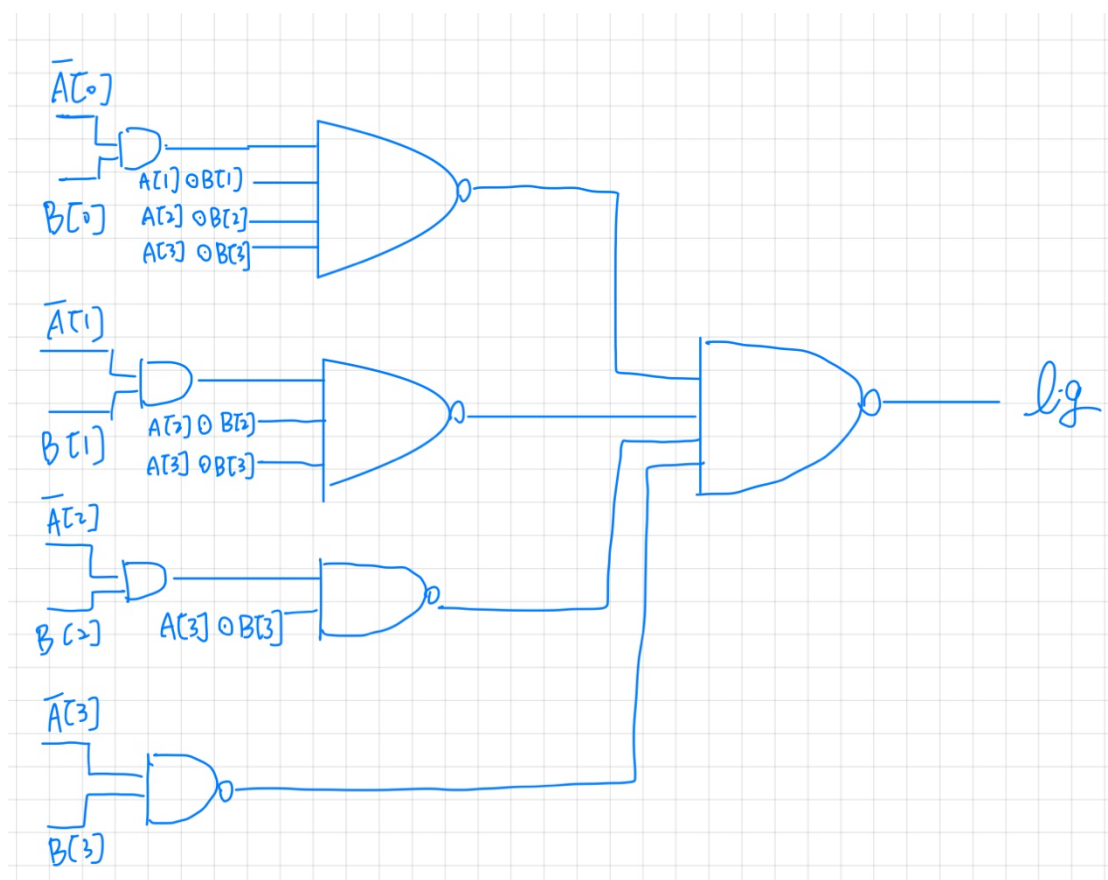
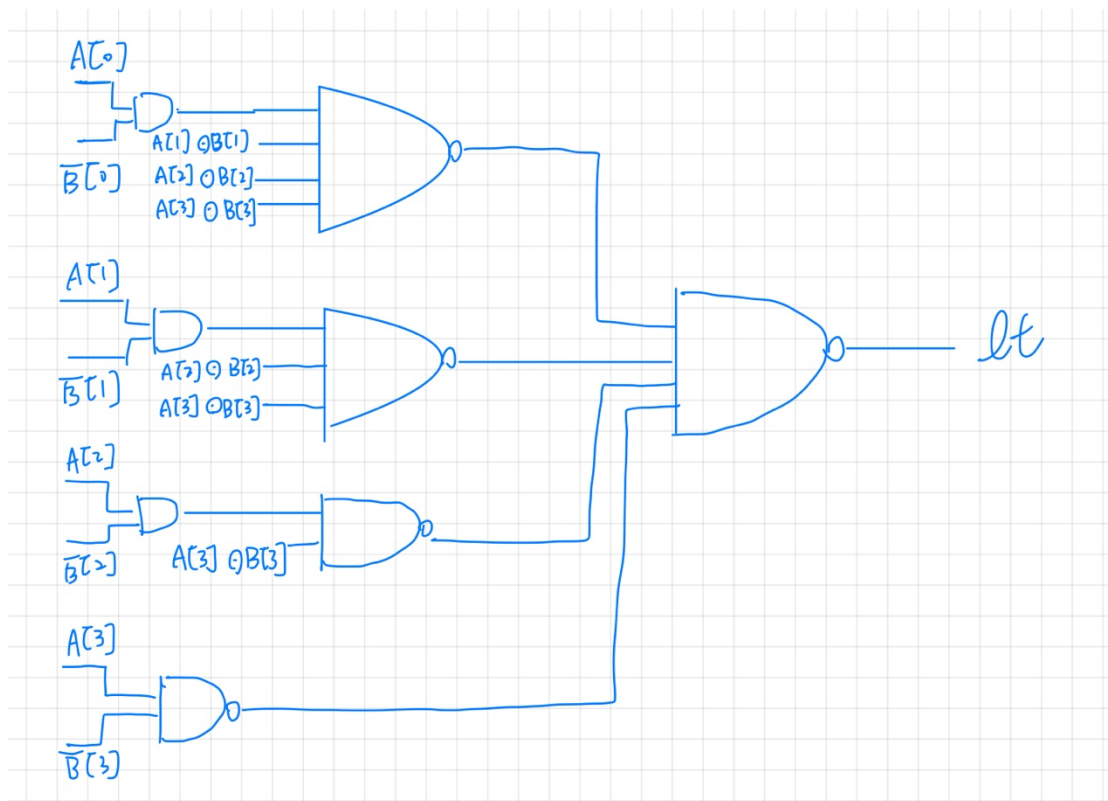
last

- OR\_S4(OR4) -> 把四種 case 中，mode 為 i0 的組合 OR 起來、mode 為 i1 的情況 OR 起來...以此類推
- final\_mux -> 由 OR4 中得到的 control 訊號，當 mode=i0 的訊號為 1 時，把 mode 設成 i0...以此類推

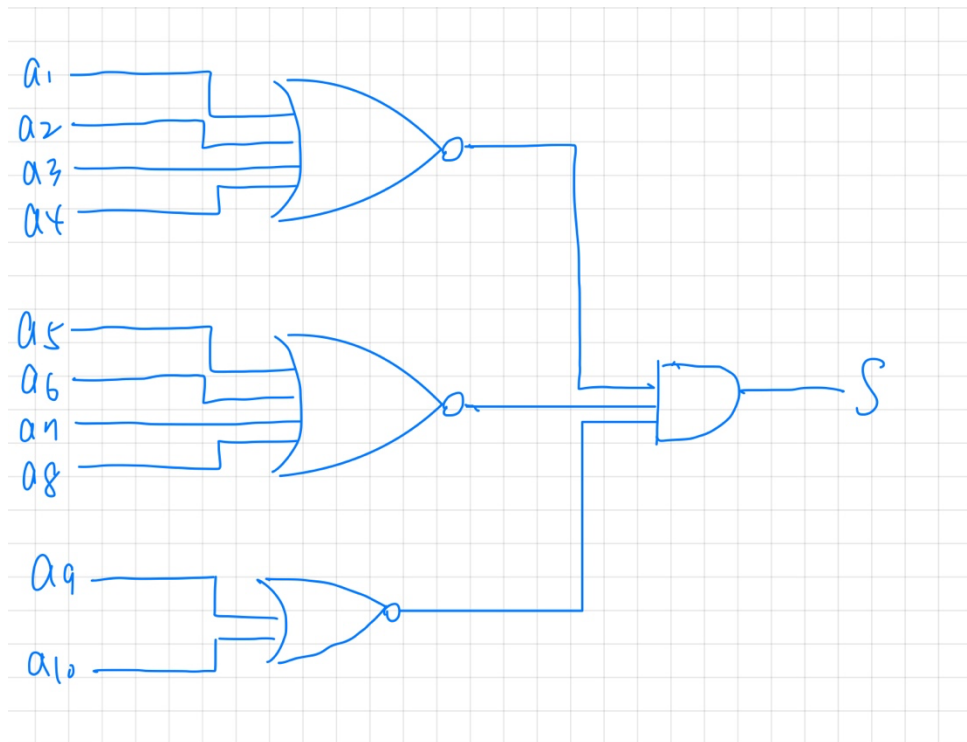
接下來是各小電路的 gate level diagram:

### 1. lg\_lt\_than(eq, lg, lt, A, B)

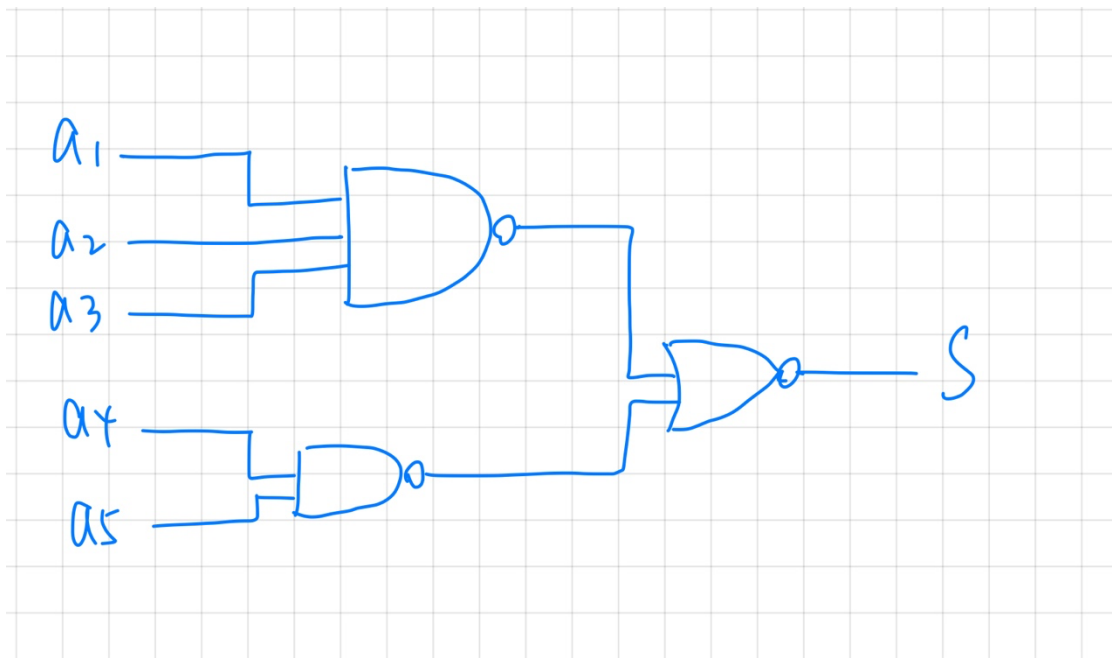




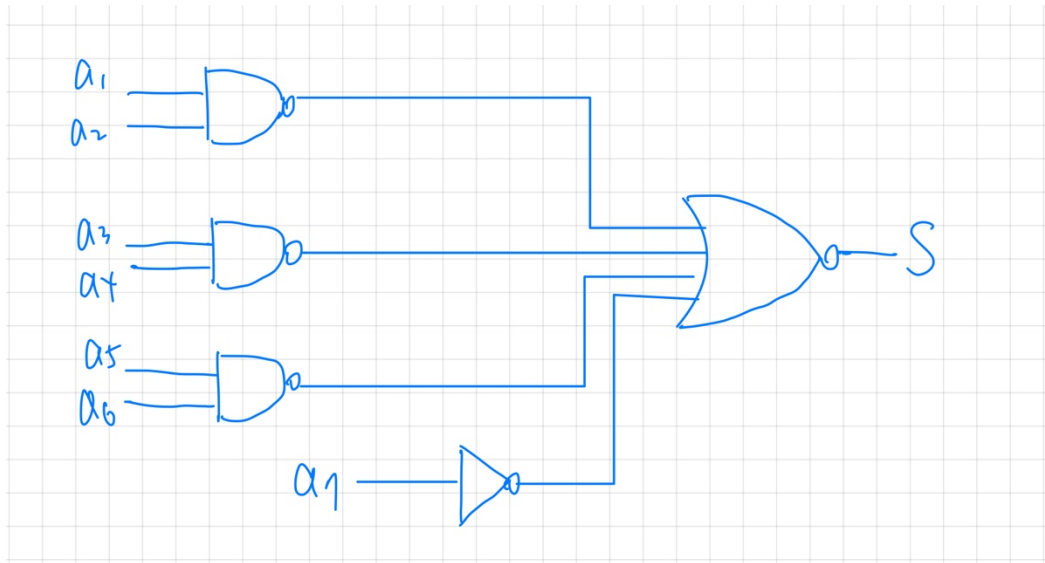
2. NOR\_S10(S, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10)



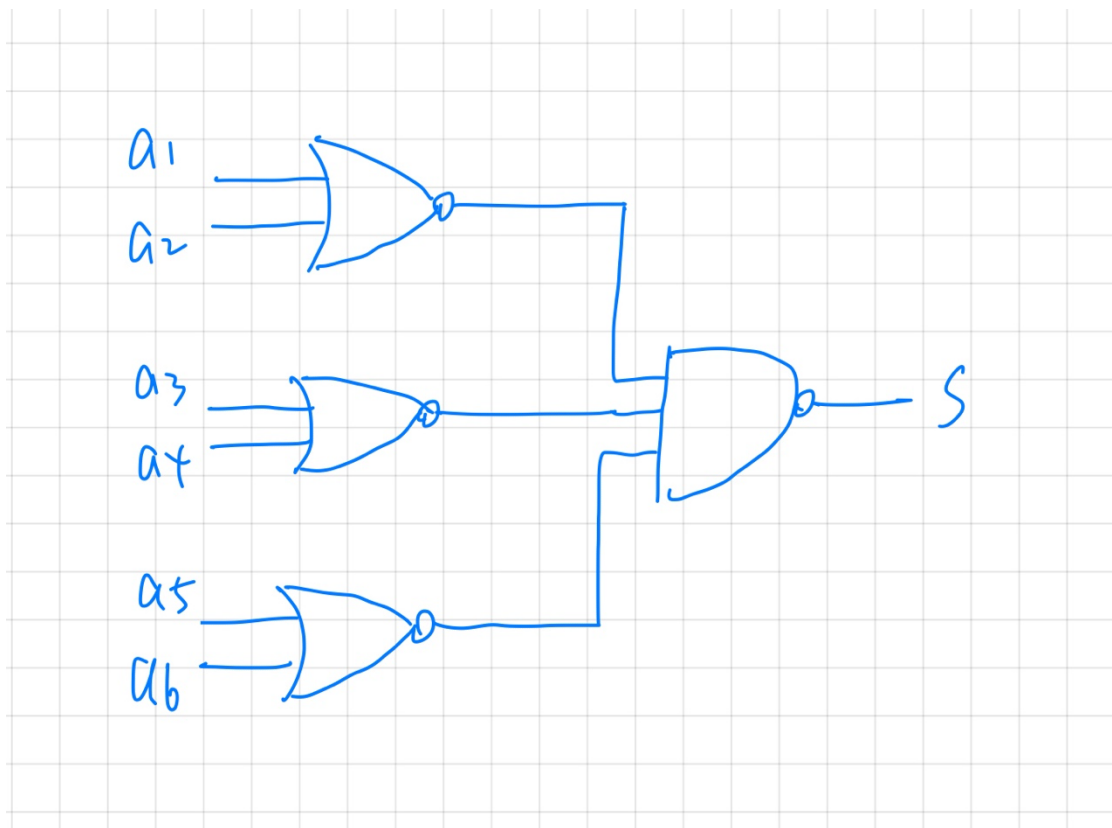
3. AN\_S5(S, a1, a2, a3, a4, a5)



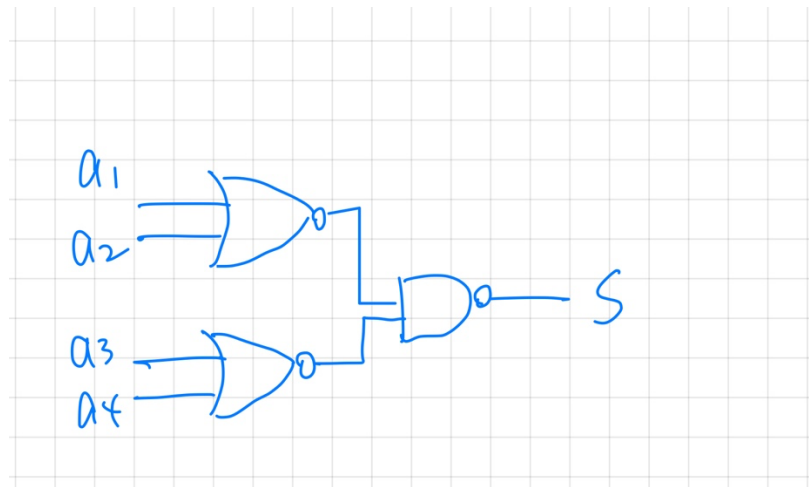
4. AN\_S7(S, a1, a2, a3, a4, a5, a6, a7)



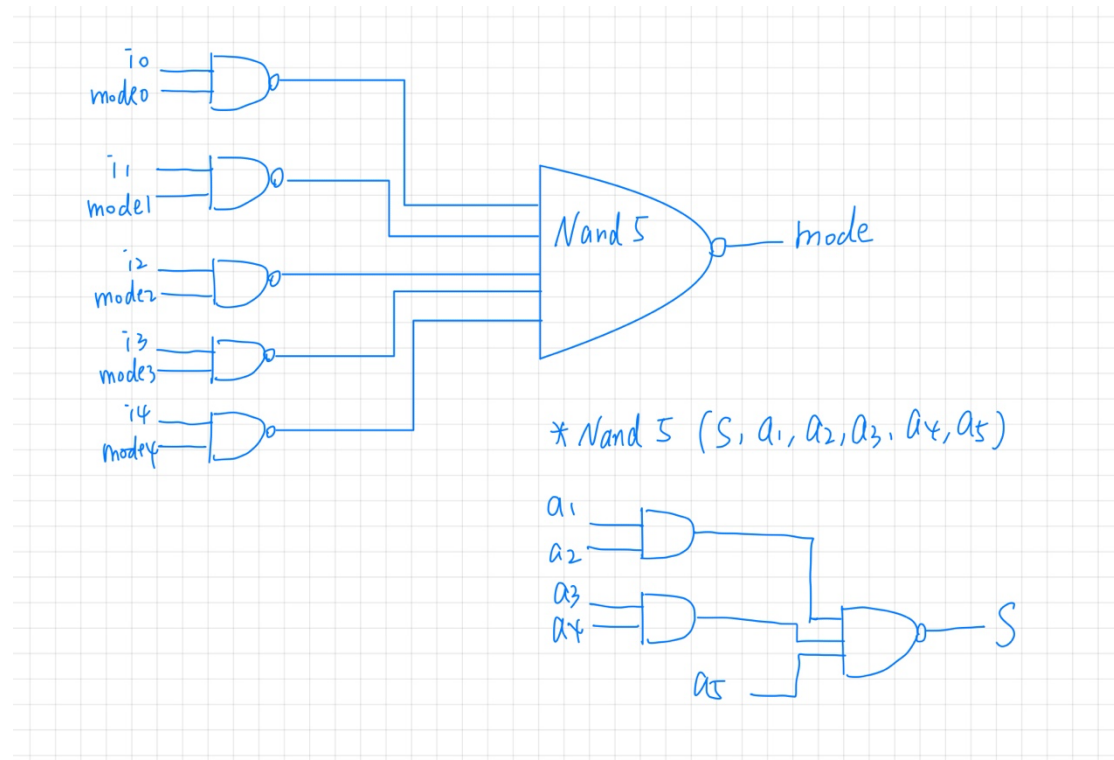
5. OR\_S6(S, a1, a2, a3, a4, a5, a6)



6. OR\_S4(S, a1, a2, a3, a4)



7. final\_mux(S, mode0, mode1, mode2, mode3, mode4, i0, i1, i2, i3, i4)



### Discussion :

在本次的作業中，我把 HA1 當作 AN2 來使用，因為 HA1 有其中一個 output 的表現和 AN2 是一樣的，而且有優於 AN2 的 latency。

在我所列出的 4 種 case 中，2 同,2 同,1 不同的 case 是 Critical path，這條 path 有最大的 latency，路途中經過

$lg\_lt\_than(EO+IV+ND4+ND4)+AN\_S5(ND3+NR2)+OR\_S6(ND3+NR2)+OR\_S4(NR2+ND2)+final\_mux(ND2+ND\_S5(HA1+ND3))$

第一次 pass 程式的時候，當時的 Critical path latency 超過 8ns，這是因為在一開始，我使用了大量的 AND, OR，而在判斷兩數大小時，都直接使用了 lib.v 中的 EN，而 lib.v 中的 EN 和 OR4 等，都是 latency 相當高的 gate，因此在之後的改良部分，我逐漸把 high latency 的 gate 換成 NAND, NOR 等的組合，利用 combinational circuit 組成整個電路。

而且還有一個部分是，在 lib.v 中的 NR3 的 latency 反倒比 NR4 來得大，因此在設計時我會以 NR4 為優先。

這次作業和其他課程的程式相當的不一樣，除了因為 Verilog 是用來描述硬體的語言之外，我們還要把電路的真實特性，像是 latency 等，列入我們設計時的考慮，收穫許多。