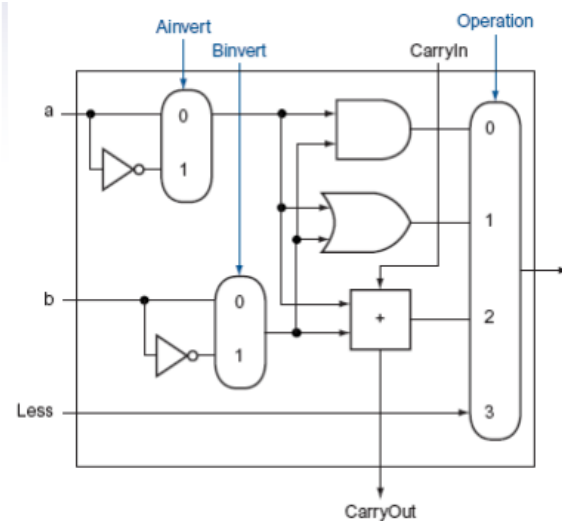
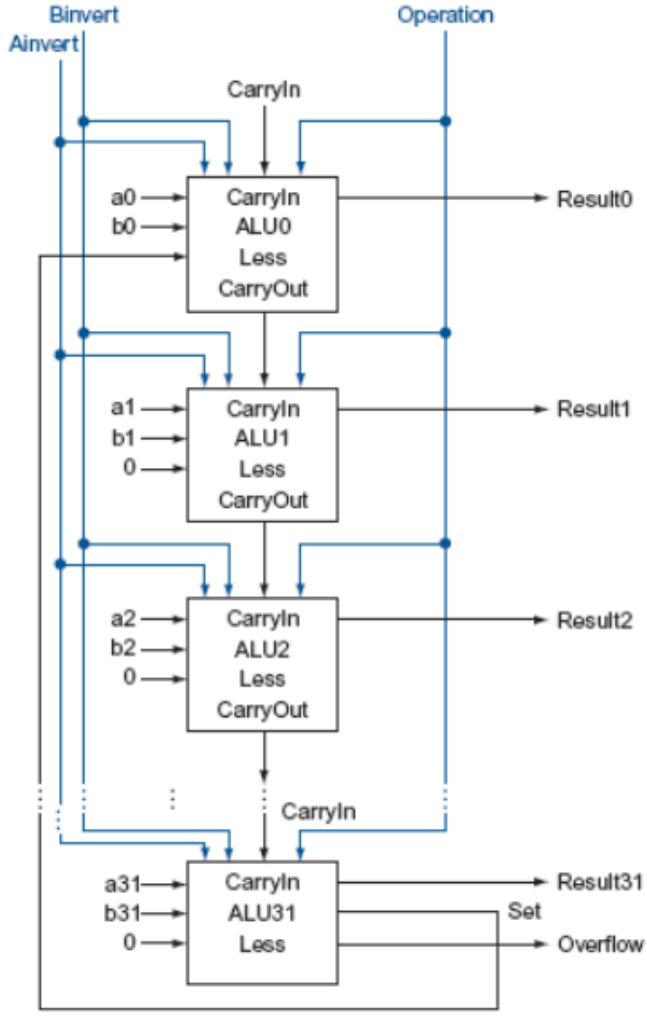
Computer Organization

Architecture diagrams:

1-Bit ALU（取自講義）:



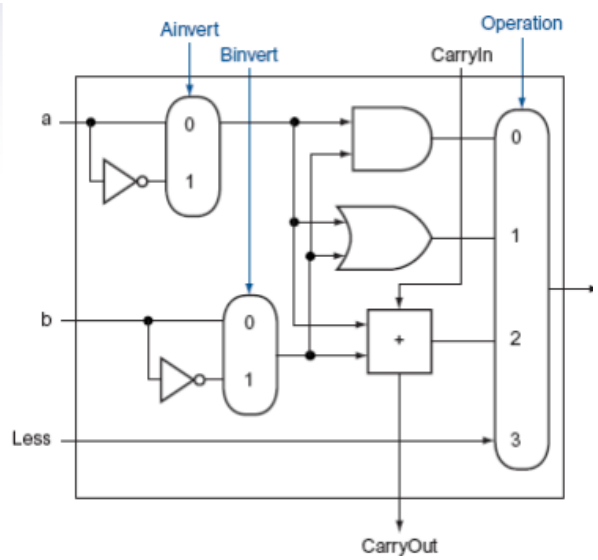
32-bit ALU（取自講義）: 使用 32 個 1-bit ALU 連結成 32-bit ALU



Hardware module analysis:

1-bit ALU 輸入設計:

ALU Action	Name	ALU Control Input
And	And	0000
OR	Or	0001
Add	Addition	0010
Sub	Subtraction	0110
Nor	Nor	1100
Slt	Set less than	0111



根據以上兩張圖可歸納出下列表格:

(此為 32-bit ALU Operation 對應每個 1-bit ALU 之 input)

	A_invert	B_invert	Cin (LSB)	operation
AND 0000	0	0	0	00
OR 0001	0	0	0	01
ADD 0010	0	0	0	10
SUB 0110	0	1	1	10
NOR 1100	1	1	0	00
SLT 0111	0	1	1	11

可得出下列等式:

(1) A_invert = Operation[3]

(2) B_invert = Operation[2]

(3) Cin (LSB) = Operation[3] & Operation[2]

(4) operation (2-bit) = {Operation[1], Operation[0]}

For LSB:

```
alu_top A0( .src1(src1[0]),
            .src2(src2[0]),
            .less(set),
            .A_invert(ALU_control[3]),
            .B_invert(ALU_control[2]),
            .cin(ALU_control[1]&ALU_control[2]),
            .operation(ALU_control[1:0]),
            .result(result_arr[0]),
            .cout(cout_arr[0]));
```

For others:

```
genvar i;
for(i=1; i<=31; i = i+1)
    begin
    alu_top A1( .src1(src1[i]),
                .src2(src2[i]),
                .less(0),
                .A_invert(ALU_control[3]),
                .B_invert(ALU_control[2]),
                .cin(cout_arr[i-1]),
                .operation(ALU_control[1:0]),
                .result(result_arr[i]),
                .cout(cout_arr[i]));
    end
```

差異:

(1) LSB 的 less 是連接 MSB 所產生的 set 訊號，而其餘的皆被設為 0

(2) LSB 的 cin 來自 32-bit ALU 的 operation，用於區分加減法

其餘的皆為計算的進位

(A - B = A + B_bar +1, cin for LSB is set to 1)

Set, Overflow, Zero 設計:

看 MSB (sign bit) 和前一 bit 之進位列出等式 (下圖為 A - B)

A (src1[31])	B (Src2[31])	C(Cout[30])	set	overflow
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	0

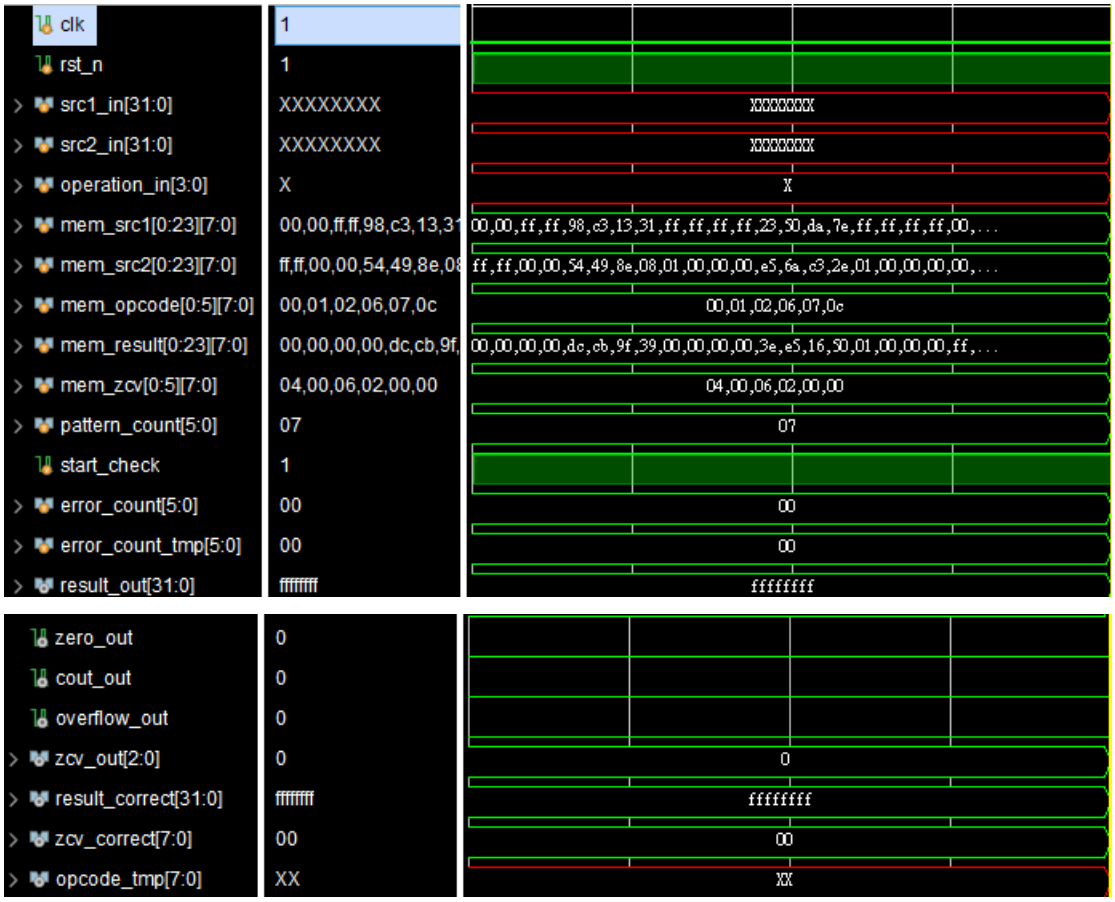
(1) Set = A' B' C' +ABC' +A' B

(2) Overflow = A' BC + AB' C'

(3) Zero = 1 if result = 0 , otherwise 0

Set, overflow 只看 src1, src2, cin 就可以得知，不用等待 result、cout

Experiment result:



Problems you met and solutions:

在討論區有人發問 ALU 是 singed number 還是 unsigned number，助教回答 unsigned，我一直沒有理解，後來我還是把 MSB 當作 sign bit 去設計 overflow 和 set。

在設計 set 和 overflow 的時候思路很不清晰，後來畫了表格，化簡等式，發現這樣比較容易。

Summary:

這次的作業，剛開始看覺得很容易，但開始寫之後，發現自己對於二進位的運算和 verilog 還是滿不熟悉的，之後多寫一些後思考應該會比現在順利。