電機一乙 數位邏輯實習

第五週實習作業報告

I.實驗目的與原理

*A.作業1*

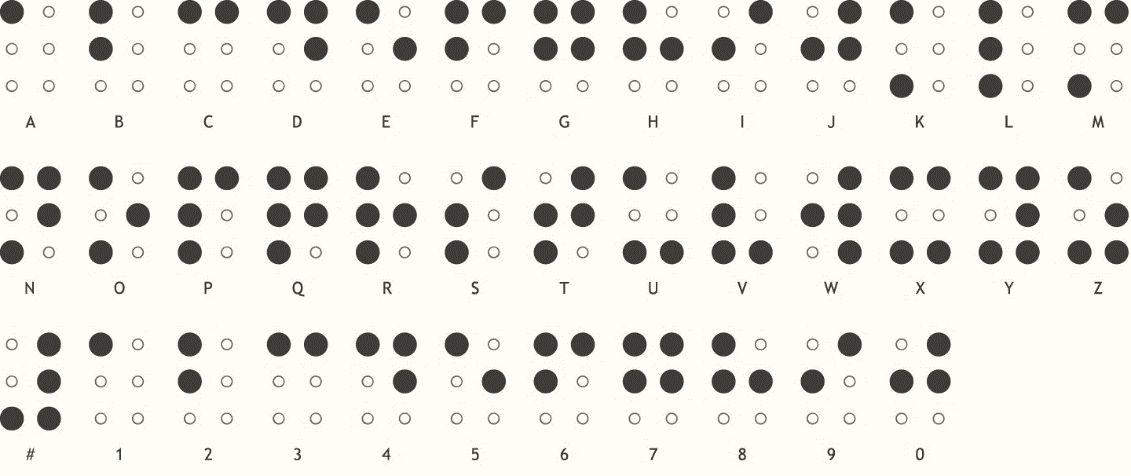
利用VHDL實作1位元的加減法器，將1位元加減法器串接為4位元加減法器。完成下列真值表，並推導Cout與Sum的布林代數式，Sub為0時進行加法運算，為1時進行減法運算。

表ㄧ：加減法器真值表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | Cin | Sub | Cout | Sum |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 |

*B.作業2*

利用VHDL實作Braille點字系統(圖一)。以A、B、C、D作為輸入，輸出參考表二，導出最簡化的布林代數式後利用VHDL實作電路。



圖ㄧ：Braille點字系統英文字母對照表

表二：點字系統輸入輸出對照表

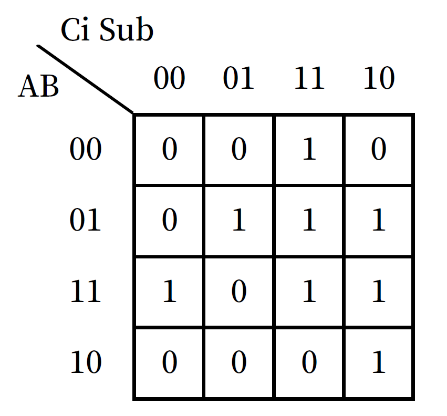
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | 輸出 |
| 0 | 0 | 0 | 0 | A |
| 0 | 0 | 0 | 1 | B |
| 0 | 0 | 1 | 0 | C |
| 0 | 0 | 1 | 1 | D |
| 0 | 1 | 0 | 0 | E |
| 0 | 1 | 0 | 1 | F |
| 0 | 1 | 1 | 0 | G |
| 0 | 1 | 1 | 1 | H |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | 輸出 |
| 1 | 0 | 0 | 0 | I |
| 1 | 0 | 0 | 1 | J |
| 1 | 0 | 1 | 0 | K |
| 1 | 0 | 1 | 1 | L |
| 1 | 1 | 0 | 0 | M |
| 1 | 1 | 0 | 1 | N |
| 1 | 1 | 1 | 0 | O |
| 1 | 1 | 1 | 1 | P |

II.實驗過程

1. *作業1*

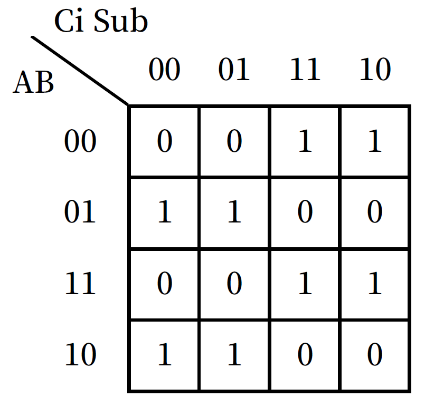
完成真值表(表三)後利用卡諾圖(圖二、三)推導出輸出Cout與Sum之布林代數式-(1)及式-(2)。



表三：加減法器真值表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | Ci | Sub | Co | Sum |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

圖二：Cout卡諾圖



圖三：Sum卡諾圖

---- (1)

---- (2)

其VHDL如下，輸入進external後，得到1位元加減法器(圖四)。

輸入輸出

布林代數式

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

USE ieee.numeric\_std.all;

entity adder\_and\_subtractor is

port (

X, Y, Cin, sub: in std\_logic;

Sum, Cout: out std\_logic);

end adder\_and\_subtractor;

architecture Equations of adder\_and\_subtractor is

begin

-- concurrent assignment statements

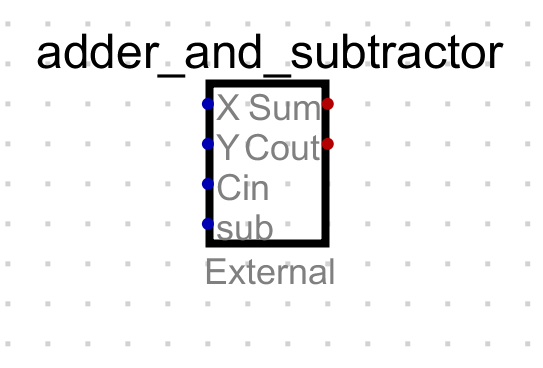
cout <= (y and cin) or (x and y and not sub) or(x and cin and not sub) or

(not x and y and sub) or (not x and cin and sub);

sum <= (not x and not y and cin) or (not x and y and not cin) or (x and y and cin) or

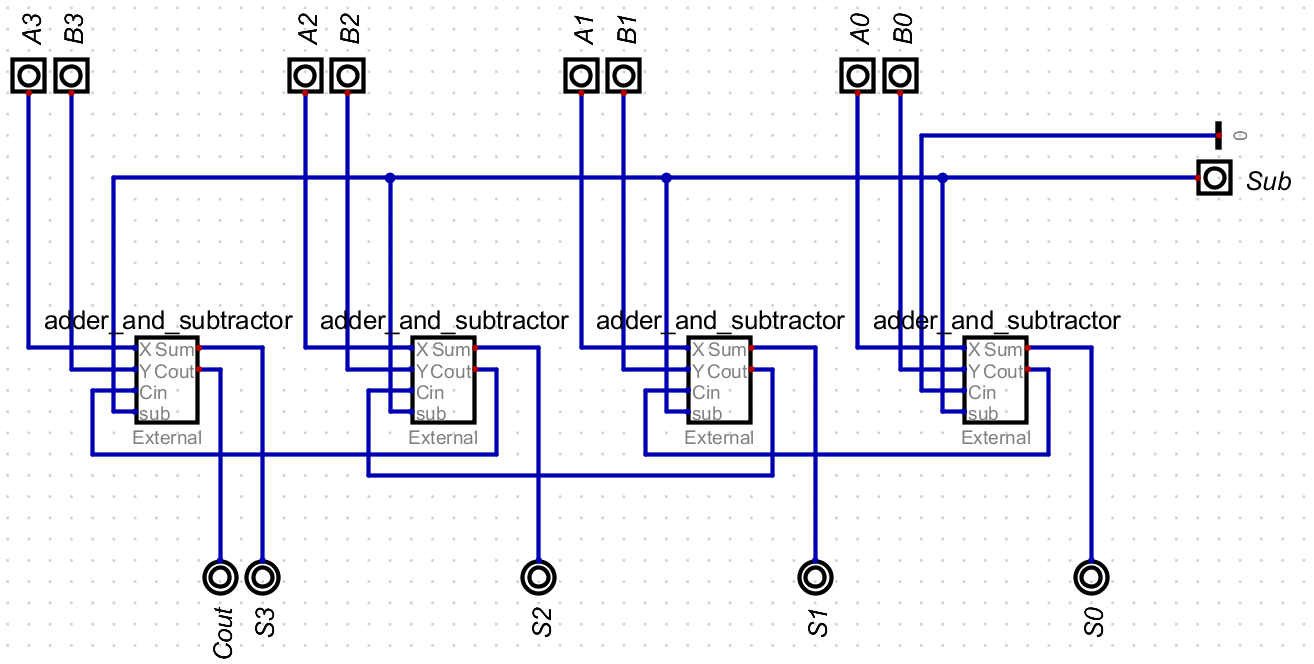
(x and not y and not cin);

end Equations;



圖四：1位元加減法器

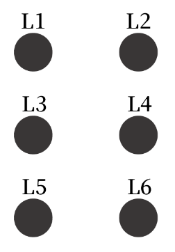
將四個1位元加減法器的Ci、Co互相連接、Sub接出，得到4位元加減法器(圖五)。



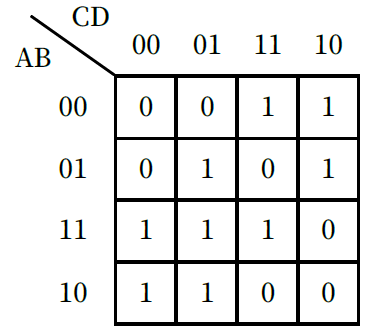
圖五：4位元加減法器

1. *作業2*

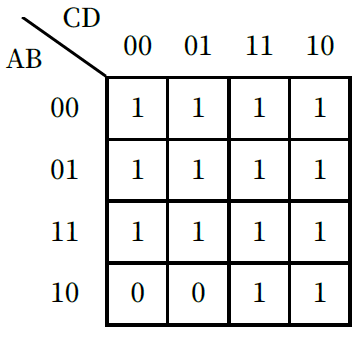
在這裡我們使用LED來代替點字卡來表示轉換後的輸出，為此我們需要求出每顆LED

(L1~L6)的布林代數式，L1~L6與原點字系統對照如下圖六，L1~L6經卡諾圖化簡(如下圖七~十二)後得布林代數式。

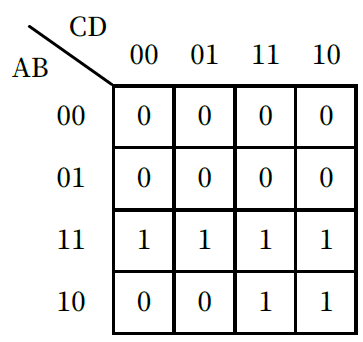
圖六：LED腳位對照



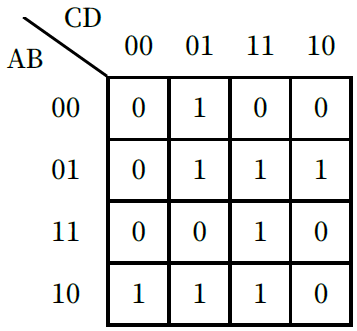
圖八：L2卡諾圖



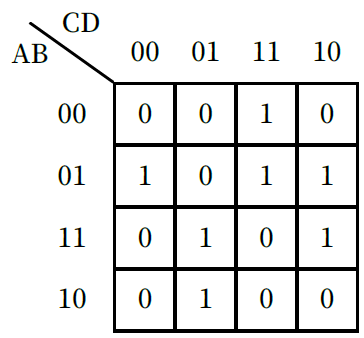
圖六：L1卡諾圖



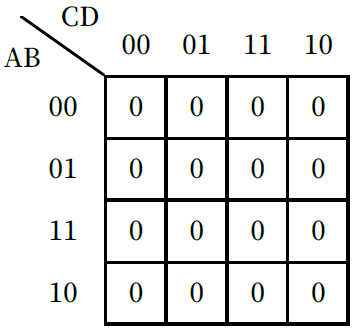
圖九：L5卡諾圖



圖十：L3卡諾圖



圖七：L4卡諾圖



圖十一：L6卡諾圖

其VHDL如下，輸入進external後，得Braille點字系統轉換器(圖十二)。

輸入輸出

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

USE ieee.numeric\_std.all;

entity hw2 is

port (

A, B, C, D: in std\_logic;

L1, L2, L3, L4, L5, L6: out std\_logic);

end hw2;

architecture Equations of hw2 is

begin

-- concurrent assignment statements

布林代數式

L1 <= not a or b or c;

L2 <= (a and not c) or (b and not c and d) or (a and b and d) or

(not a and not b and c) or (not a and c and not d);

L3 <= (not a and not c and d) or (not a and b and c) or

(a and not b and not c) or (a and c and d);

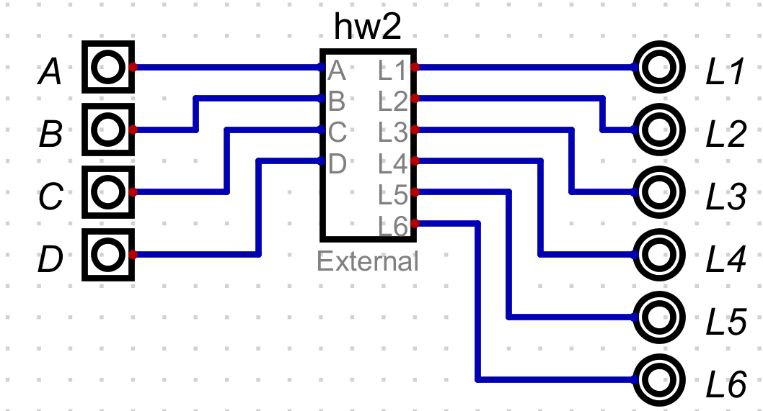
L4 <= (not a and b and not d) or (not a and c and c) or

(b and c and not d) or (a and not c and d);

L5 <= (a and b) or (a and c);

L6 <= '0';

end Equations;



圖十一：Braille點字系統轉換器

III.模擬驗證

1. *作業1*
2. *作業2*

IV.實驗結果與成果討論

1. *作業1*
2. *作業2*