

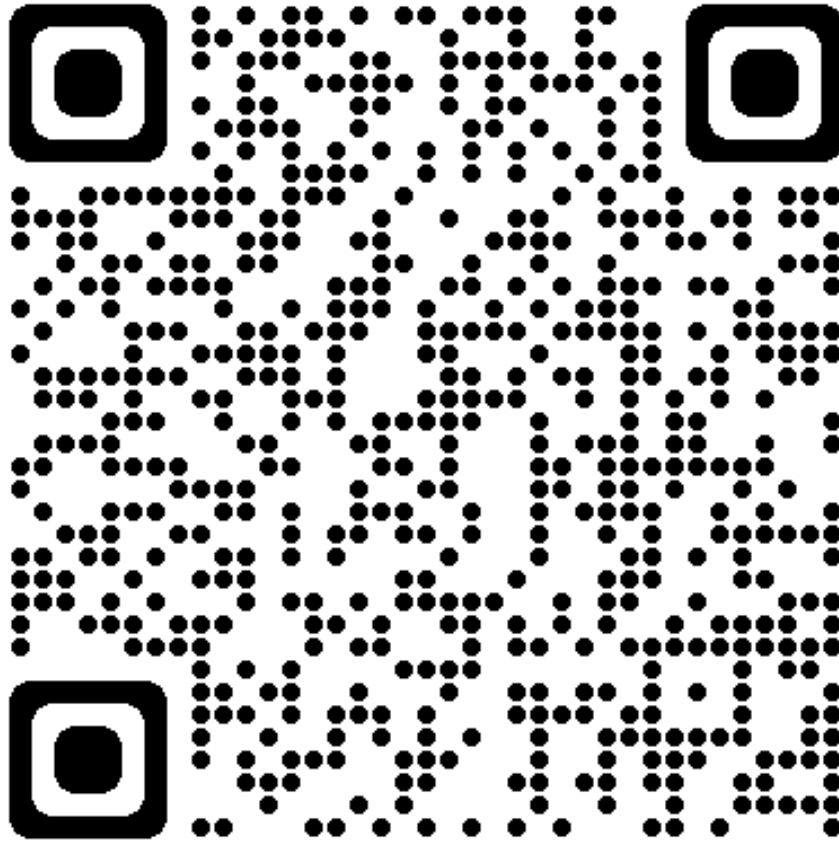
# 成大電機營

## 邏輯系統實驗 - 加法器實作



Department of Electrical Engineering  
National Cheng Kung University

## 下載投影片



[https://caslab.ee.ncku.edu.tw/dokuwiki/\\_media/edu\\_project:eecamp\\_logicsystem.pdf](https://caslab.ee.ncku.edu.tw/dokuwiki/_media/edu_project:eecamp_logicsystem.pdf)

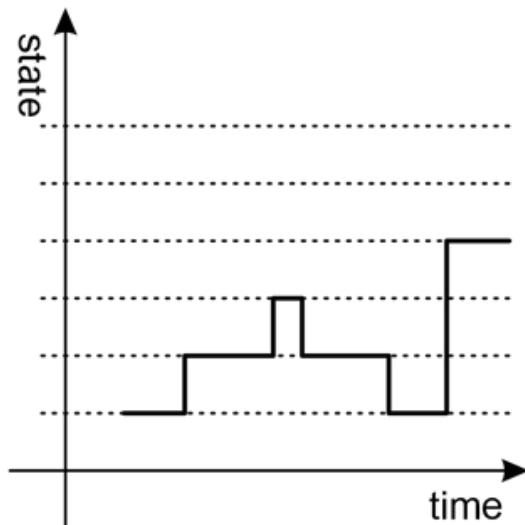
# Outline

- 何謂數位邏輯
- 二進位數值系統
- 邏輯運算
- TTL數位邏輯IC
- 七段顯示器
- 單位元全加器實作
- 多位元全加器實作
- 附錄

# 何謂數位邏輯

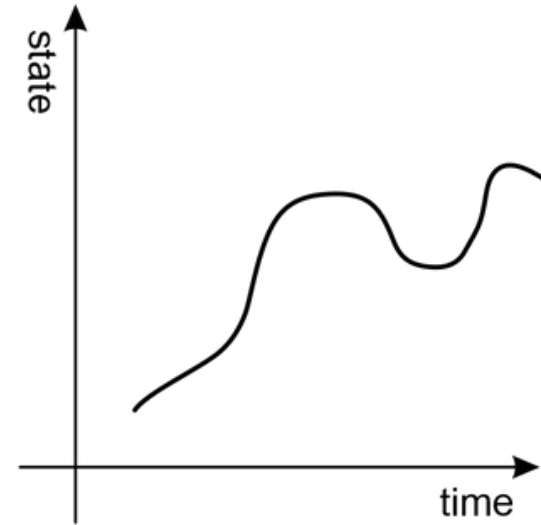
- 數位訊號

- 離散的訊號值
- 有限多的訊號值 (0與1的排列組合)
- 抵抗雜訊能力強
- 電路設計容易、便宜



- 類比訊號

- 連續的訊號值
- 無限多的訊號值 (實數稠密性)
- 抵抗雜訊能力較差
- 電路複雜、成本較高



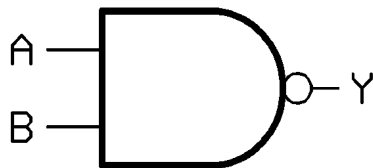
# 何謂數位邏輯

- 電腦科學的層狀結構
- 抽象 (Abstraction)
  - 去除下層的具體細節
- 舉例：排序1000個數字
  - 插入排序法、氣泡排序法、快速排序法
    - C/C++、Python、Java
      - Windows、macOS、Linux
        - Intel i9 13900K、Apple M2 Ultra

Problem
Algorithm
Program/Language
System Software
SW/HW Interface
Micro-architecture
Digital Logic
Transistors
Metal/Silicon

# 何謂數位邏輯

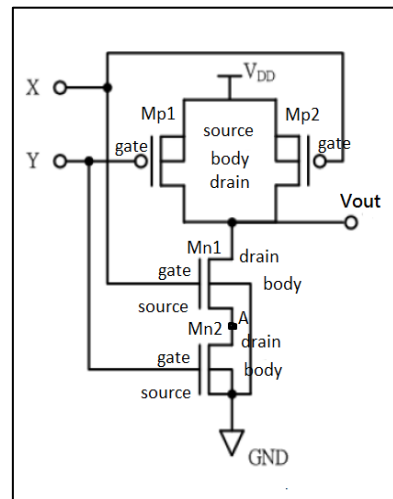
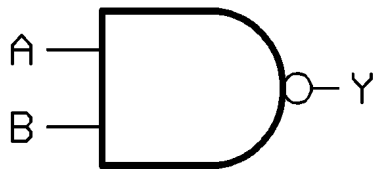
- 電腦科學的層狀結構
- 抽象 (Abstraction)
  - 去除下層的具體細節
- 舉例：排序1000個數字
  - 插入排序法、氣泡排序法、快速排序法
    - C/C++、Python、Java
      - Windows、macOS、Linux
        - Intel i9 13900K、Apple M2 Ultra



Problem
Algorithm
Program/Language
System Software
SW/HW Interface
Micro-architecture
Digital Logic
Transistors
Metal/Silicon

# 何謂數位邏輯

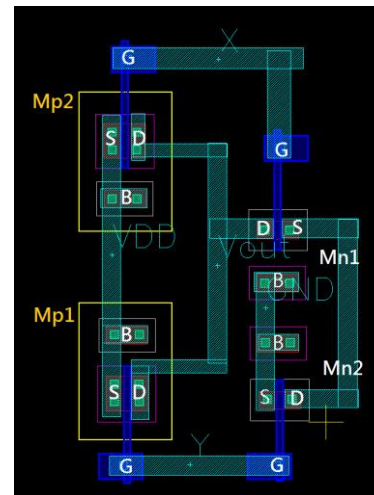
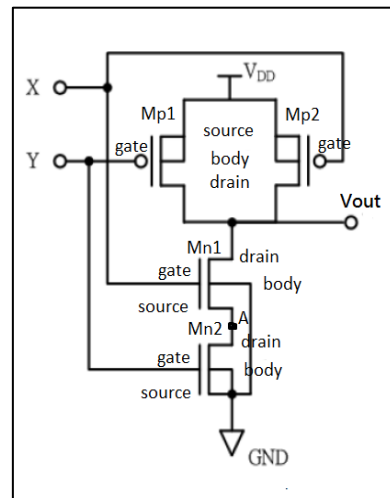
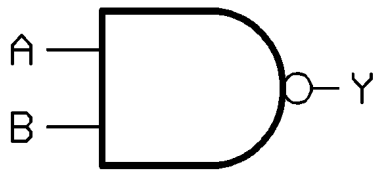
- 電腦科學的層狀結構
- 抽象 (Abstraction)
  - 去除下層的具體細節
- 舉例：排序1000個數字
  - 插入排序法、氣泡排序法、快速排序法
    - C/C++、Python、Java
      - Windows、macOS、Linux
        - Intel i9 13900K、Apple M2 Ultra



Problem
Algorithm
Program/Language
System Software
SW/HW Interface
Micro-architecture
Digital Logic
Transistors
Metal/Silicon

# 何謂數位邏輯

- 電腦科學的層狀結構
- 抽象 (Abstraction)
  - 去除下層的具體細節
- 舉例：排序1000個數字
  - 插入排序法、氣泡排序法、快速排序法
    - C/C++、Python、Java
      - Windows、macOS、Linux
        - Intel i9 13900K、Apple M2 Ultra



Problem
Algorithm
Program/Language
System Software
SW/HW Interface
Micro-architecture
Digital Logic
Transistors
Metal/Silicon

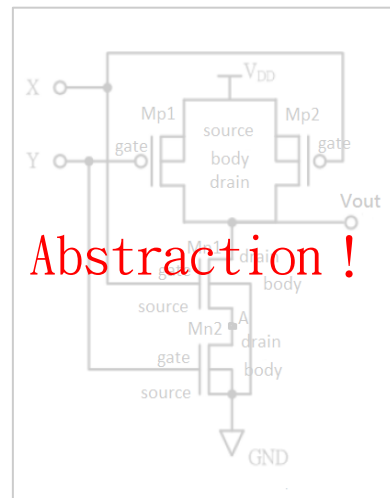
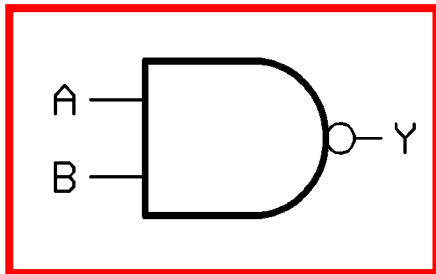


# 何謂數位邏輯

- 電腦科學的層狀結構
- 抽象 (Abstraction)
  - 去除下層的具體細節
- 舉例：排序1000個數字
  - 插入排序法、氣泡排序法、快速排序法
    - C/C++、Python、Java
      - Windows、macOS、Linux
        - Intel i9 13900K、Apple M2 Ultra

Problem
Algorithm
Program/Language
System Software
SW/HW Interface
Micro-architecture
Digital Logic
Transistors
Metal/Silicon

本日重點



# 二進位數值系統

- 我們日常生活中大部分都是使用十進制，也就是每個位元代表了10的次方
  - $1234 = 1 * 10^3 + 2 * 10^2 + 3 * 10^1 + 4 * 10^0$
- 在數位電路中，電路只看得懂「高電壓」和「低電壓」，所以我們需要用二進制表示我們所要代表的數值
  - 高電壓：邏輯1
  - 低電壓：邏輯0
- 每個位元代表2的次方
  - 1、2、4、8、16、32...
- 二進制跟十進制可以互相轉換！
  - $12_{(10)} = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 0 * 2^0 = 1100_{(2)}$
  - $1010_{(2)} = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 = 10_{(10)}$

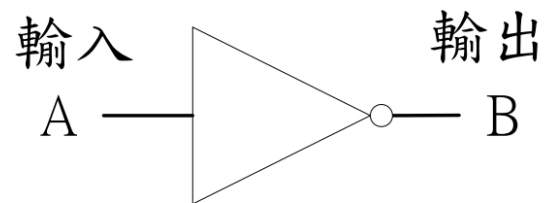
# 邏輯運算

- 我們除了可以對二進位數值進行加減等數值運算，還可以對二進位數值進行**邏輯運算**
- 邏輯運算包含
  - NOT:  $B = \sim A$
  - AND:  $C = A \& B$
  - OR:  $C = A | B$
  - XOR:  $C = A \wedge B$
  - NAND、NOR、XNOR
- 在計算機科學中，邏輯運算是所有運算的基礎
  - 這次實驗課中，我們將使用**邏輯運算**來完成一個加法器！

# 邏輯運算 - NOT

- 反向邏輯，輸出輸入的反向狀態
- NOT:  $B = \sim A$

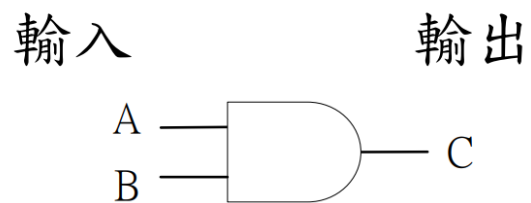
輸入 (A)	輸出 (B)
0	1
1	0



# 邏輯運算 - AND

- 與邏輯，僅在兩個輸入皆為1的情況下輸出1，其餘情況輸出0
- AND:  $C = A \& B$

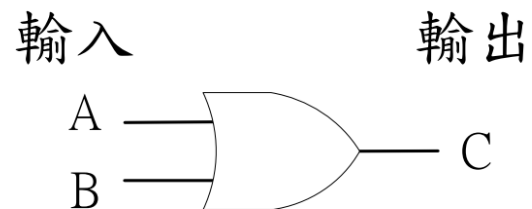
輸入 (A)	輸入 (B)	輸出 (C)
0	0	0
0	1	0
1	0	0
1	1	1



# 邏輯運算 - OR

- 或邏輯，兩個輸入其中任一為1的情況下輸出1，其餘情況輸出0
- OR:  $C = A \mid B$

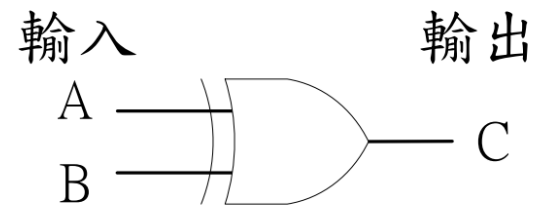
輸入 (A)	輸入 (B)	輸出 (C)
0	0	0
0	1	1
1	0	1
1	1	1



# 邏輯運算 - XOR

- 互斥或邏輯，兩個輸入不同的情況下輸出1，兩輸入相同則輸出0
- XOR:  $C = A \oplus B$

輸入 (A)	輸入 (B)	輸出 (C)
0	0	0
0	1	1
1	0	1
1	1	0



# 多個位元時？

- 每個位元單獨做邏輯運算！
- 舉例來說：
  - $A = 12_{(10)} = 1100_{(2)}$
  - $B = 7_{(10)} = 0111_{(2)}$
  - $\sim A = 0011_{(2)} = 3_{(10)}$
  - $A \& B = 0100_{(2)} = 4_{(10)}$
  - $A | B = 1111_{(2)} = 15_{(10)}$
  - $A \wedge B = 1011_{(2)} = 11_{(10)}$



# TTL 數位邏輯IC

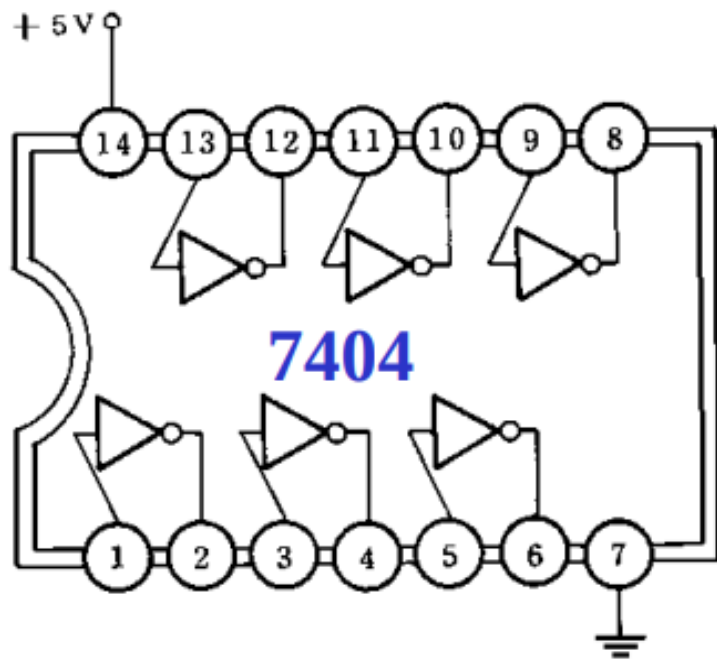
- TTL IC的特性
  - TTL系列使用的電源是直流 5V 。
  - 輸入、輸出狀態為” 0” 與” 1” 時的電壓如下表 所示。

邏輯狀態	輸入電壓	輸出電壓
0	0.8V 以下	0.4V 以下
1	2.0V 以上	2.4V 以上

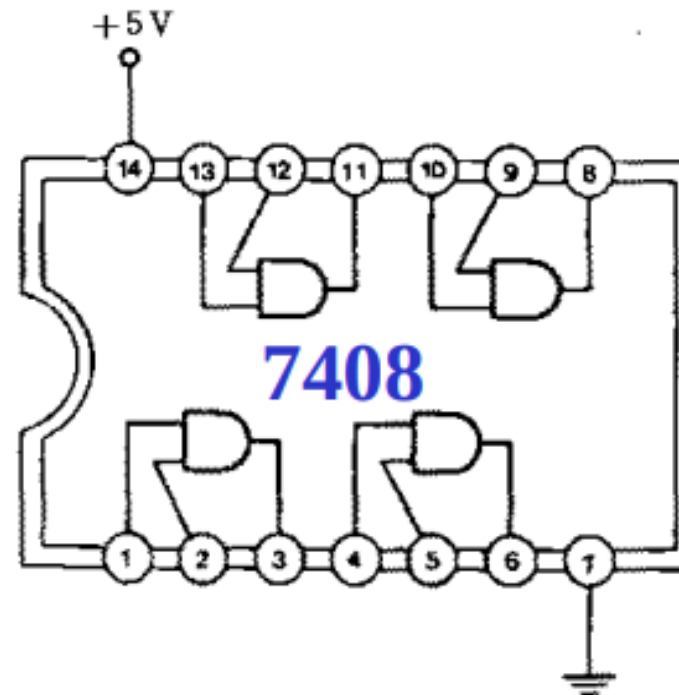
- 若要輸入邏輯1，將腳位接至5V
- 若要輸入邏輯0，將腳位接至0V (GND)

# TTL數位邏輯IC

- NOT 閘



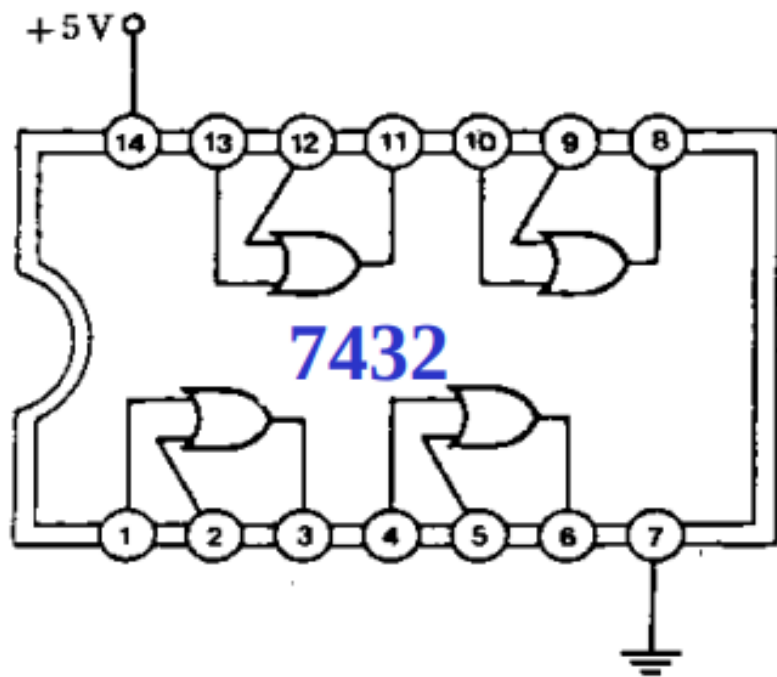
- AND閘



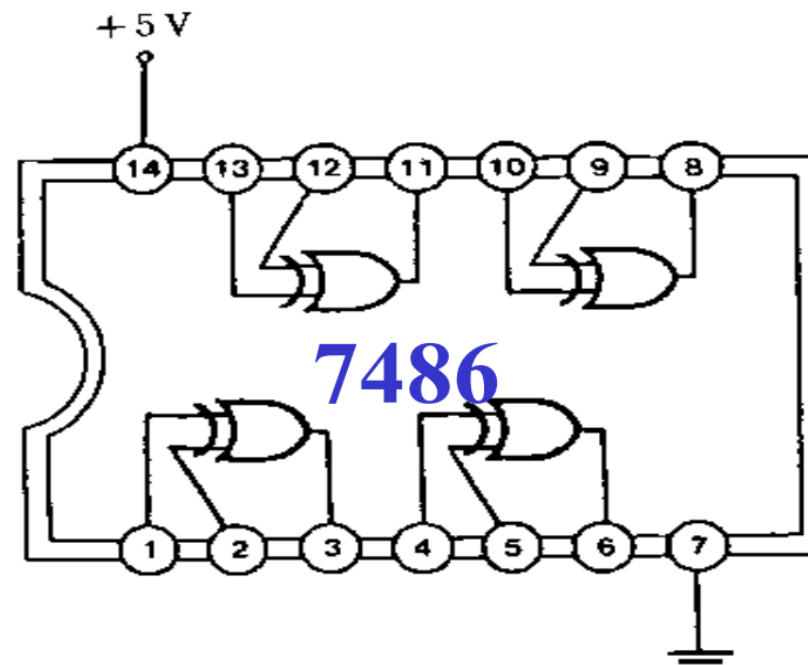
5V (14號腳位) 跟GND (7號腳位) 要記得接！！

# TTL數位邏輯IC

- OR 閘



- XOR閘



5V (14號腳位) 跟GND (7號腳位) 要記得接！！

# 單位元半加器實作

- 二進制的加法跟我們所熟知的十進制一樣，每個位數加完後，如果超過該位數的上限，就需要進位。

- 例如：

- 十進位

$$\begin{array}{r} 4 \\ + 2 \\ \hline 06 \end{array}$$

and

$$\begin{array}{r} 9 \\ + 8 \\ \hline 17 \end{array}$$

A red '1' with a curved arrow pointing to the tens place of the result (17) is shown above the second addition, indicating a carry from the units place.

- 而二進位

$$\begin{array}{r} 1 \\ + 0 \\ \hline 01 \end{array}$$

and

$$\begin{array}{r} 1 \\ + 1 \\ \hline 10 \end{array}$$

A red '1' with a curved arrow pointing to the tens place of the result (10) is shown above the second addition, indicating a carry from the units place.

# 單位元半加器實作

- 把二進位制單一個位數的加法全部列出來就會有4種組合。

$\begin{array}{r} 0 \\ + 0 \\ \hline 00 \end{array}$	$\begin{array}{r} 0 \\ + 1 \\ \hline 01 \end{array}$	$\begin{array}{r} 1 \\ + 0 \\ \hline 01 \end{array}$	$\begin{array}{r} 1 \\ + 1 \\ \hline 10 \end{array}$	<div style="color: red; font-weight: bold;">Carry</div> <div style="color: red; font-weight: bold;">1</div>
				<div style="color: red; font-weight: bold;">A</div> <div style="color: red; font-weight: bold;">B</div> <div style="color: blue; font-weight: bold;">Sum</div>

- 換個方式，我們可以把這些組合寫成一個表格，稱為真值表(truth table)。

A	B	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

# 單位元半加器實作

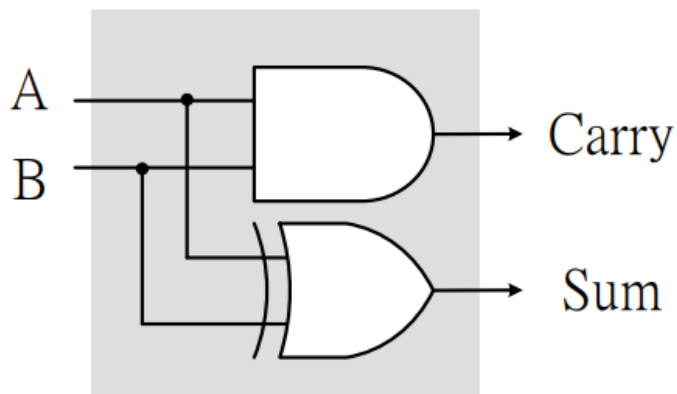
- 接著我們需要透過『邏輯』運算的方式，釐清Carry、Sum分別跟A、B的關係是什麼。

A	B	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$Carry = A \cdot B$$

$$Sum = \bar{A} \cdot B + A \cdot \bar{B} = A \oplus B$$

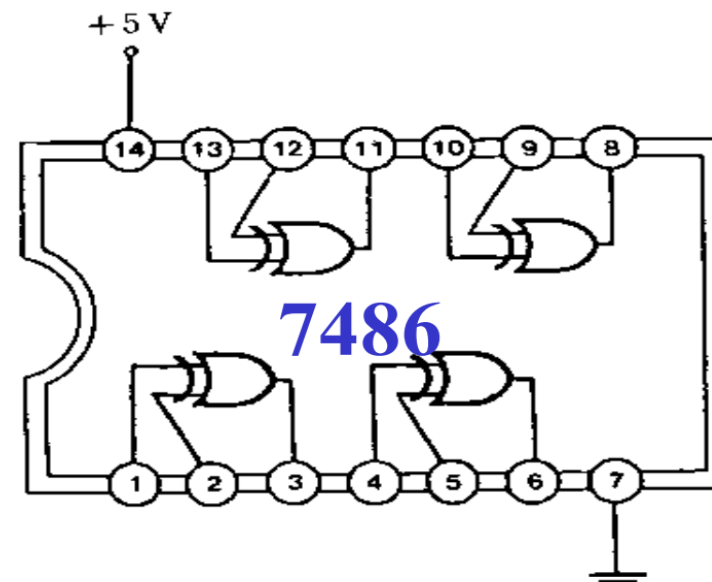
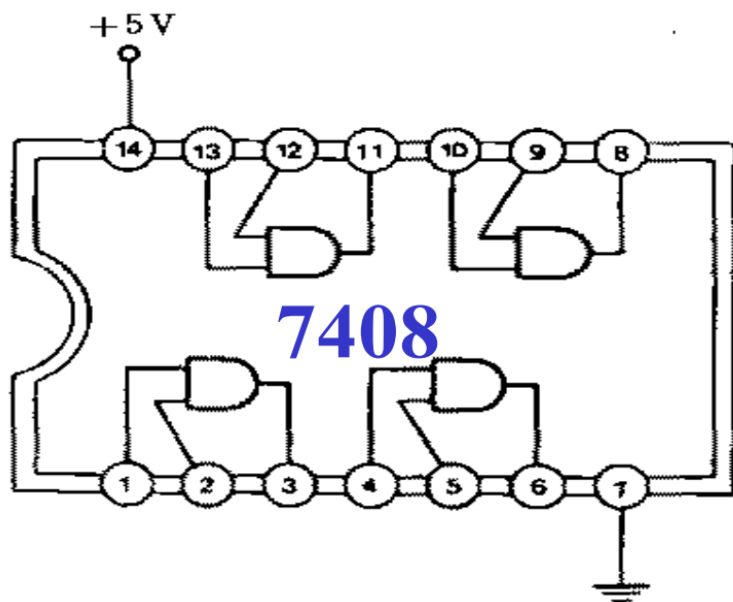
- 以邏輯閘表示：



# 單位元半加器實作

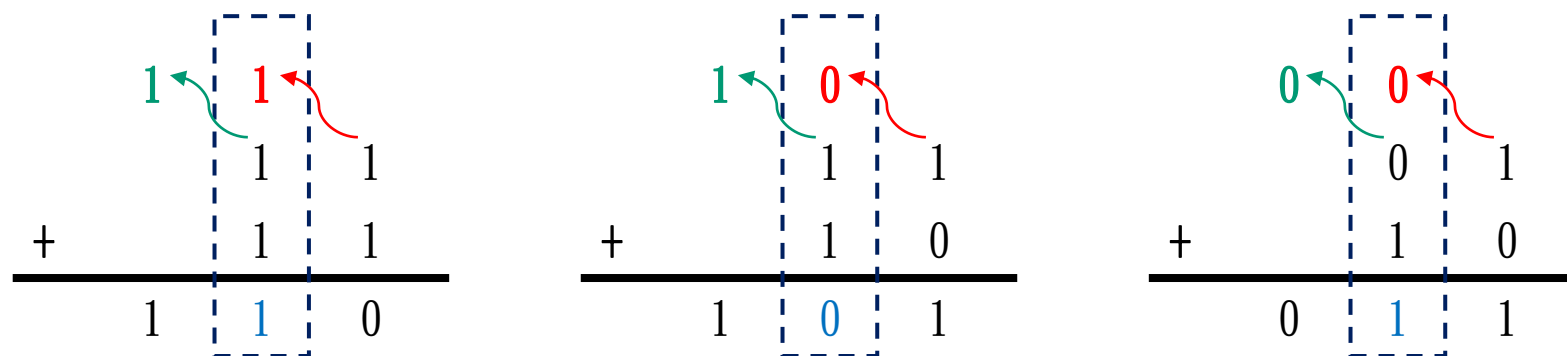
- 試著利用 IC7408、IC7486 以及指撥開關組合出半加器的電路，並使用 LED 燈顯示結果，將輸出結果填入表格，並與前一頁之真值表對照以驗證結果的正確性。

A	B	Carry	Sum
0	0		
0	1		
1	0		
1	1		



## 單位元全加器實作

- 由於加法的運算必須得考慮其他位數帶來的進位，而半加器並沒有考慮到前一個位數運算完後產生的進位，所以我們需要使用全加器來進行進一步的延伸，使整個加法運算可以被完整的實現。
- 舉例：



- 此時輸入項會有3個，分別為A、B，前一位數產生的進位C，運算後的結果Sum，以及此時產生的進位Carry。



# 單位元全加器實作

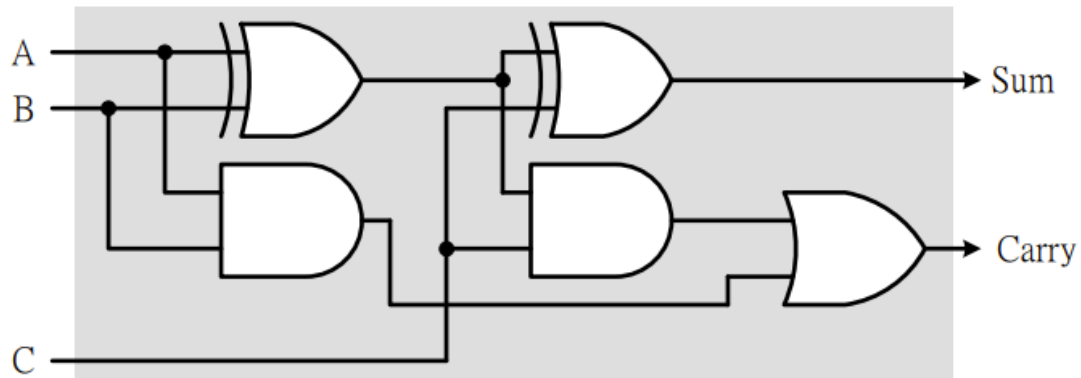
- 跟半加器一樣，我們可以列出其真值表與邏輯運算關係。

A	B	C	Carry	Sum
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

$$\begin{aligned} \text{Carry} &= A \cdot B + B \cdot C + A \cdot C \\ &= A \cdot B + (B + A) \cdot C \\ &= A \cdot B + (B \oplus A) \cdot C \end{aligned}$$

$$\text{Sum} = A \oplus B \oplus C$$

- 以邏輯閘表示：



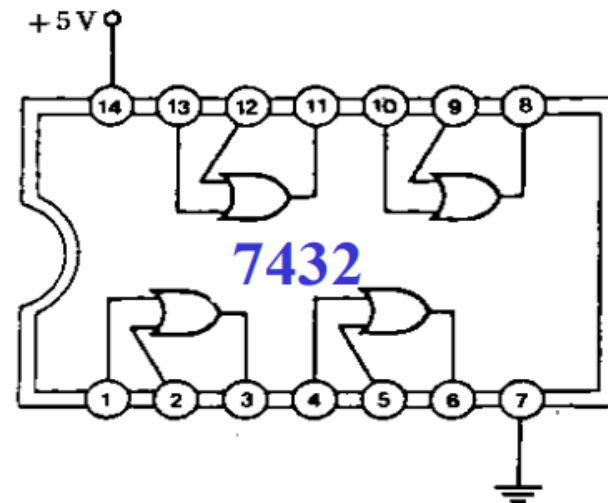
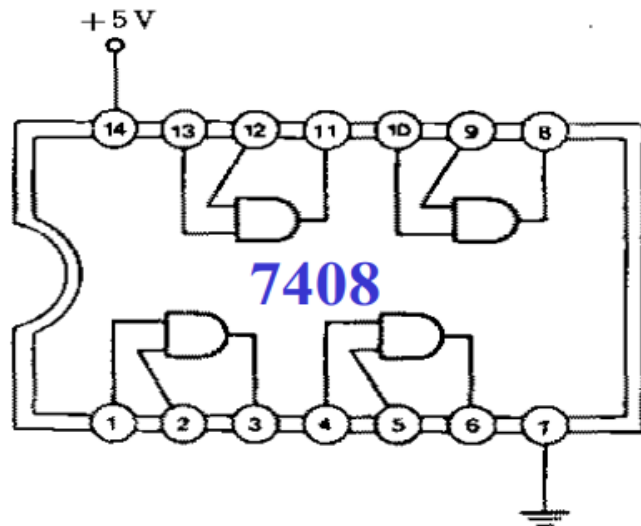
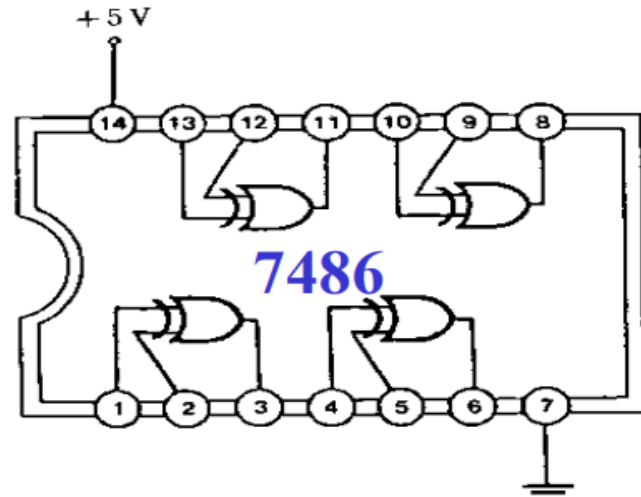
# 單位元全加器實作

- 試著利用 IC7408、IC7432、IC7486 組合出全加器的電路，並使用 LED 燈顯示結果，將輸出結果填入下表，並與前一頁之真值表對照以驗證結果的正確性。

A	B	C	Carry	Sum
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

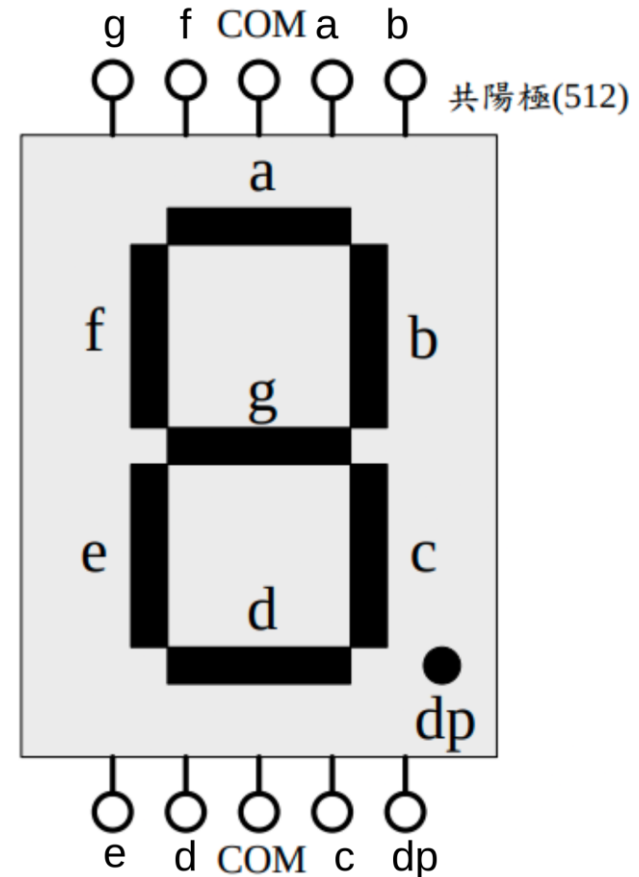
# 單位元全加器實作

- IC7408、IC7432與IC7486：

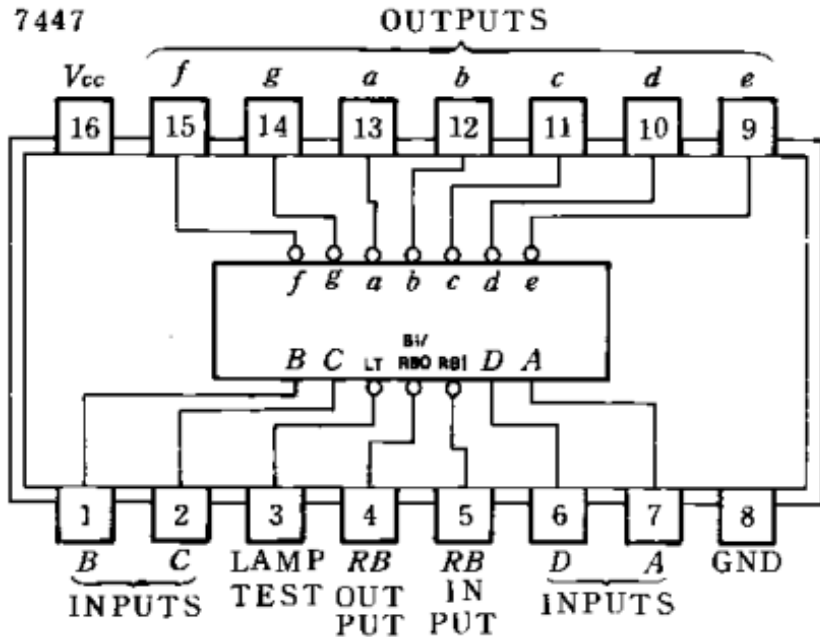
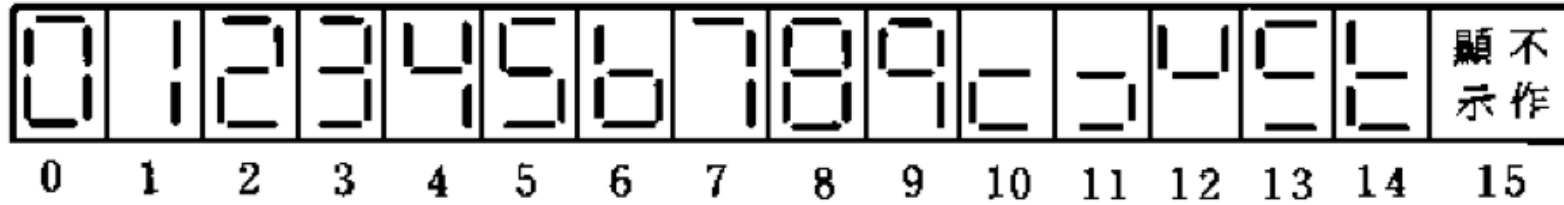


# 七段顯示器

- 七段顯示器是由LED所組合，可分為共陽極、共陰極兩種。其差別在於共陽極要使某一劃發光時必須將0電位(接地)輸入其對應之接腳，而共陰極必須將1電位(5V)輸入其對應之接腳。
- 請利用 $300\ \Omega$  電阻將COM接至 $V_{CC}$



# 七段顯示器

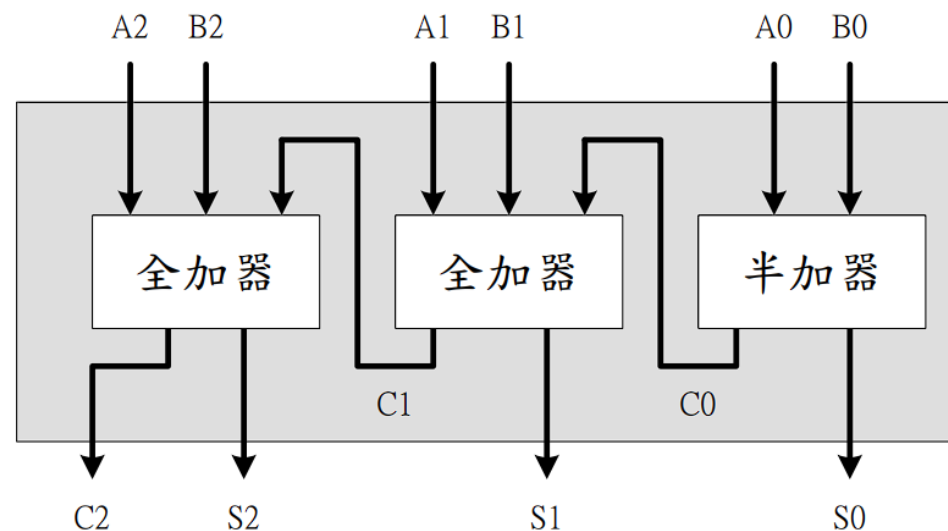
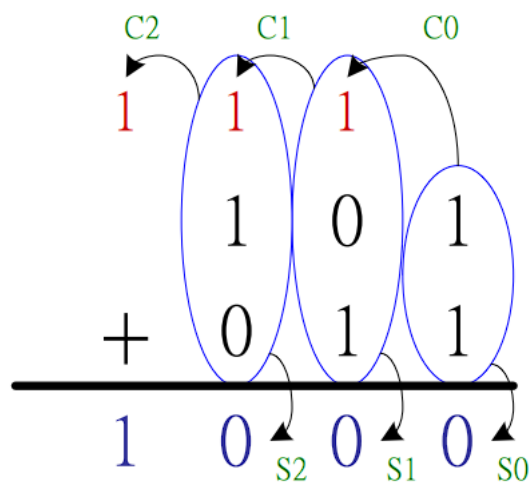


BCD	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
PIN	D	C	B	A

IC 7447的功能是將BCD碼轉至七段顯示器(共陽極)的控制電路，請同學依序將IC 7447各接腳接至七段顯示器所對應的接腳

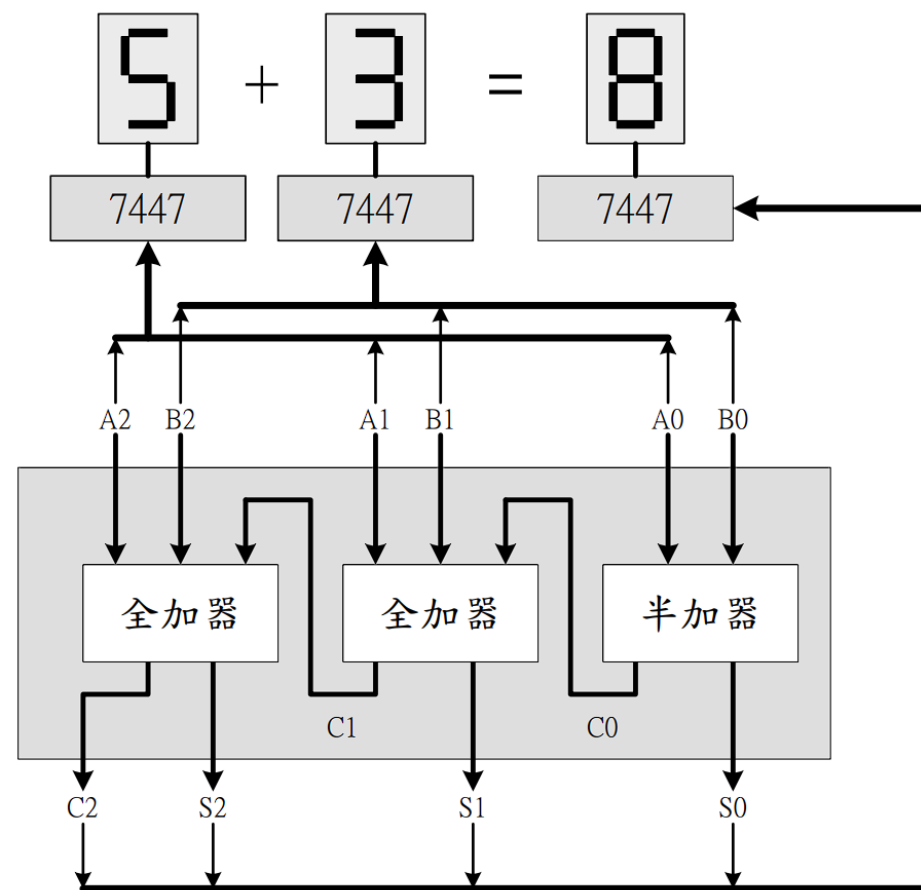
## 多位元全加器實作

- 單位元的半加器與全加器只能實現一個位元的加法，要實現多位元的加法運算，我們必須使用多個全加器和半加器組合出能實現多位元加法的電路
- 最低的位元沒有前面的進位，因此使用半加器
- 之後的位元要考慮其前一個位元的進位，因此使用全加器



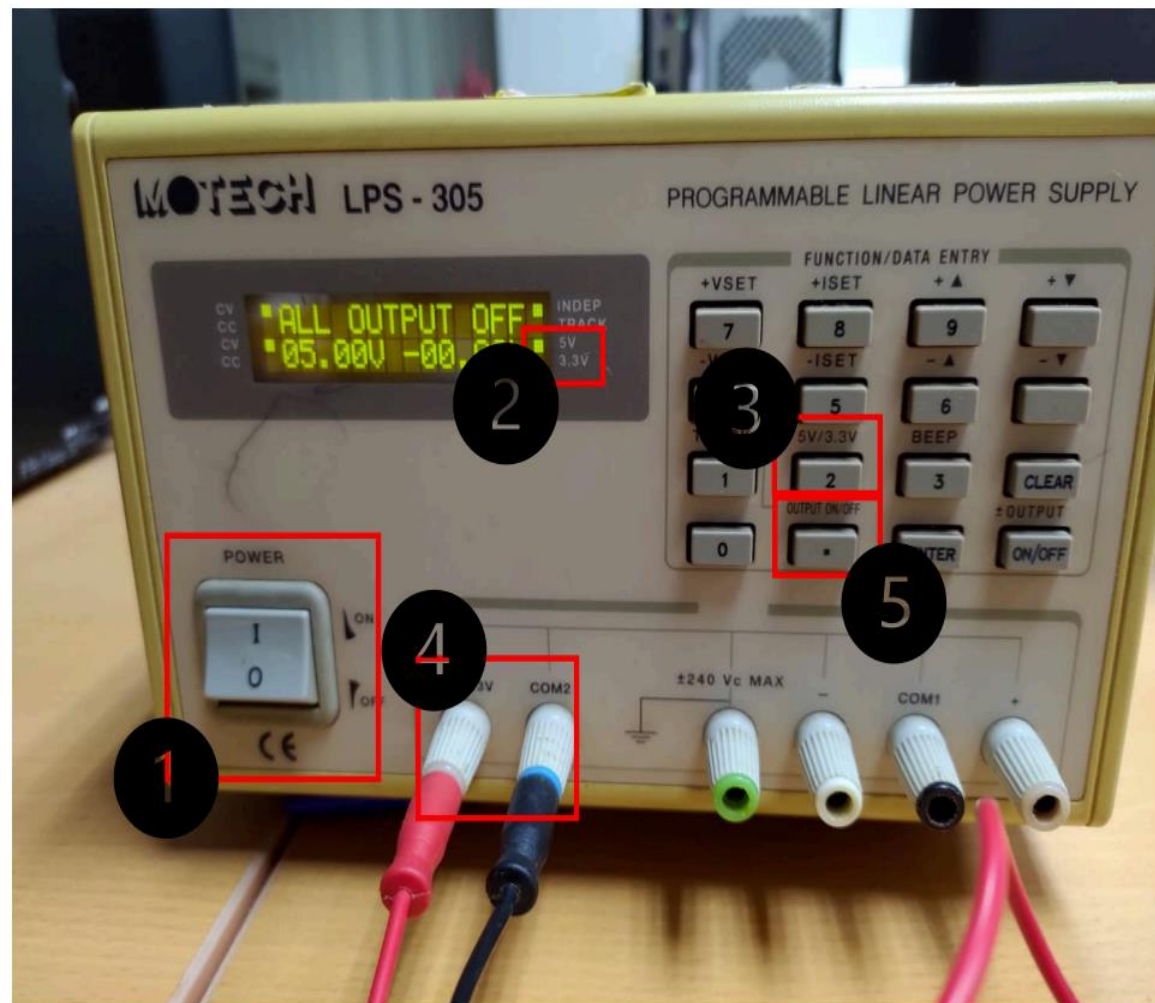
# 多位元全加器實作

- 請利用半加器與全加器組合出3位元的加法器，並將其輸入A、輸入B及輸出結果顯示在七段顯示器中



## 附錄 - 電源供應器

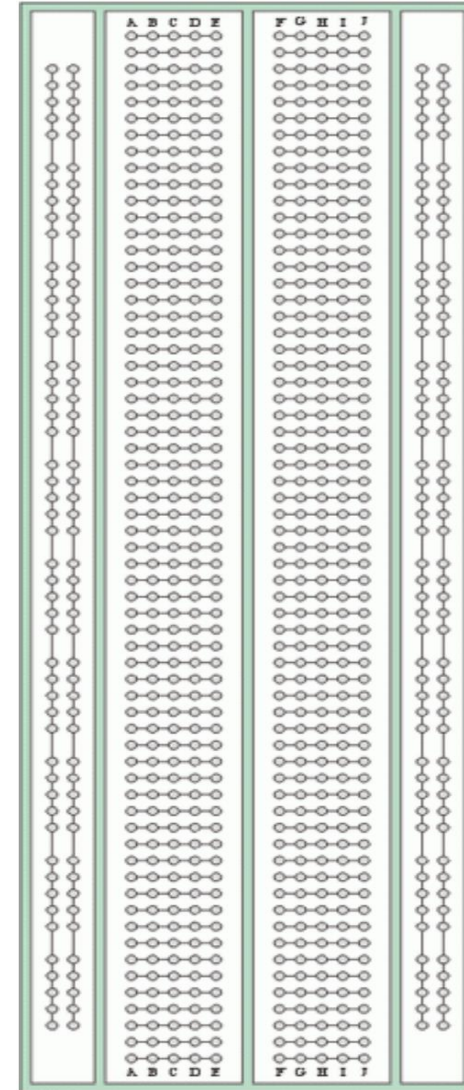
1. 將左側開關打開
2. 確定 5V/3.3V 選擇位於5V位置
3. 如果在3.3V就按一下5V/3.3V切換鍵
4. 將香蕉頭插入5V/3.3V與COM2的插座
5. 確定接線完成後，按下5V/3.3V的Power Output



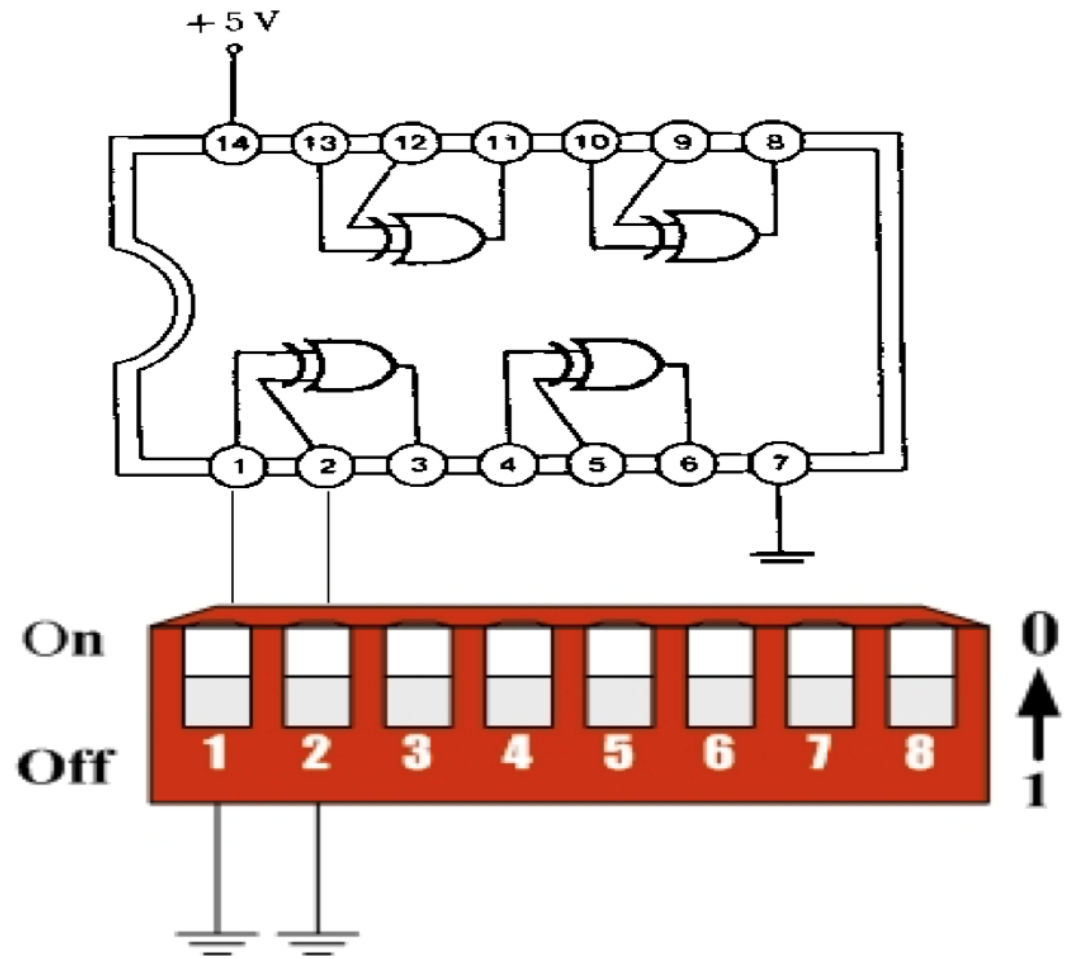


## 附錄 - 麵包板

- 紅色藍色的長條上下相通
  - 若上下有W記號在中間，則上排及下排不互通
- 橫向的 A-E 相通、F-J 相通
- 其餘不互通



# 附錄 - 指撥開關使用示意圖



## 附錄 - IC 夾

- 使用方法如圖，向上拔起

