微算機系統 Fall 2020
Microprocessor Systems

Instructor：Yen-Lin Chen(陳彥霖), Ph.D. Professor

Dept. Computer Science and Information Engineering

National Taipei University of Technology

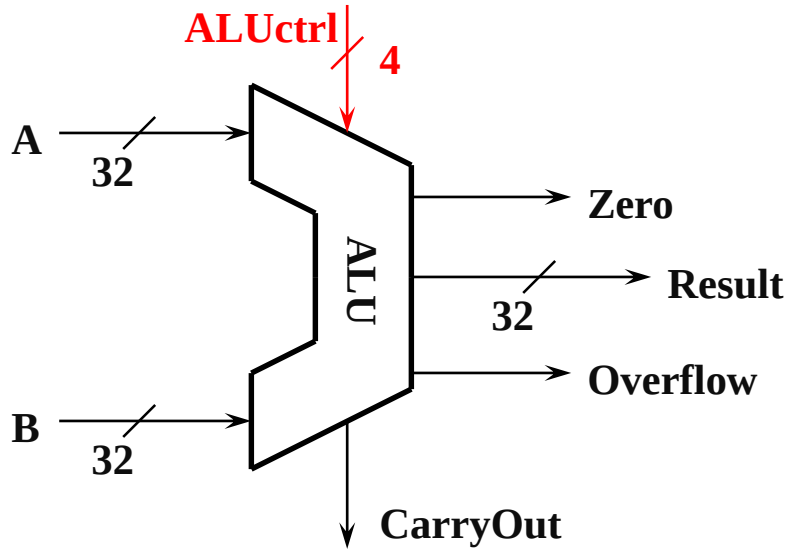
實驗三：七位元簡易運算器

Arithmetic Logic Unit(ALU)

繳交規定

- 檢查期限: 10/21(三) 中午12:30截止
- 報告繳交期限:10/26(一)上課前上傳至北科i學園PLUS->作業
- 繳交格式:北科i學園PLUS->文件(Document)->微算機系統_報告格式
- 詳細繳交規定請參照**2020 Fall 微算機系統**社團發文

Functional Specification of ALU

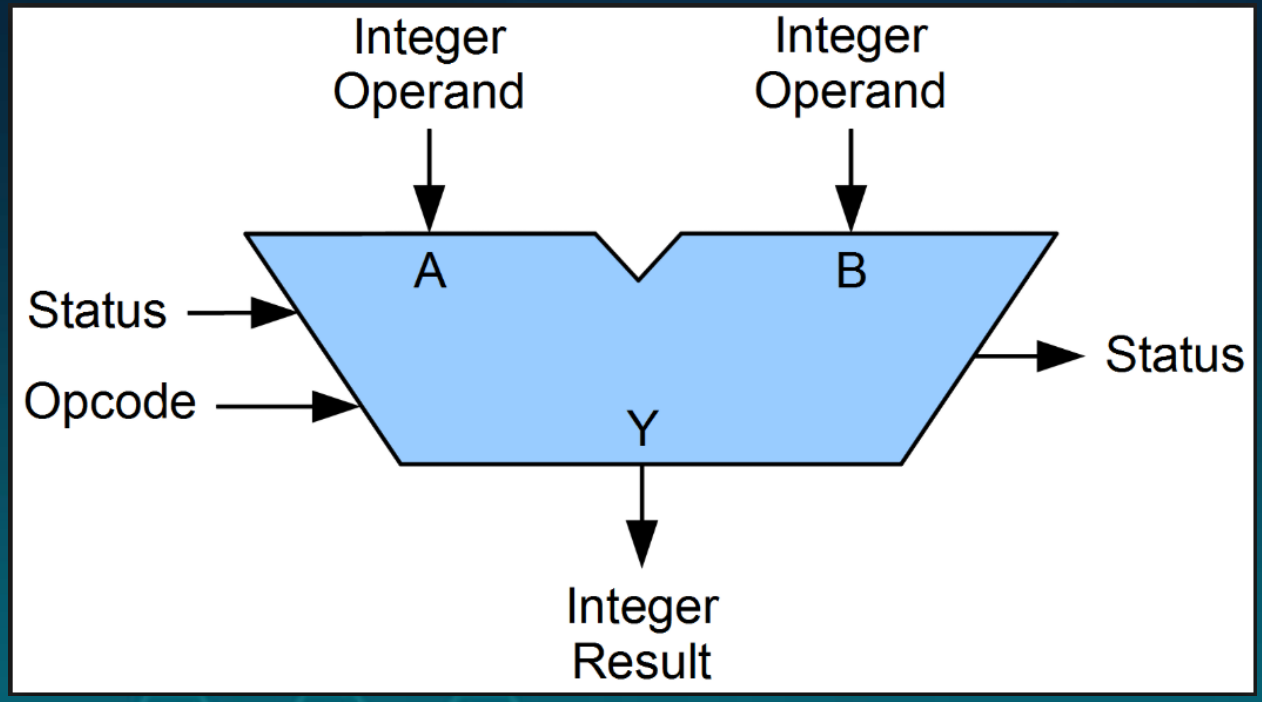


ALU Control (ALUctrl)	Function
0000	and
0001	or
0010	add
0110	subtract
0111	set-on-less-than
1100	nor

ALU基本概念

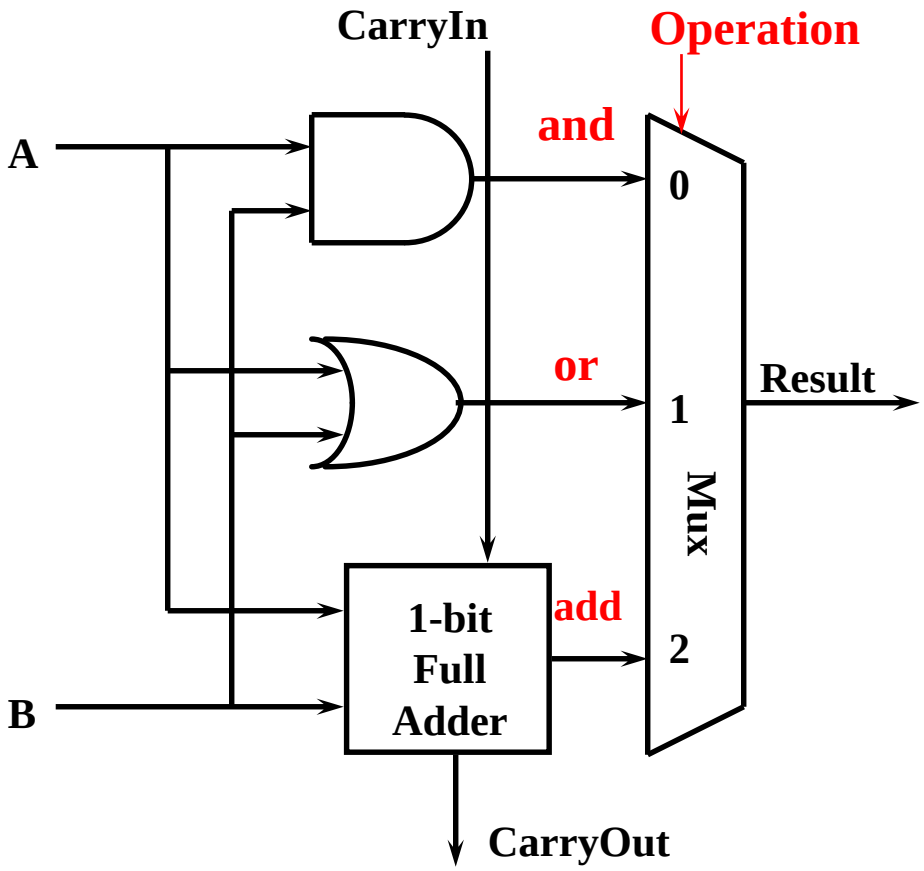
- 算術邏輯單元(Arithmetic Logic Unit, ALU)是中央處理器(CPU)的執行單元。
- 是所有中央處理器的核心組成部分，由“And Gate”和“Or Gate”構成的算術邏輯單元。
- 主要功能是進行二進制的算術運算，如加減乘(不包括整數除法)。基本上，在所有現代CPU體系結構中，二進制都以二補數的形式來表示

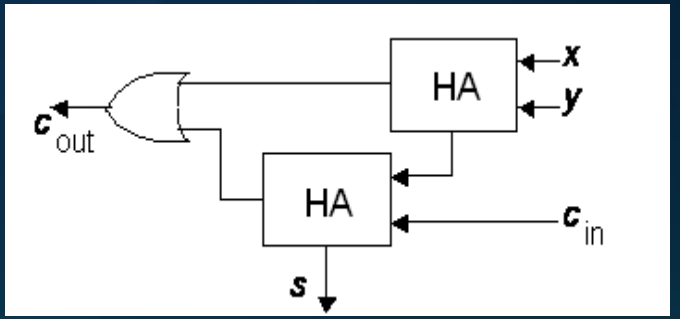
ALU基本架構



ALU基本架構示意圖

1-BIT ALU





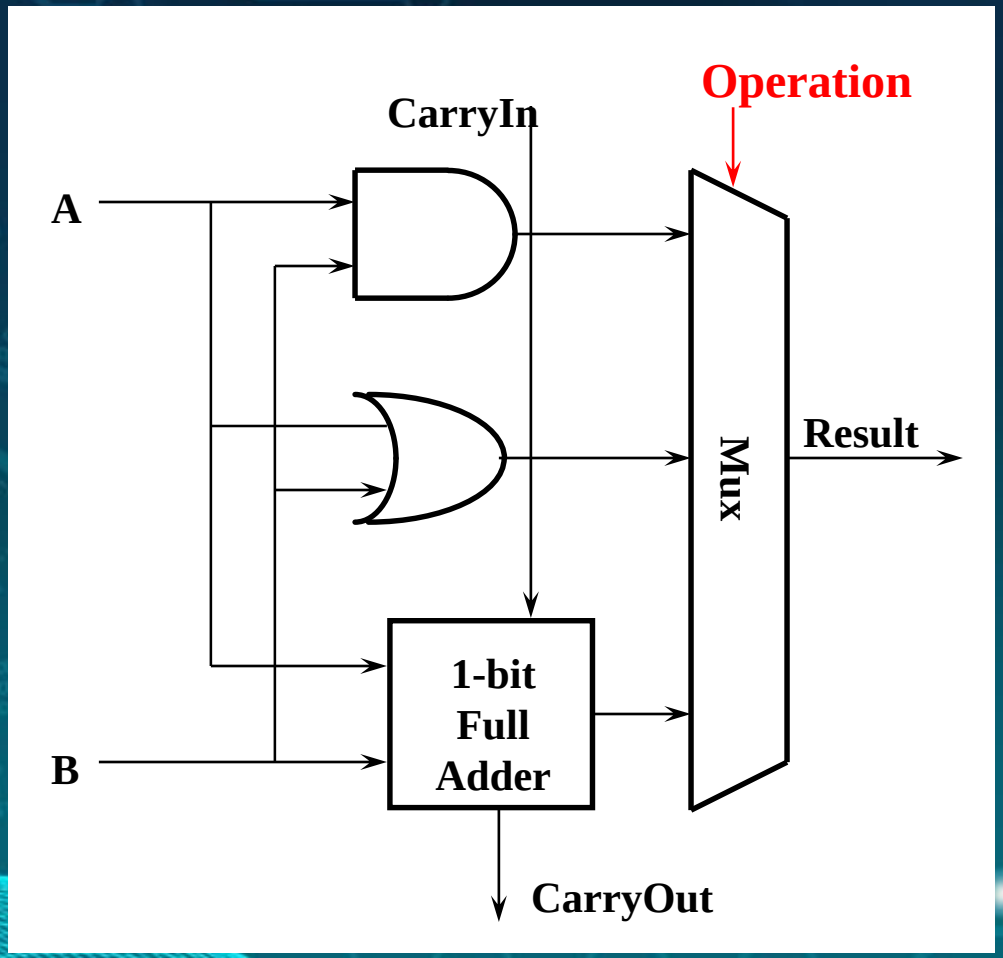
Full Adder implemented by two Half Adders

Inputs			Outputs	
X	Y	Carry In	Sum	Carry Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

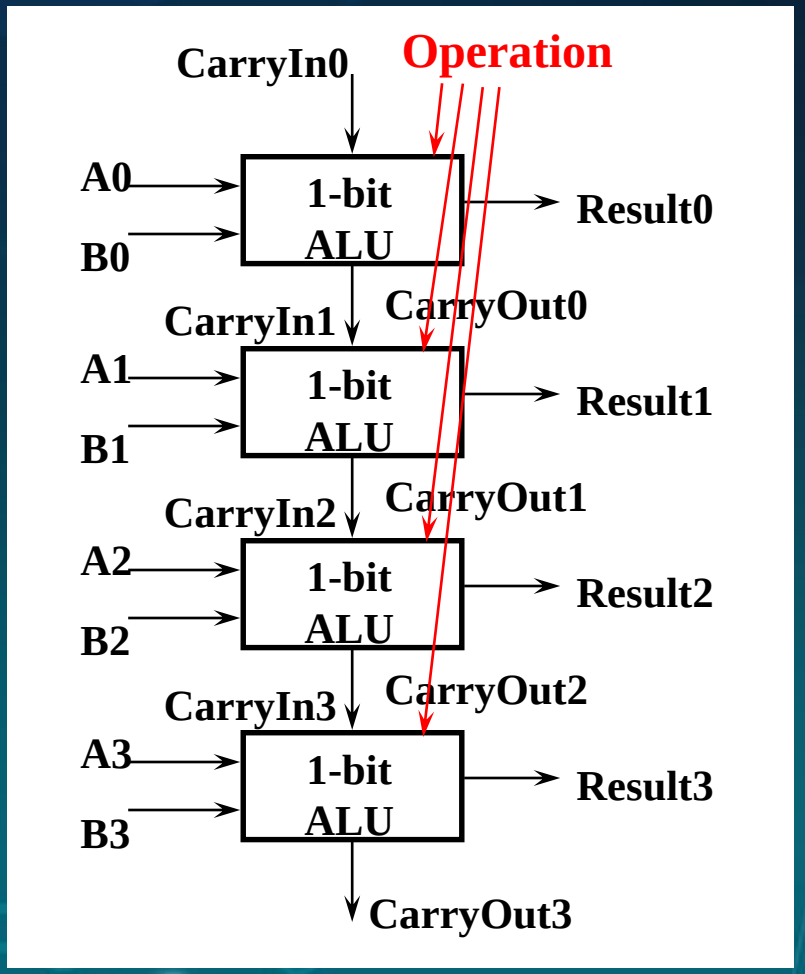
Truth Table of a Full Adder

4-bit ALU

1-bit ALU

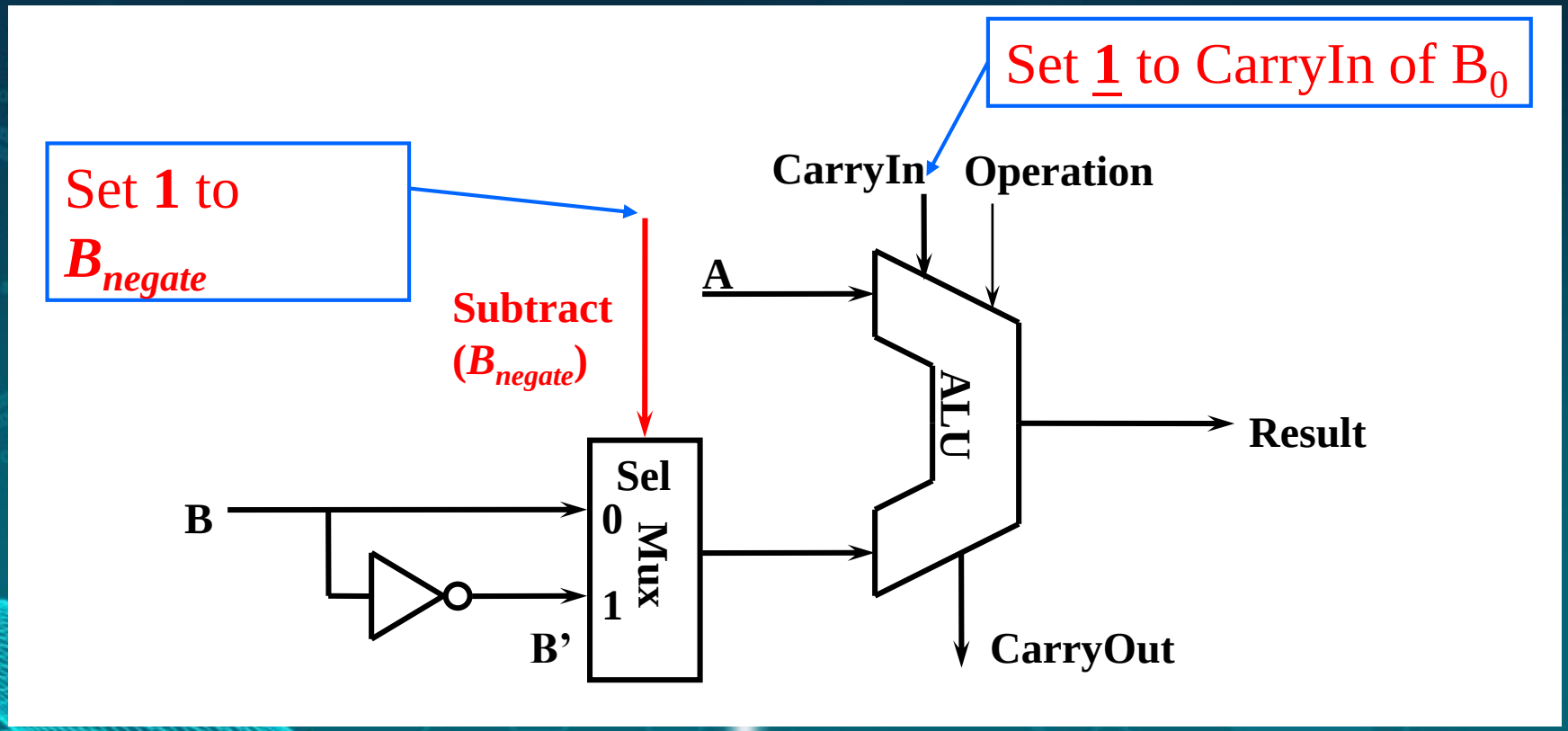


4-bit ALU



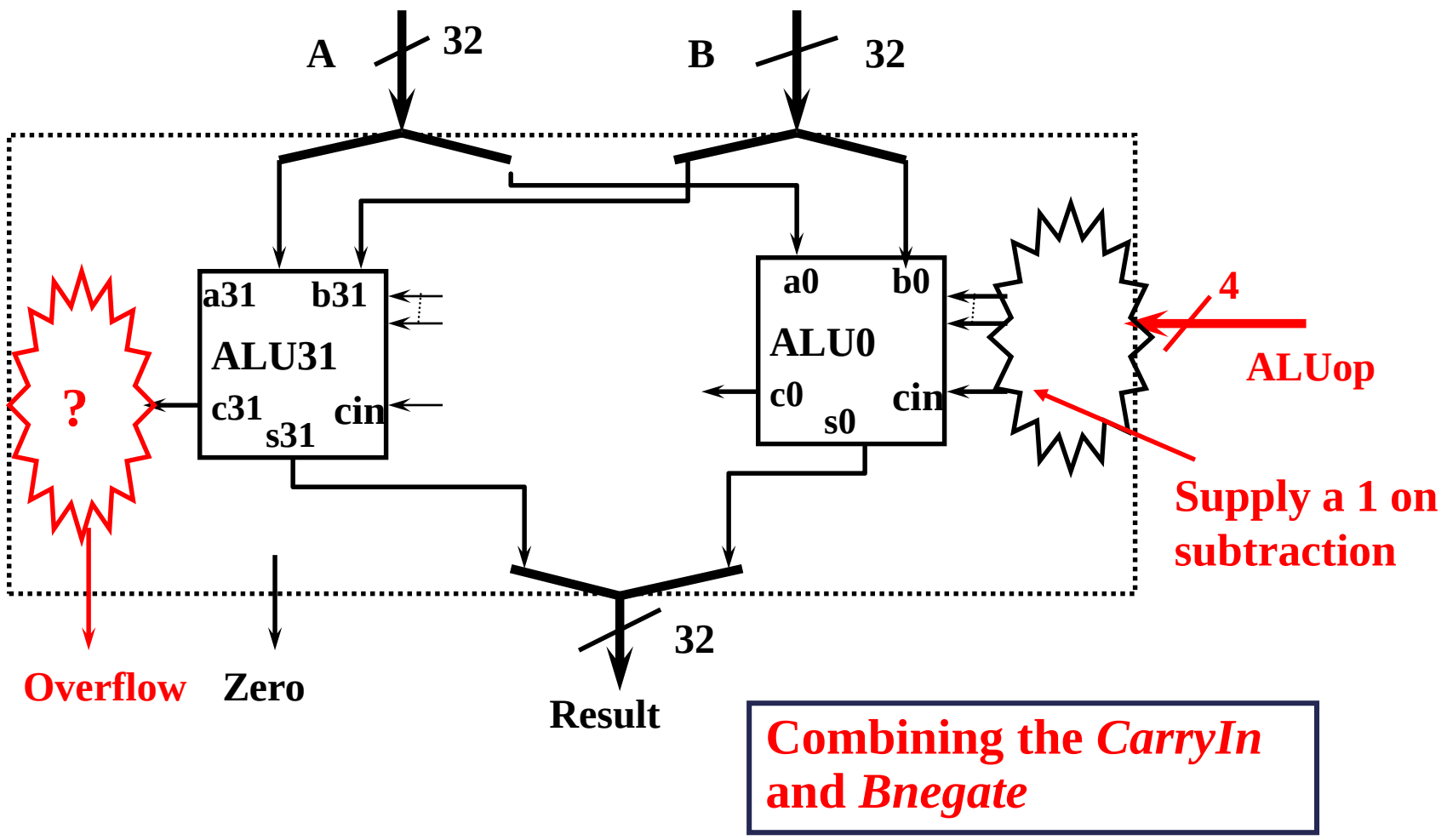
How about Subtraction?

- 2’ s complement: take inverse of every bit and add 1 (at c_{in} of first stage)
  - A - B = A + (-B) = A + (B’ + 1) = A + B’ + 1
  - B’ is *Bit-wise inverse of B*



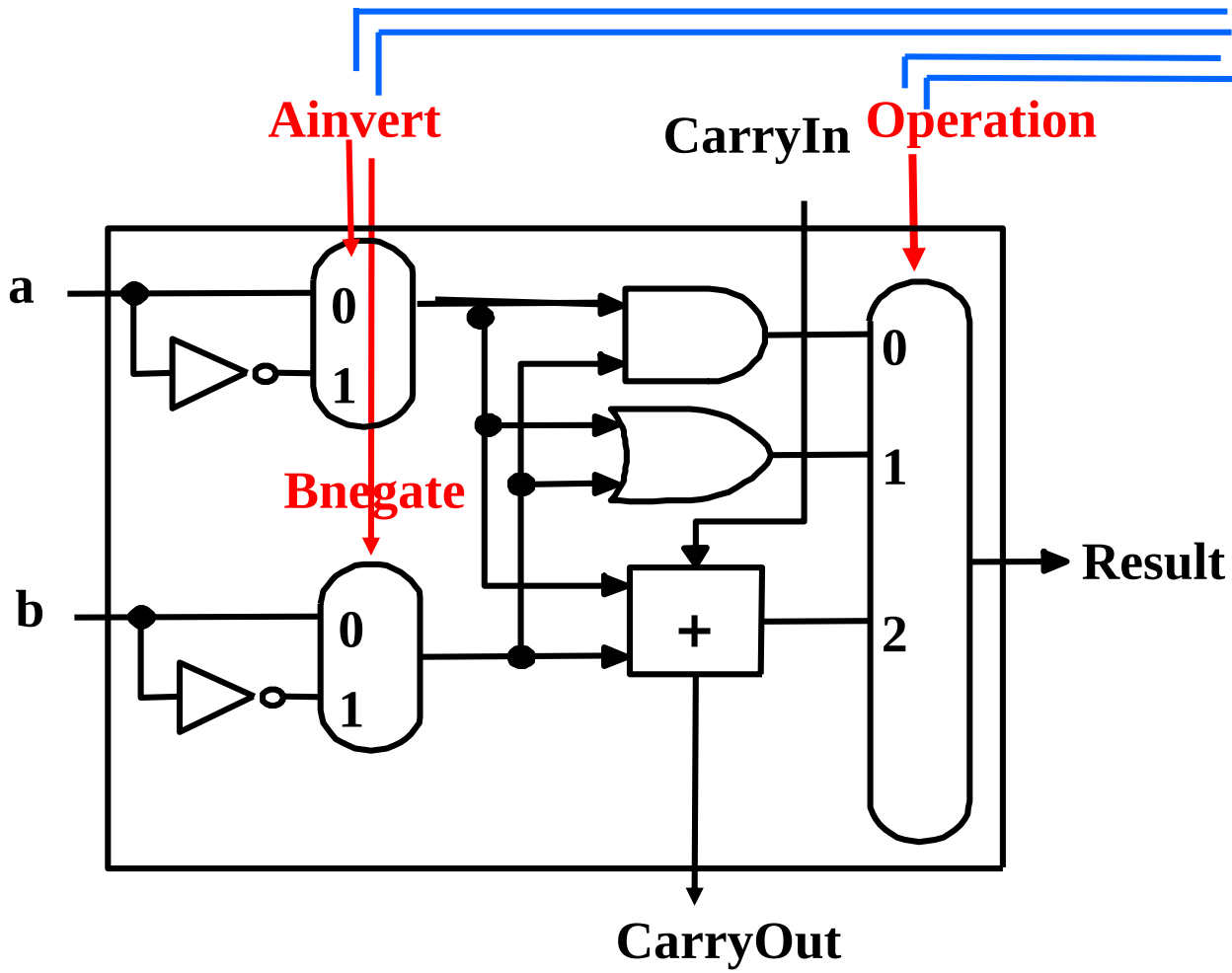
Revised Diagram

- LSB and MSB need to do a little extra



Nor Operation

- A nor B = (not A) and (not B)



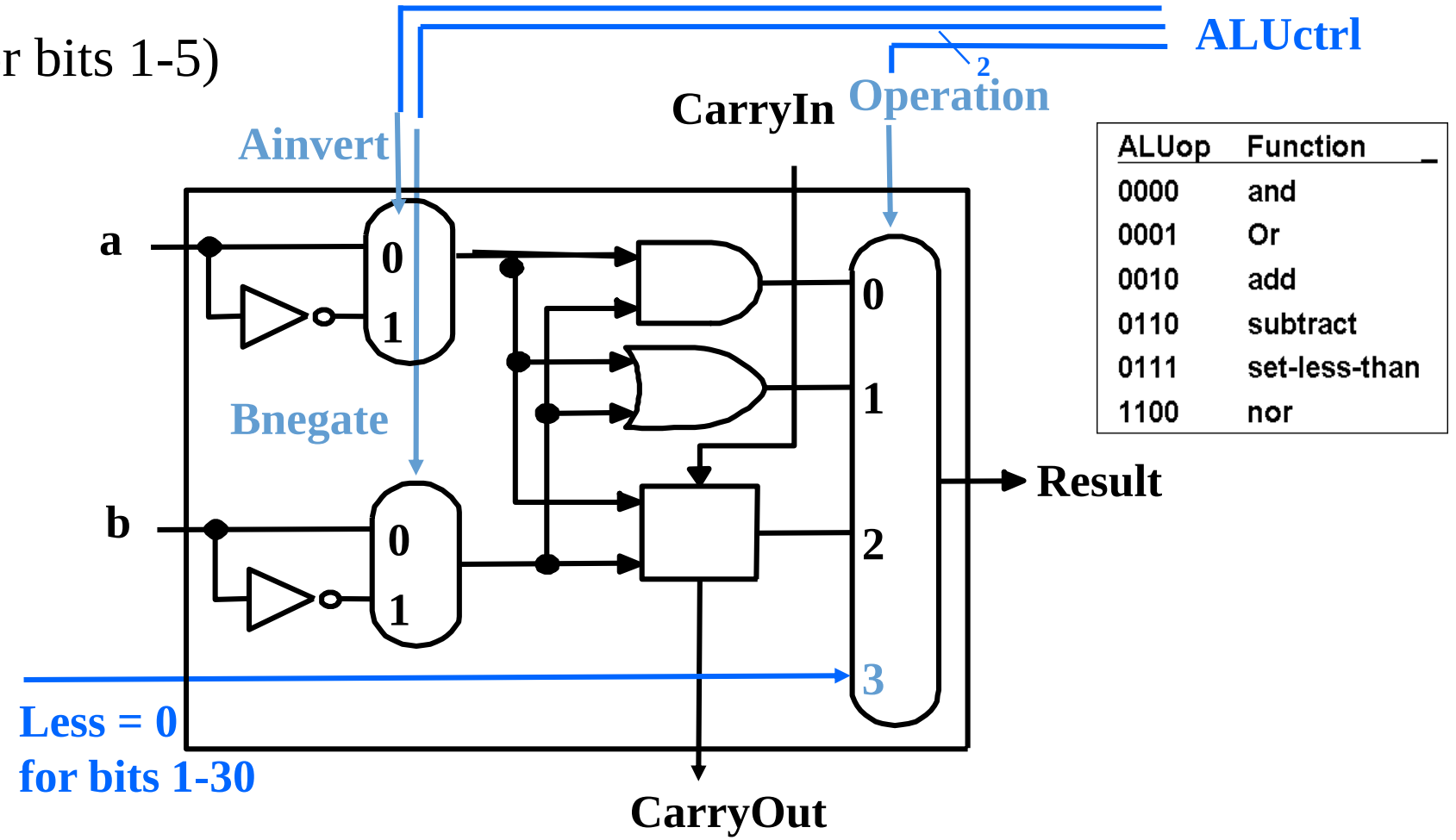
ALUop	Function
0000	and
0001	Or
0010	add
0110	subtract
0111	set-less-than
1100	nor

Set on Less Than

- **if (A-B) < 0 then Result = 1**

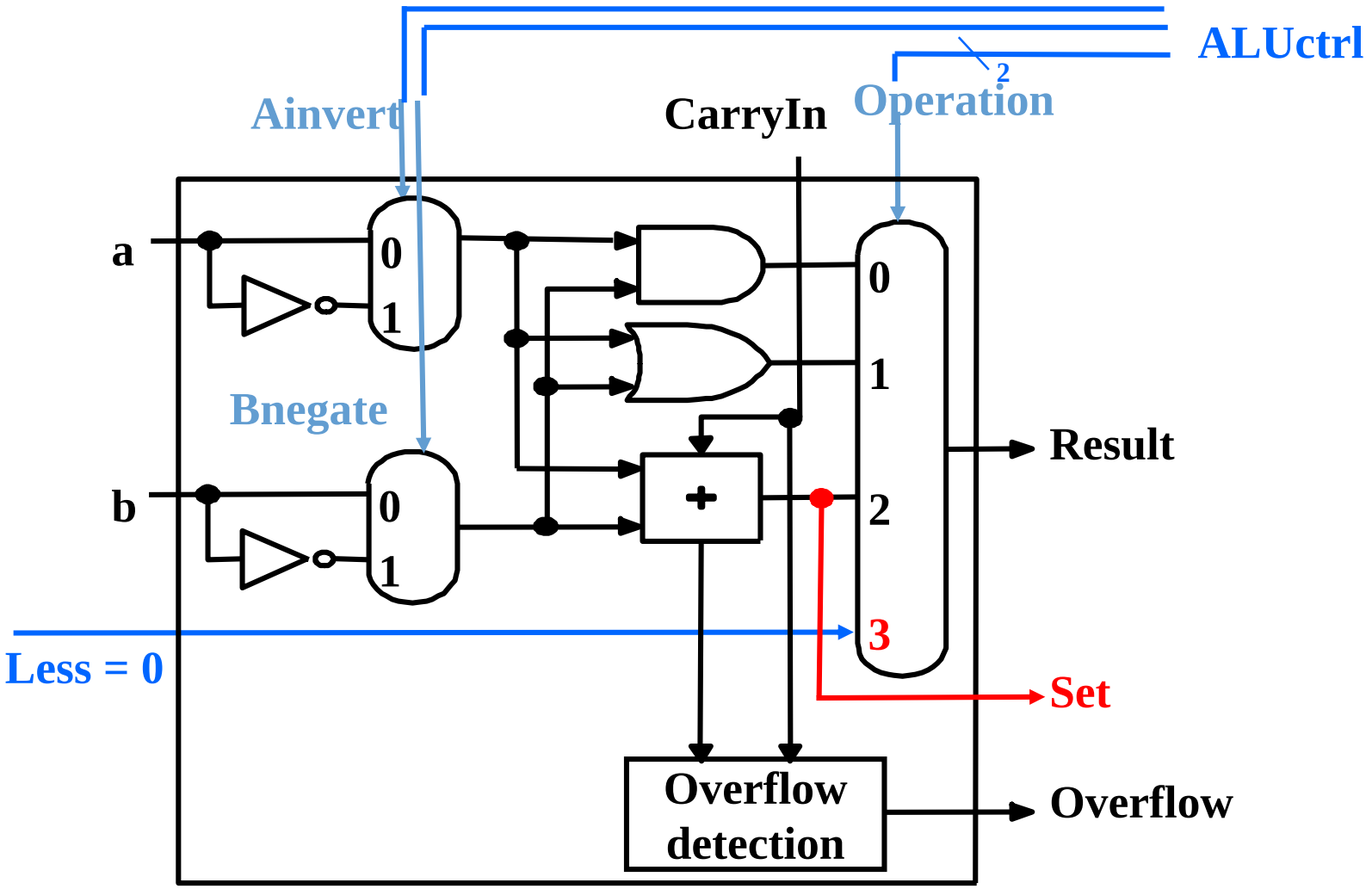
1-bit in ALU

(for bits 1-5)



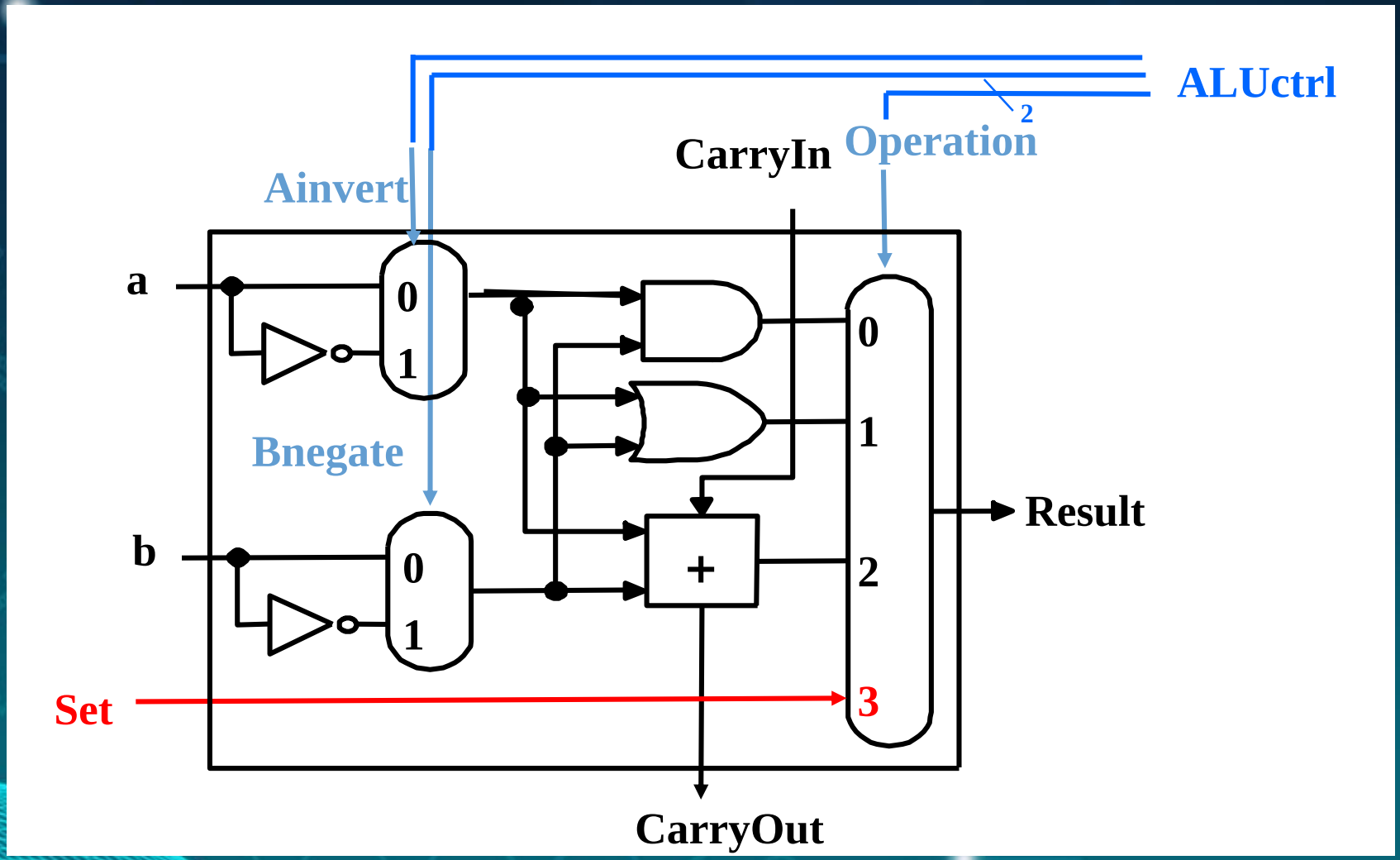
Set on Less Than (cont'd)

- Sign bit in ALU



Set on Less Than (cont'd)

- Bit 0 in ALU



GENERATE敘述 FOR i IN .. TO .. GENERATE

- VHDL提供 FOR GENERATE敘述來描述規則的結構階層性程式碼。
- GENERATE敘述必須有標籤，因此我們在程式碼中用標籤G1。

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY dec4to16 IS
        PORT (      w      : IN        STD_LOGIC_VECTOR(3 DOWNTO 0) ;
                    En     : IN        STD_LOGIC ;
                    y      : OUT       STD_LOGIC_VECTOR(0 TO 15) ) ;
END dec4to16 ;

ARCHITECTURE Structure OF dec4to16 IS
        COMPONENT dec2to4
                PORT (      w      : IN        STD_LOGIC_VECTOR(1 DOWNTO 0) ;
                            En     : IN        STD_LOGIC ;
                            y      : OUT       STD_LOGIC_VECTOR(0 TO 3) ) ;
        END COMPONENT ;
        SIGNAL m : STD_LOGIC_VECTOR(0 TO 3) ;
BEGIN
        G1: FOR i IN 0 TO 3 GENERATE
                Dec_ri: dec2to4 PORT MAP ( w(1 DOWNTO 0), m(i), y(4*i TO 4*i+3) );
                G2: IF i=3 GENERATE
                        Dec_left: dec2to4 PORT MAP ( w(i DOWNTO i-1), En, m ) ;
                END GENERATE ;
        END GENERATE ;
END Structure ;
```

四對十六 二進位解碼器的階層性程式碼

HINT

- if 有多個條件時的語法如下：
  if (signalA>signalB) and (signalB>signalC)
- 將1bit的ALU寫成一個component，輸入有四個A，B，less，carryin，輸出有三個result，set，carryout，可參考下方圖。

- 輸入的less → 當bit0時， 此less輸入bit6的set (參考P.13圖)；當bit1~6時，less輸入'0' (參考P.11,P.12圖) 。
- 輸出的set → 當bit0~5時，雖然set並沒有輸出，但是仍然需要給他設定一個值輸出(此值不會用到， 參考P.13,P.11圖)；當bit6時，就必須輸出set，並將此set給bit0的less輸入(參考P.12圖) 。

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

ENTITY onebitALU IS
    PORT( A, B, less, carryin : IN STD_LOGIC;
          opcode : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
          result, set, carryout : OUT STD_LOGIC);
END;

ARCHITECTURE LogicFunc OF onebitALU IS

BEGIN

    Your code

END LogicFunc;
```

HINT

- 用FOR迴圈來串出7-bit ALU，當bit0時輸入a,b,carryin=0及bit6的set，輸出result,carryout及set(但這個set不需要用到)；當bit6時輸入a,b,carryin及less=0，輸出result,carryout及set(這個set需給bit0的less輸入)。

實驗目標 七位元簡易運算器

- 使用1-bit ALU串成一個可進行簡易運算的7-bit ALU。
- 運算功能包括and, or, add, subtract, set-less-than, nor
- 以兩個7-bit作為輸入，並運算結果顯示於七段顯示器上。
- Operation code 以4-bit作為輸入。
- 因此輸入為 7 + 7 + 4 = 18-bit。
- 請使用for generate、if generate及component完成本次實驗。

實驗要求及配分

- 完成1-bit ALU可得30%
- 使用for generate 、if generate及component完成7-bit ALU並連接7-Segment可得40% (未使用for generate 、if generate及component只可得30%)
- 實驗報告30%
- 需驗證所有ALU指令，加減法範圍為+63 ~ -64(只驗證運算結果為正數的加減法)。
- 以Component方式整合七段顯示器顯示結果，以兩顆7-Segment顯示16進位計算結果
- 不能使用PROCESS敘述

指定腳位 – 輸入

Name	Pin Location
A0	PIN_AB28 **(SW[0])**
A1	PIN_AC28 **(SW[1])**
A2	PIN_AC27 **(SW[2])**
A3	PIN_AD27 **(SW[3])**
A4	PIN_AB27 **(SW[4])**
A5	PIN_AC26 **(SW[5])**
A6	PIN_AD26 **(SW[6])**

Name	Pin Location
B0	PIN_AB26 (SW[7])
B1	PIN_AC25 (SW[8])
B2	PIN_AB25 (SW[9])
B3	PIN_AC24 (SW[10])
B4	PIN_AB24 (SW[11])
B5	PIN_AB23 (SW[12])
B6	PIN_AA24 (SW[13])

Name	Pin Location
OP0	PIN_AA23 (SW[14])
OP1	PIN_AA22 (SW[15])
OP2	PIN_Y24 (SW[16])
OP3	PIN_Y23 (SW[17])

指定腳位 – 七段顯示器

Signal Name	FPGA Pin No.	Description	I/O Standard
HEX0[0]	PIN_G18	Seven Segment Digit 0[0]	2.5V
HEX0[1]	PIN_F22	Seven Segment Digit 0[1]	2.5V
HEX0[2]	PIN_E17	Seven Segment Digit 0[2]	2.5V
HEX0[3]	PIN_L26	Seven Segment Digit 0[3]	Depending on JP7
HEX0[4]	PIN_L25	Seven Segment Digit 0[4]	Depending on JP7
HEX0[5]	PIN_J22	Seven Segment Digit 0[5]	Depending on JP7
HEX0[6]	PIN_H22	Seven Segment Digit 0[6]	Depending on JP7
HEX1[0]	PIN_M24	Seven Segment Digit 1[0]	Depending on JP7
HEX1[1]	PIN_Y22	Seven Segment Digit 1[1]	Depending on JP7
HEX1[2]	PIN_W21	Seven Segment Digit 1[2]	Depending on JP7
HEX1[3]	PIN_W22	Seven Segment Digit 1[3]	Depending on JP7
HEX1[4]	PIN_W25	Seven Segment Digit 1[4]	Depending on JP7
HEX1[5]	PIN_U23	Seven Segment Digit 1[5]	Depending on JP7
HEX1[6]	PIN_U24	Seven Segment Digit 1[6]	Depending on JP7