

# Lab 1

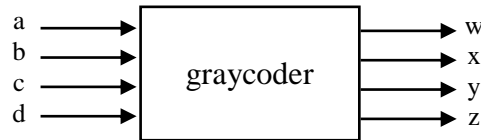
110060027 朱豐蔚

## I. exp\_1

### Design Specification

Input: a, b, c, d.

Output: w, x, y, z.



### Design Implementation

Use truth table and K-map to design the functions:

| a | b | c | d | w | x | y | z |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

$w = a$

| cd \ ab | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 0  | 0  | 0  | 0  |
| 01      | 0  | 0  | 0  | 0  |
| 11      | 1  | 1  | 1  | 1  |
| 10      | 1  | 1  | 1  | 1  |

$x = ab' + a'b = a \oplus b$

| cd \ ab | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 0  | 0  | 0  | 0  |
| 01      | 1  | 1  | 1  | 1  |
| 11      | 1  | 1  | 1  | 1  |
| 10      | 0  | 0  | 0  | 0  |

$y = bc' + b'c = b \oplus c$

| cd \ ab | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 0  | 0  | 1  | 1  |
| 01      | 1  | 1  | 0  | 0  |
| 11      | 1  | 1  | 0  | 0  |
| 10      | 0  | 0  | 1  | 1  |

$z = cd' + c'd = c \oplus d$

| cd \ ab | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 0  | 1  | 1  | 0  |
| 01      | 0  | 1  | 1  | 0  |
| 11      | 0  | 1  | 1  | 0  |
| 10      | 0  | 1  | 1  | 0  |

Code:

```
assign w = a;
assign x = a ^ b;
assign y = b ^ c;
assign z = c ^ d;
```

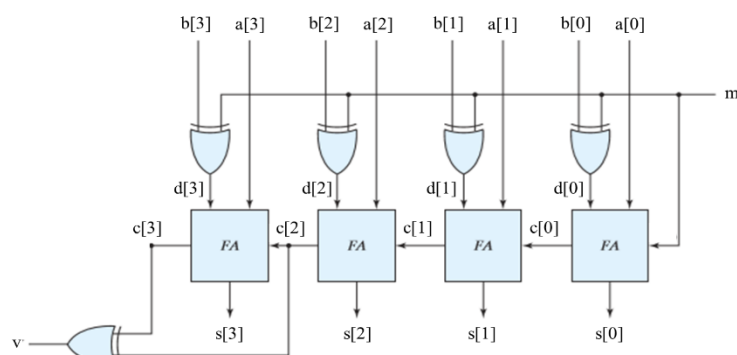
## II. exp\_2

### Design Specification

Input: a[3:0], b[3:0], m.

Output: s[3:0], v.

Wire: c[3:0], d[3:0].



## Design Implementation

增加  $c[3:0]$ ,  $d[3:0]$  以方便撰寫程式，其中：

$c[i]$  為進行第  $i$  個 FA 運算後的進位

$d[i] = b[i] \oplus m$  以方便撰寫後續程式

再套用 FA 公式：

$$s[i] = a[i] \oplus d[i] \oplus c[i-1]$$

$$c[i] = a[i]d[i] + c[i-1](a[i] \oplus d[i])$$

$$v = c[3] \oplus c[2]$$

Code:

```
assign d[0] = b[0] ^ m;
assign d[1] = b[1] ^ m;
assign d[2] = b[2] ^ m;
assign d[3] = b[3] ^ m;

assign s[0] = a[0] ^ d[0] ^ m;
assign c[0] = (a[0] * d[0]) | ((a[0] ^ d[0]) * m);
assign s[1] = a[1] ^ d[1] ^ c[0];
assign c[1] = (a[1] * d[1]) | ((a[1] ^ d[1]) * c[0]);
assign s[2] = a[2] ^ d[2] ^ c[1];
assign c[2] = (a[2] * d[2]) | ((a[2] ^ d[2]) * c[1]);
assign s[3] = a[3] ^ d[3] ^ c[2];
assign c[3] = (a[3] * d[3]) | ((a[3] ^ d[3]) * c[2]);

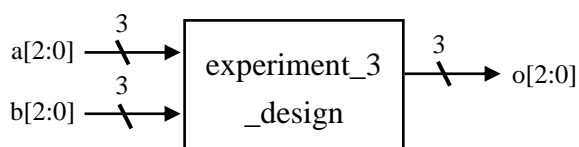
assign v = c[3] ^ c[2];
```

## III. exp\_3

### Design Specification

Input:  $a[2:0]$ ,  $b[2:0]$ .

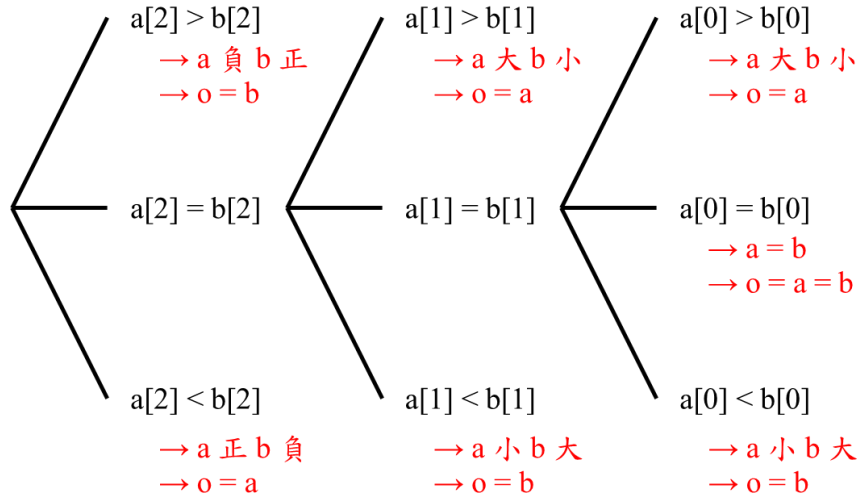
Output:  $o[2:0]$ .



## Design Implementation

為使用 if 來做撰寫，先畫出樹狀圖整理 o 和 a, b 之間的關係：

(題目要求 a, b 為 signed numbers，需特別注意 a[2], b[2] 的大小關係)



Code:

```

reg [2:0] o;

always @(a or b)
begin
    if (a[2] > b[2])
        o[2:0] = b[2:0];
    else if (a[2] < b[2])
        o[2:0] = a[2:0];
    else
        if (a[1] > b[1])
            o[2:0] = a[2:0];
        else if (a[1] < b[1])
            o[2:0] = b[2:0];
        else
            if (a[0] > b[0])
                o[2:0] = a[2:0];
            else
                o[2:0] = b[2:0];
    end
end
  
```

## Discussion

由於此三題都來自於以往的邏輯設計課，所以在設計的思路上，我沒有遇到太多的障礙，反而是在第二題 Adder/Substraterk 的 code 撰寫中，我發現了以往紙面上書寫的 function，因為需要電腦打入而變得更加冗長和難以閱讀，因此，我加入了 d[3:0]的想法，也成功簡化撰寫，使程式更加明朗，debug 上也更加順利。

在實驗結果分析中，一開始由於不熟悉操作，因此在結果的判讀上遇到了很大的障礙，例如說第三題中的數字，電腦自動顯示是 unsigned numbers，因此數字的比大小就會不明朗清楚，這一度讓我以為是結果錯誤，而後來詢問助教，也才學會透過改變模擬結果中的參數 radix，來達到更好判讀結果的方法，這也是我這次實驗的收穫之一。

## Conclusion

在 Lab 0 當中，只是讓我們照著已經寫好的程式去打，試著去熟悉 Vivado 的操作，而在 Lab 1 很不一樣的是我們必須從頭去自行設計，這個差別就讓第一次自行接觸設計 Verilog 的我們在看到繁雜的電腦介面時無從下手。

而經過了這三題的演練，我已經學會了最基本的操作，不會像第一次自行設計時那樣徬徨，我想這是我在 Lab 1 中最大的收穫。另外一方面，題目的選擇大多是來自上學期的邏輯設計課中，也讓我能夠複習邏輯設計的內容，以做好面對這學期實驗課的準備。

## References

Adder/Substrater 圖片參考來源(有自行修改):

<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.answersproject.com%2FExpertAnswers%2Fverilog-write-code-following-question-4-bit-adder-subtractor-overflow-detection-cascading-&psig=AOvVaw0UUt5byfhPkIjvEOwzJbOW&ust=1646795320956000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCKD5zebEtfYCFQAAAAAdAAAAABAD>