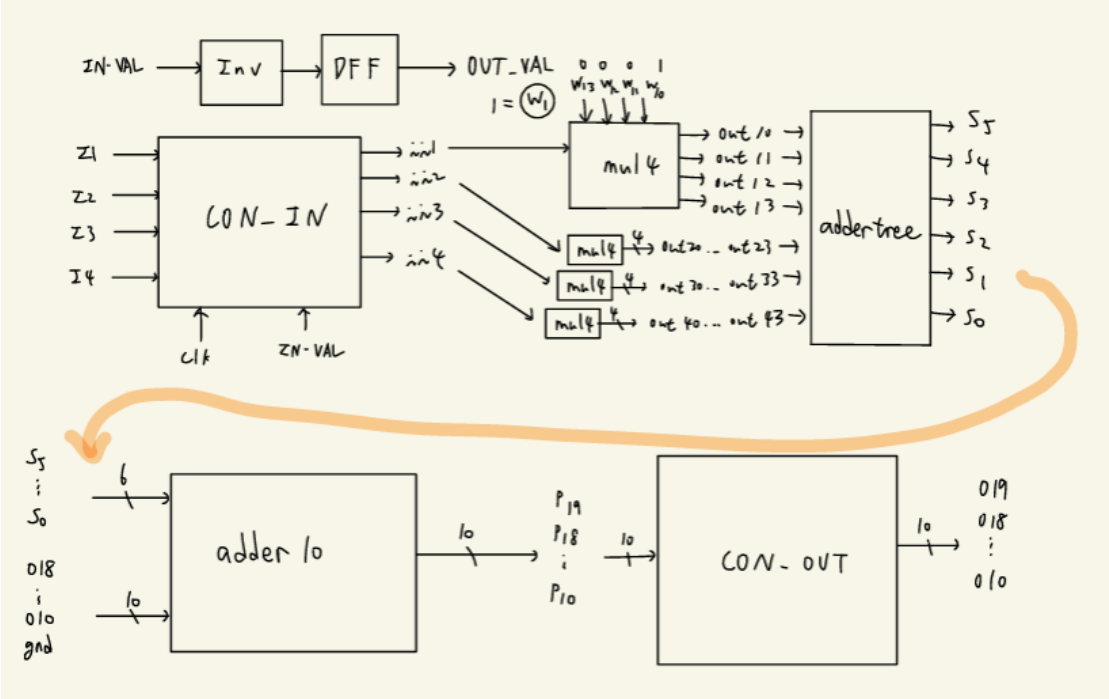
Final Report

ID：111062107

Name：鄧弘利

(1) The circuit diagram of your design and explaining your design (You can use screenshot to explain)



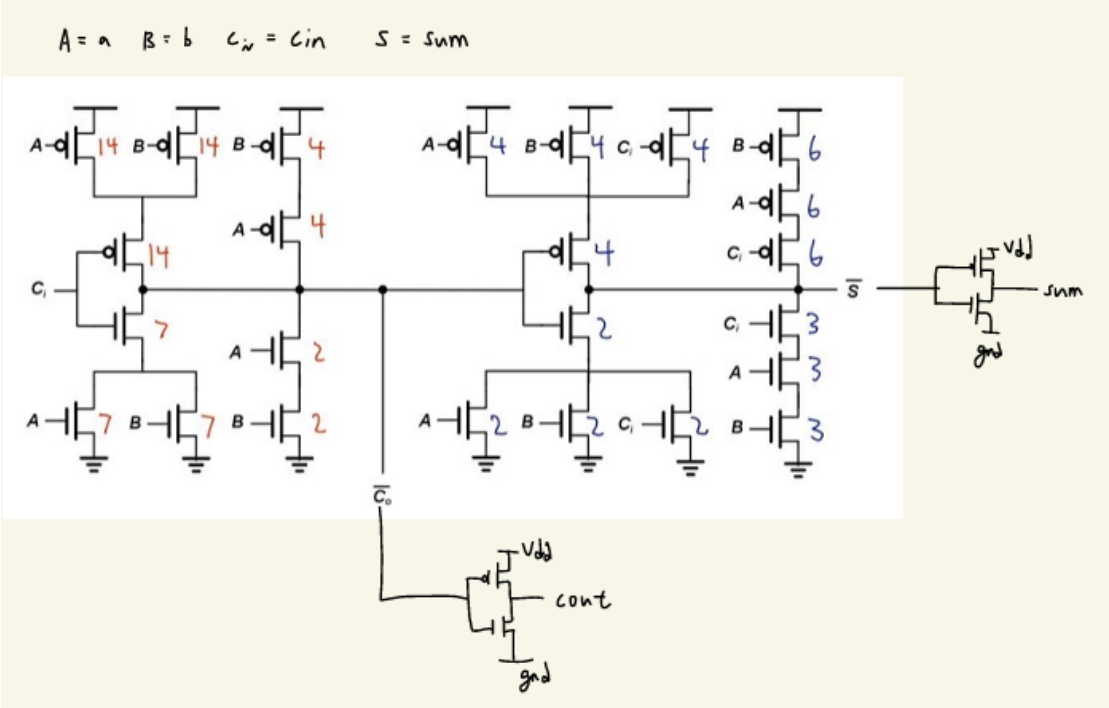
因為 O2 跟 O1 運作原理很像所以解釋 O1，兩者的差別只是在於訊號的名稱還有 weight 的設定。

我 OUT_VAL 的運作是將 IN_VAL 訊號相反之後，存到 D flip flop，做一個 cycle 的延遲。

ii1 ii2 ii3 ii4 則是透過 CON_IN 這裡去做的。當有這四個訊號之後，接著就去做乘法(也就是 mul4)，輸出的結果總共會有 16 個(out10 out11 ... out43)，就會傳給 addertree，統一做處理。加完之後的 sum，會有六個訊號來代表，分別是 s5 s4 ... s0。這些再傳給 adder10 做 10bit 的加法。那要注意的地方是我沒有用 shiftregister，我直接在傳入訊號的地方做 shift 一位，所以會是由 O18 O17 ... O10 還有 gnd 來代表 output 訊號 shift 一位的結果。加完之後會有 10 個訊號 (P19 P18 ... P10)，這些會丟給 CON_OUT 做處理，並在最後生成輸出的訊號 (O19 O18 ... O10)。

(2) The transistor level view of your CIM circuit and adder (You can just draw one of them and specified how you connect each)

我先作一個 bit 的 adder。他會做 a + b + cin 的動作，並輸出 cout 還有 sum。



再來可以看我的 addertree 的部分。

```
.subckt addertree a3 a2 a1 a0 b3 b2 b1 b0 c3 c2 c1 c0 d3 d2 d1 d0 s5 s4 s3 s2 s1 s0
Xmirroradder1 a0 b0 gnd net1 sum0 vdd gnd mirroradder
Xmirroradder2 a1 b1 net1 net2 sum1 vdd gnd mirroradder
Xmirroradder3 a2 b2 net2 net3 sum2 vdd gnd mirroradder
Xmirroradder4 a3 b3 net3 net4 sum3 vdd gnd mirroradder

Xmirroradder5 c0 d0 gnd net5 sum4 vdd gnd mirroradder
Xmirroradder6 c1 d1 net5 net6 sum5 vdd gnd mirroradder
Xmirroradder7 c2 d2 net6 net7 sum6 vdd gnd mirroradder
Xmirroradder8 c3 d3 net7 net8 sum7 vdd gnd mirroradder

Xmirroradder9 sum0 sum4 gnd net9 s0 vdd gnd mirroradder
Xmirroradder10 sum1 sum5 net9 net10 s1 vdd gnd mirroradder
Xmirroradder11 sum2 sum6 net10 net11 s2 vdd gnd mirroradder
Xmirroradder12 sum3 sum7 net11 net12 s3 vdd gnd mirroradder
Xmirroradder13 net4 net8 net12 s5 s4 vdd gnd mirroradder
.ends
```

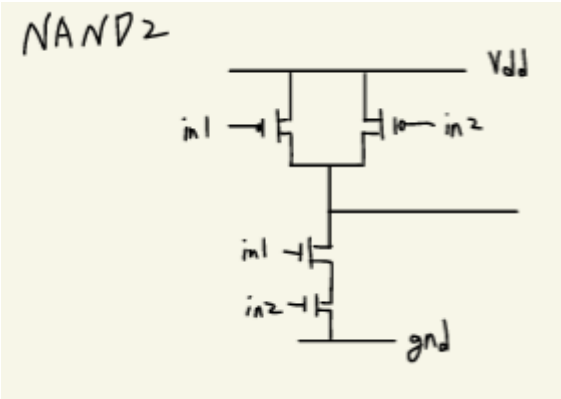
主要的概念就是第一個 bit 的 cin 是 0 所以接 gnd，然後每一個 cout 都會是下一個 bit 的 cin，sum 就是那一個位置的 sum。

最後做 shift 還有相加的地方，是 adder10，主要是做 10bit 的加法，這裡也是差不多概念，但是有點不一樣的地方在於，我第一個 bit，是用 and 還有 xor 去分別計算它的 cout 還有 sum。

```
* 10-bit
.subckt adder10 a5 a4 a3 a2 a1 a0   b9 b8 b7 b6 b5 b4 b
Xand1 a0 b0 carry0 vdd gnd AND2
Xxor a0 b0 s0 vdd gnd XOR
Xadder1 a1 b1 gnd carry1 s1 vdd gnd mirroradder
Xadder2 a2 b2 carry1 carry2 s2 vdd gnd mirroradder
Xadder3 a3 b3 carry2 carry3 s3 vdd gnd mirroradder
Xadder4 a4 b4 carry3 carry4 s4 vdd gnd mirroradder
Xadder5 a5 b5 carry4 carry5 s5 vdd gnd mirroradder
Xadder6 gnd b6 carry5 carry6 s6 vdd gnd mirroradder
Xadder7 gnd b7 carry6 carry7 s7 vdd gnd mirroradder
Xadder8 gnd b8 carry7 carry8 s8 vdd gnd mirroradder
Xadder9 gnd b9 carry8 carry9 s9 vdd gnd mirroradder
.ends
```

Xor 用了 4 個 NAND 去做。

```
.subckt XOR A B Y vdd gnd
* NAND gates for XOR logic
XNAND1 A B nand1_out vdd gnd NAND2
XNAND2 A nand1_out nand2_out vdd gnd NAND2
XNAND3 B nand1_out nand3_out vdd gnd NAND2
* Final NAND gate to produce XOR output
XNAND4 nand2_out nand3_out Y vdd gnd NAND2
.ends
```



乘法的部分我是用 AND 去做。

```
.subckt mul4 in w3 w2 w1 w0 out3 out2 out1 out0 vdd gnd
XAND0 in w0 out0 vdd gnd AND2
XAND1 in w1 out1 vdd gnd AND2
XAND2 in w2 out2 vdd gnd AND2
XAND3 in w3 out3 vdd gnd AND2
.ends
```

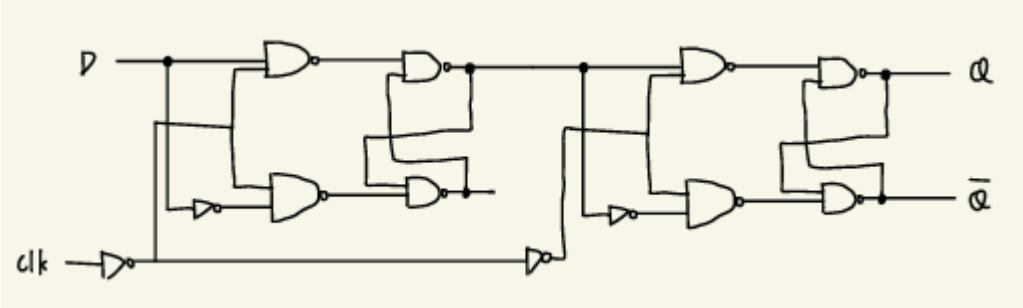
ii1 ii2 ii3 ii4 是 in，然後分別去跟他們對應的 weight 做 and。

ii1 跟 weight 乘出來，輸出的結果 out3 out2 ..out0 對到的就是 addertree a3...a0。ii2 對應輸出，則是對到 b3...b0。Ii3 對到 c3...c0，ii4 對到 d3...d0。

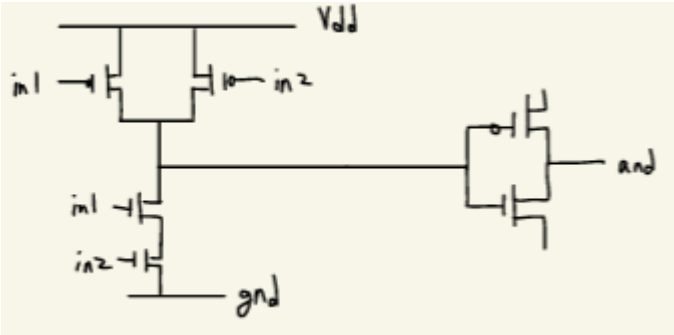
然後 addertree 輸出的 sum(s5...s0)，會去接到 adder10 的 a5...a0。然後 adder10 的 input b9...b0，是跟 output (o18...o10 還有最低位 gnd)做相接。最後加出來的結果就會是 P10...P19，會丟到 CON_OUT，去做最後的處理。

```
.subckt CON_OUT P0 P1 P2 P3 P4 P5 P6 P7 P8 P9 CLK RST O0 O1 O2 O3 O4 O5 O6 O7 O8 O9
*** Your code ***
*** Logic to handle reset functionality ***
XAND0 P0 RST reset0 vdd gnd AND2
XAND1 P1 RST reset1 vdd gnd AND2
XAND2 P2 RST reset2 vdd gnd AND2
XAND3 P3 RST reset3 vdd gnd AND2
XAND4 P4 RST reset4 vdd gnd AND2
XAND5 P5 RST reset5 vdd gnd AND2
XAND6 P6 RST reset6 vdd gnd AND2
XAND7 P7 RST reset7 vdd gnd AND2
XAND8 P8 RST reset8 vdd gnd AND2
XAND9 P9 RST reset9 vdd gnd AND2

*** Flip-flops with reset-aware inputs ***
XFF0 reset0 CLK O0 vdd gnd DFF
XFF1 reset1 CLK O1 vdd gnd DFF
XFF2 reset2 CLK O2 vdd gnd DFF
XFF3 reset3 CLK O3 vdd gnd DFF
XFF4 reset4 CLK O4 vdd gnd DFF
XFF5 reset5 CLK O5 vdd gnd DFF
XFF6 reset6 CLK O6 vdd gnd DFF
XFF7 reset7 CLK O7 vdd gnd DFF
XFF8 reset8 CLK O8 vdd gnd DFF
XFF9 reset9 CLK O9 vdd gnd DFF
.ends
```



D flip flop



and

CON_OUT 我做 reset 是用 and 去做。And 的結果再丟給 flipflop。

(3) The bonus circuit if implemented

No

(4) Waveform of your simulation

(5) The delay and the power of the circuit

```
****** transient analysis tnom=  25.000 temp=  30.000 ******
td= 543.8681p  targ=   4.7161n   trig=   4.1722n
pwr=   2.0485m  from=   0.              to=  26.0000n
```

(6) The total number of transistors (NMOS and PMOS) you use in this program

```
******  Circuit Statistics  ******
# nodes       =    6499 # elements   =    2604
# resistors   =       0 # capacitors =       0 # inductors   =       0
# mutual_inds =       0 # vccs       =       0 # vcvs        =       0
# cccs        =       0 # ccvs       =       0 # volt_srcs   =       8
# curr_srcs   =       0 # diodes     =       0 # bjts        =       0
# jfets       =       0 # mosfets    =    2596 # U elements  =       0
# T elements  =       0 # W elements =       0 # B elements  =       0
# S elements  =       0 # P elements =       0 # va device   =       0
# vector_srcs =       0 # N elements =       0
```

(7) The hardness of this assignment and how you overcome it

花比較多時間在搞懂訊號之間的關係還有先後順序。

(8) Any suggestions about this programming assignment

感覺 SPEC 有些地方可以說得更清楚一點

謝謝助教