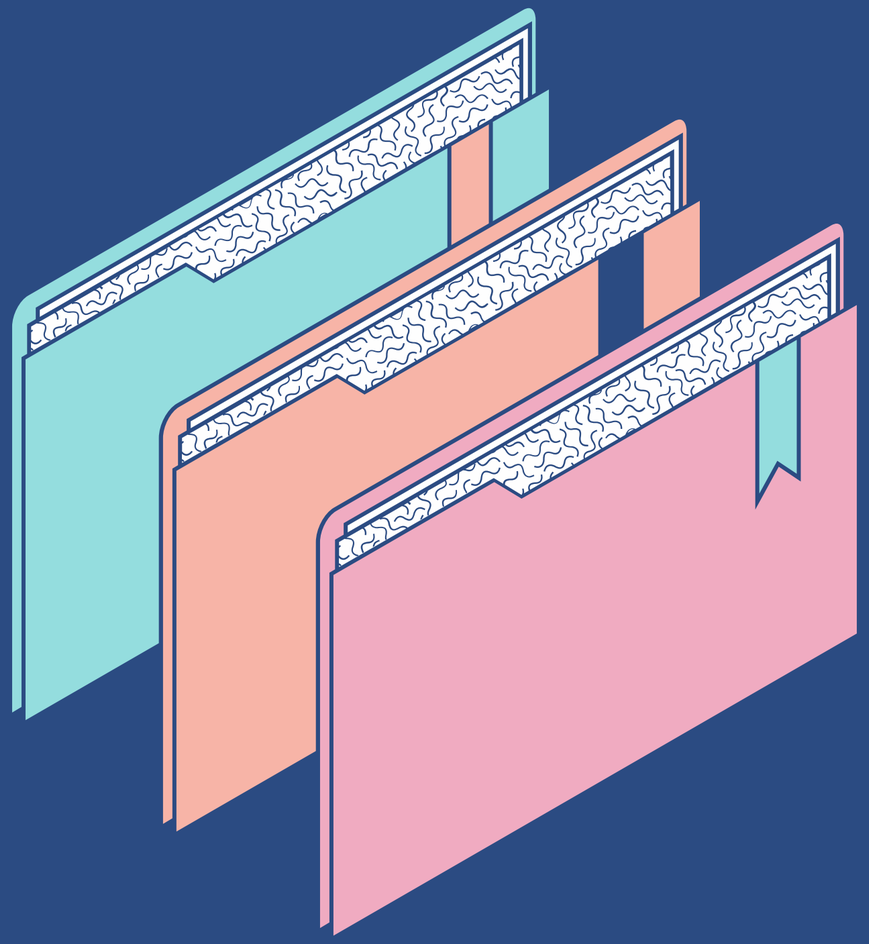
20230715~20230716

多元學習成果

晶片程式設計實作營

新竹女中　周語泠

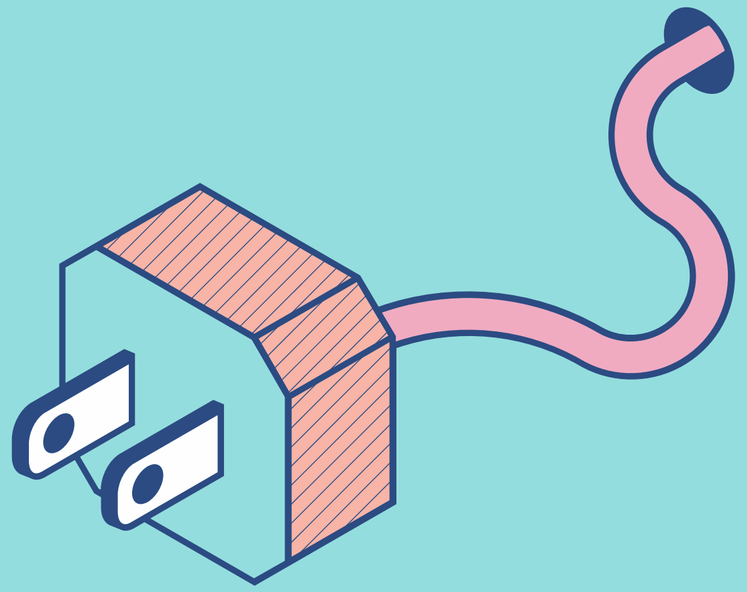


目錄

- 活動介紹
- 參與動機
- 營隊精華　(最印象深刻)
  - 數位邏輯設計實作
  - 硬體描述語言 Verilog
- 心得與反思

活動介紹

在高一的暑假期間我參與了EDC專長探索中心舉辦的晶片設計實作營，雖然跟以往需要住宿的營隊大有不同，但同時更學到了很多關於晶片和程式的專業知識。
晶片設計實作營有基本語法介紹、硬體程式語言教學，甚至還有最後蜂鳴器的實作，為時兩天的營隊也能讓我滿載而歸。



參與動機

我一直以來都對程式、晶片類的領域非常有興趣，但卻一直一竅不通。從來沒有寫過任何程式的我，在同學們互相切磋自己的程式技巧時永遠插不上話。且在這高速運轉的世界中，程式撰寫等高科技實力幾乎已成必備，為了至少對程式有初步認識和獲得真正的技能而選擇參加此次營隊。

參與營隊前給自己的目標

- 習得一種程式撰寫方式
- 認識數學邏輯閘和程式的關聯
- 了解晶片為何

營隊精華 之

數位邏輯設計實作

從應用需求到實際邏輯電路

邏輯設計流程

1 STEP

根據實際應用情形訂定所需規格

2 STEP

將所需規格列成真值表

3 STEP

利用最小項展開，將真值表轉換成布林表達式

4 STEP

將布林表達式用卡諾圖化為最簡型式

5 STEP

根據化減後的布林表達式用邏輯閘組合成實際邏輯電路

完成！

數位邏輯設計實作

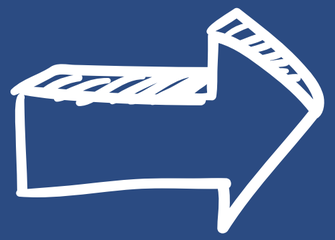
以警報器為例

要讓警報器響，警報器開關必須在開啟的狀態。
警報器開啟後，如果門沒關，或是晚上六點之後窗戶沒關，警報器就會響。

A B

理解為 => 只有在 警報器開關開啟 且 門沒關 或 (晚上六點 後 且 窗戶沒關)，警報器才會響。

C D Y



列出真值表

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	1	1

數位邏輯設計實作

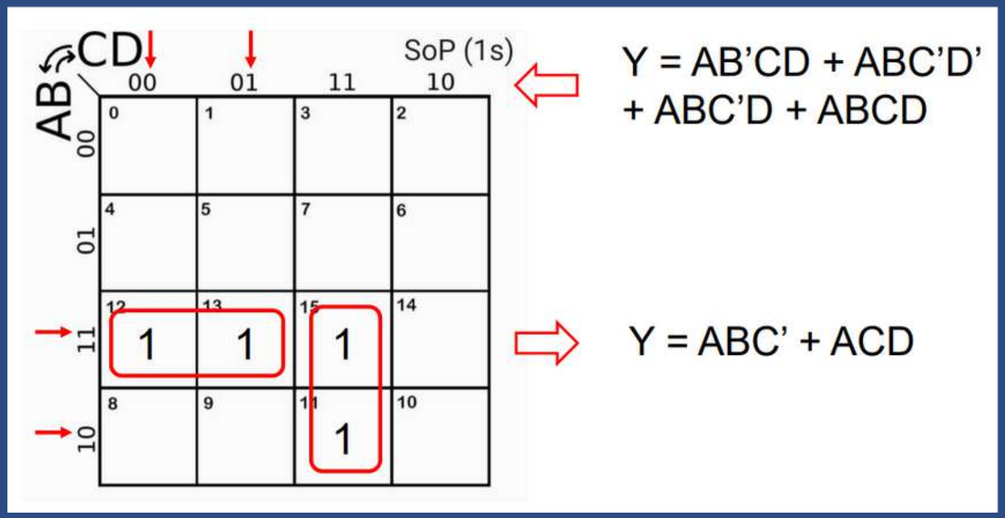
最小項展開

A	B	C	D	Y
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	1	1

卡諾圖化簡

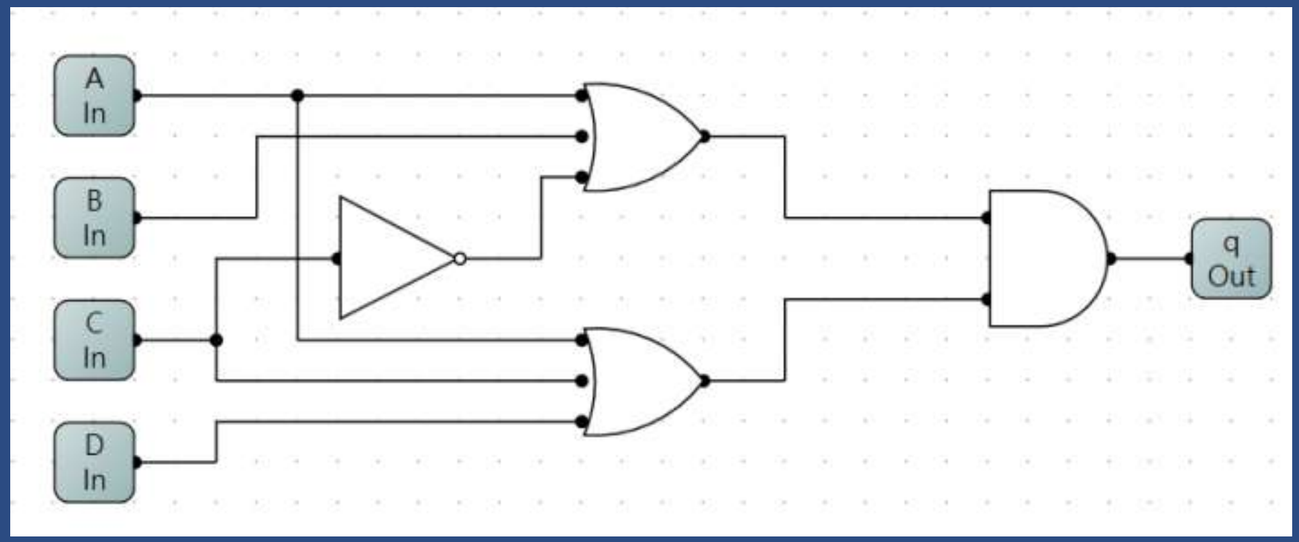
轉換成布林表達式

Y = AB‘CD + ABC’D’ + ABC’D + ABCD



數位邏輯設計實作

實際邏輯電路

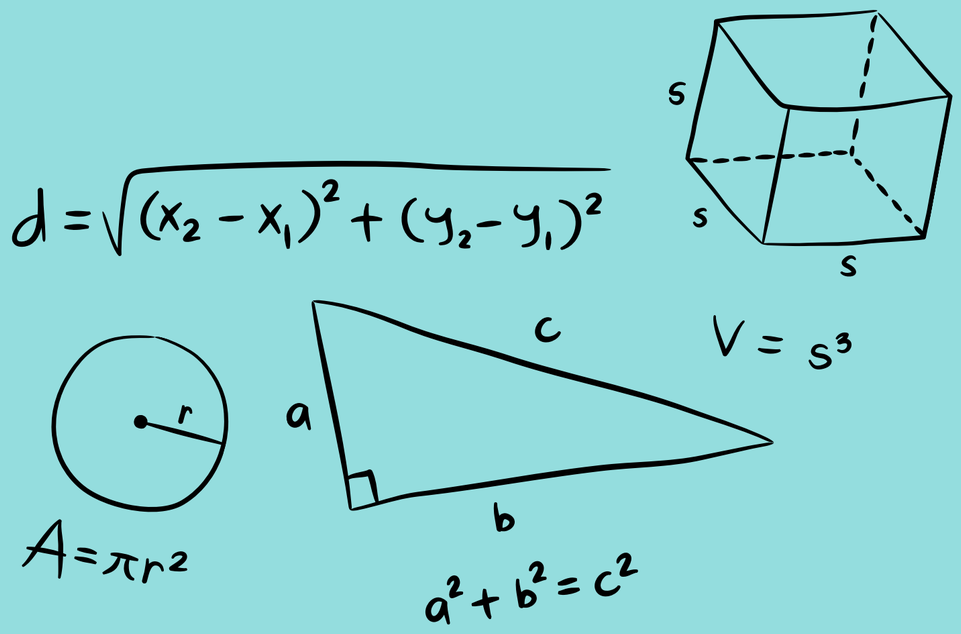


完成！

這個環節利用日常生活中出現的例子來舉例說明程式設計的最雛形。來自台大機械系的教授將所有流程做成一套SOP讓初學者們更好上手，也用清晰的數學邏輯思維一步步推導出結果。將最後結果用剛學習到的邏輯閘展現，讓我對製作程式的過程豁然開朗，愈發對程式設計的興趣。

營隊精華 之

硬體描述語言 Verilog



Write your solution here

[Load a previous submission] Load

```
1  `default_nettype none
2  module top_module(
3      input a,
4      input b,
5      input c,
6      input d,
7      output out,
8      output out_n   );
9      wire w1, w2;
10     assign w1 = a&b;
11     assign w2 = c&d;
12     assign out = w1|w2;
13     assign out_n = ~out;
14 endmodule
15
```

Submit | Submit (new window)

硬體描述語言Verilog

資料流模型的敘述

- 設計重點：說明資料如何在電路中的傳送過程
- assign
  - 等號左式只能是wire，右式可以是 wire 或 reg
  - 用於描述組合邏輯電路
  - 必須避免使用迴路式的寫法：assign a = b + a;
- 電路範例
  - X = A · C · D’ + B · C’ + B · D + C · D
  - Y = A’ + B + C

```
1  module boolean(a, b, c, d, x, y);
2      input a, b, c, d;    //wire
3      output x, y;         //wire
4
5      assign x = (a & c & (!d)) | (b & (!c)) | (b & d) | (c & d);
6      assign y = (!a) | b | c;
7
8  endmodule
```

硬體描述語言Verilog

Verilog 練習

https://hdlbits.01xz.net/wiki/Main_Page

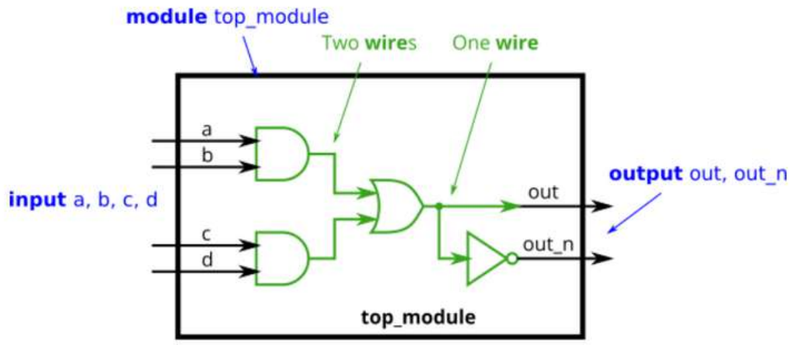
我們利用了這個網址進行多個Verilog 的基礎練習。
主要的練習模式是網頁顯示邏輯閘，再讓我們自己利用那些邏輯閘寫出與它相符合的 Verilog 運算式。

Practice

Implement the following circuit. Create two intermediate wires (named anything you want) to connect the AND and OR gates together. Note that the wire that feeds the NOT gate is really wire `out`, so you do not necessarily need to declare a third wire here. Notice how wires are driven by exactly one source (output of a gate), but can feed multiple inputs.

If you're following the circuit structure in the diagram, you should end up with four assign statements, as there are four signals that need a value assigned.

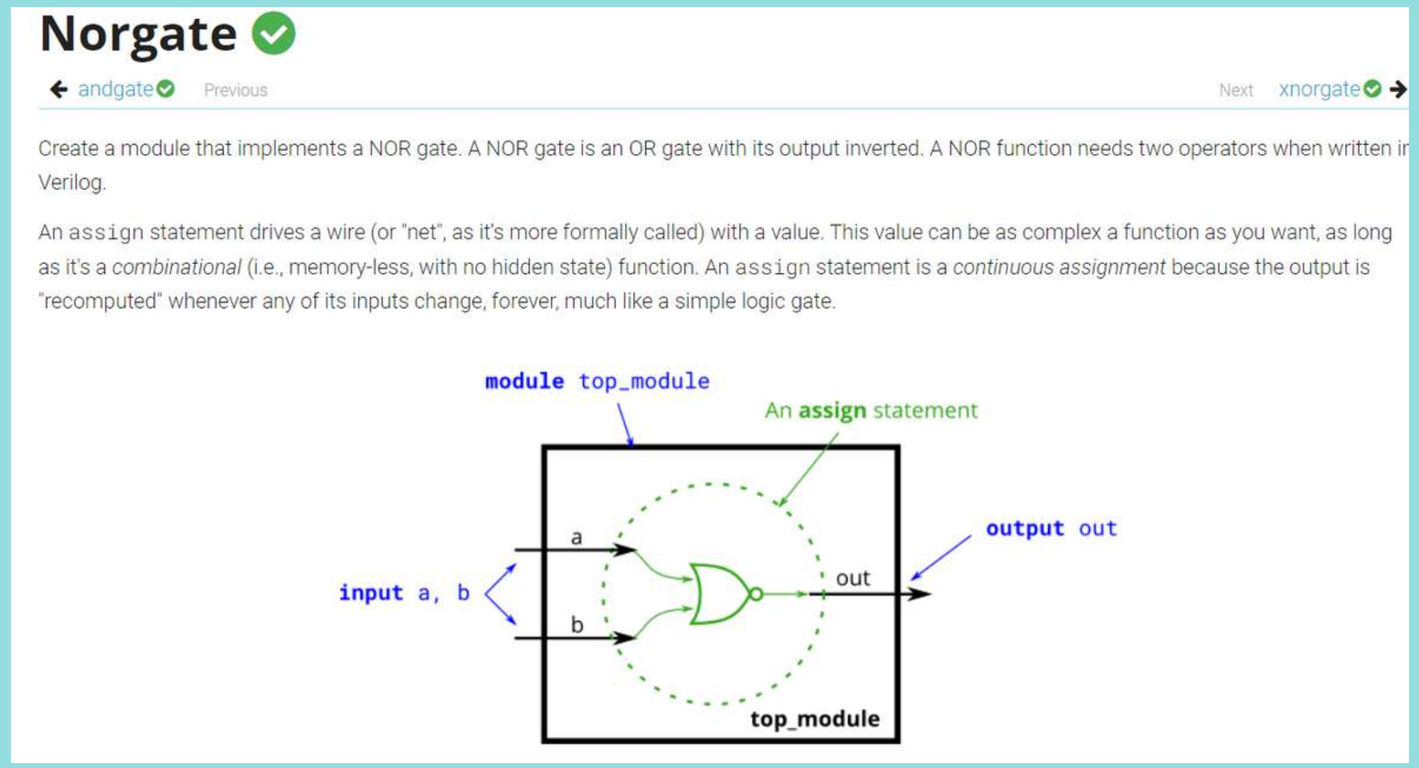
(Yes, it is possible to create a circuit with the same functionality without the intermediate wires.)



附圖為其中一練習題的螢幕截圖

硬體描述語言Verilog

我卡關最久的練習題



這題的 "nor" 和前面練習過的 "and" 不太一樣，加了否定詞。因此造成需要在 "or" 的運算子（|）前加否定運算子（!）。
但我一直卡在把程式寫成：
" assign out= a ! | b ; "
後來才發現它否定的是一整個 " a or b " 而非單純 "a"、"b"，
所以在進行調整後的運算式如下：
" assign out = ! (a | b) "
這樣才能否定一整句 a or b。
或許這在程式高手中根本是不能犯的低級錯誤，但對於第一次接觸程式的我來說真的有難度。好在最後也還是靠自己的力量把這題解出來了，也忽然意識到寫程式的樂趣和成就感！

心得與反思

完全沒寫過程式的我其實來到這個營隊之前非常驚慌失措，畢竟現在許多同學都學過程式設計了，可能全班就只有我一位沒有學習過相關技能。但在參與這個暑期活動後，更學習到硬體（七段顯示器）與軟體之間如何相互配合，也終於學會一種程式語言，好像終於和世界接軌了。其實最令我興奮的果然還是目睹在學校所學的「數學邏輯運算」和人們一直在說的「程式設計」結合起來的那瞬間，讓我終於體會到原來我們這10年來一直在學習的那些邏輯思維是這樣應用在現代科技上的，覺得非常新奇、受益良多！(最痛恨的果然是打完運算式最後都忘記加 " ; " 真的很考驗細心度，少一個；一整個都要重來了。)

唯一可惜的是蜂鳴器實作練習時我的電腦和教授供應的軟體不相容而導致我沒辦法一同進行實作，只能在旁觀摩其他同學操作。但即使如此，在教授用淺顯易懂的方式教學後，就算無法操作也心領神會。

為時兩天的晶片設計營也終於告一個段落，在營隊結束後我期許自己能持續鑽研程式設計，將現在我唯一學會的程式語言練得爐火純青，也許未來有相似邏輯問題時也能運用這種能力破解他！也希望自己可以繼續完成Verilog 的練習題，將當時無法破解的問題全都弄懂。

專長探索中心
Expertise Discovery Center

結 業 證 書

茲證明

學員　　周語泠

自中華民國 112/07/15 至 112/07/16

參加 晶片程式設計實作營 課程，

修習 12 小時訓練期滿。

此證

中 華 民 國 一 一 二 年 七 月 十 六 日

附圖為參與證明

THANK YOU

新竹女中 周語泠