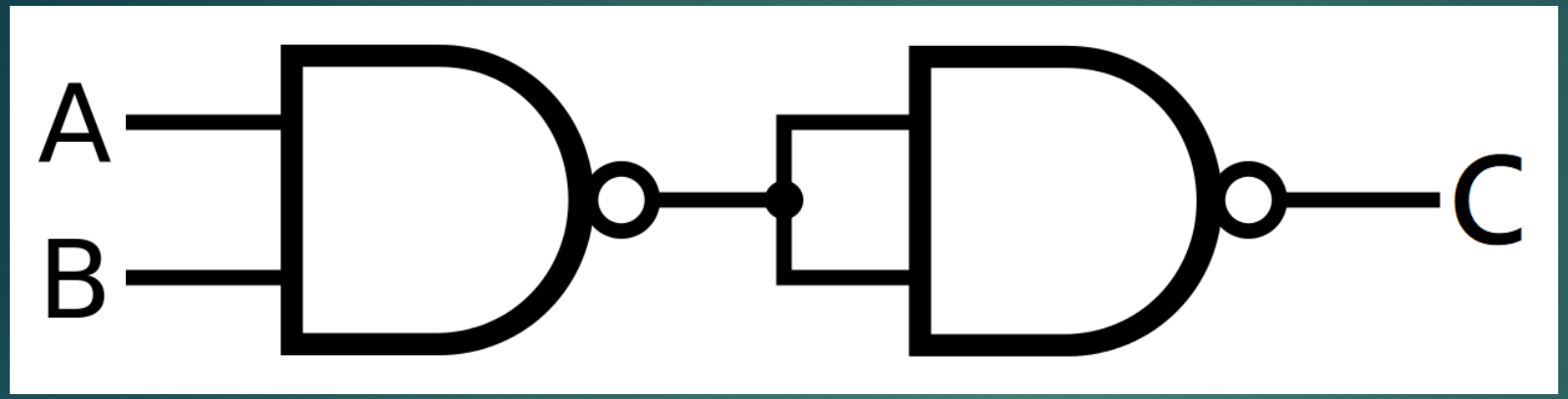
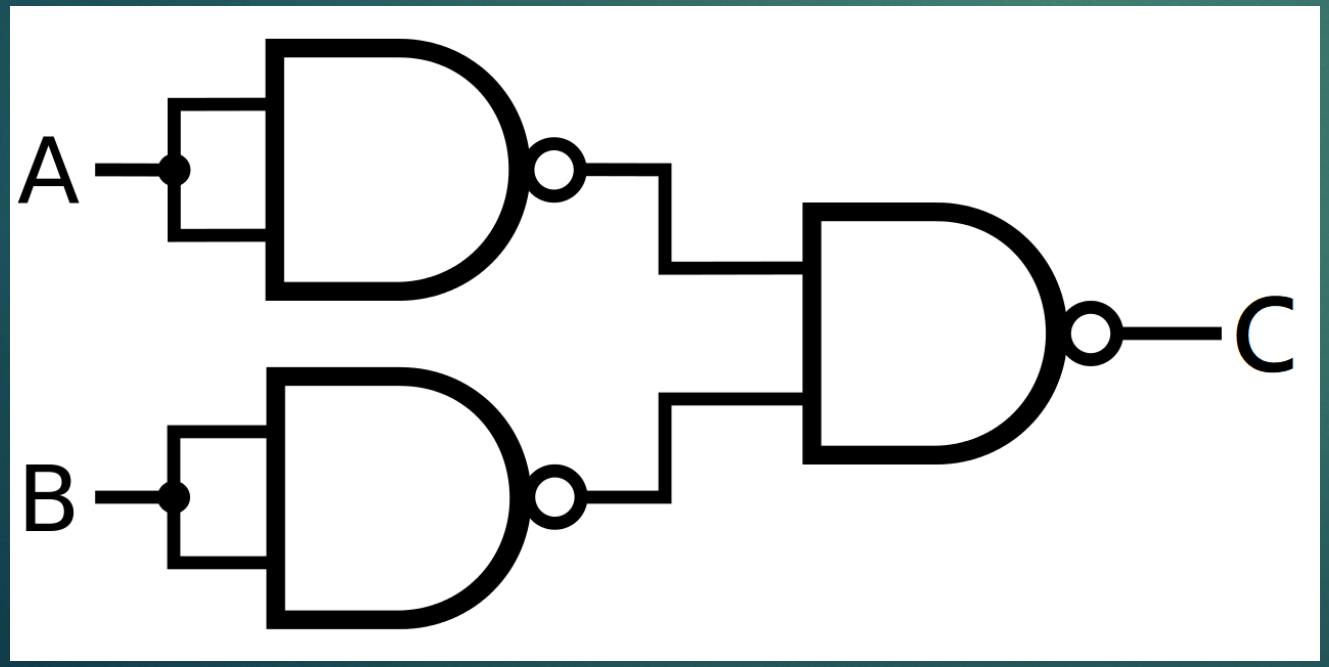
Lab2 ALU

106000215 江承翰

需要用NAND gate實作部分的電路圖

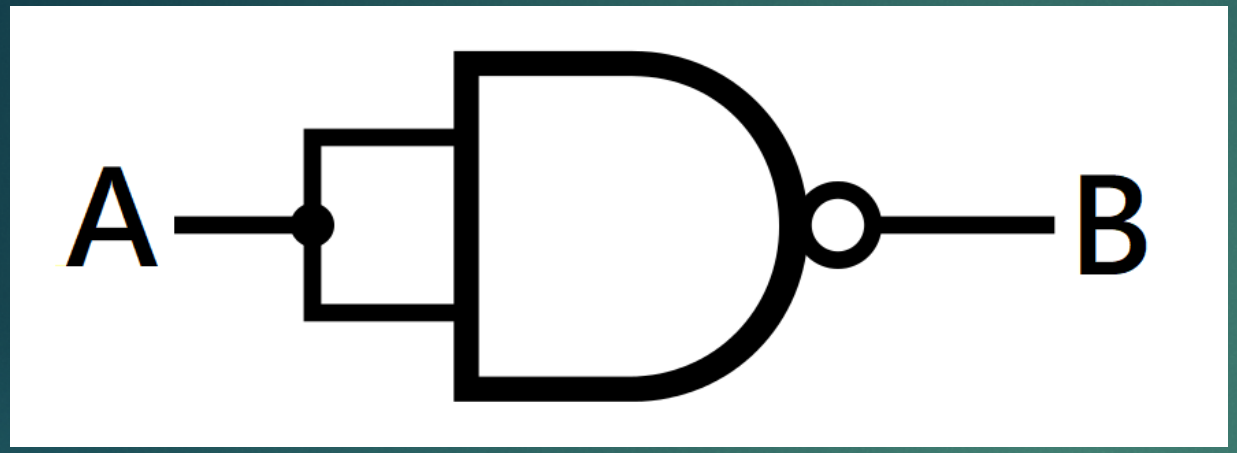


AND gate

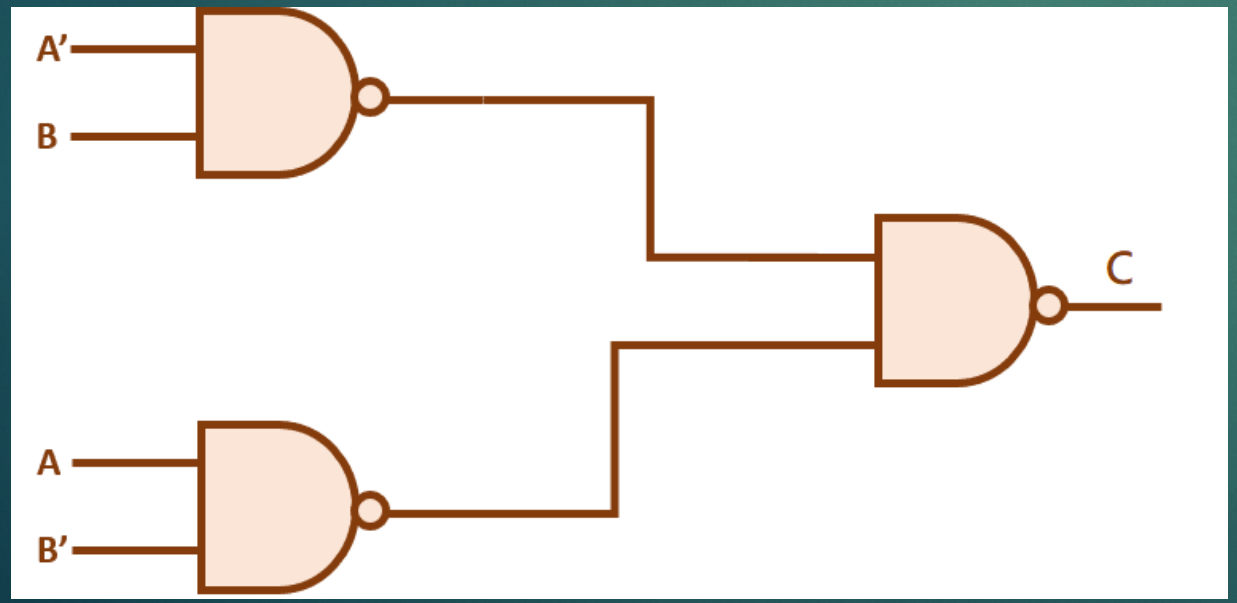


OR gate

需要用NAND gate實作部分的電路圖



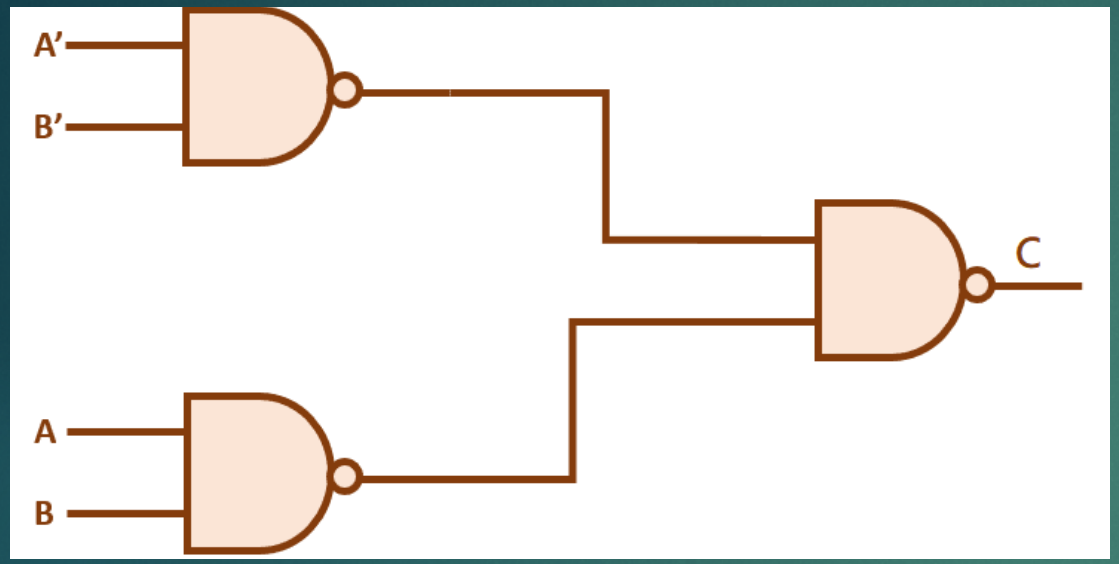
NOT gate



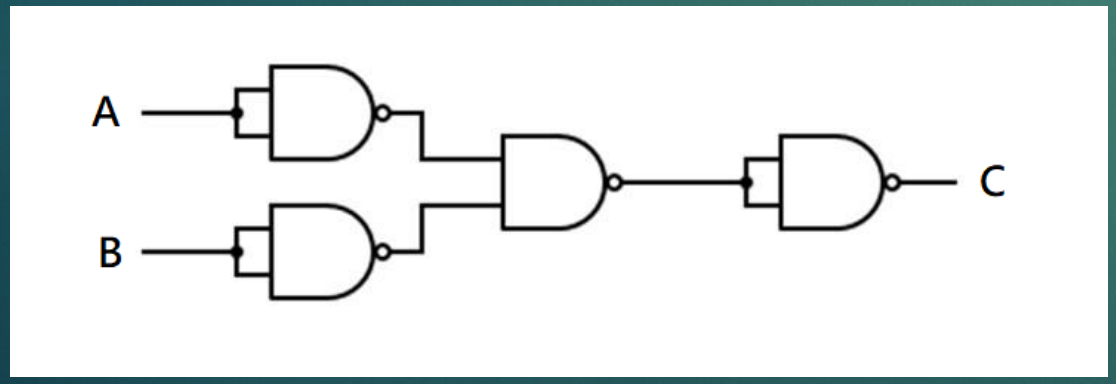
XOR gate

Remark: A' and B' are also made by NAND gate using the way above

需要用NAND gate實作部分的電路圖



XNOR gate



NOR gate

ncverilog 執行結果(最終make syn)

```
                tsmc13.XOR3X1:v <0x0ec0956a>
                        streams:  10, words:  1277
                worklib.ALU:v <0x4c2c6a61>
                        streams:   0, words:     0
                worklib.AND:v <0x0f5bfce5>
                        streams:   0, words:     0
                worklib.Arbiter:v <0x3d526739>
                        streams:   0, words:     0
                worklib.Decoder:v <0x48f2d2ce>
                        streams:   0, words:     0
                worklib.NOR:v <0x669949ab>
                        streams:   0, words:     0
                worklib.tb:v <0x234f6d74>
                        streams:   8, words: 81468
        Building instance specific data structures.
        Loading native compiled code:     ..................... Done
        Design hierarchy summary:
                                Instances  Unique
                Modules:             1078      56
                Primitives:          2592       8
                Timing outputs:      1244      17
                Registers:              7       7
                Scalar wires:        1252       -
                Expanded wires:        68       3
                Always blocks:          1       1
                Initial blocks:         3       3
                Cont. assignments:      0       5
                Pseudo assignments:     4       4
                Interconnect:        2967       -
                Simulation timescale:  1ps
        Writing initial simulation snapshot: worklib.tb:v
Loading snapshot worklib.tb:v ..................... Done
*Verdi3* Loading libsscore_ius141.so
*Verdi3* : Enable Parallel Dumping.
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run
-------------------------------

score =         41 / 41
Congratulations!!!!!
You pass all of the testcases. :))))

-------------------------------

Simulation complete via $finish(1) at time 1410 NS + 0
./ALU_tb.v:479  $finish;
ncsim> exit
TOOL:   ncverilog       14.10-s005: Exiting on Apr 18, 2019 at 15:16:37 CST  (total: 00:00:02)
(END)
```

Report Area

```
****************************************
Report : area
Design : ALU
Version: K-2015.06-SP1
Date   : Thu Apr 18 15:14:22 2019
****************************************

Library(s) Used:

    slow (File: /theda21_2/CBDK_IC_Contest/cur/SynopsysDC/db/slow.db)

Number of ports:                          400
Number of nets:                          1606
Number of cells:                         1077
Number of combinational cells:           1065
Number of sequential cells:                 1
Number of macros/black boxes:               0
Number of buf/inv:                        202
Number of references:                      39

Combinational area:              13185.403310
Buf/Inv area:                      879.253186
Noncombinational area:               0.000000
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:                 13185.403310
Total area:                 undefined

```

問題討論

關於Arbiter

- 在查過維基百科後，大概了解Arbiter(仲裁器)大致上是在多CPU時常用的Unit，負責在有多個程序同時索求電腦資源或記憶體時，將資源優先分配給優先度較高的程序。這次實作的find first bit arbiter就是類似的概念，以A的bit順序模擬處理器程序的優先順序。

關於always block

- 助教在講解時有跟我們說，always block中盡量不要針對多個變數做控制，但我在實作時並沒有考慮到這點，導致使用舊版的testbench會在A-B時出現unexpected result(左邊的幾個bit會變成1)。雖然使用新版的testbench之後問題已經改善，這次也來不及改，但我以後會養成良好的coding習慣，將不同的變數分開處理。

關於makefile

- 這次我有在助教的makefile中加入了一些自訂的指令，讓make cleaning變得更好用些(加入*.sdf, Report_area.txt, ALU_syn.v，讓cleaning清得更乾淨)，另外想順便請問一下助教:
  - 在SRC_syn中的第三行(-v /theda21_2/CBDK_IC_Contest/...)的意義
  - header.v的意義
- 由於以上功能這次並沒有用到，所以把疑問放在這裡

Thank you