

PDSD

HW1-2

Voting circuit and Median circuit design

班級:113 電機甲

學號:B093011055

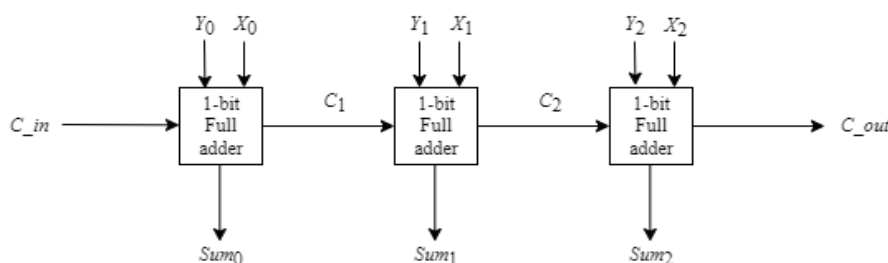
姓名:劉浩崴

# 壹、設計原理

## 一、 Voting circuit

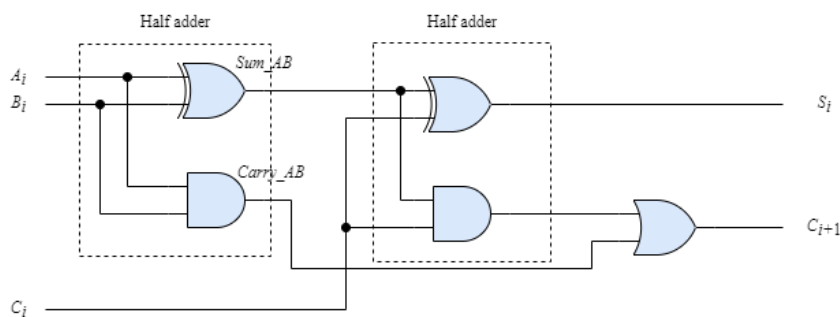
此 Voting circuit 設計為讓五個 3-bit 1-hot(只能有一個 bit 為 1)的數值輸入後其輸出為這五個數之中出現最多次的數值，若有相同數量的話，則輸出相同數量數值較大的輸入。此次作業設計將以數個 3-bits ripple carry adder 加上用來比較數量並輸出的邏輯閘來達成此 Voting circuit 的功能。

### (一)、3-bits ripple carry adder

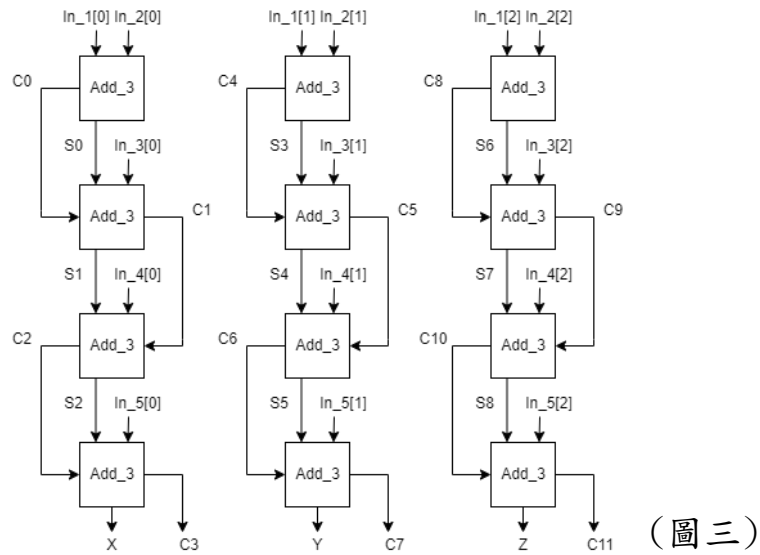


(圖一) 3-bits ripple carry adder

3-bits ripple carry adder 為三個 Full adder(圖二)所組成，如圖一。而此 voting circuit 設計利用 3-bits ripple carry adder 來將各個輸入的第一、二、三個位元分別進行相加，而其第一、二、三個位元相加完的結果分別代表 001、010、100 的數量，因此要完成此功能的電路共需 12 個 3-bits ripple carry adder，如圖三。(註:因為輸出數值不超過 5，故使用三位元加法器即可)



(圖二)



## (二)、比較數量的邏輯閘

在經過圖三的電路後可得到 001、010、100 的數量分別為 X、Y、Z，

在分析 X、Y、Z 數值組合後可以得到以下表格。

X	Y	Z	Out
5	0	0	001
4	1	0	
4	0	1	
3	1	1	
3	2	0	
3	0	2	
0	5	0	010
0	4	1	
1	4	0	
1	3	1	
0	3	2	
2	3	0	
2	2	1	
0	0	5	100
1	0	4	
0	1	4	
1	1	3	
2	0	3	
0	2	3	
1	2	2	
2	1	2	

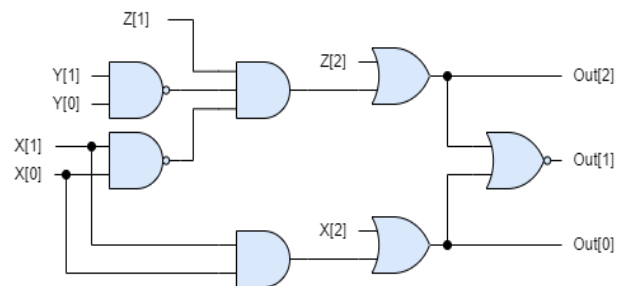
由表格的輸入輸出結果可得輸出：

$$\text{Out}[0] = X[0] \& X[1] \mid X[2]$$

$$\text{Out}[2] = \{ \sim(X[0] \& X[1]) \& \sim(Y[0] \& Y[1]) \& Z[1] \} \mid Z[2]$$

$$\text{Out}[1] = \sim(\text{Out}[0] \mid \text{Out}[2])$$

其邏輯電路圖如圖四

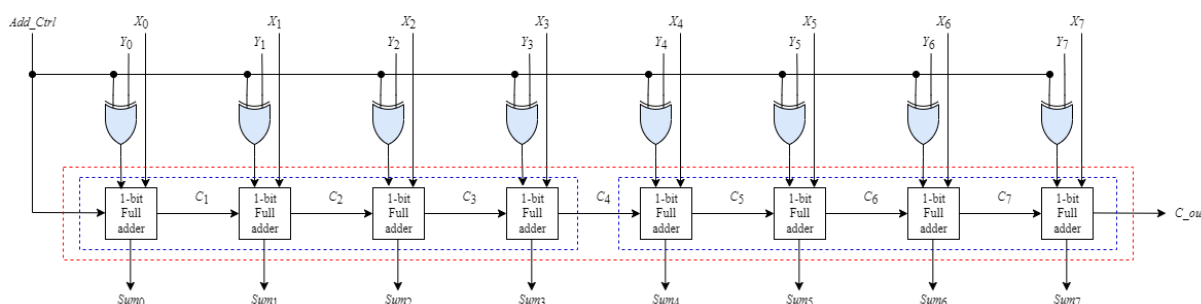


(圖四)

## 二、 Median circuit

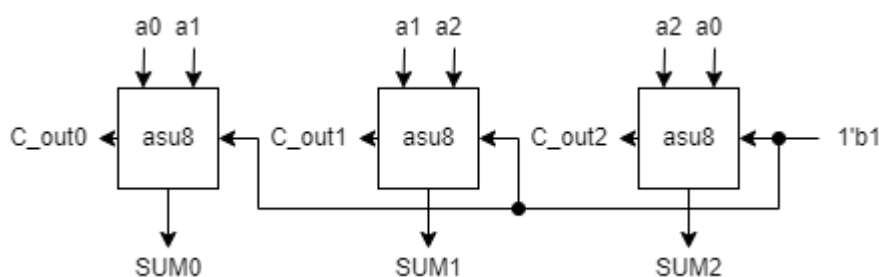
此 Median circuit 設計為讓三個 8-bit 1-hot(只能有一個 bit 為 1)的數值輸入後其輸出為這三個數的中位數。此次作業設計將使用三個 8-bit 的加減法器來比較三個輸入的大小，接著把比完大小的特徵位元之真值表利用 K-map 進行化簡產生決定輸出中位數的 Select bit，最後再利用數個 2 to 1 Mux 來輸出三個 8-bit 1-hot 的中位數。

### (一)、8-bit ripple carry adder/subtractor(asu8)



(圖五) 8-bit ripple carry adder/subtractor (asu8)

圖五為 8-bit 的加減法器，其為 8 個 Full adder 和控制加減法功能的 Add\_ctrl(圖四中藍色方框為 4 個 Full-adder 所組成的 Add4，2 組 Add4 可組成 1 個紅色方框的 Add8)，由於要比較大小，故此設計的加減法器的 Add\_ctrl 皆設為 1。至於加減法器能判斷大小的功能在其輸出的最高位元，當最高位元為 0 時，代表被減數大於等於減數，反之，最高位元為 1 時，則代表被減數小於減數。因此將上述功能套入此設計，其電路圖如(圖六)。



(圖六)

## (二)、產生決定輸出中位數的 Select bit 的邏輯電路

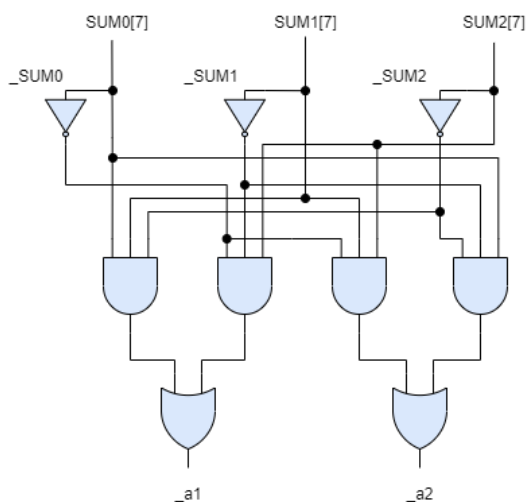
由圖六可得到判斷大小的特徵位元分別為  $SUM0[7](a0-a1)$ 、 $SUM1[7](a1-a2)$ 、 $SUM2[7](a2-a0)$ ，將其與決定中位數的 Select bit 輸出  $\_a0$  (輸出  $a0$ )、 $\_a1$  (輸出  $a1$ )、 $\_a2$  (輸出  $a2$ ) 之關係化為真值表(圖七)後利用 K-map 化簡可得到以下關係式：

A	B	C	$\_a0$	$\_a1$	$\_a2$
0	0	0	1	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	0	1
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	1	0
1	1	1	0	0	0

$$\begin{aligned} SUM0[7] &\Rightarrow A \\ SUM1[7] &\Rightarrow B \\ SUM2[7] &\Rightarrow C \\ \_a0 &= A'C' + AB'C \\ \_a1 &= A'B'C + ABC' \\ \_a2 &= A'BC + AB'C' \end{aligned}$$

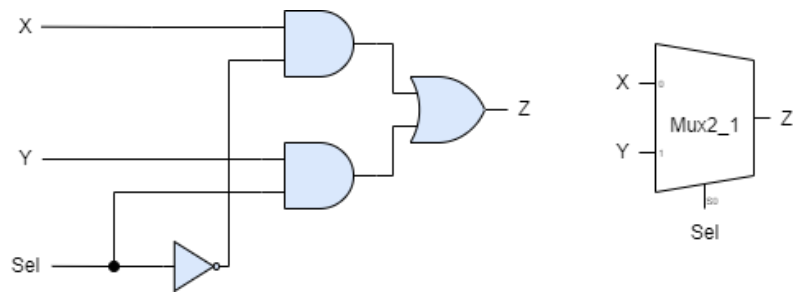
(圖七)

在得到輸入輸出的關係式後，可利用邏輯閘實現關係式的邏輯電路，但由於決定輸出的三個條件只需要兩個 Select bit 決定，故邏輯電路只需取兩條關係式即可，電路圖如圖八。



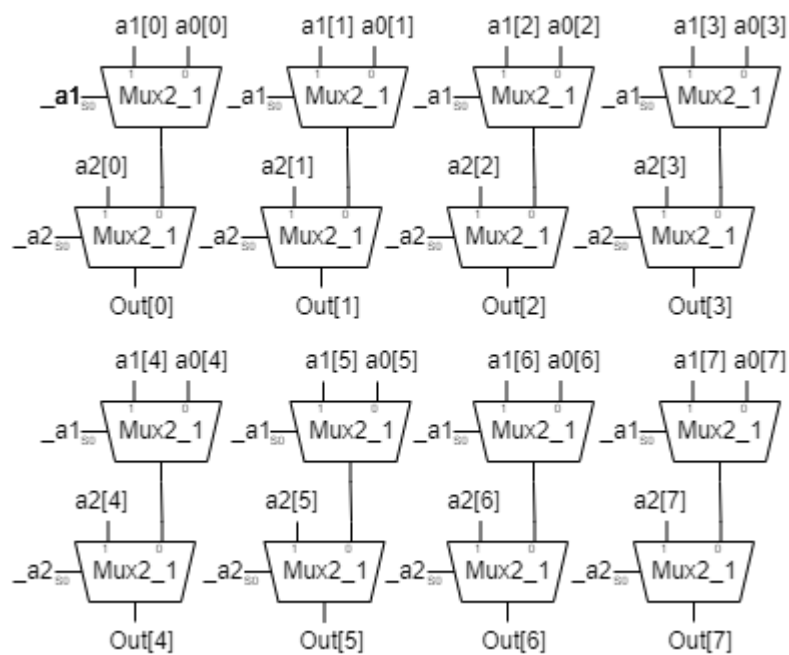
(圖八)

### (三)、2 to 1 Mux



(圖九) 2 to 1 Mux(Mux2\_1)

圖九為 2 to 1 Mux 的架構圖，其功能為讓兩個輸入利用一個選擇位元來決定其輸出。從圖八的邏輯電路可得到決定中位數輸出的 Select bit：a1、a2，利用兩層 2 to 1 Mux 來篩選便可得到此 Median circuit 三個輸入的中位數，電路圖如圖十。



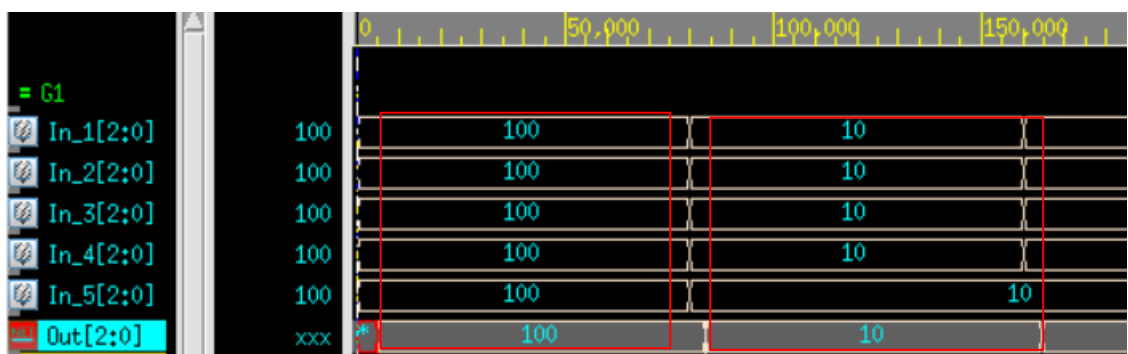
(圖十)

## 貳、結果分析

### 一、 Voting circuit

此次結果將檢測輸入數值 1.皆相同 2.四同一不同 3.三同 4.兩組相同平手等四種輸入組合，若以上檢測皆無誤，則此 Voting circuit 是可行的。每項檢測使用兩組數據確保正確。

#### (一)、輸入皆相同



#### 〈分析一〉

In\_1=100，In\_2=100，In\_3=100，In\_4=100，In\_5=100，Out=100

#### 〈結論一〉

全部輸入皆為 100，輸出結果為 100。計算結果正確

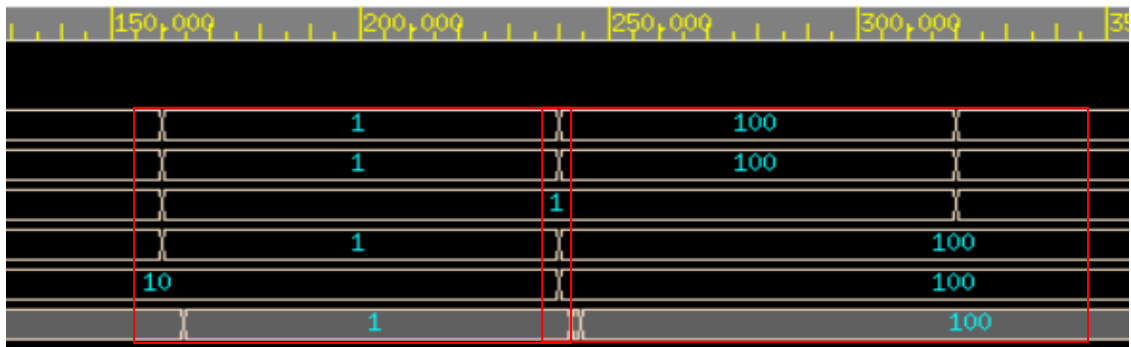
#### 〈分析二〉

In\_1=010，In\_2=010，In\_3=010，In\_4=010，In\_5=010，Out=010

#### 〈結論二〉

全部輸入皆為 010，輸出結果為 010。計算結果正確

## (二)、輸入四同一不同



### 〈分析一〉

In\_1=001 , In\_2=001 , In\_3=000 , In\_4=000 , In\_5=010 , Out=001

### 〈結論一〉

有四個輸入為 001，數量佔多數，輸出結果為 001。計算結果正確

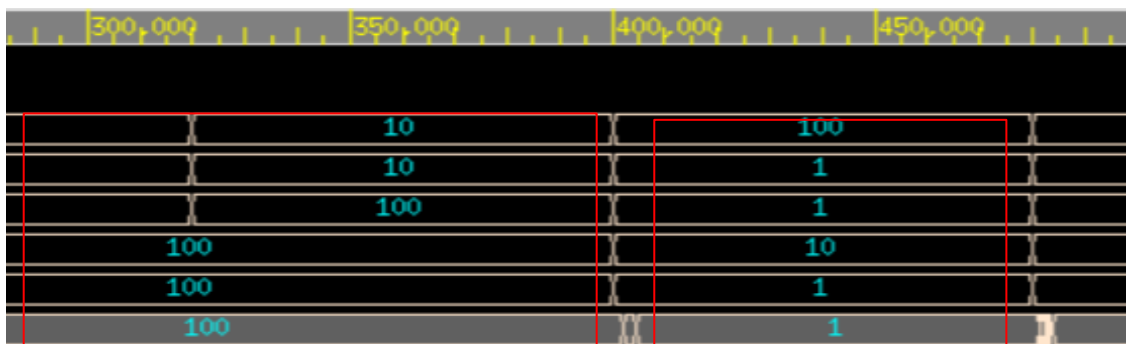
### 〈分析二〉

In 1=100 , In 2=100 , In 3=001 , In 4=100 , In 5=100 , Out=100

## 〈結論二〉

有四個輸入為 100，數量佔多數，輸出結果為 100。計算結果正確

### (三)、輸入三個相同



### 〈分析一〉

In 1=010 , In 2=010 , In 3=100 , In 4=100 , In 5=100 , Out=100

### 〈結論一〉



有三個輸入為 100，數量佔多數，輸出結果為 100。計算結果正確

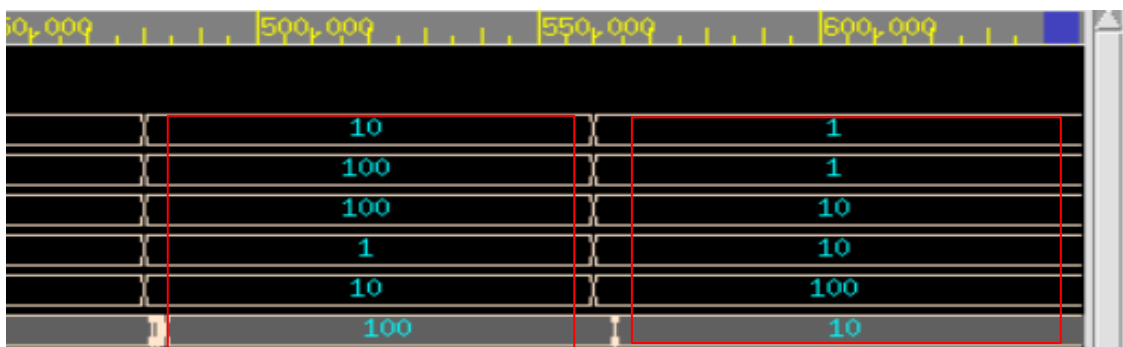
#### 〈分析二〉

In\_1=100，In\_2=001，In\_3=001，In\_4=010，In\_5=001，Out=001

#### 〈結論二〉

有三個輸入為 001，數量佔多數，輸出結果為 001。計算結果正確

### (四)、輸入兩組相同平手



10	1
100	1
100	10
1	10
10	100
100	10

#### 〈分析一〉

In\_1=010，In\_2=100，In\_3=100，In\_4=001，In\_5=010，Out=100

#### 〈結論一〉

輸入為 100 和 010 的數量相同，100 數值較大，輸出結果為 100。計算結果正確

#### 〈分析二〉

In\_1=001，In\_2=001，In\_3=010，In\_4=010，In\_5=100，Out=010

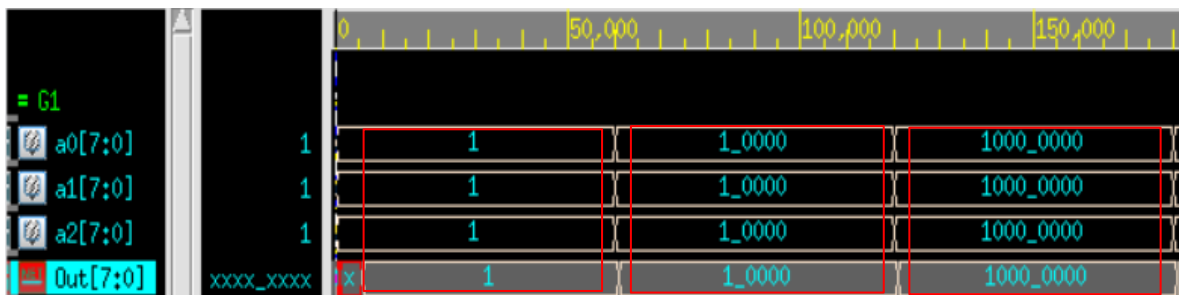
#### 〈結論二〉

輸入為 001 和 010 的數量相同，010 數值較大，輸出結果為 010。計算結果正確

## 二、 Median circuit

此次結果將檢測輸入數值 1.皆相同 2.兩同一不同 3.皆不同等三種輸入組合，若以上檢測皆無誤，則此 Median circuit 是可行的。每項檢測使用三組數據確保正確。

### (一)、輸入皆相同



#### 〈分析一〉

`a0=00000001`，`a1=00000001`，`a2=00000001`，`Out=00000001`

#### 〈結論一〉

輸入皆為 00000001，輸出結果為 00000001。計算結果正確

#### 〈分析二〉

`a0=00010000`，`a1=00010000`，`a2=00010000`，`Out=00010000`

#### 〈結論二〉

輸入皆為 00010000，輸出結果為 00010000。計算結果正確

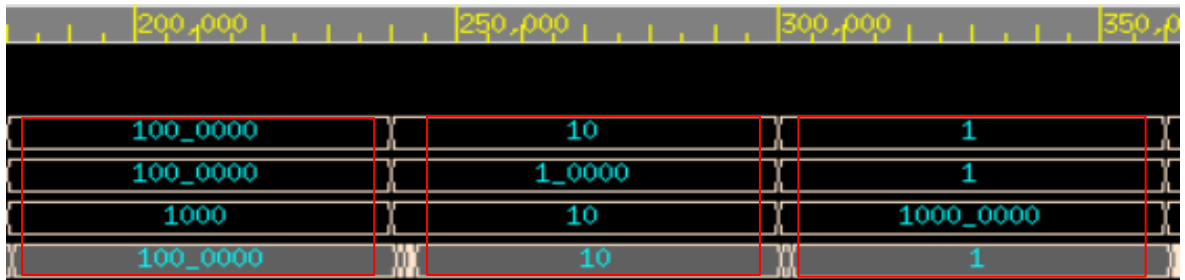
#### 〈分析三〉

`a0=10000000`，`a1=10000000`，`a2=10000000`，`Out=10000000`

#### 〈結論三〉

輸入皆為 10000000，輸出結果為 10000000。計算結果正確

## (二)、輸入兩同一不同



100_0000	10	1
100_0000	1_0000	1
1000	10	1000_0000
100_0000	10	1

### 〈分析一〉

$a_0=01000000$ ， $a_1=01000000$ ， $a_2=00001000$ ， $Out=01000000$

### 〈結論一〉

有兩個輸入為 01000000，因此中位數為輸入數值相同者，輸出結果為 01000000。計算結果正確

### 〈分析二〉

$a_0=00000010$ ， $a_1=00010000$ ， $a_2=00000010$ ， $Out=00000010$

### 〈結論二〉

有兩個輸入為 00000010，因此中位數為輸入數值相同者，輸出結果為 00000010。計算結果正確

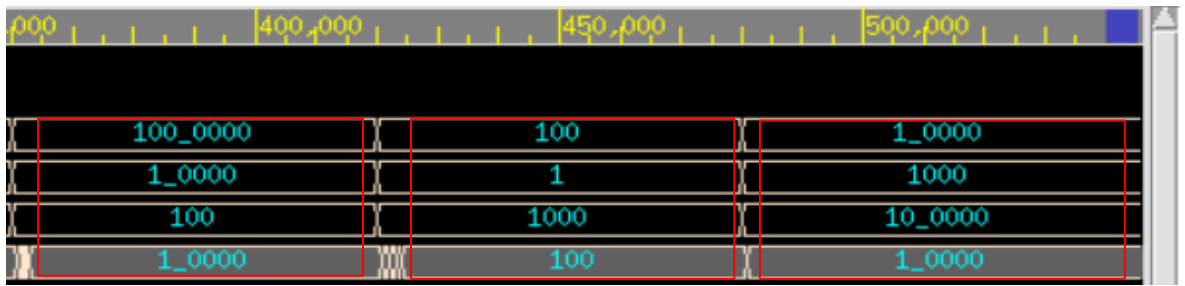
### 〈分析三〉

$a_0=00000001$ ， $a_1=00000001$ ， $a_2=10000000$ ， $Out=00000001$

### 〈結論三〉

有兩個輸入為 00000001，因此中位數為輸入數值相同者，輸出結果為 00000001。計算結果正確

### (三)、輸入皆不同



100_0000	100	1_0000
1_0000	1	1000
100	1000	10_0000
1_0000	100	1_0000

#### 〈分析一〉

$a_0=01000000$ ， $a_1=00010000$ ， $a_2=00000100$ ， $Out=00010000$

#### 〈結論一〉

中位數為 00010000，輸出結果為 00010000。計算結果正確

#### 〈分析二〉

$a_0=00000100$ ， $a_1=00000001$ ， $a_2=00001000$ ， $Out=00000100$

#### 〈結論二〉

中位數為 00000100，輸出結果為 00000100。計算結果正確

#### 〈分析三〉

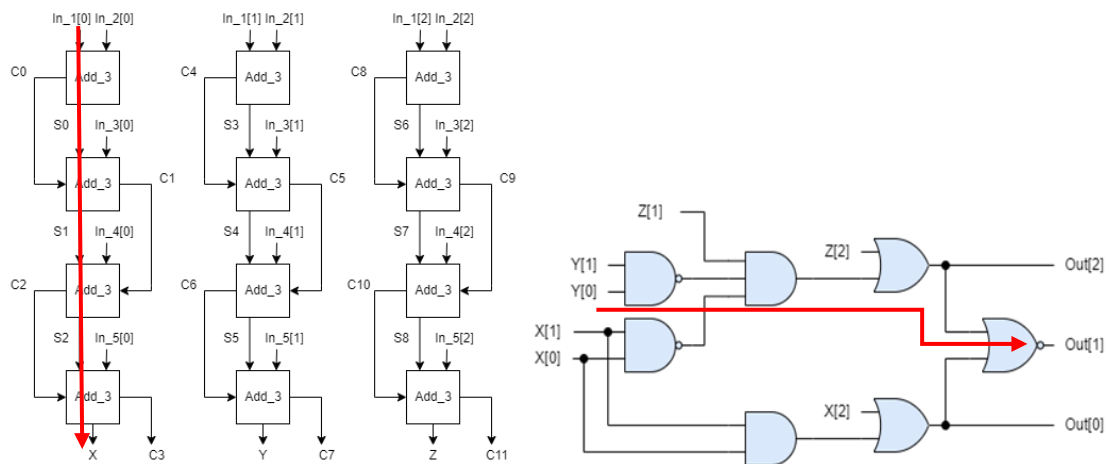
$a_0=00010000$ ， $a_1=00001000$ ， $a_2=00100000$ ， $Out=00010000$

#### 〈結論三〉

中位數為 00010000，輸出結果為 00010000。計算結果正確

# 參、電路架構分析

## 一、 Voting circuit



(Voting circuit 架構圖)

### (一)面積分析

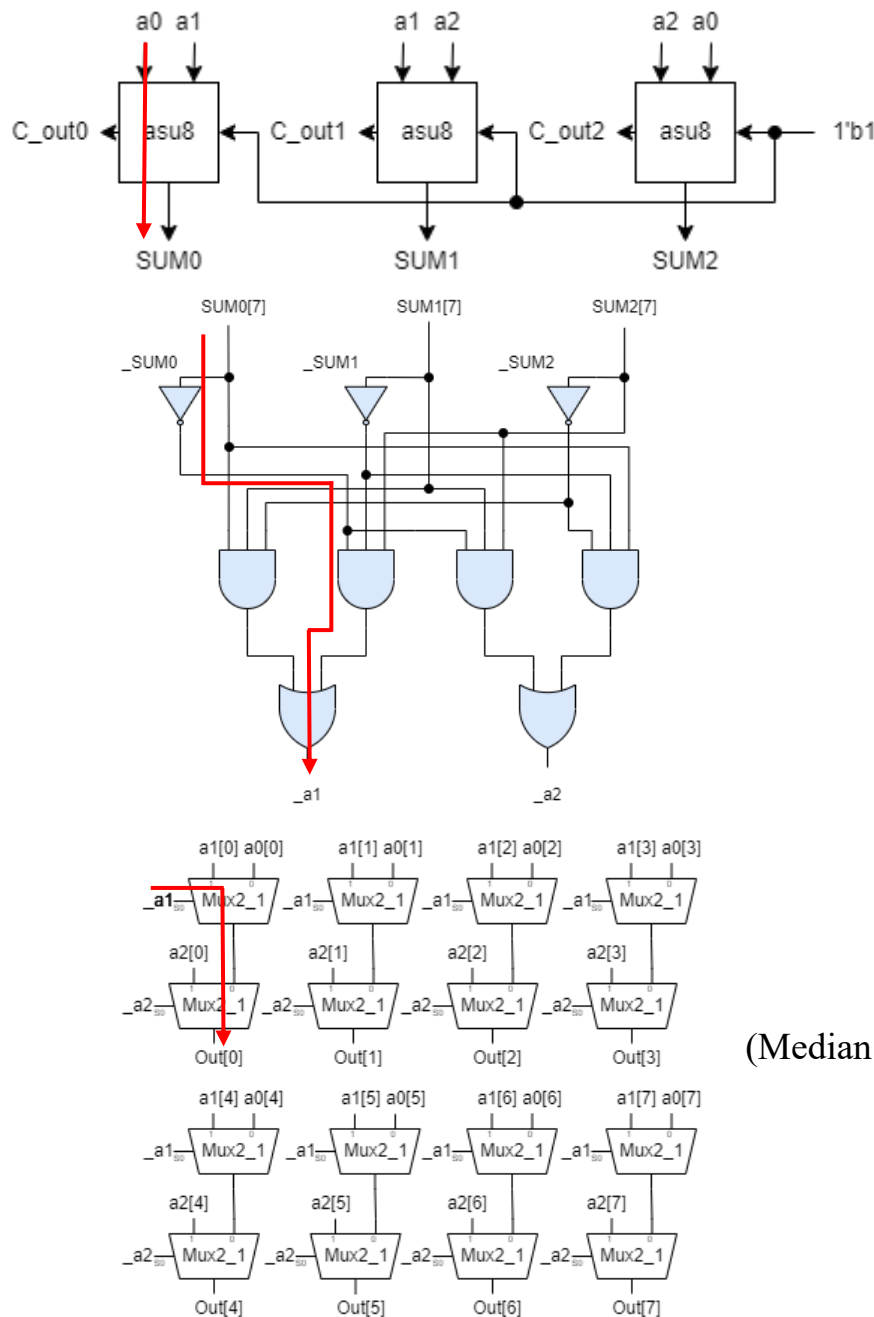
這次作業的 Voting circuit 利用 12 個 3-bits ripple carry adder(Add\_3) 再加上比較數量的邏輯閘電路所組成，其中 1 個 Add\_3 由 3 個 Full adder 所組成，1 個 Full adder 有 5 個邏輯閘，如圖二，因此 1 個 Add\_3 共有 15 個邏輯閘，12 個 Add\_3 則便有 180 個邏輯閘。而比較數量的邏輯閘電路有 7 個邏輯閘，因此整個 Voting circuit 的邏輯閘數量為  $180+7=187$  個邏輯閘。

### (二)最長路徑分析

這次作業的 Voting circuit 的最長路徑為上面架構圖的紅色箭頭所示，其中 1 個 Add\_3 由 3 個 Full adder 所組成，而 Full adder 的 delay 為 2 個 xor 閘  $\Rightarrow 5 \times 2 = 10$  delay，又由上一級進位到產生下一級進位的 delay 為 1 個 and 和 1 個 or 閘  $\Rightarrow 2 \times 2 = 4$  delay，因此 1 個 Add\_3 的 delay 為  $10+4 \times 2 = 18$  delay。接著要計算出輸入位元相加完的結果需要經過 4 個 Add\_3，

其 delay 為  $18 \times 4 = 72$  delay。而比較數量的邏輯閘電路的最長路徑 delay 為 1 個 nand、1 個 and、1 個 or 和 1 個 nor 閘  $\Rightarrow 2 \times 4 = 8$  delay，最後把 4 個 Add\_3 的 delay 和比較數量的邏輯閘電路的 delay 相加便可得到整個 Voting circuit 的最長路徑 delay 為  $72 + 8 = 80$  delay。此電路設計 testbench 的 timescale 為 10ns/100ps，由此估計從輸入到得到計算結果的最大 delay 時間大約為  $80 \times 10 = 800$ ns。

## 二、Median circuit



(Median circuit 架構圖)

## (一) 面積分析

這次作業的 Median circuit 利用 3 個 8-bit ripple carry adder/subtractor (asu8)、產生決定輸出中位數的 Select bit 的邏輯電路以及 16 個 2 to 1 Mux(Mux2\_1)所組成，其中 asu8 由 8 個 Full adder 和 8 個 xor 閘所組成，邏輯閘數量為  $5 \times 8 + 8 = 48$  個，3 個的話則有  $48 \times 3 = 144$  個，而產生決定輸出中位數的 Select bit 的邏輯電路的邏輯閘數量有 9 個，16 個 2 to 1 Mux 邏輯閘數量為  $4 \times 16 = 64$  個，因此整個 Median circuit 的邏輯閘數量為  $144 + 9 + 64 = 217$  個。

## (二) 最長路徑分析

這次作業的 Median circuit 的最長路徑為上面架構圖的紅色箭頭所示，其中 1 個 asu8 由 8 個 Full adder 和 8 個 xor 閘所組成，能得到其 delay 為  $5 + 10 + 4 \times 7 = 43$  delay，而產生決定輸出中位數的 Select bit 的邏輯電路 delay 為 1 個 not、1 個 and 和 1 個 or 閘  $\Rightarrow 1 + 2 \times 2 = 5$  delay。1 個 2 to 1 Mux 的 delay 為 1 個 not、1 個 and 和 1 個 or 閘  $\Rightarrow 1 + 2 \times 2 = 5$  delay，因此可以得到篩選 Median circuit 三個輸入的中位數的兩層 2 to 1 Mux 的 delay 為  $5 \times 2 = 10$ ，最後將各個部件的 delay 相加可得到整個 Median circuit 的最長路徑 delay 為  $43 + 5 + 10 = 58$  delay。此電路設計 testbench 的 timescale 為 10ns/100ps，由此估計從輸入到得到計算結果的最大 delay 時間大約  $58 \times 10 = 580$ ns。

# 肆、問題與討論

## (一) Verilog code 寫法

在這次的 Voting circuit 和 Median circuit 的設計中，Verilog 必須以 combinational 的方式來寫，原先我打算以 data flow 的方式來進行，但後來想了想覺得 data flow 的方式不太好對電路的面積和最長路徑進行分析，因此最後還是回到了 gate level 的方式來建構邏輯電路。然而在想完演算法，寫完 Verilog 後，發現 data flow 相對於 gate level 更容易以直觀的方式來表現複雜的電路，對於之後設計更抽象且複雜的電路，gate level 和 data flow 這兩種方式勢必得交互使用來讓設計電路的過程更有效率。

## (二) 邏輯電路的面積與延遲

在完成這次的 Voting circuit 和 Median circuit 的設計後，對於電路面積和延遲我認為都有蠻大的改進空間，像是 Voting circuit 要計算各個位元總和的 4 個 Add\_3，其實只要 2 個 Full adder 和 1 個 Add\_3 就可以完成，其原因在於當三個位元要用 Full adder 進行相加時，其中 1 個位元可以當成 C\_in 來使用，在加完後只要調整位元順序，便可進行更高位元的加法，這樣不但能減少邏輯閘的數量，對於電路的最長路徑也有明顯的改善，因此之後的電路設計中，在完成初步且最直觀的演算法後，還得再思考是否有更好的方法來優化電路，不然電路的面積和延遲可能不小心就暴增了不少。



## 伍、心得

在完成這次的作業後，深感演算法對邏輯電路的影響，在先前修硬體描述語言時，不論怎麼寫，只要能夠符合電路要求就行，但是一旦要考慮邏輯電路的面積和延遲，寫起來便會綁手綁腳，儘管這些參數對電路會有很大的影響。希望在之後的課程中能學會更多優化電路的技巧以便在設計電路時能夠更有彈性，且提供更多演算法的思路。