2010 Summer Camp

ATAU
CSIE

課程內容

程式設計

講師 | 郭明鑫

＜Part 1 ＞

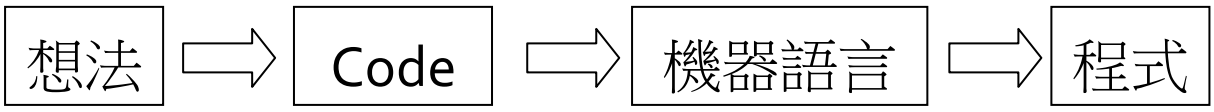
什麼是程式設計？

用現有的語法及指令組合，讓電腦做我們想做的事。
→當你進到電腦的世界，你就必須用使它能懂的語言讓程式運作。
舉個例子，如果外國人用俄語法語或其他語言向你問路，便會溝通不良，無法理解。

如何將指令轉換成程式？

Compiler：把原始碼(Source Code)翻譯成電腦看得懂的語言(即機器語言)的程式。

利用 Compiler(編譯器)，將程式碼轉成電腦能懂的語言。
→將想法寫成程式語言，再用工具轉成機器語言



程式語言的種類

需要程式幫忙翻譯的：C, C++, Java

腳本語言，寫一行跑一行：Python, Ruby, Perl

低階的語言，意義上比較沒這麼直觀，不容易理解：Assembly (組合語言)

機器語言，電腦運作時的型態：Binary Code (二進位碼，01010101..)

寫 C 語言要用到的 Compiler 是？

這裡推薦使用 Dev C++這個 IDE(整合開發環境，內含文字編輯器 + 編譯器 + 除錯器)，來寫我們的程式。

Dev C++內建的 MinGW 就是一個我們能夠利用的 compiler。

下載 Dev C++

官網 http://www.bloodshed.net/devcpp.html

找到下載頁面 http://www.bloodshed.net/dev/devcpp.html，找到以下

Downloads

***Dev-C++ 5.0 beta 9.2 (4.9.9.2)* (9.0 MB) with Mingw/GCC 3.4.2**
Dev-C++ version 4.9.9.2, includes full Mingw compiler system with GCC 3.4.2 and GDB 5.2.1 See NEWS.txt for changes in this release.

Download from:

C 語言概觀

歡迎進入 C 語言的世界。在這個世界中，每行指令都必須以分號 **;** 結尾，而括號也必須像算數時一樣，一左一右配對。

你還會知道使用變數前必須「宣告」，讓電腦知道你想用什麼變數(容器)來存資料。

寫程式就像堆積木一樣，是從基礎的工具(例如 main，即程式開始執行的地方，可以想像成大塊的積木)開始寫起，然後在括號裡面慢慢加東西(把積木往上堆)。一般的程式都有一個大致固定的整體架構，**例如黑框圈起的部分是每個程式都要有的**，而內容再依實際需要去改變。

[*] Untitled1

```
#include<stdio.h>

int main() {

    /*被夾起來的部分是註解。這塊區域要放的，就是你的程式要做的內容。*/

    return 0;
}
```

PS:
暫停程式的方法：方便觀察輸出
在程式開頭加上　　**#include<stdlib.h>**
在要暫停的地方加上　**system("pause");**

程式的組成

宣告部分　→告訴電腦，我想要使用的變數有哪些。

讀取輸入　→讓使用者提供給程式資訊。

運算　　→讓電腦依據程式的指示，去產生想要的結果。

輸出結果　→讓我們看到執行的結果。

宣告：告訴電腦，我們想使用的變數名稱。

形式：**變數型態 變數名稱;**

例：

```
int my_int;
float my_float;
```

這裡的 float(浮點數)，就是小數的意思。

輸入：從鍵盤讀入資料，讓使用者提供程式需要的資訊

形式：**scanf("資料型態", 對應的變數);**

例：

```
scanf("%d", &int_1);  d 指的是 decimal(十進位數)
scanf("%f", &float_1);
scanf("%c", &char_1);
```

運算

形式：**接受結果變數 = 變數 1 [操作] 變數 2**

例：裝在盤子裡的東西 = 食物 + 調味料

這邊的「操作」，可以是+, -, ×, ÷, %等等。

其中%是取餘數，例如 5%2 就是 5 去對 2 取餘數，也就是 1。

特別注意，這邊的 = 和數學上的「等於」並不完全相同，

而是將右方的運算結果「指定」給左邊的意思。

例如 x = x + y，做的是先把 x 和 y 相加，再把這個答案「指定」給 x，

而不像數學上有移項的操作。

需要判斷條件的時候：

用 if 來控制要執行的程式區域

if(符合這邊的敘述) {就做這裡面的事}

例：一元二次方程式 $ax^2+bx+c=0$

```
if (判別式大於 0) {
 一元二次方程式有兩個實根;
}
else if(判別式等於 0) {
 一元二次方程式有重根;
}
else {
 一元二次方程式沒有實數解;
}
```

關係運算子：判斷兩個要比較的條件時使用。

1. 大於 >
2. 小於 <
3. 等於 = =

例：

判別式大於零→ D > 0

判別式等於零→ D == 0

當條件不只一個的時候

用 && 表示 AND，前後都符合時才會去做。

if((條件 1)&&(條件 2)) { …做這些事… }

用 ‖ 表示 OR，前後條件有一個對就行。

if((條件 1) ‖ (條件 2)) { …做這些事… }

重覆迴圈—for ：用來簡化重覆的工作

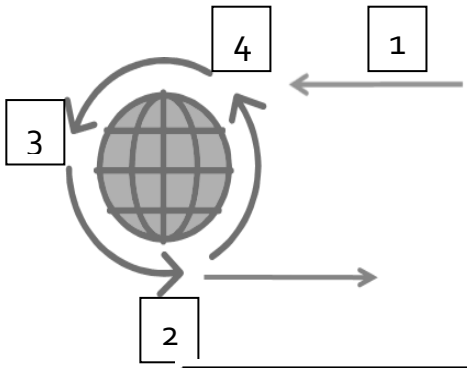
用 for 迴圈把要重覆的部分包起來

1 2 3

for(起始狀態 ； 繼續的條件 ； 每做完一個循環的改變) {
在結束之前，做這個大括號裡面的事;

4

}



例：從 1 加到 10，用 sum 紀錄總和。

```
Number = 0;
sum = 0; //先把 sum 歸零
for(number=1; number<=5; number++) {
  sum = sum + number ;
}
```

```
number      sum
0           0
1           1
2           3
3           6
4           10
5           15
請按任意鍵繼續 . . .
```

輸出：用 printf 把想要的結果輸出到螢幕上

形式：**printf(“輸出的內容”, 對應的變數);**

例：

```
printf("%d\n", int_1);
printf("%f\n", float_1);
printf("%s\n", string_1);
```

＜Part 2＞

形式： 型態　array 名稱[數量]

例：

int　number[10]

→宣告 number[**0**], number[1] … , number[9]等十個變數。

注意這邊是從 0 開始編號。這是規定。

Array ：用來宣告一系列同性質變數

Array：Number

0	1	2	3	4	5	6	7	8	9

Break/Continue：用來控制迴圈執行的流向

在迴圈進行中，用 **break** 跳出一層，或是用 **continue** 略過該迴圈剩下的部分，直接進入下次的迴圈。用這種方式，來改變程式運行的途徑。

例：

```
for (開始拍賣;如果還沒賣完就繼續;下一項)
{
  if(出完價了) {
      一手交錢一手交貨;
      continue;
  }
  else if(天塌下來了!!!!) {
      break; //逃命囉~~~~~
  }
`
```

Overflow(溢位)

變數可以當作是儲存數值的「容器」。
當數字的大小超過變數能儲存的範圍時，便會發生溢位的現象。

例：**int** 型態的變數只能存到 $2^{31}-1$，如果超過的話，它的值便不會是預期的結果。

如果要存更大的數或做更大的運算要怎麼辦？

- **用其他型態的變數**：例如 **long long**，或是 **double**

- **大數運算**：將運算的動作寫入程式，用很多個變數去儲存原本的數。
  [參考圖]

```
  1  2  3  4  5  6
+7  8  9  9  9  9
----------------------
  8  10 12 13 14 15
----------------------
  9  1  3  4  5  5
```

Library：用 google 找 C 語言的函式工作時，可以用的關鍵字。

例如 math.h 裡面有 log, sin, cos 和其他現成的函式可以使用。

[網站例] C/C++ Reference：http://www.cplusplus.com/reference/

[網站例]The C Library Reference Guide：
http://www.acm.uiuc.edu/webmonkeys/book/c_guide/

[網站例]中文參考用文件

用已知的方法化簡問題

・適度的運算

・運用已知的結果或方法

－數學公式

[例] 1+2+...+500 = ?
這邊的 500 看起來可能沒什麼，但是如果換成 500 億呢？

我們知道連加的和就是梯形面積公式，(上底+下底)乘高除 2
因此，應該先用已知的公式稍作化簡，以節省計算的時間。

－消去公因數

[例] C(18,9)

Dynamic Programming (動態規劃)：

保留運算過程中的資訊，在之後繼續使用。

[例] 捷徑問題：我們已經知道在 n×m 的地圖上，捷徑有 C(n+m,m)條。

現在，我想知道，在有缺陷的地圖上，規定只能往右或往上，從甲到乙有幾種走法？

1	■	4	12	乙
1	4	4	8	12
1	3	■	4	4
1	2	3	4	■
甲	1	1	1	1

解決問題的想法：

在工具有限的情況下，儘可能設計方法去嘗試！

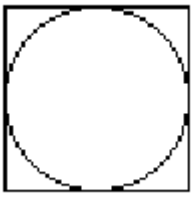
[例] 逼近法：錐體體積=柱體體積*1/3 ？

因為我們知道圓柱體積的算法，但不知道錐體的體積是怎麼來的，
所以將錐體看成一連串細小的圓柱疊成的形狀。

當分割非常小的時候，這種近似的誤差比例，應該會相對地變小。
→利用電腦的速度處理複雜的重覆運算。

[例] 蒙地卡羅法：算圓周率／圓面積

因為用逼近法要用商高定理去求出長寬，中間過程可能會再度造成誤差，
因此我們有一個想法。



將圓內接在正方形內，然後對著正方形的範圍內「隨機」取點。
當取的點非常多的時候，落在圓內的點對上所有點的比例，應該會接近圓面積

Note

硬體概論

講師 — 蔡宗翰

電腦簡介：

電腦（Computer）全名是電子計算機（Electronic computer）
以複雜的電路(包括積體電路,電阻,電線,各種IC半導體)所構成。

電腦是一部能接收資料，並且將資料加以運算處理，產生結果的機器。此外電腦具有運算十分快速，儲存資料容量大，處理資料結果正確的特性。

電腦由很多零件組成，我們通稱這些零件為「電腦硬體」。
ex：中央處理器（CPU, Central Processing Unit）、主機板（Mother Board）、記憶體（Memory）、硬碟（Hard Disk）、鍵盤（Keyboard）、滑鼠（Mouse）、螢幕（Monitor）、印表機（Printer）等。

電腦五大單元：

1. 控制單元

控制單元如同人的中樞神經系統，主要功能在指揮執行順序並控制協調其他單元之動作

2. 算術與邏輯單元

算術邏輯單元(ALU)，是中央處理單元之一部分，它負責執行二進位資料之運算

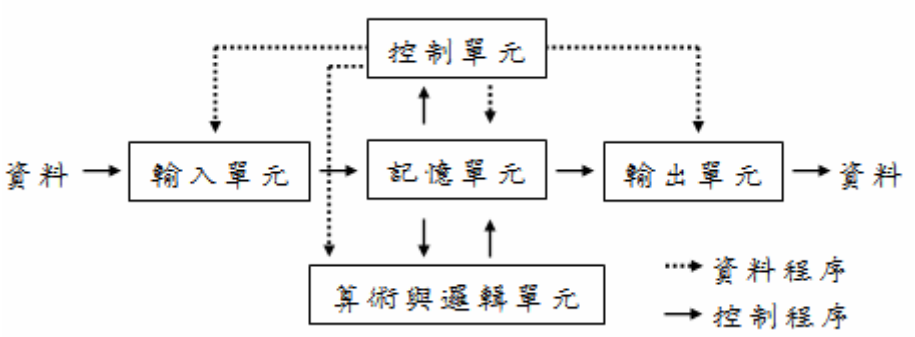
3. 記憶單元

記憶單元是具有記憶功能的設備，又區分為主記憶體與輔助記憶體

4. 輸入單元

輸入單元是用來把資料或程式轉換成CPU 所能接受與處理的設備。ex:

下圖為電腦處理資料之流程簡圖



主記憶體與輔助記憶體：

記憶體可分為兩大類：**主記憶體**與**輔助記憶體**。
輔助記憶體又稱儲存裝置（Storage Devices）。
主記憶體可分層分為：CPU暫存器(register)、CPU快取記憶體(cache)、隨機存取記憶體(RAM: Random Access Memory)。因為前兩項是做在CPU內部，無法改變，所以一般所說的「記憶體」是指隨機存取記憶體(RAM)。

電腦零件：

一、主機板

主機板（Motherboard、Mainboard；又稱**主板**、**系統板**、**邏輯板**、**母板**、**底板**等，英文簡稱「Mobo」），是構成複雜電子系統例如電子計算機的中心或者主電路板。

典型的主板能提供一系列接合點，供處理器、顯示卡、聲效卡、硬碟、記憶體、對外裝置等裝置接合。

它們通常直接插入有關插槽，或用線路連接。主板上最重要的構成元件是晶元組（Chipset）。

而晶元組通常由北橋和南橋組成，也有些以單晶片設計，增強其效能。這些晶元組為主板提供一個通用平台供不同裝置連接，控制不同裝置的溝通。

它亦包含對不同擴充插槽的支援，例如處理器、PCI、ISA、AGP，和 PCI Express。晶元組亦為主板提供額外功能，例如整合顯核，成聲效卡（也稱內置顯核和內置聲卡）。一些高價主板也整合 IrDA、藍芽和 802.11（Wi-Fi）等功能。

晶片組:-負責整個主機板上所有的東西的溝通和控制
北橋 － 高速裝置 － CPU、記憶體、顯示卡
南橋 － 低速裝置 － SATA、USB、PCI、音效、網路、滑鼠、鍵盤

二、中央處理器

中央處理器（Central Processing Unit，**CPU**），是電腦的主要裝置之一。其功能主要是解釋電腦指令以及處理電腦軟體中的資料。

所謂電腦的可編程性主要是指對 CPU 的編程。CPU、內部記憶體和輸入／輸出裝置是現代電腦的三大核心部件。

由積體電路製造的 CPU，20 世紀 70 年代以前，本來是由多個獨立單元構成，後來發展出微處理器 CPU 複雜的電路可以做成單一微小功能強大的單元。

三、記憶體

隨機存取記憶體（Random access memory，RAM），屬於揮發性記憶體，即斷電後會失去所儲存之資料。

主要有以下兩種：

DRAM（Dynamic random access memory，動態隨機存取記憶體）

SRAM（Static random access memory，靜態隨機存取記憶體）

四、 硬碟

硬碟（Hard Disk Drive，簡稱 HDD）是電腦上使用堅硬的旋轉碟片為基礎的非揮發性（non-volatile）儲存裝置。

它在平整的磁性表面儲存和檢索數位資料。資訊透過離磁性表面很近的寫頭，由電磁流來改變極性方式被電磁流寫到磁碟上。

資訊可以透過相反方式回讀，例如磁場導致線圈中電力改變或讀頭經過它的上方。

硬碟按資料介面不同，可以分成數種，常見桌上型電腦之資料介面為 ATA（IDE）和 SATA。

ATA 全稱 Advanced Technology Attachment，是用傳統的 40-pin 並列資料線連線主板與硬碟的，外部介面速度最大為 133MB/s，因為並列線的抗干擾性太差，且排線佔空間，不利電腦散熱，將逐漸被 SATA 所取代。

SATA，全稱 Serial ATA，也就是使用串列埠的 ATA 介面，因抗干擾性強，且對資料線的長度要求比 ATA 低很多，支援熱插拔等功能，已越來越為人所接受。SATA-I 的外部 介面速度為 150MB/s，SATA-II 更達 300MB/s，SATA 的前景很廣闊。而 SATA 的傳輸線比 ATA 的細得多, 有利於機殼內的空氣流通。

現行主流為SATA-II，IDE已經瀕臨淘汰。

現時，SATA 分別有 SATA 1.5Gbit/s、SATA 3Gbit/s 和 SATA 6Gbit/s 三種規格。

五、顯示卡

顯示介面卡（Video card、Graphics card、Video adapter），台灣與香港簡稱為**顯示卡**，是個人電腦最基本組成部分之一。顯示卡的用途是將電腦系統所需要的顯示資訊進行轉換驅動顯示器，並向顯示器提供行掃描訊號，控制顯示器的正確顯示。

六、電源供應器

電源供應器 (Power)
提供整部電腦電源所需的電源，此部分非常重要，因為一但供電不穩定，長期下來會讓電腦零件「慢性燒毀」。

七、光碟機

光碟機 (DVD/CD-Rom)
可分為：CD-Rom(只能讀CD)、CD-RW(CD燒錄機，只能燒CD)、DVD-Rom(只能讀DVD)、DVD-RW(DVD燒錄機)和Combo機(CD-Rom+CD-RW+DVD-Rom)。下一代技術為Blu-ray Disc(簡稱BD、藍光)和HD-DVD(High Definition DVD)。

八、I/O 裝置

I/O裝置為Input/Output裝置，是電腦與外界，外界與電腦溝通的裝置，或外接設備。EX:螢幕、鍵盤、滑鼠。少了它們，電腦可說就是一部廢鐵，所以他們非常重要。注意，沒有鍵盤的話是無法開機的。

Note

數位邏輯概論

講師 | 李昱璇

數位邏輯

數位邏輯的探討:

「邏輯的推論」和「數位電路元件」是學習數位電子的基礎。

1.人類對於邏輯的觀念早已熟析，常利用邏輯的觀念推論一些事物，例如：解決數學問題、根據事實做事務的推論、根據新資料修正原本推論。

2.數位邏輯探討的是以 1 和 0 兩個數位作合理的推論。任何事情不是 0 就是 1。

3.數位電路即是利用數位邏輯表示電子電路的高電壓〈1〉和低電壓〈0〉以控制電路。

什麼是數位(digital)？什麼是類比(analog)？

數位(digital)：表示某些事物具備不連續的、離散的性質。

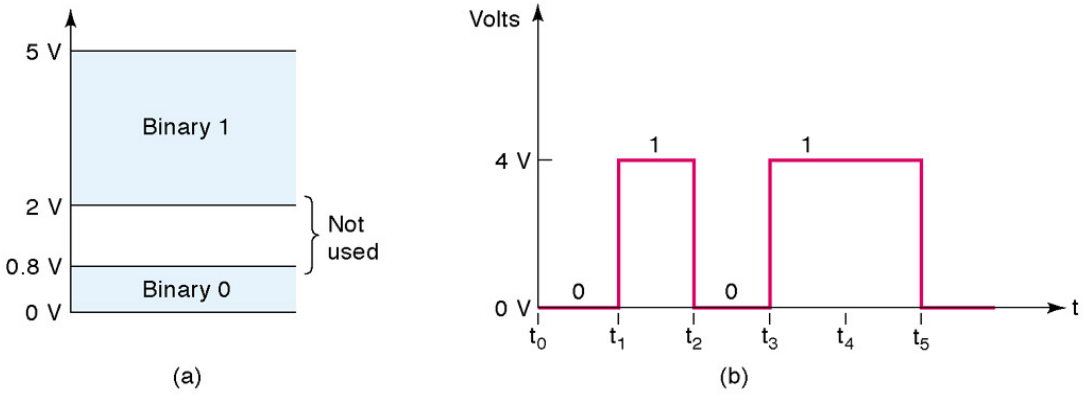
類比(analog)：表示某些事物是有連續的性質。

數位的色譜(普普風格的色塊)

類比的色譜(水彩渲染、彩虹)

電子學上，電壓和電流的大小常是需要測量的值，類比式的電子設備其電壓和電流的變化為連續性的；而數位式的電子設備，其電壓和電流的變化是跳躍性的，也就是有分離性，具有階梯性，也就是由某一定值跳到另一值。

所以數位電路必須定義在某個電壓準位是 1，而再另一個電壓準位是 0。



數位和類比的溝通

真實的世界是類比的，為了處理的方便，常被轉換為數位式的。

數位(Digital)轉成類比(Analog) ，簡稱 D/A。

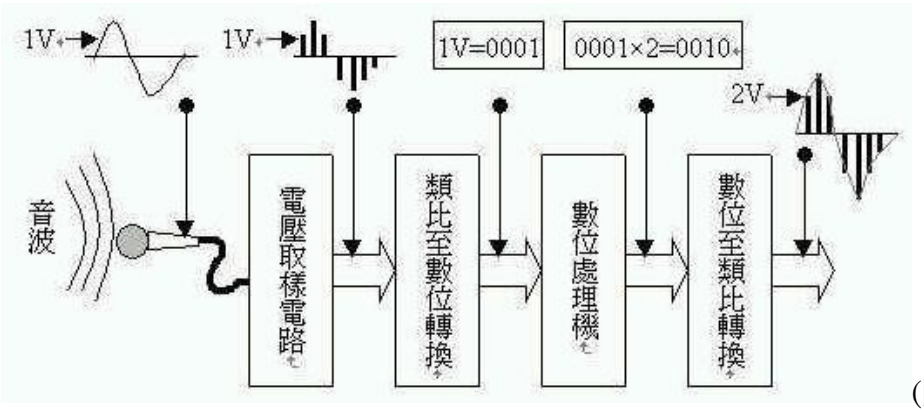
類比(Analog)轉成數位(Digital) ，簡稱 A/D。

數位系統與類比系統的運作方式

我們以音波放大器為例

數位系統

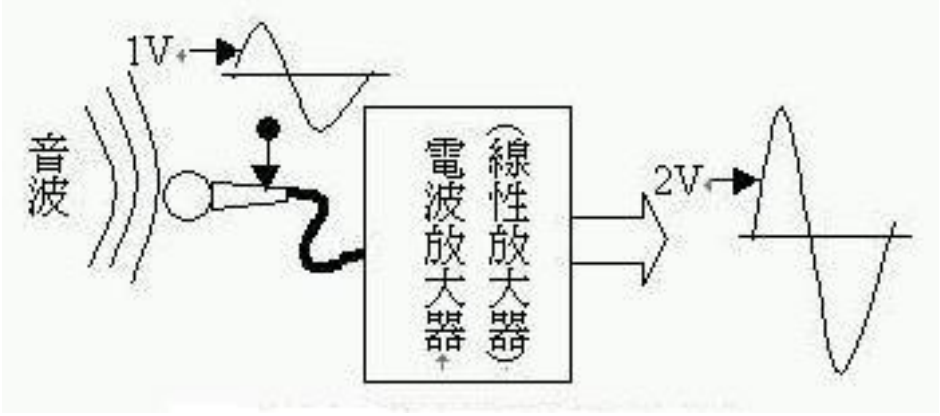
如圖 1 所示，當一個人對著麥克風說話的時候，麥克風會將音波轉換成連續的電波，而數位系統首先要做的就是將此連續的電波分成很多的片段，每一片段得到的電壓稱為「取樣電壓」，然後將取樣電壓依其大小付予一個相對的二進制的值(數碼)，這樣的處理稱為「類比至數位轉換(A/D)」，轉換後的數碼再經過數位處理機加以運算，以此例而言運算的目的在將輸入的數碼乘以使用者要求的倍數，因此經過數值處理機運算後得到另一組數值較大的數碼，此數碼再經由「數位至類比轉換(D/A)」電路轉換成電壓，一個連續的輸入電壓經由處理後至類比輸出端已是被放大的電波了，由於此種系統負責處理放大倍數的電路是數位處理機(一般電腦包含的功能)，主要作用在於數碼的運算及處理，因此本例可稱為是一個數位系統的放大器了。



(圖１)

類比系統

如圖 2 所示，麥克風輸入的電波經由一個電波放大器，直接將輸入的電波以電晶體原有的放大特性加以放大，此種音波放大的過程未經任何的數位處理，而且輸入至輸出電波都是「連續性的」，不像數位系統中會將輸入電波分成許多「非連續性的」片段來處理，因此我們可稱此放大器是屬於類比式的放大器。



(圖２)

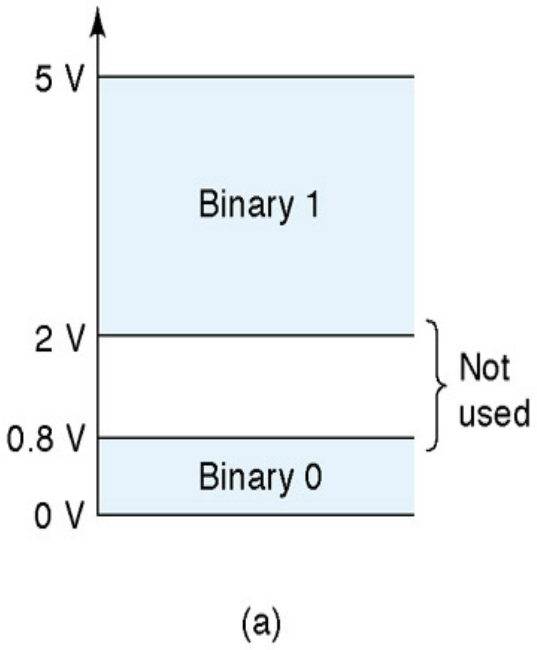
數位與類比的比較就是為什麼需要數位化系統的原因

(1) 數位系統的運算精確而類比系統誤差較大

以前例而言，在**數位系統**中假設輸入電波經取樣後的電壓是 1V，經類比至數位轉換後的編碼是 0001(此碼表示數目 1)，經過數位處理機加以運算後的值是 0010(此碼表示數目 2)，再經由數位至類比轉換至輸出端就得到 2V 的電壓。同理，若數位處理機所設定的放大倍數不變，數位處理系統對於每一個取樣電壓皆可做相同倍數的放大，在此例中為 2 倍。然而**類比系統**，同樣的將 1V 輸入類比放大器，並調整放大器的增益(放大倍數)為 2，我們得到的放大電壓可能是 1.8V 或 2.1V，而非應有的放大電壓 2V，此種誤差乃電晶體放大電路先天的特性使然，尤其是溫度變化較大的環境之下，運算值(本例是指放大倍數)就不如數位系統來得穩定可靠，所以精確的處理對於類比系統考慮就較為困難了。

(2)數位系統較類比系統不容易被雜訊干擾

數位系統在運算的過程中所處理的信號電壓不是高(代表 1 的電壓)就是低(代表 0 的電壓)，高低之間會留有一段容易區分的距離，此種距離容忍了一些雜訊的重疊干擾，使得數位系統分辨代表數值的高低信號不至錯亂，所以運算的結果也是穩定精確的。如右圖所示，一般把 0V~0.8V 間視為低電壓，電壓記為 0，2V~5V 間視為高電壓，電壓記為 1，介於 0.8V~2V 間的電壓則為錯誤訊息，表電路發生問題需作檢查。然而，**類比系統**將小信號直接透過(電晶體)放大器放大，在放大的同時雜訊也跟著被放大了，其放大的結果就可想而知了。



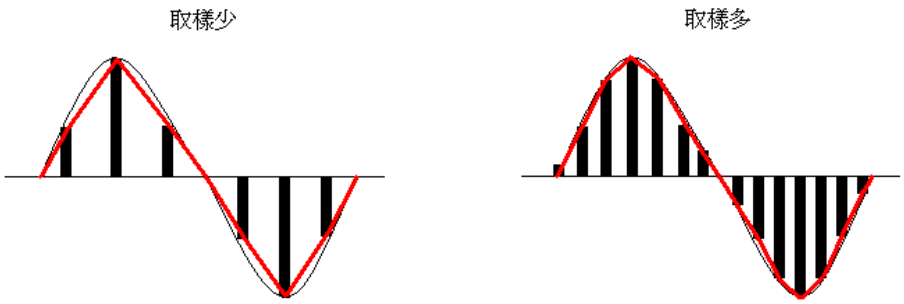
(3)數位系統的信號儲存較類比系統容易

數位系統儲存信號時，儲存的是代表信號的數碼，可由任何 1 或 0 的型態組合，例如磁場的「強」與「弱」或「N 極」與「S 極」，電壓的「高」與「低」，光線的「有」與「無」，所以數位系統可儲存信號的裝置種類很多，包括磁帶機、磁碟機、隨機存取記憶體(RAM，一種以電壓儲存的記憶體)、 光碟機，甚至以打孔區分有無的紙帶，以鉛筆塗抹的答案卡。然而，**類比系統**是要依振幅比例將信號電壓儲存下來，在市面上可以看到用來儲存的方式，就只有錄音或錄影帶了，早期也用金屬板或塑膠板刻下音波的振幅做成唱片，但是現在已經很難找到了，因為儲存後的效果和保存期限實在不能和數位系統的 CD(compact disk)相比。

(4)數位系統的信號編輯較類比系統容易

所謂信號編輯是指信號源的複製、修改、剪接、加回音特效、兩個以上信號源的混合等等，這對於**數位系統**而言只是對於一連串編碼的移動或再運算，通常一部桌上型電腦即可完成，但對於**類比系統**的音源編輯而言，就可能需要多台的錄音機、混音器、可程式編輯控制機等等，而且操作上對於時間點的掌握是相當麻煩的。

綜合以上的分析比較，我們知道選擇數位系統通常是優於選擇類比系統的，但是數位系統是不是就沒有缺點了呢？我們仔細觀察圖 1 中數位至類比轉換後之輸出，它仍然是由很多片段所組合成的波形，嚴格的說它與未放大前的波形相比是失真的波形，只有取樣的次數（頻率）增加時，它會更接近原來的波形，但隨著輸入電波的頻率增高，取樣頻率就要更高，這樣一來數位系統中所有電路的處理速度都要增高，儲存取樣資料的記憶容量也得要增大，這些都是我們以後在研究數位電路必須留意的地方。



數位邏輯的發展

數位邏輯裝備的體積和價格降低的難以想像，例如：電腦。在新方向的發展中，同時取代舊有的類比電路才能製造的產品，數位電視就是一個明顯的例子。

產品分成三種類型：電腦、周邊元件、測試、量度、家用電子產品。

數位 IC 未來目標：變小、多功能化、便宜、便利、節源。

二、數位邏輯的二進位舉例邏輯的二元相對的名詞

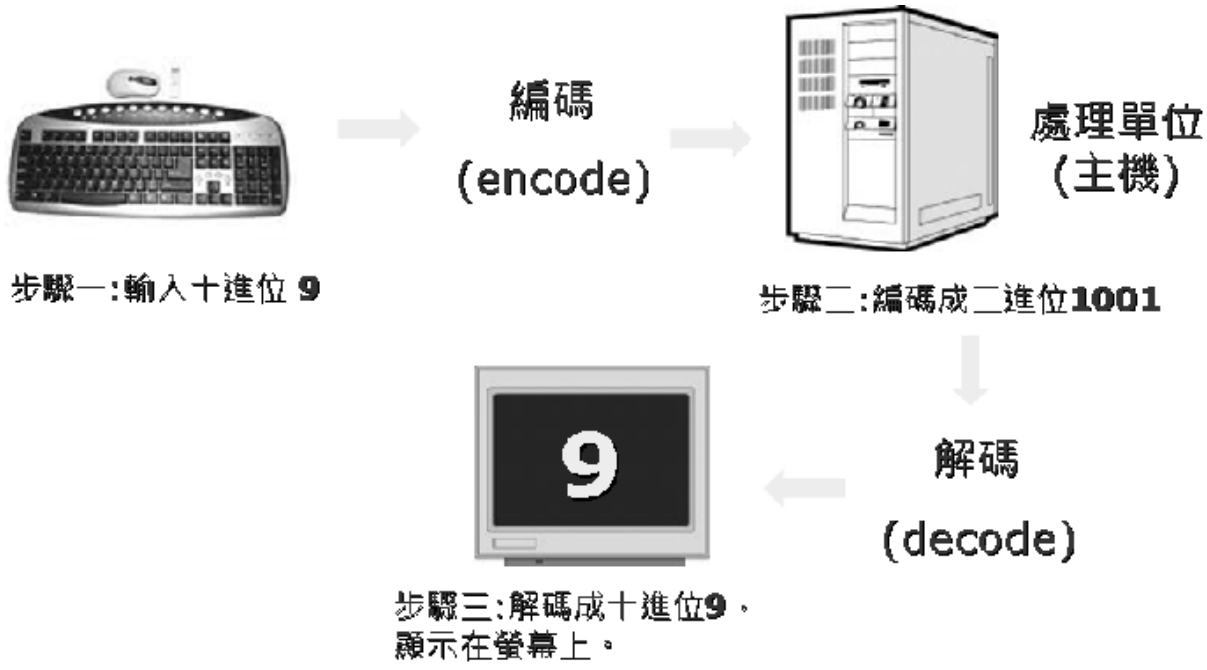
數位邏輯的二進位應用

數位邏輯的二進位特性，即使用兩種邏輯狀態，就能使邏輯電路發揮其所具有的推論和記憶的功能。數位邏輯的學習，需要配合信號的編碼〈encode〉和解碼〈decode〉的過程。

編碼〈encode〉是把人類習用的信號編成二進碼。

解碼〈decode〉是把二進碼還原成人類能懂的信號。

我們用十進位數利用鍵盤輸入，經過編碼器編成二進碼，然後經由處理單元處理後，經解碼器還原成十進位，再利用顯示幕顯示出來。



二、數目系統

數目系統有可以有無限多種，但在我們日常生活中常用到的卻相當有限，除了十進制之外，還有以六十進位的分、秒計時，十二支為一打，十二打為一筐的十二進制，再來就是電腦相關設計者需要熟悉的二進制了。

二進制:用 2 個數字表示數值，0~1。
八進制:用 8 個數字表示數值，0~7。
十進制:用 10 個數字表示數值，0~9。
十六進制:用 16 個數字代表數值，0~9、A(10)、B(11)、C(12)、D(13)、E(14)、F(15)。

在計算機中, 二進制用以表示:
(1)數值 (2)記憶體地址 (3)指令碼 (4)文字或符號 (5)內部狀態

十進制轉換成二進制、十六進制

十進制轉二進制

整數部份以 2 當作除數做連除法，除到商小於 2 為止，餘數由右至左排成的數目即為二進制的整數。以 44 為例，44 的二進制表示法則為 101100。

$$44_{10} = 32+8+4+0 = 0*2^6+1*2^5+0*2^4+1*2^3+1*2^2+0*2^1+0*2^0$$
$$= 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0_{\,2}$$

2| 44 ----- 0
2| 22 ----- 0
2| 11 ----- 1
2| 5 ----- 1
2| 2 ----- 0
2| 1 ----- 1 ↑
0

[練習一下吧]
$79_{(10)}$ =
$214_{(10)}$ =

十進位轉十六進位

整數部份以 16 當作除數做連除法，除到商小於 16 為止，餘數由右至左排成的數目即為二進制的整數。以 751 為例，751 的十六進制表示法則為 2EF。

$$751_{10} = 2*16^2+14*16^1+15*16^0$$
$$= 2 \quad E \quad F_{16}$$

16| 751 ----- F
16| 46 ----- E
2 ↑

[練習一下吧]
$79_{(10)}$ =
$214_{(10)}$ =

[備註] A=10 B=11 C=12 D=13 E=14 F=15

其他進位轉成十進位

二進位轉十進位

以 1101 為例，因為是二進位，所以要以 2 為基底來換算成十進位。

1 1 0 1

$= 1\times2^3+1\times2^2+0\times2^1+1\times2^0$

$= 8 + 4 + 0 + 1$

$= 13$

[練習一下吧]

$10010_{(2)}$ =

$110111_{(2)}$ =

十六進位轉十進位

以 12AB 為例，因為是十六進位，所以要以 16 為基底來換算成十進位。

1 2 A B

$= 1\times16^3 + 2\times16^2 + A\times16^1 + B\times16^0$

$= 1\times4096 + 2\times256 + 10\times16 + 11\times1$

$= 4779$

[練習一下吧]

$39C_{(16)}$ =

$A80_{(16)}$ =

[備註] A=10 B=11 C=12 D=13 E=14 F=15

三.BCD Code(Binary Code of Decimal)

將十進位數字中的每一個數字以二進位表示，以 4 位元為一組，從 0 編碼至 9 一共十個編碼，可用於運算或資料記錄，便於電腦從業人員對於數目的辨識，但以位元利用而言比較浪費記憶體空間。

$874_{(10)}$ = 1000 0111 0100

$943_{(10)}$ = 1001 0100 0011

四.0~15 用各數字系統表示

十進位 Decimal	二進位 Binary	八進位 Octal	十六進位 Hexadecimal	BCD
1	1	1	1	0001
2	10	2	2	0010
3	11	3	3	0011
4	100	4	4	0100
5	101	5	5	0101
6	110	6	6	0110
7	111	7	7	0111
8	1000	1 0	8	1000
9	1001	1 1	9	1001
10	1010	1 2	A	0001 0000
11	1011	1 3	B	0001 0001
12	1100	1 4	C	0001 0010
13	1101	1 5	D	0001 0011
14	1110	1 6	E	0001 0100
15	1111	1 7	E	0001 0101

三、0 和 1 的世界

基本邏輯閘

邏輯閘:特定的電路線路,可以將一組或一組以上的輸入轉換為特定需求的輸出。
真值表：將所有輸入與輸出的關係，以逐條方式分析列舉出來，所得的表格稱為真值表。

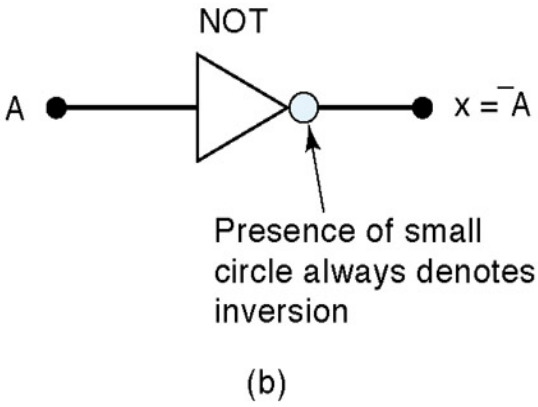
1.NOT Gate

有一個輸入端和一個輸出端，輸出端的狀態永遠與輸入端相反。

NOT

A	$x = \overline{A}$
0	1
1	0

(a)



(b)

2.AND Gate

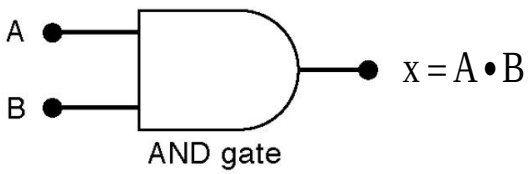
有兩個以上的輸入端和一個輸出端，當任何一個輸入端為邏輯 0 時，輸出端必為邏輯 0 ，僅輸入端皆為邏輯 1 時，輸出端才會為邏輯 1 。

口訣:任一輸入為 0，輸出為 0。

AND

A	B	$x = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

(a)



(b)

3.OR Gate

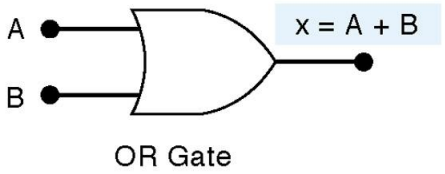
有兩個以上的輸入端和一個輸出端，當任何一個輸入端為邏輯 1 時，輸出端必為邏輯 1 ，僅輸入端皆為邏輯 0 時，輸出端才會為邏輯 0 。

口訣:任一輸入為 1，輸出為 1。

OR

A	B	x = A + B
0	0	0
0	1	1
1	0	1
1	1	1

(a)

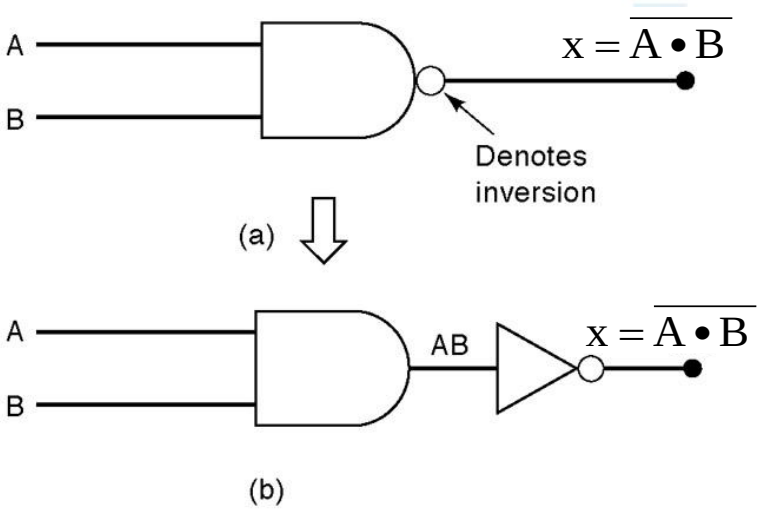


OR Gate

(b)

4.NAND Gate

功能相當於 AND Gate 的輸出端加上一個 NOT Gate，有兩個以上的輸入端和一個輸出端，當任何一個輸入端為邏輯 0 時，輸出端必為邏輯 1 ，僅在輸入端為邏輯 1 時，輸出端才會為邏輯 0 。



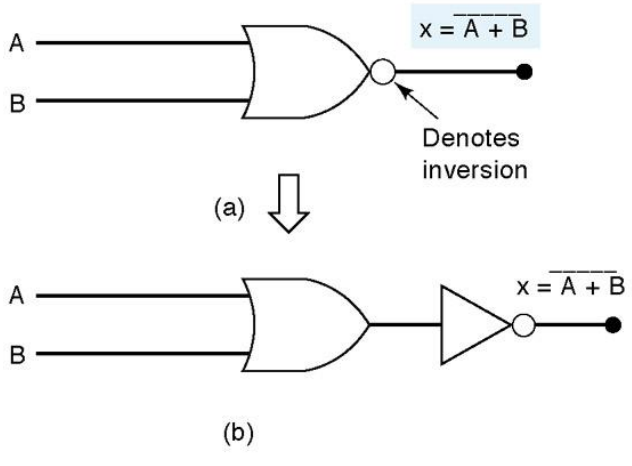
(b)

		AND	NAND
A	B	$A \bullet B$	$\overline{A \bullet B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

(c)

5.NOR Gate

功能相當於 OR Gate 的輸出端加上一個 NOT Gate，有兩個以上的輸入端和一個輸出端，當任何一個輸入端為邏輯１時，輸出端必為邏輯０，僅在輸入端為邏輯０時，輸出端才會為邏輯１。



		OR	NOR
A	B	A + B	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

(c)

布林代數 (Boolean Algebra)

布林代數是英國數學家喬治布林(George Boolean)於 1854 年發表處理數位邏輯的代數運算式，只是依種邏輯結果與判斷條件之間的關係表達，就像是算數一樣是屬於符號語言的一種。

a.布林運算

(1) X・0 = 0	AND 的運算，在變數條件均為 1 時結果方為 1 ，此定理中的一個變數已經固定為 0，所以不管 X 為 1 或 0 其結果必為 0。
(2) X・1 = X	AND 的運算，在變數條件均為 1 時結果方為 1 ，此定理中的一個變數已經固定為 1，若 X 為 1 則結果為 1，若 X 為 0 則結果為 0，所以 X・1 = X。
(3) X・X = X	AND 的運算，在變數條件均為 1 時結果方為 1 ，若 X 為 1 則 1・1 = 1，若 X 為 0 則 0・0 =0，所以 X・X = X。
(4) $X + \overline{X} = 0$	AND 的運算，在變數條件均為 1 時結果方為 1 ，而 X 與 $\overline{X}$ 總是相反的，亦即 1・0 = 0 或 0・1 =0，所以 $X + \overline{X} = 0$ 。
(5) X + 0 = X	OR 的運算，在變數條件任何一者為 1 時結果為 1 ，若 X 為 1 則 1+0 = 1，若 X 為 0 則 0 + 0 =0，所以 X + 0 = X。
(6) X + 1 = 1	OR 的運算，在變數條件任何一者為 1 時結果為 1 ，此定理中的一個變數已經固定為 1，所以 X+1 = 1。
(7) X + X = X	OR 的運算，在變數條件任何一者為 1 時結果為 1 ，若 X 為 1 則 1+1 = 1，若 X 為 0 則 0 + 0 =0，所以 X + X = X。
(8) $X + \overline{X} = 1$	OR 的運算，在變數條件任何一者為 1 時結果為 1 ，而 X 與 $\overline{X}$ 總是相反的，亦即 1 + 0 = 1 或 0 + 1 = 1，所以 $X + \overline{X} = 1$

b.狄摩根定律(Demorgan’ s Theorems)

1. $\overline{(A+B)} = \overline{A} \bullet \overline{B}$
2. $\overline{(A \bullet B)} = \overline{A} + \overline{B}$

口訣:橫線段，OR、AND 交換

下表以 $\overline{A+B} = \overline{A} \bullet \overline{B}$ 為舉例真值表，證明之。

A	B	A+B	$\overline{A+B}$	A’	B’	$\overline{A}+\overline{B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

註:狄摩根定律不只適用於二變數，同時也適用於多變數。

Note

數位邏輯實作

講師 | 楊曜賓

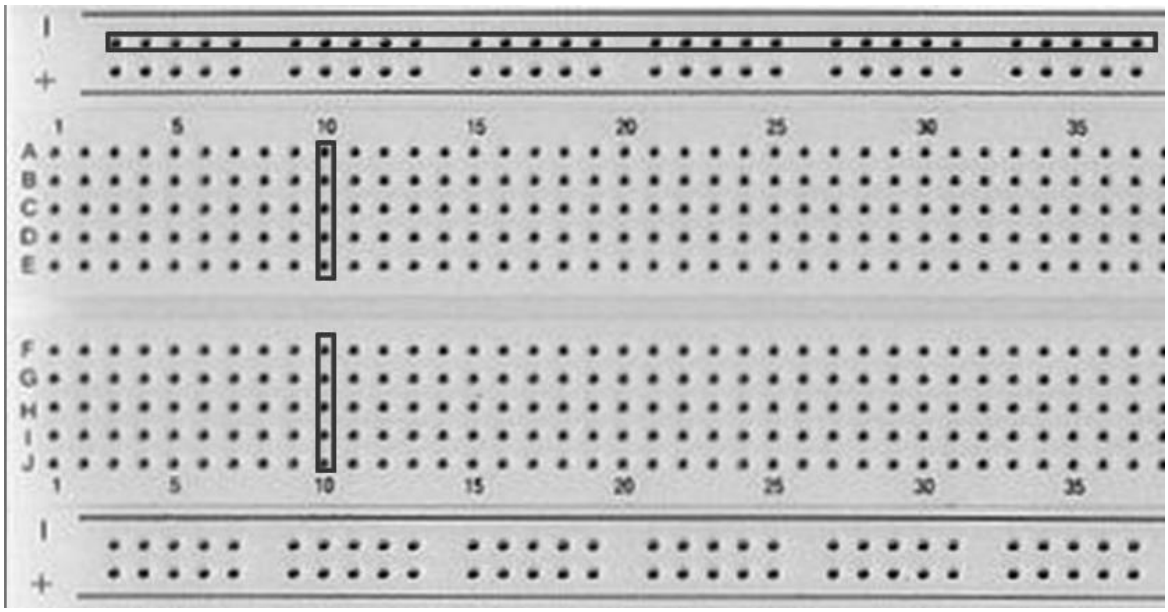
電路概要:

我們將使用兩個 74283 全加器及 1 個 7447 解碼器來製做 1 個 4-bit BCD 加法器

器材介紹:

1. 麵包板：

實驗室裡常用來接電路的器材，比起印刷電路板，麵包板具有修正電路方便的好處。即使電路接錯，只需要將電子零件拔起即可修正，無須焊接電路。



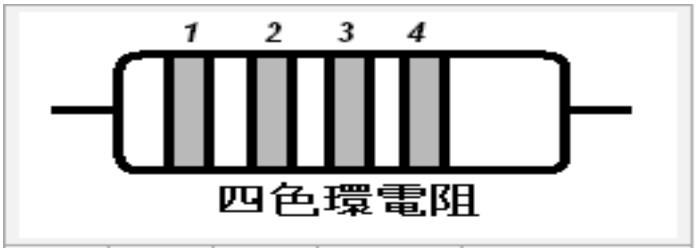
2. 電源供應器：

(1) Vcc：以電壓 5V 來代表邏輯 1(High)
(2) GND：接地(0V)，相當於邏輯 0(Low)
基本上這次的實作只會用到這兩種電壓

3. 電阻器：

電阻值：(10a + b) × 10c　(第四個顏色代表誤差)
※這次實驗會用到的電阻為 330 歐姆，前三個顏色為「橙橙棕」

a　b　c



顏色	1 十位	2 個位	3 倍數	4 誤差值
黑	0	0	×1	
棕	1	1	$\times 10^{1}$	±1% (F)
紅	2	2	$\times 10^{2}$	±2% (G)
橙	3	3	$\times 10^{3}$	
黃	4	4	$\times 10^{4}$	
綠	5	5	$\times 10^{5}$	±0.5% (D)
藍	6	6	$\times 10^{6}$	±0.25% (C)
紫	7	7	$\times 10^{7}$	±0.1% (B)
灰	8	8	$\times 10^{8}$	±0.05% (A)
白	9	9	$\times 10^{9}$	
金			$\times 10^{-1}$	±5% (J)
銀			$\times 10^{-2}$	±10% (K)
透明				±20% (M)

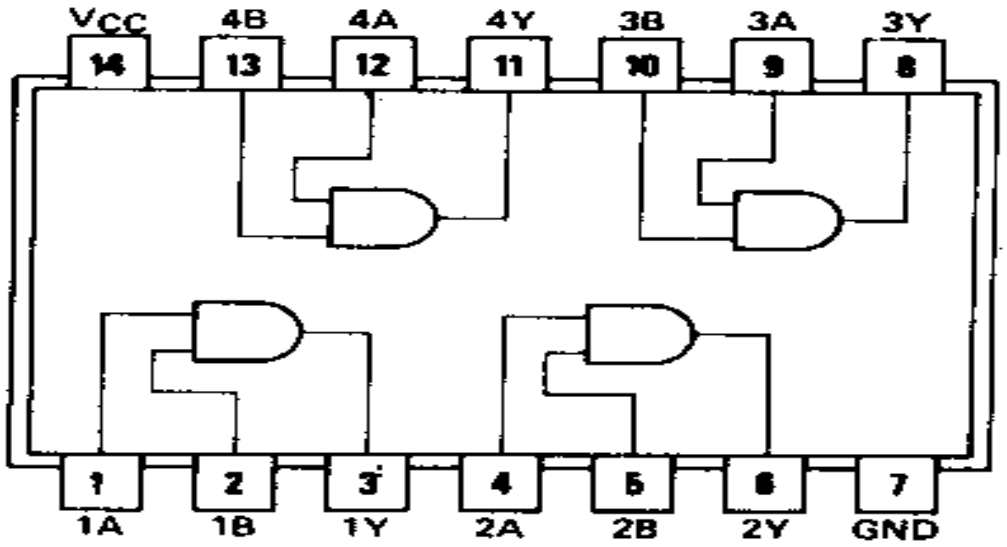
4. 積體電路(integrated circuit，簡稱 IC)：

(1) 是指將多種電子元件的電路集成在晶片上的一種高級電子元件。

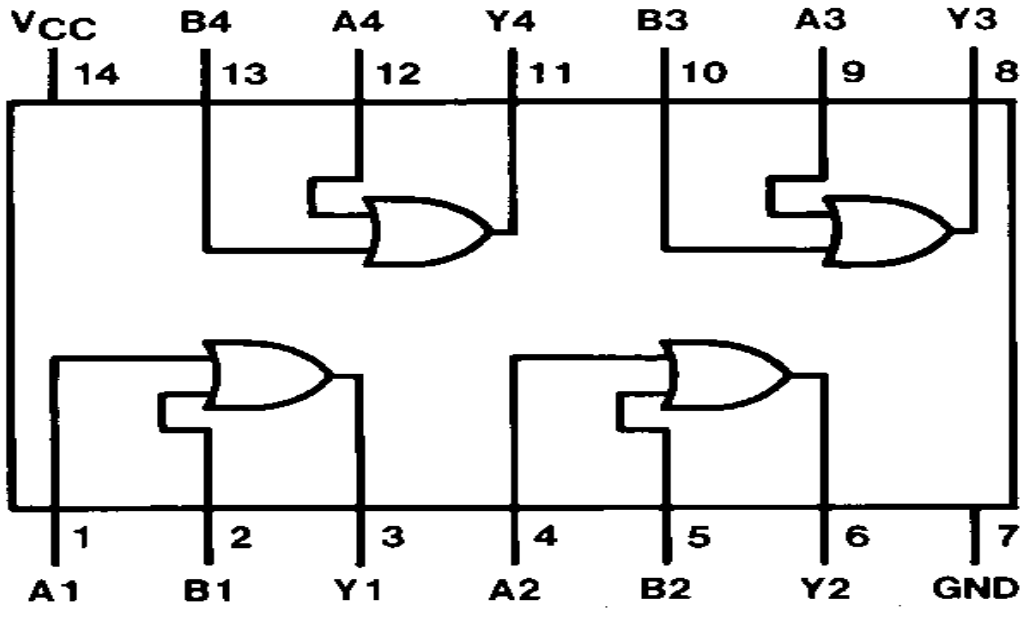
(2) 接腳：以 IC 的缺口朝上為基準，左上開始為 1 號接腳，逆時針數到右上

※基本上會用到的 IC 為 1 個 7408、1 個 7432、1 個 7447、2 個 74283

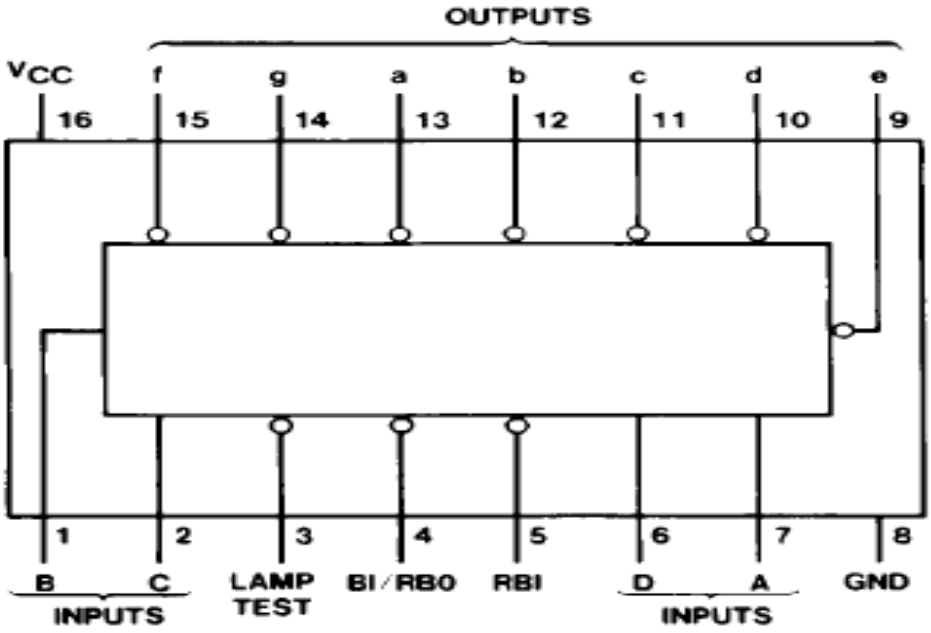
7408：2 輸入 AND 邏輯閘



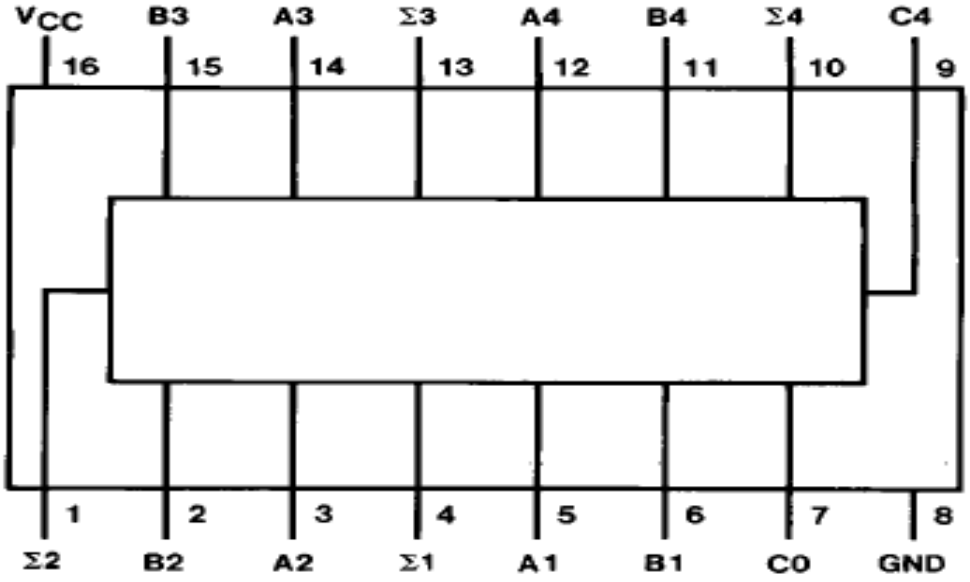
7432：2 輸入 OR 邏輯閘



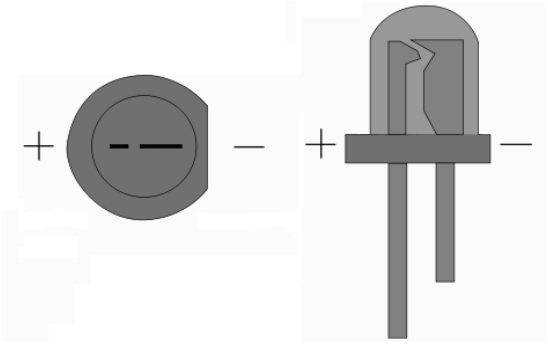
7447：BCD code 七段顯示器解碼器



74283：4-bit 二進位全加器



5. 發光二極體(Light-Emitting Diode，簡稱 LED)：

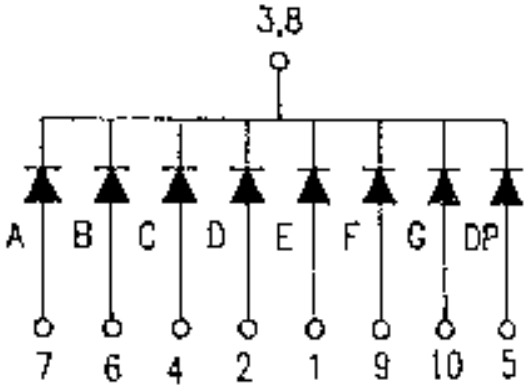


※這次會用到 LED 來代表輸出的十位數數字

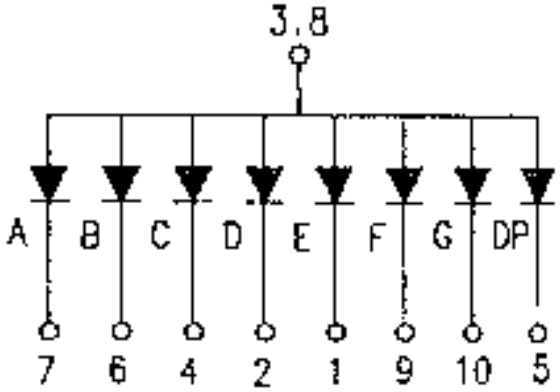
6. 七段顯示器：藉由七個發光二極體以不同組合來顯示數字，有共陽極與共陰極兩種形式，為常用來顯示數字的電子元件 。

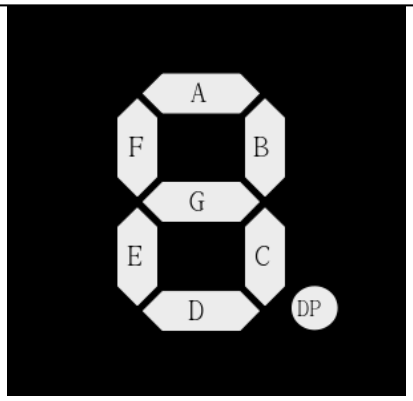
※這次基本上會用到 1 個七段顯示器

共陽極：所有 LED 的陽極接在一起



共陽極：所有 LED 的陽極接在一起

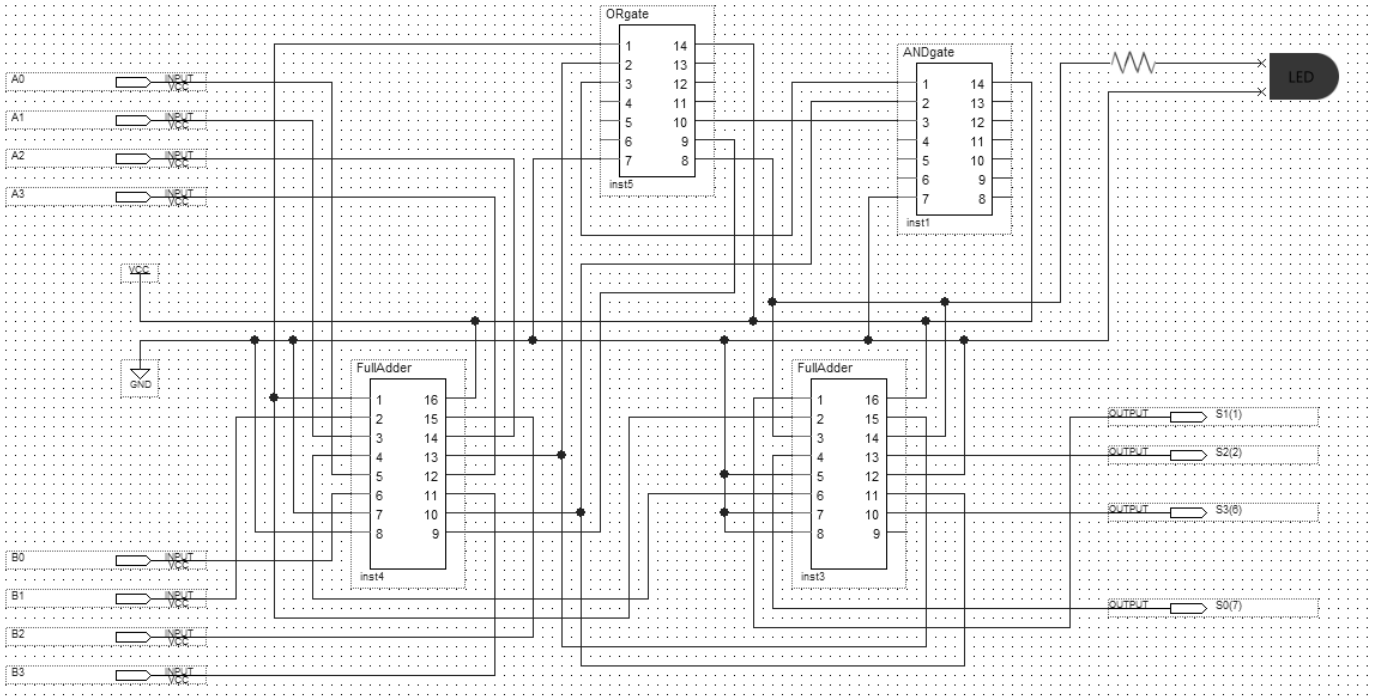




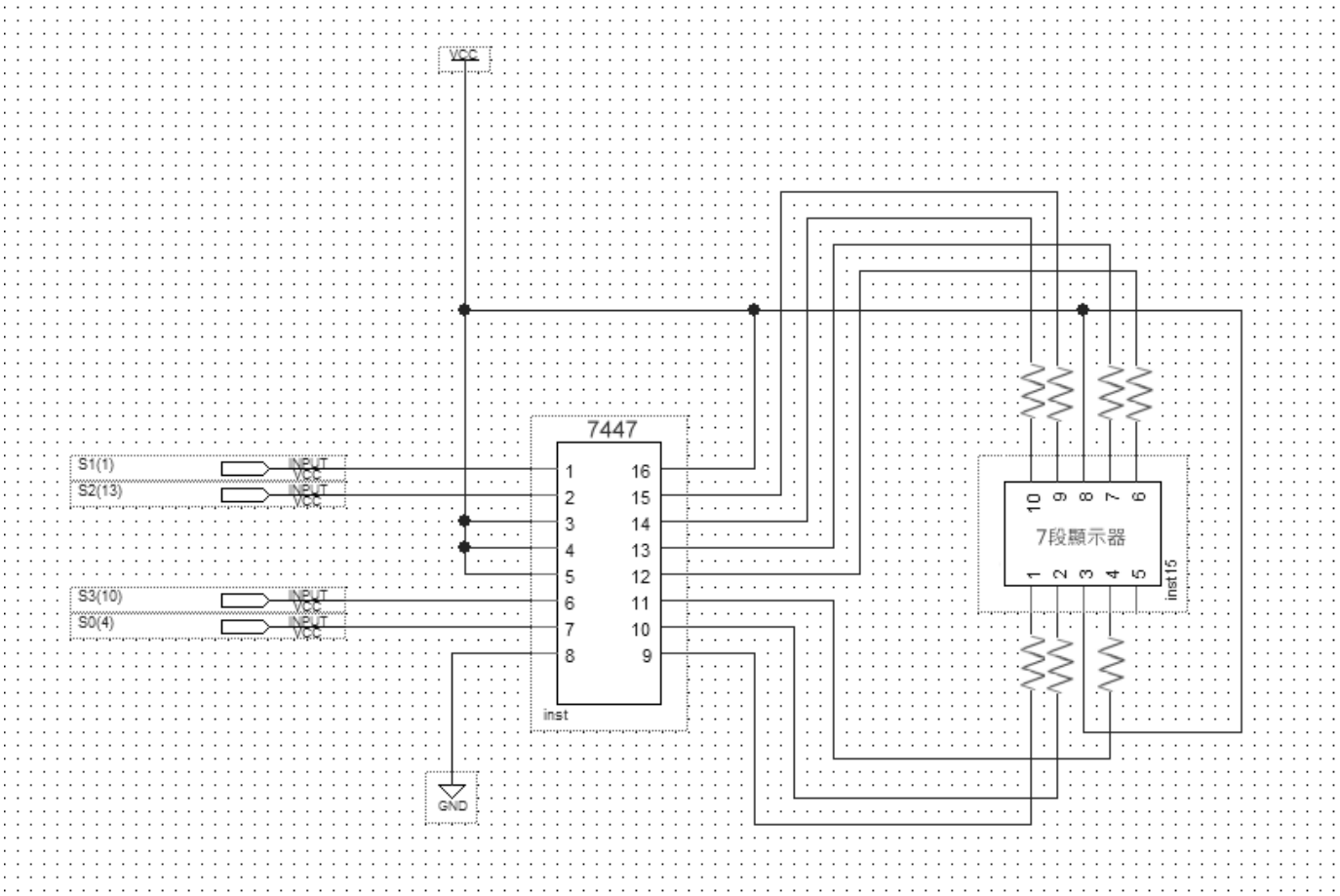
電路原理：

1. 以+5V 代表邏輯信號 1，以接地(0V)代表邏輯信號 0
2. 將兩個二進位數字(範圍從 0~9)相加，再藉由解碼器轉換成十進位數字並在七段顯示器上顯示
3. 由於最大顯示數字為 18，十位數的數字只有 0 或 1，為了避免電路過於複雜，以 LED 燈表示十進位數字(LED 燈亮為 1，不亮為 0

電路圖：電路圖拆成兩部分，每組發兩個麵包板，並藉由四條電線將這兩部分結合



↑ BCD 加法器電路



↑解碼器電路圖

注意事項：

1. 電路圖中交錯的線，有點代表兩條線有接在一起，沒有點代表沒接在一起
2. 電路圖中 A0 為二進位的最低位，A1 為第二位，以此類推，B 輸入也是如此(假設輸入 9，二進位表示為 10012，即 A0=1、A1=0、A2=0、A3=1)
3. 請先確認電路沒接錯後再打開電源，以免發生短路，以避免燒壞零件
4. 如果發生短路，請別馬上去碰 IC(溫度會非常高)
5. 假如聞到燒焦味，請馬上關掉電源

加分題：

1. 將十位數的 LED 燈換成七段顯示器
2. 用 1 個 74283 及 5 個 LED 燈接出 4-bit 加法器，並將 2 進位數字用 LED 燈來表示(燈亮代表 1，不亮代表 0)
3. 請接出 8-bit 加法器，並將 2 進位數字用 LED 燈來表示

Note

電腦網路概論

講師 | 陳紹中

Introduction to Computer Networks　電腦網路概論

一、網際網路歷史

網際網路（Internet）為目前規模最大的電腦網路，其前身為 1970 年代由美國國防部所開發的軍事導向網路阿帕網（ARPANet）。

1980 年，ARPANet 將其核心協定（Protocol）由 NCP 改變為 TCP/IP，成為當今網際網路的核心，ARPANet 採用的 RFC（Request for Comments），亦是網際網路一直以來所使用的標準。

二、電腦網路分類（規模）

1. WAN（Wide Area Network）
   廣域網路，是指一個在很大地理範圍裡，由許多小型的網路所共同組成的大規模網路。網際網路就是一個最為著名，規模最大的廣域網路。
2. LAN（Local Area Network）
   區域網路，是指覆蓋局部地理範圍（家庭、辦公室等）的小型網路。

三、全球資訊網（World Wide Web）

全球資訊網，經常簡稱為 WWW，也是許多網址的開頭，是由許多互相鏈接的 HTML 網頁所組成的龐大資訊系統，藉由網際網路來鏈接並存取（Access）這些資源。值得特別注意的是，全球資訊網經常被當作是網際網路的同義詞，事實上這是一個嚴重的誤解，全球資訊網僅是依賴網際網路來運作的其中一項服務。

四、超文本傳輸協定（Hyper Text Transfer Protocol）

HTTP 是網際網路上應用最廣泛的一種通訊協定，所有 WWW 上的文件都必須遵守這個標準。HTTP 的用途是提供一組發布與接收 HTML 頁面的方法。

五、檔案傳輸協定（File Transfer Protocol）

FTP 是在電腦網路中傳輸檔案的一套標準協定。FTP 能傳送與接收任何類型的文件而不需要另外處理，經常被使用在傳輸大量、或較大的檔案。使用者可以稱過圖形介面（GUI）的 FTP 客戶端（Client）軟體來輕易操作 FTP 通訊，此類軟體的使用方法一般與在本機（Local host）搬移檔案大同小異。

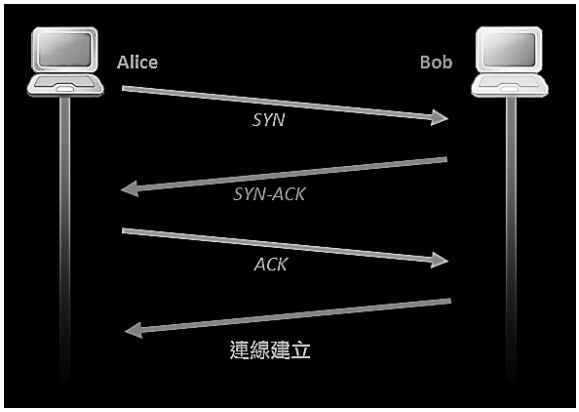
六、Telnet Protocol

Telnet 協定提供使用者由本機登入遠端主機，並且透過網路執行工作，就像直接在遠端主機的前面輸入指令來操作一般。一般的 Telnet 客戶端亦可用來連接到其餘協定的伺服器，手動鍵入請求(Request)來操作該協定。例如，可以 Telnet 客戶端連接到一台 HTTP 伺服器，鍵入「GET /index.html HTTP/1.1」來要求「/index.html」這個頁面的資料。

七、傳輸控制協定（Transmission Control Protocol）

TCP 是一種「連線導向」（Connection-oriented）的、可靠（Reliable）的通訊協定。TCP 使用一個校驗和（Checksum）函數來檢驗所傳輸的數據是否有誤，若傳輸失敗則會繼續嘗試。

TCP 使用的連線建立方式是「三方交握」（Three-way Handshake），詳細可參考如圖所示的連線模型。



八、用戶數據報協定（User Datagram Protocol）

UDP 是一個簡單的「數據導向」（Message-oriented）的、不可靠的通訊協定。由於 UDP 缺乏可靠性以及屬於非連線導向協定，因此 UDP 的應用（程式）必須允許一定量的丟包、出錯和複製（資料重覆）。UDP 以這些缺點換取簡單及高效率的優點，正好與 TCP 成為互補的通訊協定，開發者應視應用程式需要選擇適合的傳輸方式（TCP 或 UDP）。

九、開放式通信系統互聯參考模型（OSI/RM）

OSI 的全名是 Open System Interconnection Reference Model，是由國際標準組織（ISO，International Standard Organization）所提出的概念模型，試圖統一全世界電腦網路的通訊標準。OSI 模型將電腦網路結構分為七層，由第七層開始分別為「應用層」（Application Layer）、「展現層」（Presentation Layer）、「會談層」（Session Layer）、「傳輸層」（Transport Layer）、「網路層」（Network Layer）、「資料連結層」（Data Link Layer）、「實體層」（Physical Layer）。

各層之主要用途可參閱附表。

	Data Unit	No	Layer	Function
Host Layers	Data	7	Application	Network process to application.
		6	Presentation	Data representation, encryption, and decryption.
		5	Session	Interhost Communication.
	Segments	4	Transport	End-to-end connections and reliability. Flow Control.
Media Layers	Packet	3	Network	Path determination, and logical addressing.
	Frame	2	Data Link	Physical addressing.
	Bit	1	Physical	Media, signal, and binary transmission.

表格資料來源：http://en.wikipedia.org/wiki/OSI_model

十、路由器（Router）

路由器在坊間一般被稱為「IP 分享器」，是一種可以將封包通過網路傳到目的地的網路設備，因此路由器通常位於兩個或多個網路的交匯處。路由工作是在 OSI 的第三層（網路層）完成。路由器透過內部的路由表決定封包的流向，路由表裡記載的是 ARP 對應表（即 IP/MAC 位址對應表）與其網路的路由路徑。

十一、集線器（Hub）

集線器是將多條乙太網路（Ethernet）連線集合在同一段物理介質下的網路設備，運作在 OSI 的第一層（實體層），集線器可視為是將多條網路線接成一條的中繼器。

十二、網路協定位址（Internet Protocol Address）

IP 位址是一種在網路中為主機（Host）編址（Addressing）的方式。常見的 IP 位址可分為 IPv4 與 IPv6 兩大類。

IPv4 位址由 32 位元的整數組成，為方便使用，常表示為四組整數的形式，如 140.122.185.138，每組數字均為 0~255 的 10 進制數。在整個網際網路中，每個 IPv4 都是唯一的。以 IPv4 的方式為全球電腦標址，約可有 42 億台電腦同時上線，但隨著資訊科技發展，IPv4 的編址也即將用盡，因此提出了新一代的 IPv6 標準。

十三、介質訪問控制地址（Media Access Control Address）

MAC 提供定址與媒體存取的控制方式，使得不同設備或網路上的節點可以在多點的網路上通訊而不會互相衝突。這個特性在區域網路（LAN）中格外重要，因為不同區域網路中主機的 IP 可能相同，但 MAC 為網路設備出廠時就已決定，為世界獨一無二的地址，因此可藉由 MAC 來辨認兩台主機的異同。

十四、連接埠（Port）

在同一台主機上的不同服務（Service）會對應到不用的連接埠，使得客戶端連線時可以透過不同的連接埠來決定要連線到哪一個服務，例如輸入網址「http://example.org:80」與網址「http://example.org:8080」僅管連接到同一台主機，但因為連接埠不同，所以 HTTP 客戶端所連接上的是兩個截然不同的服務。

連接埠的埠號通常介於 1~65535 之間，其中作業系統通常會保留 1~1024 作為特權連接埠使用。在客戶端建立連線時也會自動選擇一個較高號的連接埠與對方主機連線，可藉由指令「netstat -a」來了解目前的通訊狀況。

在一個伺服軟體上線服務之後，會監聽（Listen）其所設定的特定連接埠，通常特定的通訊協定會有標準的連接埠號，常見的連接埠可以見附表所示。

連接埠號	服務
21/TCP	FTP
22/TCP	SSH
23/TCP	Telnet
25/TCP	SMTP
80/TCP	HTTP
110/TCP	POP3
443/TCP	HTTPS
8080/TCP	HTTP Proxy

十五、網路線製作

1. 材料：Cat-5e 網路線一段、壓線鉗、兩個 RJ45 空接頭。
2. 步驟：
   (a) 將網路線外皮去掉，露出八條不同顏色的細線，分開並去掉不必要的包覆材料
   (b) 將八條線以「橙白、橙、綠白、藍、藍白、綠、棕白、棕」由左至右排列
   (c) 排列固定之後將線頭剪齊，將接頭凸起面朝下
   (d) 將線插入空接頭，確認每條線都插到底
   (e) 以壓線鉗壓緊空接頭，確認線不會脫落

Note

網路服務應用

講師 | 陳紹中

一、Web 2.0

Web 2.0 的應用代表目前全球資訊網的一種轉變，從若干單方向的資訊網站一個為終端使用者（End User）提供網路應用的服務平台，這種概念的支持者期望 Web 2.0 所提供的服務最終可以取代目前的桌面應用程式（Desktop Application）。

其實 Web 2.0 並不是一個新興技術標準，而且利用各種技術架構，建立一個雙向的分享平台，將資訊內容的提供者（Provider）由站方轉變為每位使用者的參與（Participation），產生個人化（Personalization）的內容，藉由分享（Sharing）的概念構成整個 Web 2.0 的分享社群（Community）

二、網路日誌（Blog）

Blog 是 Web Log 的簡稱，代表的意義是「網路上的日誌」，經常被音譯為「部落格」。Blog 通常是一種由個人管理的網站，在上面不定期張貼文章、影音來記錄個人生活或是一些筆記。

許多 Blog 的主題都是個人生活記錄，也有一些特定主題的 Blog，例如技術筆記或是新聞、藝術等等。

三、微網誌（Micro-Blog）

Micro-Blog 是一種允許使用者即時更新簡短文字（140 字左右），並可以公開發佈的部落格形式。微網誌的特色是不需像傳統 Blog 一般長篇大論，而利用簡短的文件記錄一件事、或是發表一則個人看法的言論。

除了文字之外，亦可以發表連結、圖片、影音等等，使得微網誌的應用層面愈來愈廣。微網誌也因為其簡單易用性而被大眾使用者廣為接受。

四、社交網路服務（Social Network Service）

SNS 主要作用是一群擁有相同興趣與活動的人一同建立的網路社群，這類社群為用戶提供各種聯繫與交流，如電子郵件、即時通訊（IM，Instant Message）服務等等。

五、常見 Web 服務列表

類型	名稱	網址
Blog	Blogger	http://blogger.com/
	Wordpress	http://wordpress.com/
SNS	Facebook	http://facebook.com/
Micro-Blog	Plurk	http://plurk.com/
	Twitter	http://twitter.com/
檔案分享	Miroko	http://miroko.tw/
	Dropbox	http://dropbox.com/
線上購物	Amazon	http://amazon.com/
	PChome 線上購物	http://shopping.pchome.com.tw/
	Yahoo! 奇摩拍賣	http://tw.bid.yahoo.com/
	露天拍賣	http://www.ruten.com.tw/
搜尋引擎	Google	http://google.com/
	Bing	http://bing.com/
辦公軟體	Google Docs	http://docs.google.com/
生活工具	Google Maps	http://maps.google.com/

Note

資訊安全概論與實做

講師 | 陳紹中

Introduction to Information Security

資訊安全概論與實做

2010 NTNU CSIE Summer Camp | S.C. Chen

一、資訊安全

隨著資訊科技的日漸進步，各種資訊系統如雨後春筍般不斷出現，對於個人或團體組織來說，「資訊」就是一種資產。新一代的資訊系統改變我們處理資訊的方式，但同時也帶來一個新的議題，就是「資訊安全」。

資訊安全的種類一般可概分為以下三大類：

- 硬體安全：硬體環境控制與人為管理等
- 軟體安全：資料安全、程式安全、通訊安全
- 個人防護：人身安全、個人隱私權、秘密通訊安全

然後，影響資訊安全的因素也相當多，大部分為：

- 未經授權者利用非法手段，竊取或變更資料與設定
- 合法的電腦使用者利用職務之便蓄意竊取或破壞資料
- 資訊傳輸過程中途遭到截取與竄改
- 惡意軟體的感染與破壞

二、電腦犯罪

隨著資訊的傳遞愈趨方便與快速，電腦犯罪的年齡層與知識需求也愈來愈低，所謂電腦犯罪指的就是「利用電腦系統來從事未經授權的非法行為」。

三、電腦駭客

駭客一詞為原文「Hacker」音譯，原意是指「對電腦科學具有高度理解與能力的人」，但後來遭到社會大眾與媒體的誤用，因而產生負面的解釋。

因此在資訊安全領域為了定義出相關人員的角色，以「帽子顏色」衍生出「白帽駭客」、「黑帽駭客」與「腳本小子」等等（在許多電影或小說裡，穿戴白色帽子者通常是正派角色）。

白帽駭客即是資訊安全從業人員，這類人士擁有廣博的資訊安全知識與技術，利用所學協助自己或他人保護資訊系統的安全，也經常從事「滲透測試」。相對於白帽，黑帽駭客則是指擁有此類技術，但利用於攻擊他人的犯罪者，又稱為Cracker。此外，腳本小子原文為「Script Kiddie」，意指那些剛入門資訊安全領域，擁有平庸的知識與技術，只利用現成工具與他人發現的漏洞來實施惡意攻擊的人。

四、滲透測試（Penetration Test）

滲透測試為許多資訊安全從業人員（白帽駭客）的主要工作，以模擬入侵者攻擊的手段來檢測一個資訊系統（電腦系統或人員體系）的安全性。依照測試要求不同，又可分為「黑箱測試」、「白箱測試」與「灰箱測試」等。

滲透測試一般會檢測「資訊洩露」與「系統弱點」等。資訊洩露指的是相關系統或人員在疏忽之下洩露機敏資訊給未經授權的第三者;而系統弱點指的是相關資訊系統因為開發的疏失，而存在漏洞，造成未經授權人士（黑帽駭客或腳本小子等）可經由該弱點攻擊系統或許存取機敏資料。

由於測試範圍經常包含商業機密與敏感資訊，滲透測試的評估與執行需要由資深、有良好信譽的團隊所參與，坊間有若干公司配合專業的白帽駭客團隊提供相關服務。

五、釣魚攻擊（Phishing）

釣魚攻擊之原文「Phishing」與「Fishing」發音相同，是指在資訊通訊過程中偽造合法人士的身份來騙取使用者的機敏資訊，如帳號密碼、信用卡號、身份證帳號（或社會安全碼）等等的攻擊手段。

常見的釣魚攻擊為「偽造登入頁面」與「偽造官方信件」，兩者皆是利用一些仿冒的手段來騙取使用者上鉤。因此使用這些有關機敏資訊的服務時必須要再三檢查資訊系統的真偽性。

六、惡意軟體

惡意軟體一般有如下的幾個特性，符合其中之一者即可稱為惡意軟體：

- 騙取使用者安裝、或強制安裝，並且抵制移除
- 強制修改使用者設定，如瀏覽器首頁、作業系統設定等
- 莫名其妙彈出廣告、或干擾使用者操作系統
- 侵犯或向外傳送使用者儲存在系統裡的機敏資料
- 停用防毒軟體或系統管理程式
- 打開「後門」供未經授權的攻擊者存取電腦

這類惡意軟體通常會配合「釣魚攻擊」來欺騙使用者上鉤安裝，或配合瀏覽器漏洞強行在電腦裡安裝。也有不少惡意軟體偽裝成防毒軟體的案例。

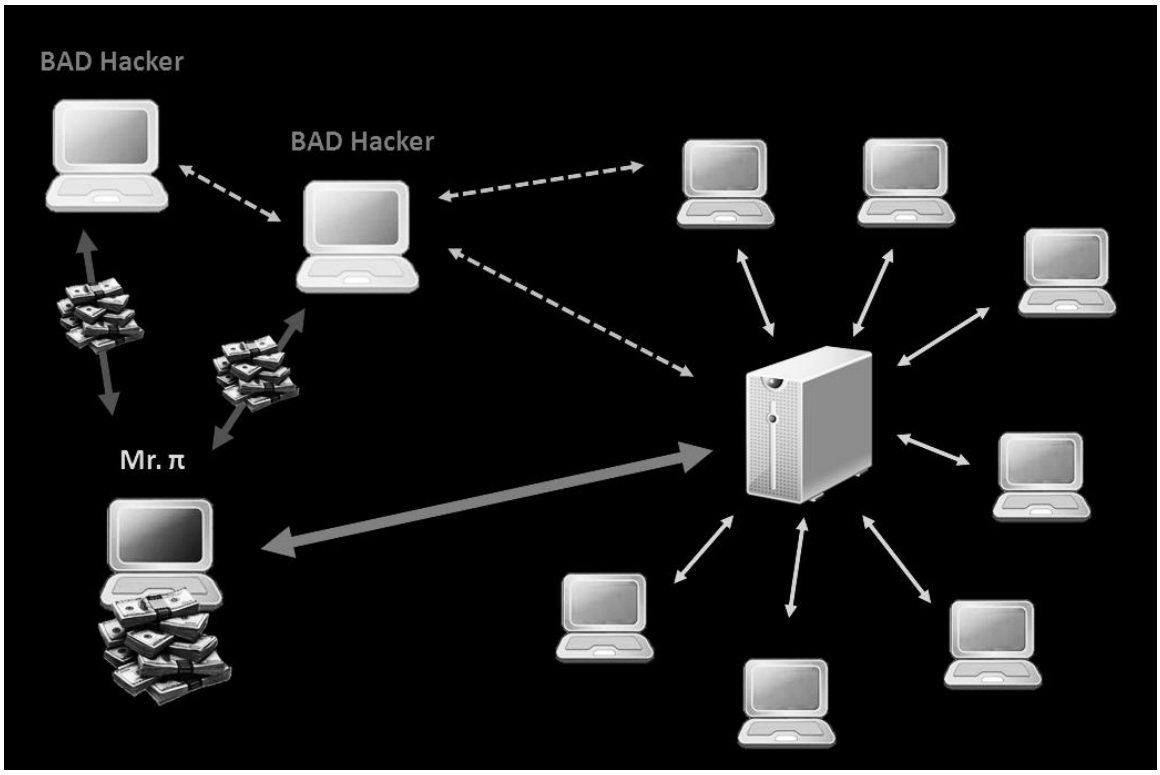
要防範惡意軟體的首要條件即是不隨便下載安裝軟體，尤其盜版與非法軟體最常與惡意軟體一同包裹。此外，定期更新作業系統與防毒軟體也是防範的重要手段。

七、殭屍網路（Botnet）

殭屍網路為「惡意軟體」的一個擴展應用，意指未行授權的攻擊者（黑帽駭客）利用惡意軟體控制一定數量的受害者電腦，收集機敏資訊或對其它伺服器發動攻擊的非法手段。

這些淪陷的受害者電腦即駭客們黑話裡的「肉雞」，通常是大規模的分散在不同的地理位置，透過攻擊者編寫的程式與中央伺服器（C&C Server）連接，接受攻擊者所下的指令來發動攻擊或回傳資訊。

附圖即為一個典型的 Botnet 模型。



八、加密演算法

資訊系統為保障資訊的傳輸安全，廣泛使用加密演算法將資訊加密後傳送。依照加解密的方法不同，常見的加密演算法可分為兩大類：

- 單向加密：資訊僅能由「明文」被加密為「密文」，無法還原，又被稱為是雜湊式加密演算法。
- 雙向加密：資訊經由對應的「金鑰」可由密文被解譯成明文，為一般認知裡的加密。

然而「雙向加密」又可分為：

- 對稱式加密：加解密時使用同一把金鑰
- 非對稱式加密：加密時使用「公鑰」，解密時使用「私鑰」。公鑰可公開給任何人知道，私鑰為自行持有

常見的加密演算法有下列數種：

- 單向加密：MD4、MD5、SHA-1
- 雙向加密：
    - 對稱式加密：變位字謎、Caesar Cipher、(3)DES、AES、Blowfish
    - 非對稱式加密：RSA、DSA

以攻擊者的角度看來，要防制加密演算法就必須發展破密的手段，常用的破譯方法有以下幾種：

- 暴力猜解法
- 字典猜解法
- 頻率分析法

其中前兩項適用於保護密碼用的單向加密演算，而「頻率分析法」為透過分析一個語言中字母或字詞的出現頻率（如：英文中以字母 E 的出現頻率最高），來猜解密文與明文之間的對應關係，破譯 Caesar Cipher 時可以得到不錯的效果。

九、連接埠掃描攻擊

連接埠掃描攻擊即 Port Scanning，指的是利用工具掃描目標電腦所開啟（監聽）的連接埠，進而判斷目標伺服器所提供的服務。利如使用工具掃描目標電腦有監聽 TCP 連接埠 21 與 80，即猜測可能是一台提供 FTP 存取的 Web 伺服器。

進階的 Port Scanning 工具（如常見的 Nmap）除了典型的連接埠掃描之外，提供了進階的掃描方式來規避 IDS（Intrusion Detection System，入侵偵測系統）的檢測；也提供了其它的掃描，包含依照 Fingerprint（特徵）來猜測目標電腦的作業系統等等。

十、常見的 DOS 指令

在 Windows 還沒有發展之前，微軟公司為個人電腦（PC）開發了一套稱為「MS-DOS」的純文字作業系統。在 Windows 出現之後，仍然保留了類似的執行環境，即是「命令提示字元」（Command Prompt）。

這邊介紹常用、常見的 DOS 指令：

- cd：進入一個資料夾，如「cd C:\Windows」
- dir：列出目前資料夾下的檔案
- copy：以「Copy <來源> <目標>」的形式複製檔案，如「copy foo.txt bar.txt」就是複製一份 foo.txt 並改名為 bar.txt
- del：刪除檔案（為 Delete 的縮寫），如「del hello.exe」
- cls：清空目前螢幕（命令提示字元視窗）的資訊
- md：建立一個新的資料夾，如「md myfolder」建立一個名為 myfolder 的資料夾
- telnet：使用 telnet 連線到遠端伺服器，使用方法為「telnet <目標電腦位址> <連接埠>」，如「telnet www.csie.ntnu.edu.tw 80」即是連接到「師大資工系」的 web 伺服器。

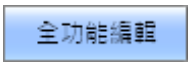
Note

影像處理實作

講師 － 翁仁一

快速修片模式： 

- 右邊的 快速修片面板：提供許多的預設功能可以快速上手。
- 編輯 功能表、調整 功能表、相片 功能表、特效 功能表

全功能編輯模式： 

◆選取一小部分區域

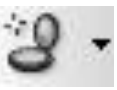
標準選取工具 ： 選取有規則形狀的區域。

套索工具 ： 選取物體的邊緣時相當有用。

魔術棒工具 ： 可選取相似顏色的區域，背景和主體顏色差異很大時會很有用。

◆路徑工具  、文字工具  、變形工具 

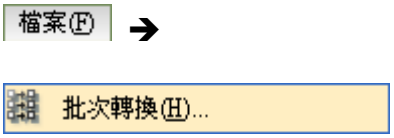
◆編修工具  、繪圖工具  、印章工具  、

仿製工具  、填充工具 

◆百寶箱：提供一堆現成的範本直接點選利用。

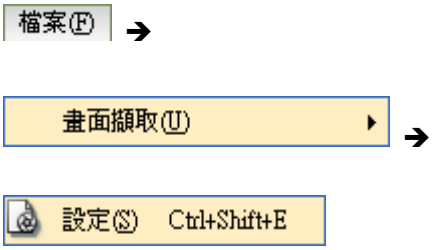
◆其他小工具

●批次轉換：



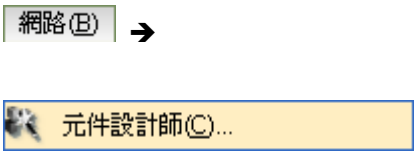
一次轉換許多檔案，可轉換檔案格式和色彩。

● 畫面擷取：



可一次擷取許多畫面，也可以選擇擷取的範圍。

●元件設計師：



提供一些圖示、按鈕等的範本可供修改。

沒有 PhotoImpact 的解決方法

■找免費軟體或是共享軟體

Photoscape	http://photoscape.org/
PhotoFiltre	http://photofiltre.free.fr/
GIMP	http://www.gimp.org/

■一些線上服務

Pixlr	http://pixlr.com/express/	http://pixlr.com/editor/
Splashup	http://www.splashup.com/	
Sumo Paint	http://www.sumopaint.com/app/	
Picnik	http://www.picnik.com/	

Note