

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

實驗二：

多位元加法器 & BCD加法器電路設計

組合電路設計流程 & 階層化設計

繳交規定

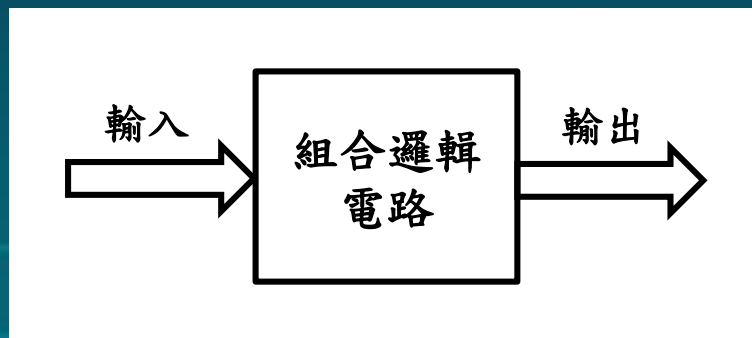
- 檢查期限：10/7 (三) 中午12:30截止
- 報告繳交期限：10/12 (一)上課前上傳至北科i學園PLUS->作業
- 繳交格式:北科i學園PLUS->文件(Document)->微算機系統_報告格式
- 詳細繳交規定請參照2020 Fall 微算機系統社團發文

配分方式

- 實驗目標一 40%
- 實驗目標二 30%
- 實驗報告 30%
- 實驗與報告遲交一週內打8折，第二週6折
- 之後不接受補交實驗與報告

基本概念

- 組合邏輯電路 (Combination Logic Circuit)：主要在於它的輸出只受輸入函數的控制,而不受先前輸入及狀態記憶影響。
- 下圖為組合邏輯電路之概念圖

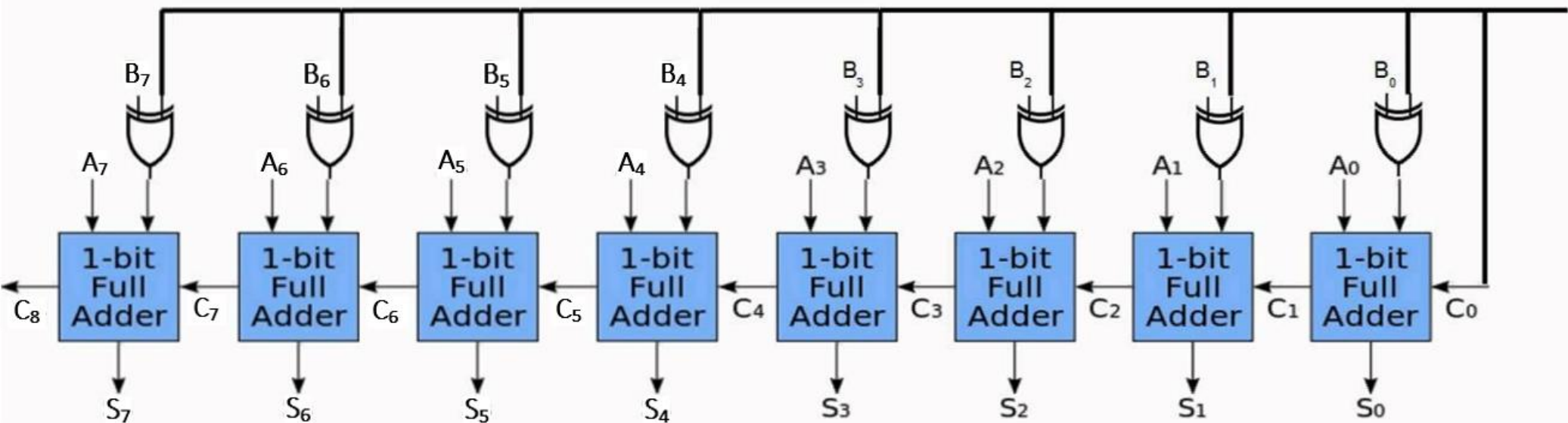


組合邏輯電路的設計流程

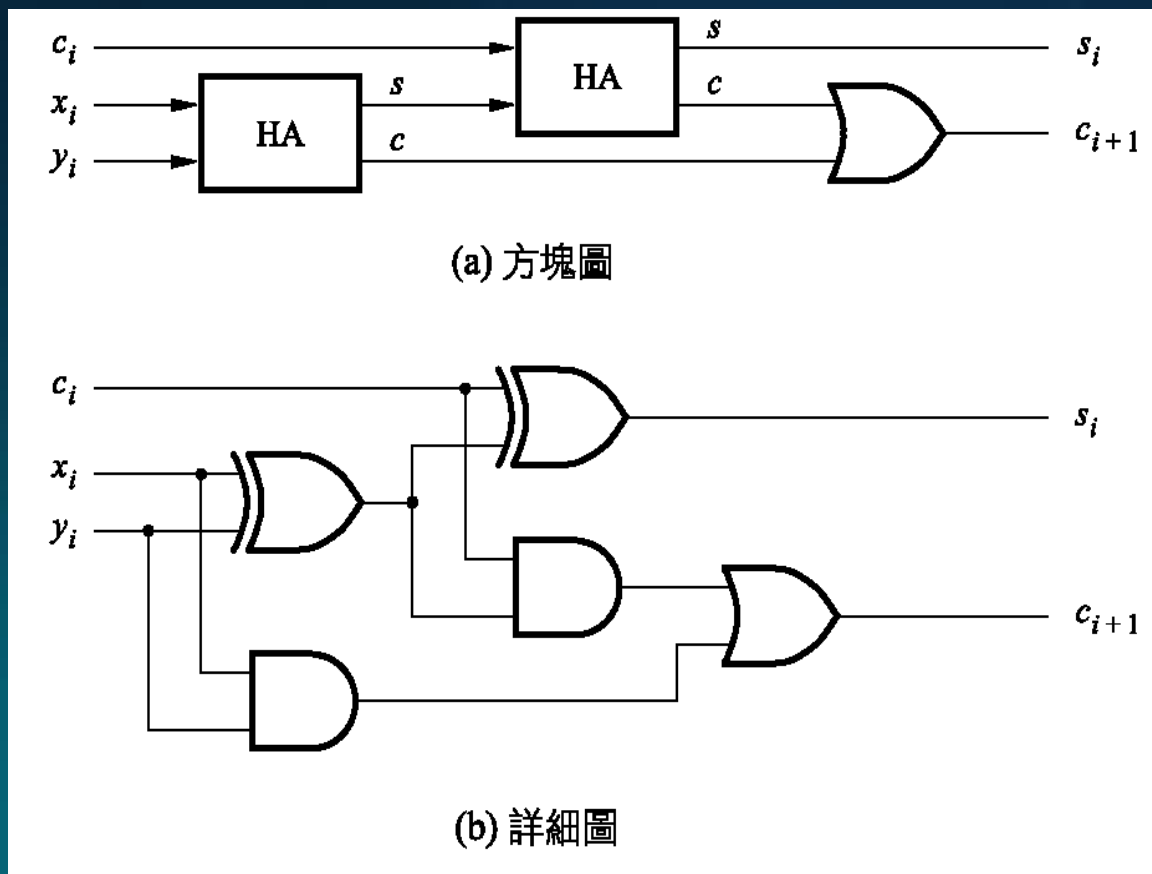
1. 需求敘述:將所需問題加以闡述說明。
2. 分析:將需求加以分析並以真值表描述輸入與輸出的關係,然後再由真值表導出布林函數。
3. 布林函數化簡:將找出來的布林函數加以化簡成最簡式,藉以縮減邏輯閘的使用。
4. 撰寫VHDL程式碼。
5. 實驗。

多位元加法器電路之設計

- 設計一個8 bit（1bit符號 7bit有效位元）多位元加法器，
輸入腳位為A7、A6、A5、A4、A3、A2、A1、A0、B7、



全加器電路圖



真值表

輸入			輸出	
C_i	x_i	y_i	C_{i+1}	S_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

布林等式

- 利用真值表寫出對等的布林等式並化簡。

$$s_i = x_i \oplus y_i \oplus c_i$$

$$c_{i+1} = x_i y_i + x_i c_i + y_i c_i$$

VHDL參考程式架構

```
LIBRARY IEEE;

USE IEEE.STD_LOGIC_1164.all; ← Use宣告區&標準定義宣告庫

ENTITY example IS

PORT(
    A : in std_logic_vector(7 downto 0);
    B : in std_logic_vector(7 downto 0);
    S : out std_logic_vector(7 downto 0));

END example;

ARCHITECTURE a OF example IS

    SIGNAL C : std_logic_vector(7 downto 0);

BEGIN

    (output pins)<=(input pins)

    (signals)<=(input pins)

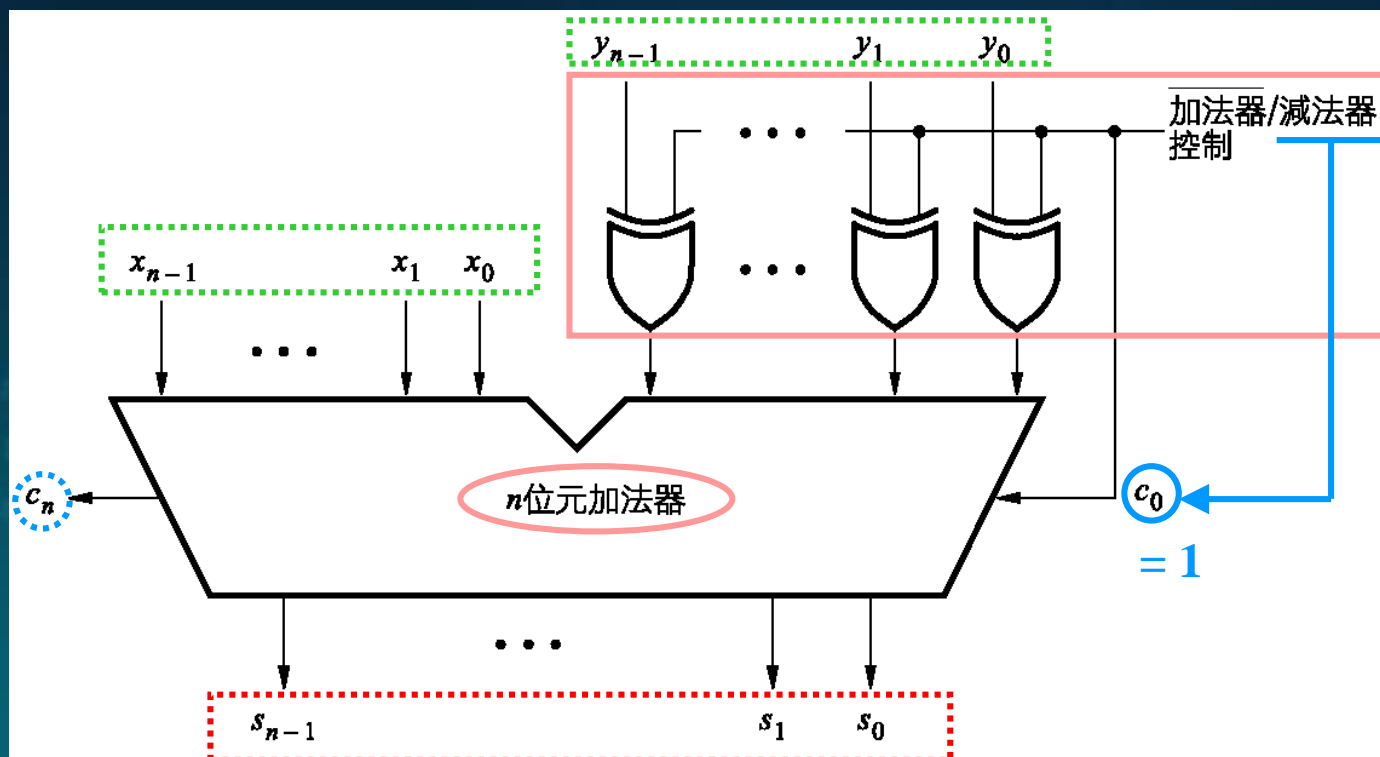
    (output pins)<=(signals)

END a;
```

STD_LOGIC表現的型態是實際數位電路在輸入或輸出位準所呈現的邏輯特性，它比BIT訊號所能描述的邏輯特性更為真實且完整。**STD_LOGIC為BIT的強化訊號型別，可以與BIT互換。**

Signal指令式宣告電路內部自行使用的訊號，因為這類訊號沒有傳送到電路的外部界面，所以通常我們都在架構(architecture)中宣告它。

電路設計



合併加法器／減法器單元。

實驗目標一 多位元加法器

1. 以1-bit之全加器為基礎，將上述之8-bit多位元加法器邏輯函數以 Package 與 component 語法包裝，並以實驗板上兩顆七段顯示器來呈現結果。
2. 測試時，將以 Switch 開關輸入，而兩顆七段顯示器上必須可正確顯示出運算的結果(十六進位)，結果皆為正整數，範圍為0~FF(需偵測Overflow，Overflow時亮LED燈)。
3. 禁用IF和SWITCH，請以加法器(邏輯閘)方式實現，使用其他方式則酌以扣分。

指定腳位

- 右表僅為範例
- 可自行選擇腳位

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])
A7	PIN_AB26 (SW[7])
B0	PIN_AC25 (SW[8])
B1	PIN_AB25 (SW[9])
B2	PIN_AC24 (SW[10])
B3	PIN_AB24 (SW[11])
B4	PIN_AB23 (SW[12])
B5	PIN_AA24 (SW[13])
B6	PIN_AA23 (SW[14])
B7	PIN_AA22 (SW[15])

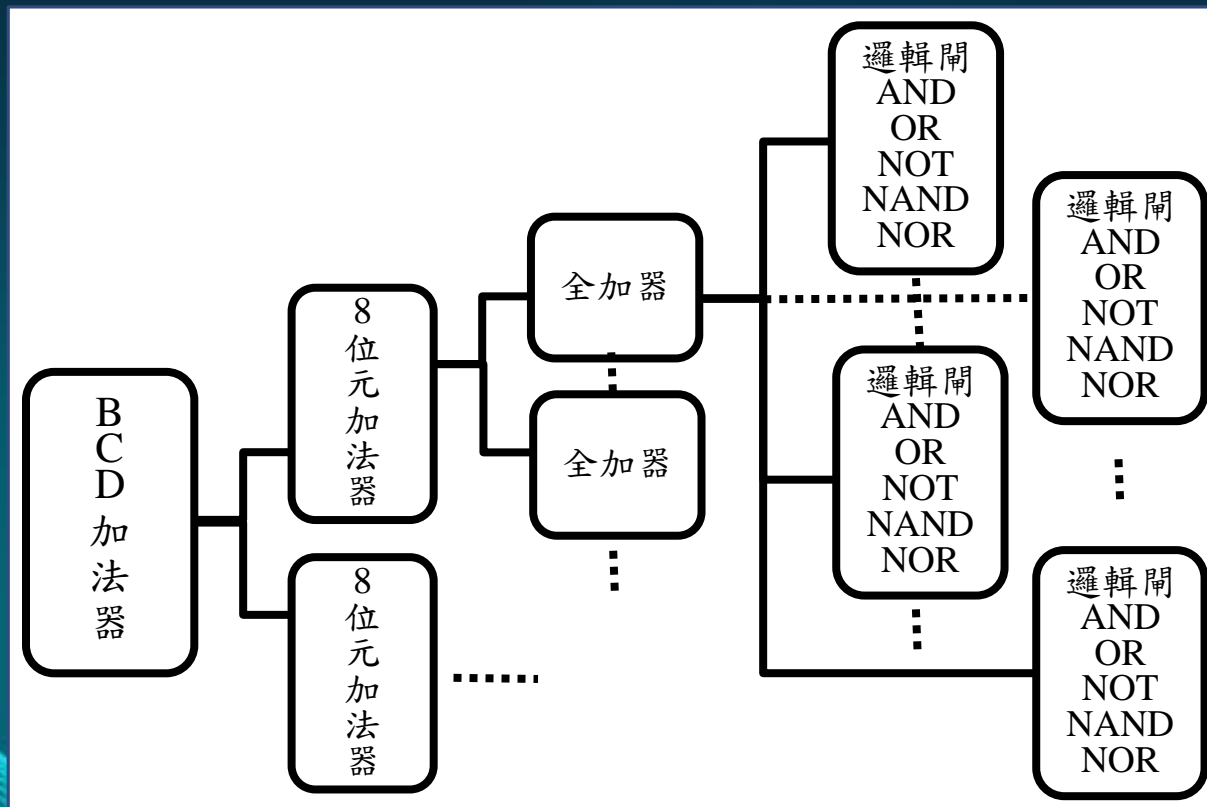
Name	Pin Location
a0 (HEX0[0])	PIN_G18
b0 (HEX0[1])	PIN_F22
c0 (HEX0[2])	PIN_E17
d0 (HEX0[3])	PIN_L26
e0 (HEX0[4])	PIN_L25
f0 (HEX0[5])	PIN_J22
g0 (HEX0[6])	PIN_H22
a1 (HEX1[0])	PIN_M24
b1 (HEX1[1])	PIN_Y22
c1 (HEX1[2])	PIN_W21
d1 (HEX1[3])	PIN_W22
e1 (HEX1[4])	PIN_W25
f1 (HEX1[5])	PIN_U23
g1 (HEX1[6])	PIN_U24

Name	Pin Location
led (LEDR[0])	PIN_G19

目標二 – BCD加法器

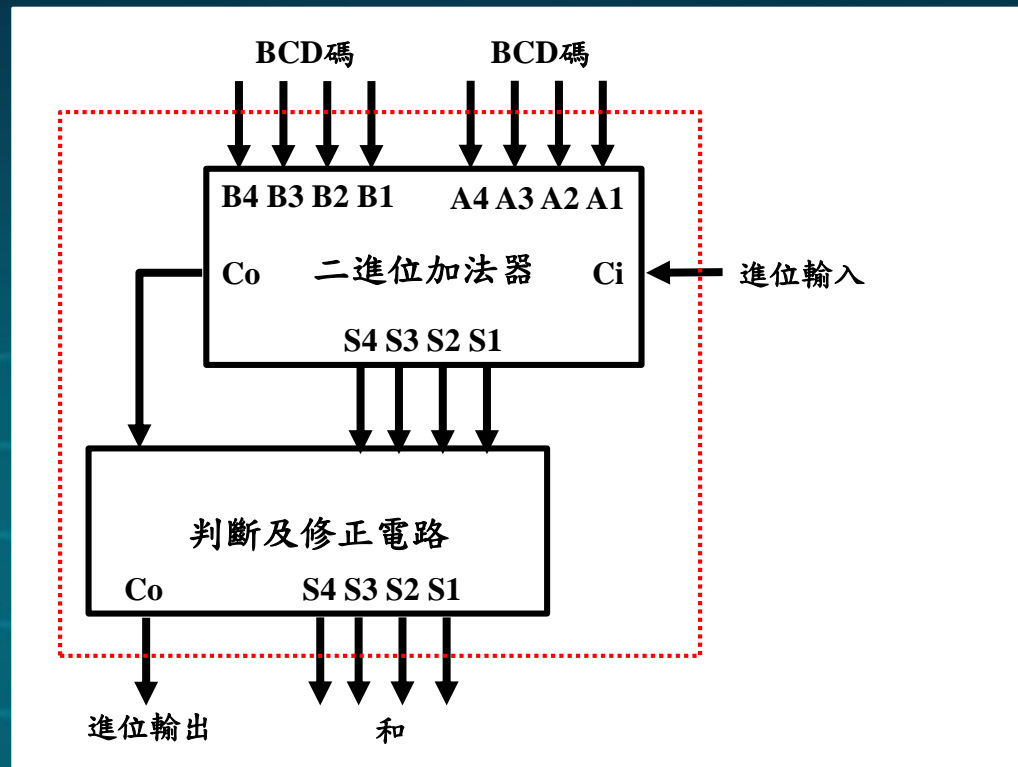
基本概念：階層化的設計概念

- 上層電路是由下層電路所合成，而本實習以實現一個 BCD 加法器，呈現電路階層化設計的概念，並以兩顆 7-Segment 七段顯示器顯示 0~99 相加之結果。
- 設計由上而下，實現電路由下而上。



BCD加法器之系統架構

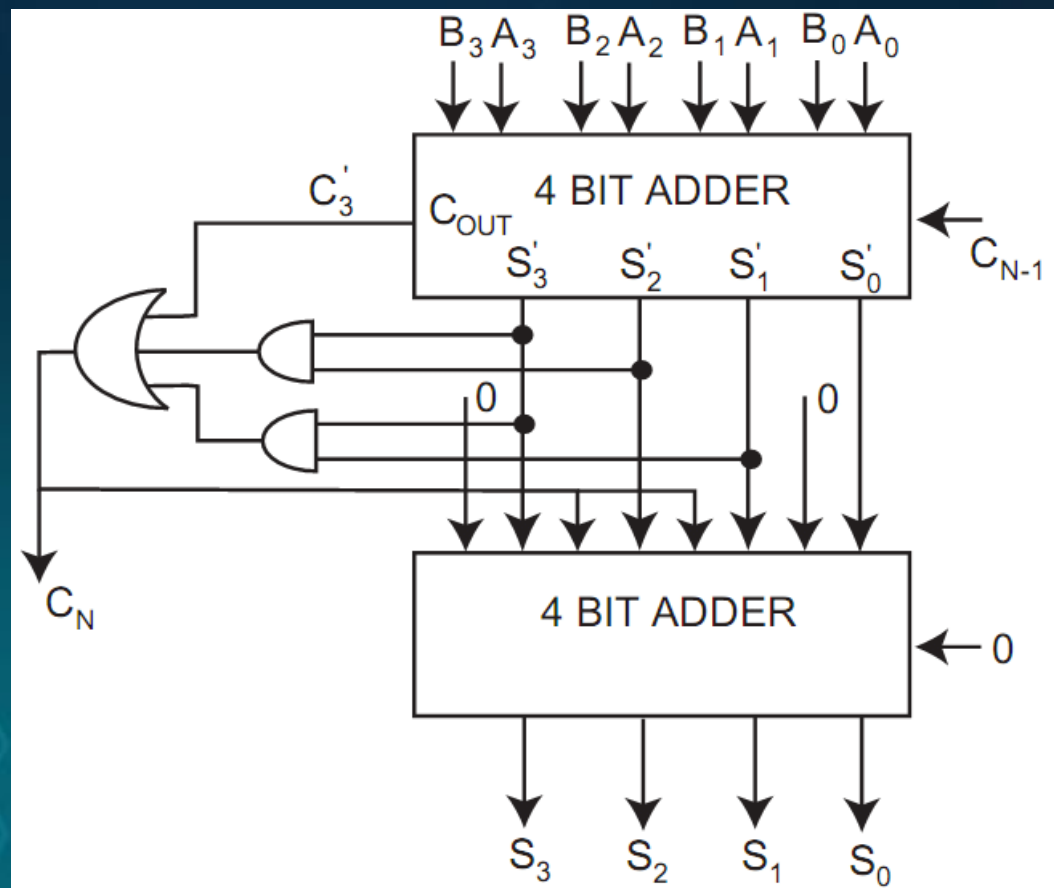
- A1-A4 與 B1-B4 相加若 S1-S4 是否超過 1001 (9) 時則必須由『判斷電路』判斷。
- 而超過 9 再利用『修正電路』作修正其『和』及產生『進位輸出』



BCD加法器之電路

$$U_n = (S_4 \cdot S_1) + (S_3 \cdot S_2) + C_3$$

- 加法進位的條件：
 - $8 + 4 > 9 \Rightarrow$ 進位。
 - $8 + 2 > 9 \Rightarrow$ 進位。
- BCD加法器進位條件：
 - $S_3' \text{ AND } S_2'$ 是 1 就進位。
 - $S_3' \text{ AND } S_1'$ 是 1 就進位。
- C_N 代表是否進位 (Carry)。
 - 若有進位 ($C_N=1$)，要在最終結果再加一個 0110 的值轉為 Decimal。



VHDL code 撰寫與實現

- 在完成 BCD 加法器電路之後，以階層化設計整合上次實驗一所實作之 7-Segment 七段顯示器電路之 VHDL 實作檔，再以 **Package 與 component 語法包裝** 並整合入 BCD 加法器電路中，而且在實驗板上以點亮兩顆七段顯示器驗證 BCD 加法電路是否正確。

VHDL code

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

ENTITY BCDAdder IS
    PORT(
        A, B          : IN STD_LOGIC_VECTOR(3 downto 0);
        S              : OUT      STD_LOGIC_VECTOR(3 downto 0);
        Co             : OUT      STD_LOGIC);
END BCDAdder;
ARCHITECTURE MAIN OF BCDAdder IS
    SIGNAL temp : STD_LOGIC_VECTOR(4 downto 0);
BEGIN

END MAIN;
```

包裝 7-Segment Component 的範例

```
PACKAGE SevenSegment_package IS
    COMPONENT SevenSegment
        PORT ( D0 , D1 , D2 ,D3 : IN STD_LOGIC;

              a, b, c, d, e, f, g, h: OUT STD_LOGIC);
    END COMPONENT ;
END SevenSegment_package;
```

實驗目標二—BCD加法器

1. 將上述之BCD加法器邏輯函數以Package與component語法包裝，並以實驗板上面兩顆七段顯示器驗證BCD加法電路是否正確。
2. 測試時，將以Switch開關輸入要相加的數字，而兩顆七段顯示器上必須可正確顯示出BCD十進位加法運算之結果。
3. 在實驗二中，只須使用A0~A3與B0~B3，完成BCD十進位之正整數相加，亦結果皆為正整數，範圍為0~18。
4. 禁用IF和SWITCH，請以加法器(邏輯閘)方式實現，使用其他方式則酌以扣分。

指定腳位

Name	Pin Location
a0 (HEX0[0])	PIN_G18
b0 (HEX0[1])	PIN_F22
c0 (HEX0[2])	PIN_E17
d0 (HEX0[3])	PIN_L26
e0 (HEX0[4])	PIN_L25
f0 (HEX0[5])	PIN_J22
g0 (HEX0[6])	PIN_H22
a1 (HEX1[0])	PIN_M24
b1 (HEX1[1])	PIN_Y22
c1 (HEX1[2])	PIN_W21
d1 (HEX1[3])	PIN_W22
e1 (HEX1[4])	PIN_W25
f1 (HEX1[5])	PIN_U23
g1 (HEX1[6])	PIN_U24

指定腳位

Name	Pin Location
A0	PIN_AB28 (SW[0])
A1	PIN_AC28 (SW[1])
A2	PIN_AC27 (SW[2])
A3	PIN_AD27 (SW[3])
B0	PIN_AB27 (SW[4])
B1	PIN_AC26 (SW[5])
B2	PIN_AD26 (SW[6])
B3	PIN_AB26 (SW[7])

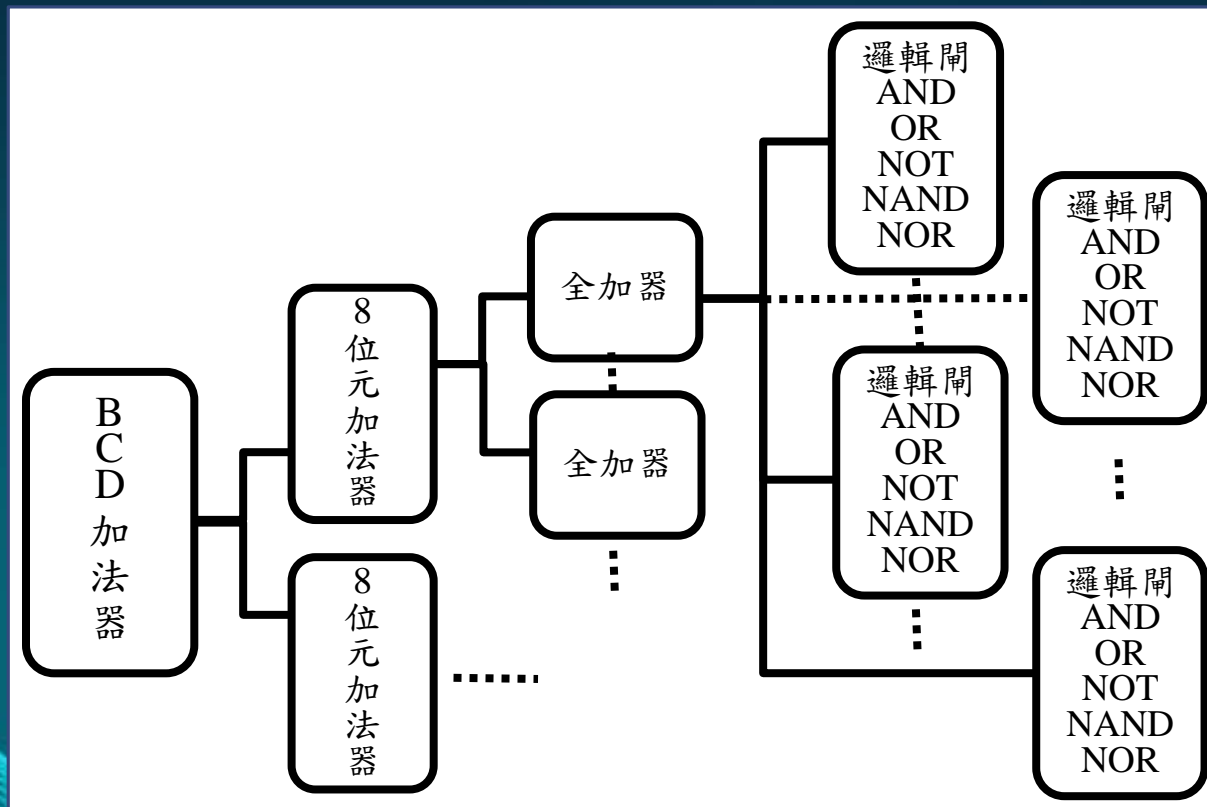
加分—8位元 BCD加法器

實驗目標加分—8位元 BCD加法器

1. 將上述之 BCD 加法器邏輯函數以 Package 與 component 語法包裝，並以實驗板上面兩顆七段顯示器驗證 BCD 加法電路是否正確。
2. 測試時，將以 Switch 開關輸入，而兩顆七段顯示器上必須可正確顯示出運算的結果(十進位)，結果皆為正整數，範圍為 00~99(需偵測 Overflow，Overflow 時亮 LED 燈)。
3. 禁用 IF 和 SWITCH，請以加法器(邏輯閘)方式實現，使用其他方式則酌以扣分。

基本概念：階層化的設計概念

- 上層電路是由下層電路所合成，而本實習以實現一個 BCD 加法器，呈現電路階層化設計的概念，並以兩顆 7-Segment 七段顯示器顯示 0~99 相加之結果。
- 設計由上而下，實現電路由下而上。



指定腳位

- 右表僅為範例
- 可自行選擇腳位

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])
A7	PIN_AB26 (SW[7])
B0	PIN_AC25 (SW[8])
B1	PIN_AB25 (SW[9])
B2	PIN_AC24 (SW[10])
B3	PIN_AB24 (SW[11])
B4	PIN_AB23 (SW[12])
B5	PIN_AA24 (SW[13])
B6	PIN_AA23 (SW[14])
B7	PIN_AA22 (SW[15])

Name	Pin Location
a0 (HEX0[0])	PIN_G18
b0 (HEX0[1])	PIN_F22
c0 (HEX0[2])	PIN_E17
d0 (HEX0[3])	PIN_L26
e0 (HEX0[4])	PIN_L25
f0 (HEX0[5])	PIN_J22
g0 (HEX0[6])	PIN_H22
a1 (HEX1[0])	PIN_M24
b1 (HEX1[1])	PIN_Y22
c1 (HEX1[2])	PIN_W21
d1 (HEX1[3])	PIN_W22
e1 (HEX1[4])	PIN_W25
f1 (HEX1[5])	PIN_U23
g1 (HEX1[6])	PIN_U24

Name	Pin Location
led (LEDR[0])	PIN_G19