

NCTU-EE DCS-2021

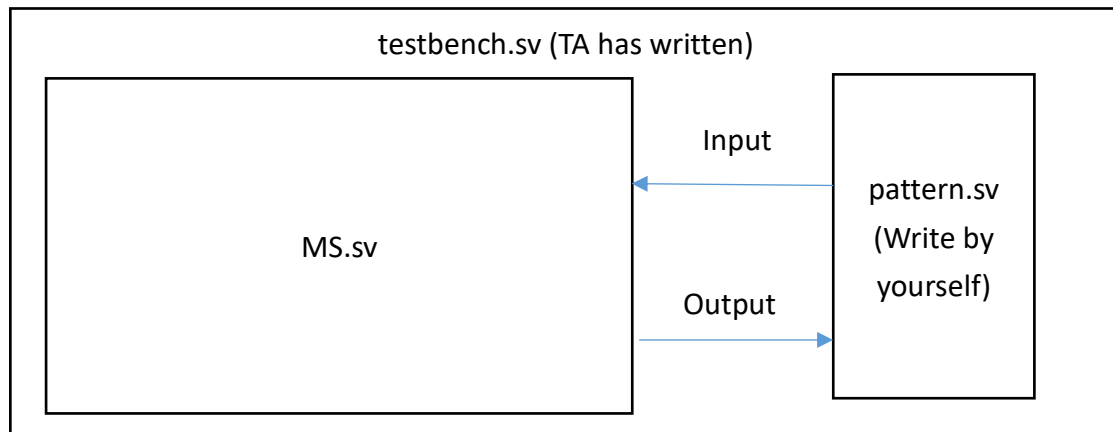
Final Project

Design: MS

資料準備

1. 從 TA 目錄資料夾解壓縮
% tar -xvf ~dcsta01/Final.tar
2. 解壓縮資料夾 OT 包含以下：
 - A. 00_TESTBED/
 - B. 01_RTL/
 - C. 02_SYN/

Block Diagram



設計描述

此次 Final Project 主要是實作走迷宮的演算法 – 廣度優先 Breadth-First Search (BFS)，此次 final 會稍微介紹 BFS 跟此次作業相關規則，而更詳細的演算法過程請各位同學上網搜尋與查詢。

此次作業可以分成三個步驟，**1.接收迷宮資料 2.廣度優先查詢 3.輸出迷宮路徑。**

1. 接收迷宮資料:

一開始 pattern 會輸出 16*16=256 個 1bit 資料進入 Design，資料進入的順序，如下圖所示，由左至右由上而下

EX : (先(0,0)~(15,0) 再輸入(0,1)~(15,1))

輸入進的 256 個數字皆為 1bit，分別 1 代表牆壁，0 代表可走的路，並且迷宮的起始都在 (1,1) 終點都在 (14,14)，詳細可看下圖。

最外圍一定是牆壁，不需考慮是 0 的狀況，而(1,1)~(14,14) 都有可能為牆壁。

2. 廣度優先查詢:

依照以下三種規則

1. 從起點開始，遇到岔路時會同時向岔路擴展開來。
2. 如果岔路走到死路，則不會再擴展
3. 直到擴展到終點為止
4. 若找不到路徑、終點，代表尋找失敗、地圖無效，則需將 `maze_not_valid` 輸出 1。

3. 輸出迷宮路徑

由終點往回走尋找路徑(走剛剛走過的路)

尋找的優先順序為”上” → ”左” → ”下” → ”右”。

最終會找到路徑，以座標軸表示，再輸出路徑，從(14,14)輸出至(1,1)，分別輸出至 `out_x` 與 `out_y`

我們的 `pattern` 有提供一個地圖範例

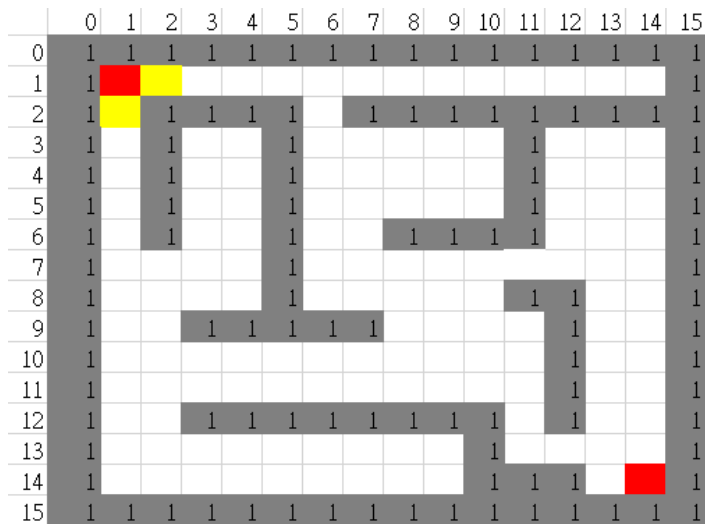
以下舉該地圖作範例，紅色為出入口

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1															1
2	1		1	1	1	1		1	1	1	1	1	1	1	1	1
3	1		1			1						1				1
4	1		1			1						1				1
5	1		1			1						1				1
6	1		1			1			1	1	1	1				1
7	1					1										1
8	1					1						1	1			1
9	1			1	1	1	1	1					1			1
10	1												1			1
11	1												1			1
12	1			1	1	1	1	1	1	1	1		1			1
13	1											1				1
14	1											1	1	1		1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

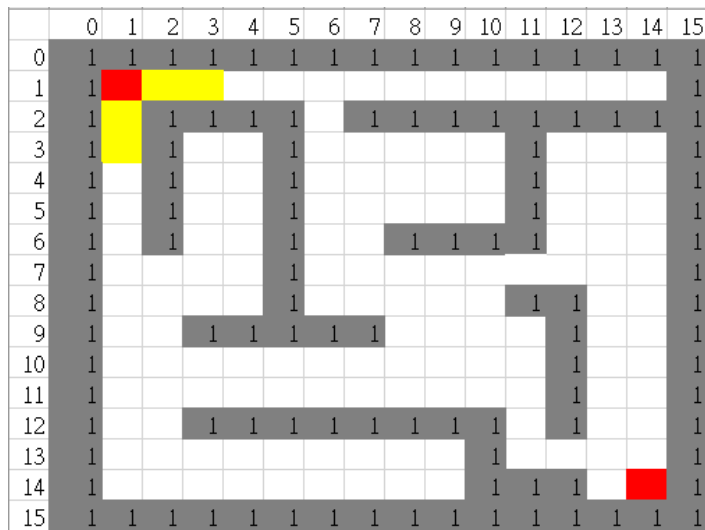
BFS:

以上圖為例，黃色代表搜尋到的區域

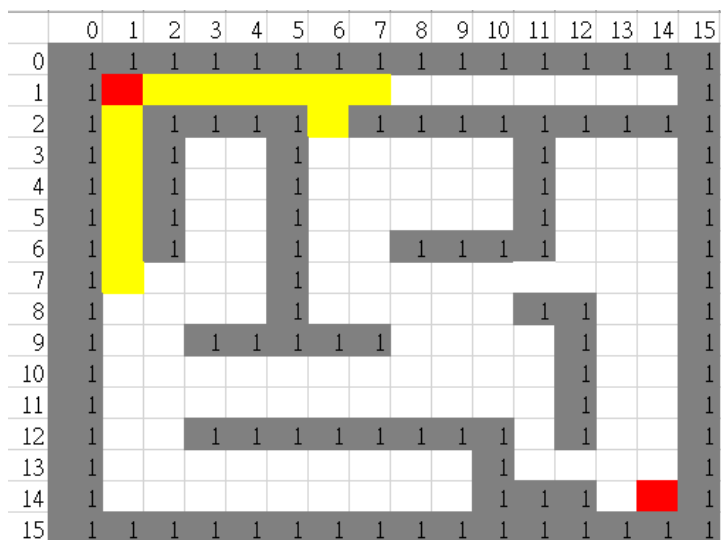
1. 第一步



2. 第二步



4. 第六步，遇到岔路繼續擴散



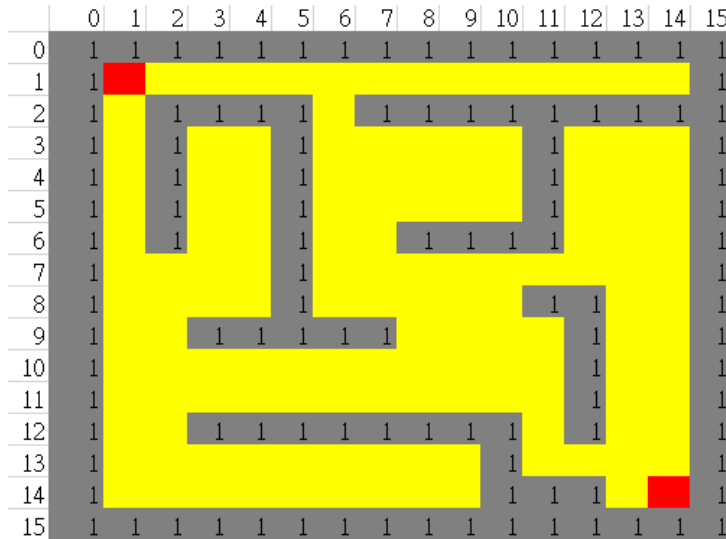
5. 第七步

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

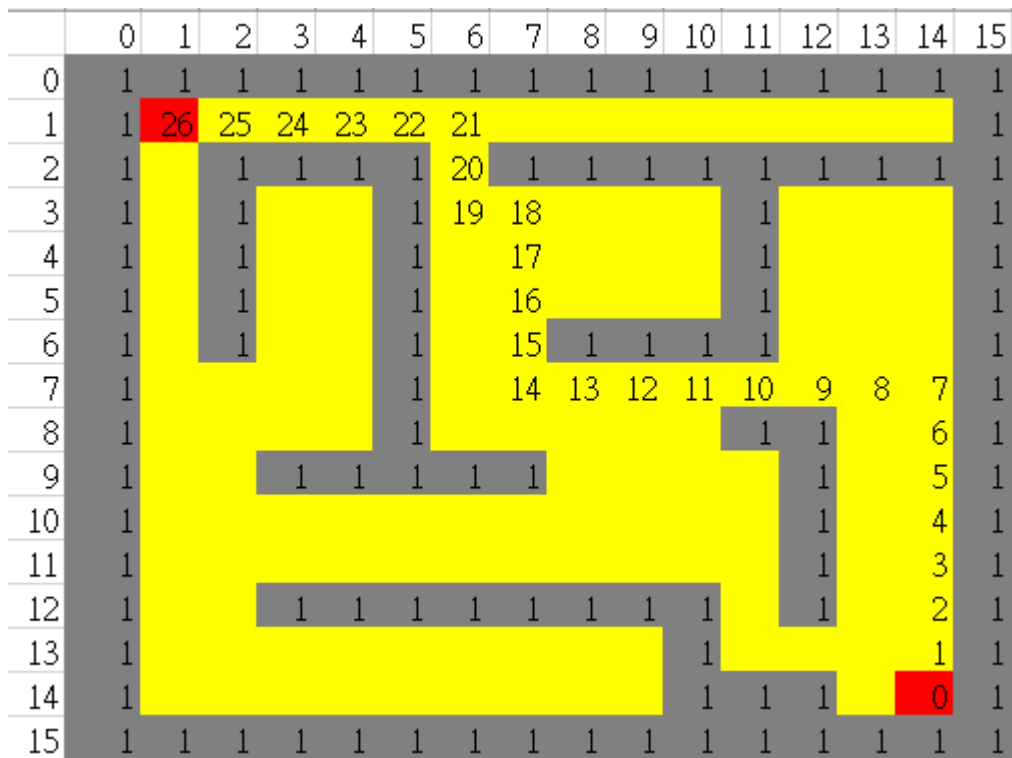
6. 重複這些步驟，遇到死路則不會擴散

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

7. 最後搜尋到終點，以此範例來看，會剛好走過所有地圖，但不一定每個地圖都會走完才找到終點，可能會走到終點後，還有未走過的區域，要自行判斷。



8. 由終點往回走，依照上左下右規則



9. 由5.找到路徑，依照 0~24 吐出路徑值，分別將 X 軸 Y 軸給

out_x,out_y，如下

out_x = [14, 14, 14, 14, 14, 14, 14, 14, 13, 12, 11, 10, 9, 8, 7, 7, 7,
7, 7, 6, 6, 6, 5, 4, 3, 2, 1]

out_y = [14, 13, 12, 11, 10, 9, 8, 7, 7, 7, 7, 7, 7, 7, 6, 5, 4, 3, 3, 2,
1, 1, 1, 1, 1, 1]

動畫可參考下網址，但尋找路徑的規則與這次作業不全相同!

<http://bryukh.com/labyrinth-algorithms/>

此次作業會使用到 Queue 的概念，請同學自行上網查閱相關想法。

Input

Signal name	Number of bit	Description
clk	1 bit	Clock 5ns
rst_n	1 bit	Asynchronous active-low reset
in_valid	1 bit	為 1 時代表給 maze 資料，連續給滿 256 cycle
maze	1 bit	地圖資料，依序給滿 256 個，由上至下由左至右

Output

Signal name	Number of	Description
out_valid	1 bit	為 1 時代表 output out_x,out_y, maze_not_valid 等值，若路徑無效則 1cycle。in_valid 後 3000cycle 內要吐值。
maze_not_valid	1 bit	如果找不到路徑，則 output 1，相反則為 0。路徑無效時，1 維持 1cycle。
out_x	4 bit	依序吐路徑的 X 值，由終點開始。路徑無效時為 0。吐完值後須歸 0。

out_y	4 bit	依序吐路徑的 Y 值，由終點開始。路徑無效時為 0。吐完值後須歸 0。
-------	-------	-------------------------------------

本次提供一組迷宮 pattern，為前幾頁的圖，請自行使用讀檔方式讀進去，並且自行創造新的迷宮 pattern。

Input.txt: 由左至右 由上而下

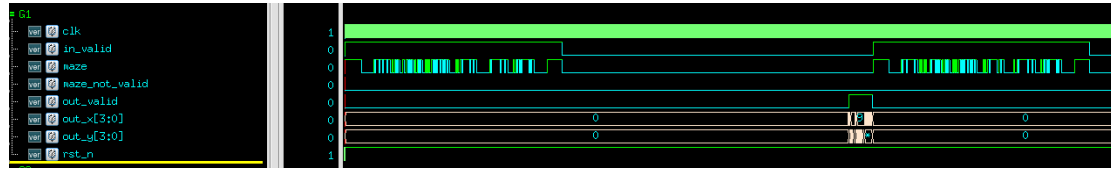
out_x.txt & out_y.txt: 第一個值為答案的長度，之後的數值為 XY 的答案

1	27	1	27
2	14	2	14
3	14	3	13
4	14	4	12
5	14	5	11
6	14	6	10
7	14	7	9
8	14	8	8
9	14	9	7
10	13	10	7
11	12	11	7
12	11	12	7
13	10	13	7
14	9	14	7
15	8	15	7
16	7	16	7
17	7	17	6
18	7	18	5
19	7	19	4
20	7	20	3
21	6	21	3
22	6	22	2
23	6	23	1
24	5	24	1
25	4	25	1
26	3	26	1
27	2	27	1
28	1	28	1
29		29	

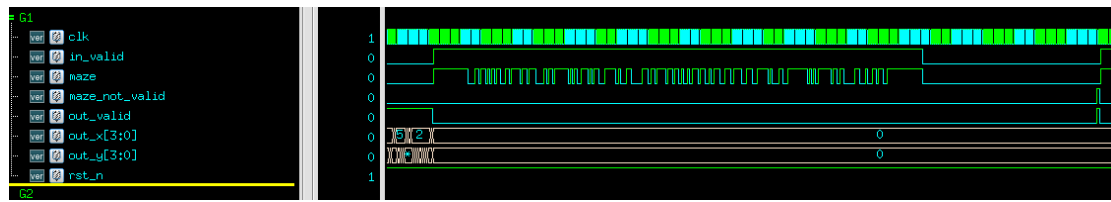
答案的長度

Example Waveform

有路徑



無路徑



Specification

1. Top module name : MS(File name: MS.sv)
2. 所有 output 必須為 0，在非同步負準位 reset。
3. 02_SYN result 不行有 error 且不能有 latches。
4. Clock period 5ns。
5. Input delay = 0.5 * clock period; output delay = 0.5* clock period;

上傳檔案

1. Code 使用 09_upload 上傳。
2. 1de 請 **6/17 23: 59** 之前上傳
3. 2de 請 **6/24 23: 59** 之前上傳
4. 一組上傳一個檔案就好
5. 報告注意:
一組上傳一份，請在第一頁上方任一處寫上組別(參考表單)，報告中心得的部分請組員分別寫各自的心得。

Grading Policy

1. Pass the RTL & Synthesis simulation.50%
2. Performance(A*T) 40%
3. Report. 10%

Note

Template folders and reference commands:

1. 01_RTL/ (RTL simulation) ./01_run
2. 02_SYN/ (Synthesis) ./01_run_dc