112 邏輯設計實驗二

Lab1 4-bits (unsigned) Add/Subtract Unit Design

實驗報告

組別：第三組

學號：10927202 / 10927207

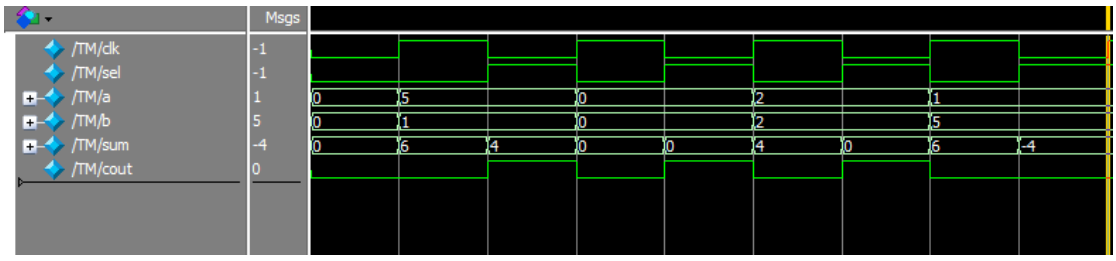
姓名：陽彩柔 / 蒲品憶

1. Verilog Code

```
module TM;

reg clk, sel;
reg [3:0] a, b;
wire [3:0] sum ;
wire cout ;

AddSub addsub(a, b, sum, cout, sel);

initial clk = 1'd0;
always #5 clk = ~clk;

initial
begin
    a = 4'd0;
    b = 4'd0;
    sel = 0;

#5  a = 4'd5;
    b = 4'd1;
    sel = 0;

#5  a = 4'd5;
    b = 4'd1;
    sel = 1;

#5  a = 4'd0;
    b = 4'd0;
    sel = 0;

#5  a = 4'd0;
    b = 4'd0;
    sel = 1;

#5  a = 4'd2;
    b = 4'd2;
    sel = 0;

#5  a = 4'd2;
    b = 4'd2;
    sel = 1;

#5  a = 4'd1;
    b = 4'd5;
    sel = 0;

#5  a = 4'd1;
    b = 4'd5;
    sel = 1;
end
endmodule
```

```
module FA(a, b, c, sum, cout);

    input a, b, c;
    output cout, sum;
    wire tmpAB, cANDtmpAB, aANDb;

    assign tmpAB = a ^ b;
    assign cANDtmpAB = c & tmpAB;
    assign aANDb = a & b;
    assign sum = tmpAB ^ c;
    assign cout = cANDtmpAB | aANDb;
endmodule


module AddSub(a, b, sum, cout, sel);

    output [3:0] sum ;
    output cout ;
    input [3:0] a, b;
    input sel;
    wire[3:0] c, tmpB;

    assign tmpB[0] = ( sel == 0 ) ? b[0] : (b[0] ^ sel) ;
    assign tmpB[1] = ( sel == 0 ) ? b[1] : (b[1] ^ sel) ;
    assign tmpB[2] = ( sel == 0 ) ? b[2] : (b[2] ^ sel) ;
    assign tmpB[3] = ( sel == 0 ) ? b[3] : (b[3] ^ sel) ;

    FA fa0(a[0], tmpB[0], sel, sum[0], c[0]);
    FA fa1(a[1], tmpB[1], c[0], sum[1], c[1]);
    FA fa2(a[2], tmpB[2], c[1], sum[2], c[2]);
    FA fa3(a[3], tmpB[3], c[2], sum[3], cout);


endmodule
```

2. 模擬 waveform



說明：因為是在做 unsigned 加減法，若夠減的話 cout 為 1，反之 cout 為 0，例如最後一個測試檔(1 － 5) = 0001 + 1010 + 1 = 1100 ，cout = 0，代表不夠減；(5 －1) = 0101 + 0001 + 1 = 0111，cout = 1，代表夠減。

3. 心得報告

陽彩柔：雖然每學期都會碰到 Verilog，但只要過了一個假期，幾乎就會忘掉很多，在這次的實驗中，我們花了一些時間討論並複習該怎麼打，過程中，我們還額外寫了一個轉換 2 補數的 function，但後來發現是多此一舉，還有我們在參考圖的時候並沒有直接按照圖上邏輯，而是相信自己的判斷，結果嘗試了很久還是沒成功，最後只好去看圖並照著邏輯做出來。透過這次實驗，讓我對 verilog 受熟悉及了解。

蒲品憶：因為過了一個寒假，所以 verilog 的程式寫法都忘光光了，需要重新複習，因此這次 Lab01 做的比較慢，而且一開始我們有些方向錯誤，所以繞了一很多彎路，像是 FA 我們應該要按照圖片用邏輯判斷，但是我們寫成條件判斷，所以有一些漏洞，還有加減法器只要用 sel 判斷就好，結果我們又另外寫一個 function 獨立出來把 b 轉成 2 補數，雖然沒有錯，但是多此一舉，造成程式碼重複有點累贅，所以我們之後也做修改，這次實驗讓我撿回很多記憶，非常充實。