

Project 2

Sprite Game Display Engine

1.問題描述

請完成一Sprite Game Display Engine (後文以SGDE 表示) 電路設計。如圖 1 所示，本顯示引擎 (Display Engine)，接收數個遊戲物件之種類及座標資訊，最後結果輸出在 64*64 大小的顯示器記憶體。

本控制電路各輸入輸出信號的功能說明，請參考表一。各組同學必須根據第二節所給的設計規格及附錄 C 中的測試樣本完成設計驗證。

本次專案的作答時間為 12 月 6 日上午09:00 到 12 月 13 日下午9:00。當作答時間結束後，課程助教會根據第三節中的評分標準進行評分。為了評分作業的方便，各組同學應參考附錄 E 中所列的要求，附上評分所需要的檔案。

本題目完成後請上傳至 “140.118.121.247/Computer/File system/Project/Group_X” (The group list has already been announced in Moodle.)，詳情請參照附錄 E、F。

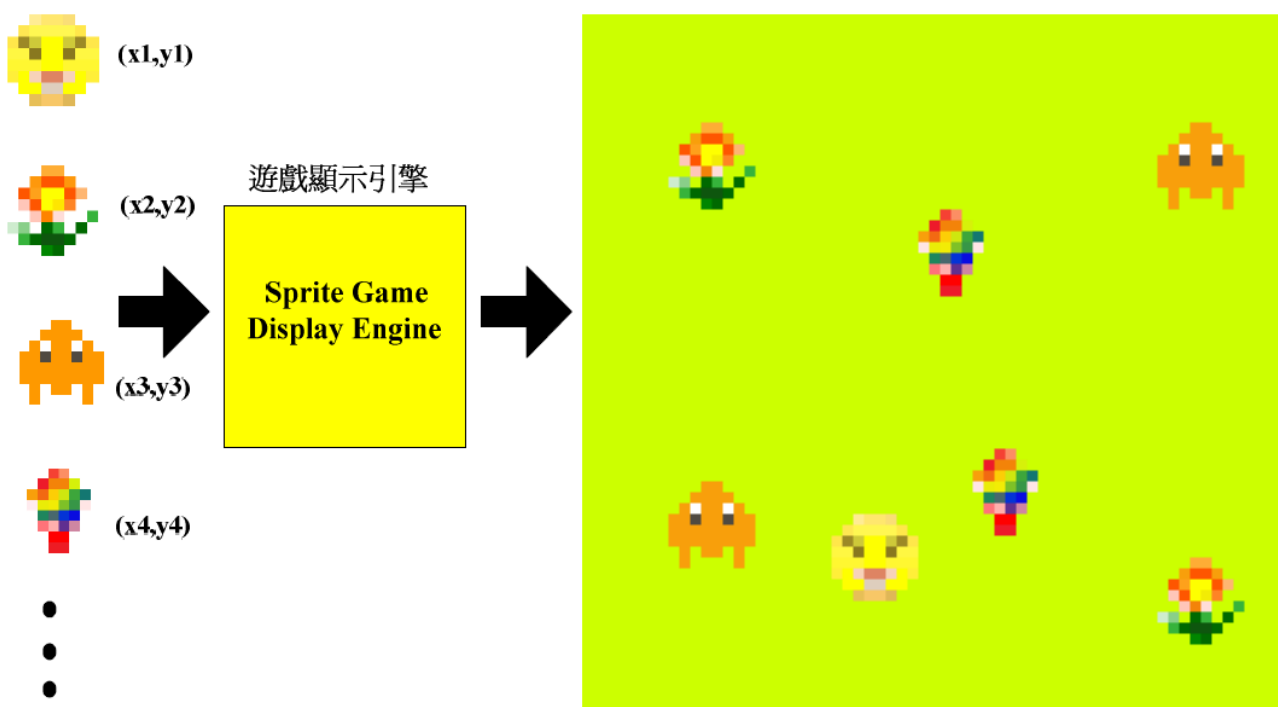


圖 1 系統示意圖

2.設計規格

2.1 系統方塊圖

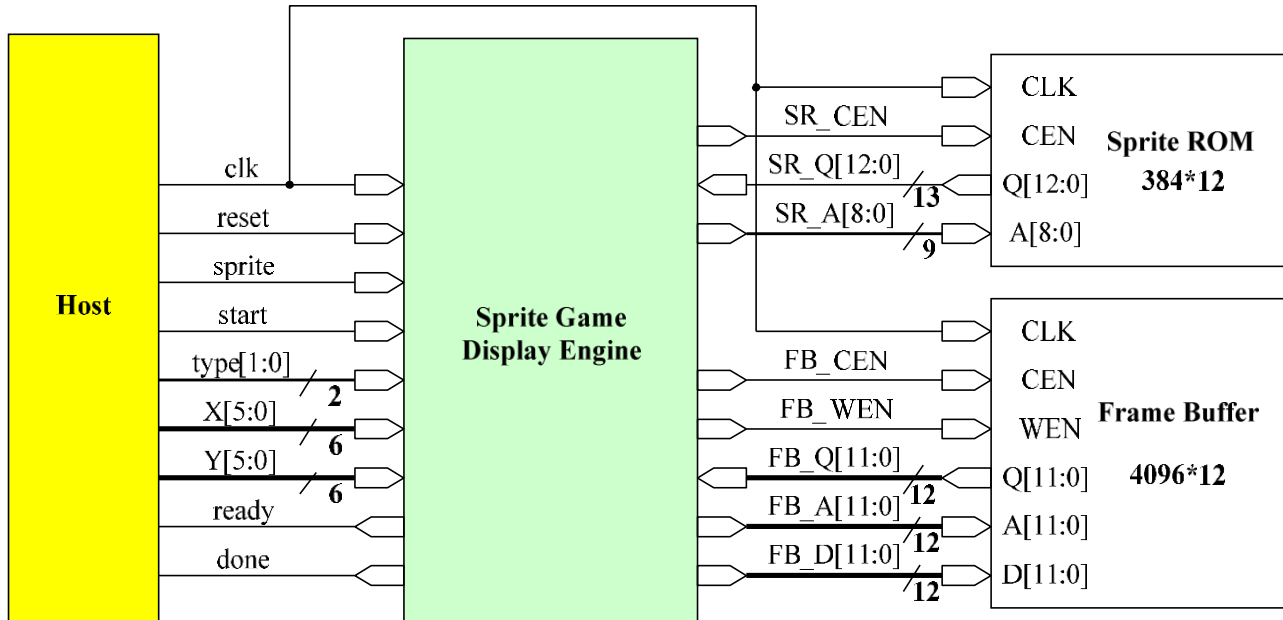


圖 2 系統方塊圖

2.2 輸入/輸出介面

表一 -輸入/輸出訊號

Signal Name	I/O	Width	Simple Description
clk	I	1	本系統為同步於時脈正緣之同步設計。
reset	I	1	高位準非同步(active high asynchronous)之系統重置信號。
sprite	I	1	當 HIGH 時表示正輸入一筆物件資料(物件種類 type 及物件座標 X,Y)。
start	I	1	當 HIGH 時表示所有的物件資料都已經輸入完畢，Game Engine 可以開始產生結果畫面至 Frame Buffer 中。
type	I	2	物件種類，2'b00 代表 Man，2'b01 代表 Ghost，2'b10 代表 Candy，2'b11 代表 Flower。
X	I	6	物件原點(左上角)在畫面上的X座標。
Y	I	6	物件原點(左上角)在畫面上的Y座標。
ready	O	1	當HIGH時代表SGDE 可以接受輸入新物件資料。
done	O	1	當HIGH時代表SGDE 已經完成動作，已將整個結果

			畫面產生至 Frame Buffer 中
SR_CEN	O	1	Sprite ROM 的Chip Enable 控制訊號，當 LOW 時可以讀取 Sprite ROM 內部的資料。
SR_Q	I	13	Sprite ROM 的輸出資料匯流排。
SR_A	O	9	Sprite ROM 的位址匯流排。
FB_CEN	O	1	Frame Buffer 的 Chip Enable 控制訊號，當 LOW 時可以操作 Frame Buffer。
FB_WEN	O	1	Frame Buffer 的Write Enable 控制訊號，當 LOW 時可以寫資料進入 Frame Buffer。
FB_Q	I	12	Frame Buffer 的資料輸出匯流排。
FB_A	O	12	Frame Buffer 的位址匯流排。
FB_D	O	12	Frame Buffer 的資料輸入匯流排。

2.3 系統功能描述

SGDE 負責將遊戲畫面寫入顯示器之記憶體中，由主機端(Host)輸入數個遊戲物件的種類及座標，SGDE 收集這些資訊將顯示圖形寫入顯示器之記憶體中(Frame Buffer)。每一個遊戲物件都是8*8 大小的圖片資料，圖片資料已儲存在Sprite ROM 內，如圖3-2 所示。SGDE 接收到由Host 端輸入的數個遊戲物件座標後，必須從Sprite ROM 取得該遊戲物件之圖片資料，然後根據要求將結果畫出(寫入Frame Buffer)。

輸入的遊戲物件的個數不定，但數目最多不會超過 20 個遊戲物件。

2.3.1 輸入資料圖片定義

遊戲物件的圖片資料都儲存在Sprite ROM。每個遊戲物件為一8*8 大小的圖片，總共64 個pixels。每一個pixel 由13bits 組成，代表RGB 資訊與mask 判斷。圖3-1 為Sprite ROM 示意圖，Sprite ROM 橫座標位置12~9bit 代表R 顏色；8~5bits 代表G 顏色；4~1bits 代表B 顏色；第0bit 表示此pixel 的mask。

mask 指的是該 pixel 透明度的設定，當顯示圖片時，僅顯示 mask=1 的 pixel，而 mask=0 的部份，則顯示底色或其底層的物件顏色，如圖4-1 為Flower 的mask 示意圖，而圖4-2 則為Flower 實際顯示圖。關於底色之定義，請參考2.3.2(遊戲畫面底色定義)。

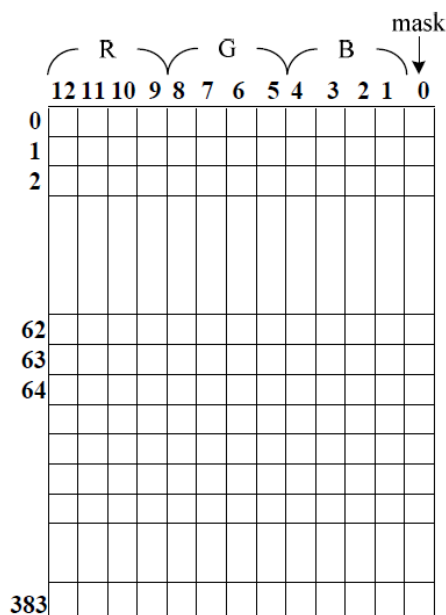


圖 3-1 Sprite ROM

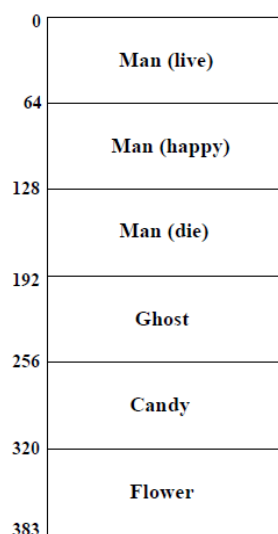


圖 3-2 Sprite ROM 儲存物件內容

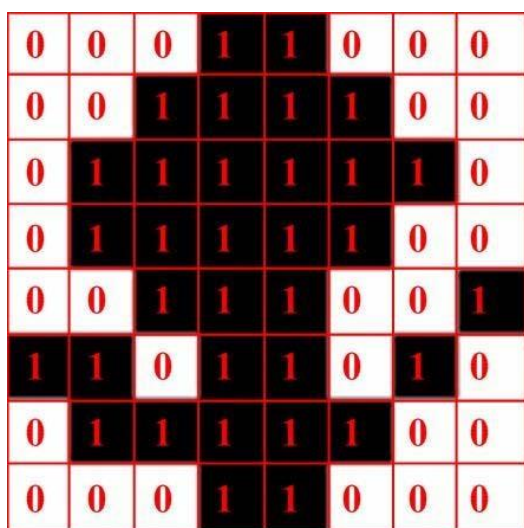


圖 4-1 mask 設定

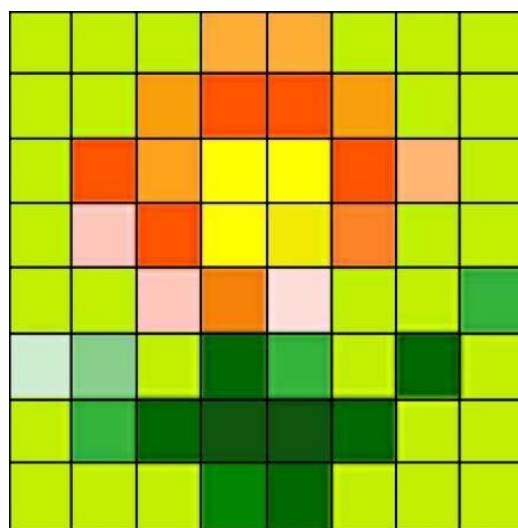


圖 4-2 Flower 實際顯示圖

遊戲物件總共分為四大類型，分別是 Man(type=0)，Ghost(type=1)，Candy(type=2)與 Flower(type=3)。

- ◆ Man：Man 分為三種狀態，分別是 live、happy 及 die，如圖5-1 是live 狀態的圖形；圖 5-2 是live 狀態下mask 的圖示。圖6-1 是happy 狀態的圖形；圖6-2 是happy 狀態下mask 的圖示。圖7-1 是die 狀態的圖形；圖7-2 是die 狀態下mask 的圖示。**Man 的狀態受到 Man 和其他物件重疊狀況所影響**，詳見2.3.3 遊戲規則定義。
- ◆ Ghost：Ghost 只有一種狀態，如圖8-1 所示；圖8-2 是Ghost 狀態下mask 的圖示。
- ◆ Candy：Candy 只有一種狀態，如圖9-1 所示；圖9-2 是Candy 狀態下mask 的圖示。
- ◆ Flower：Flower 只有一種狀態，如圖10-1 所示；圖10-2 是Flower 狀態下mask 的圖示。

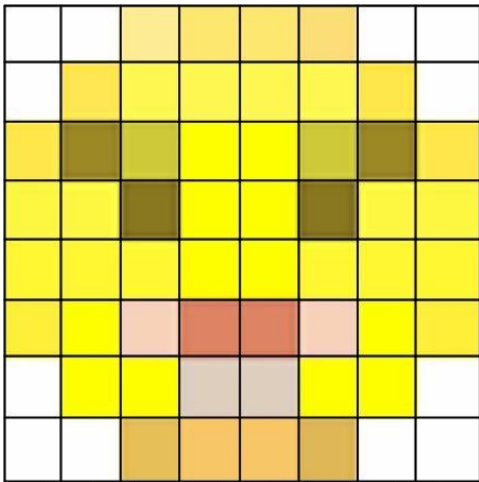


圖 5-1 live 狀態

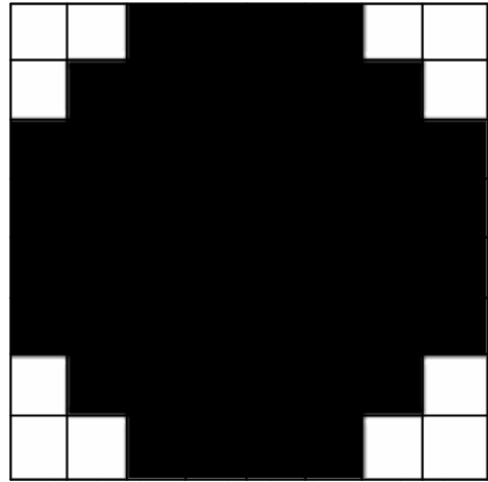


圖 5-2 live 狀態的 mask 圖示

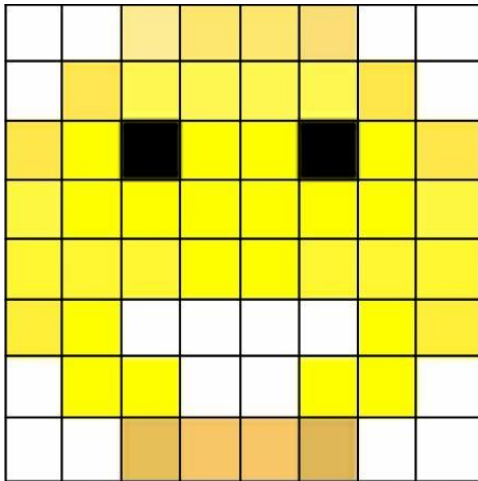


圖 6-1 happy 狀態

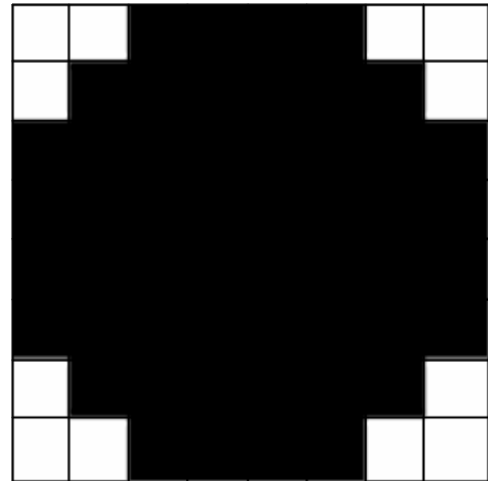


圖 6-2 happy 狀態的 mask 圖示

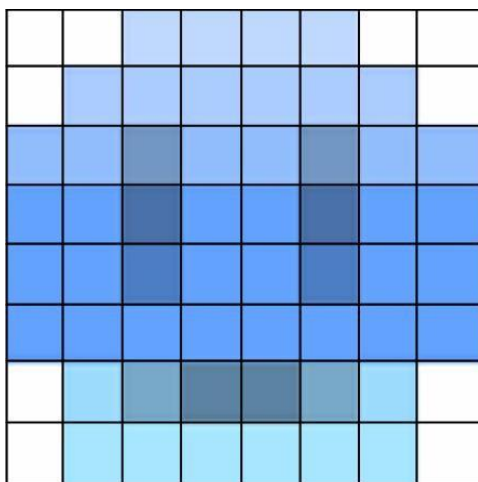


圖 7-1 die 狀態

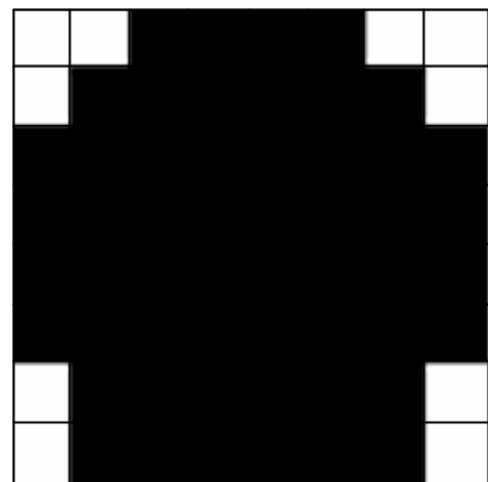


圖 7-1 die 狀態的 mask 圖示

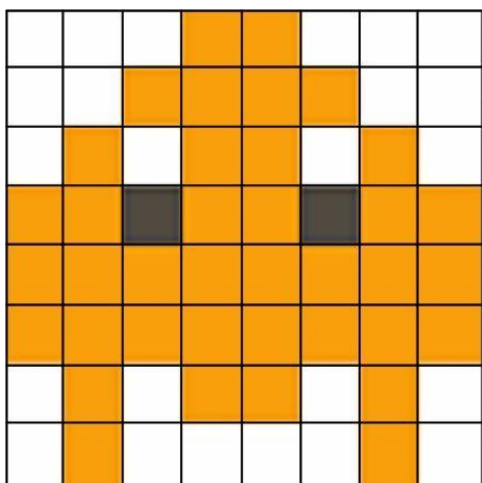


圖 8-1 Ghost 圖示

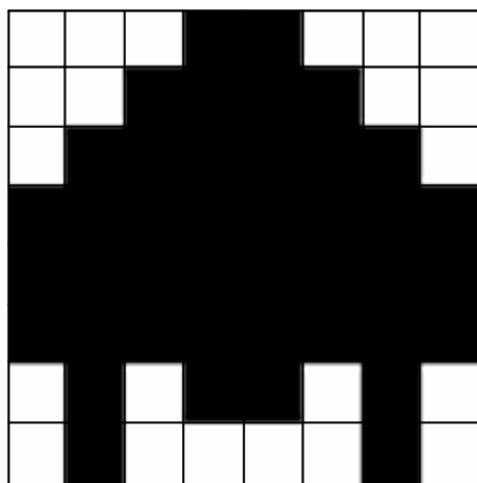


圖 8-2 Ghost 狀態的 mask 圖示

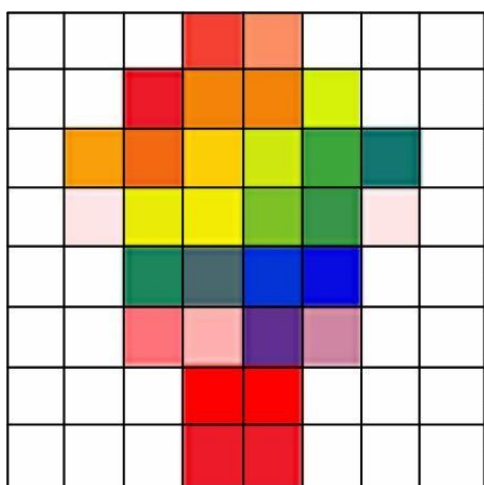


圖 9-1 Candy 圖示

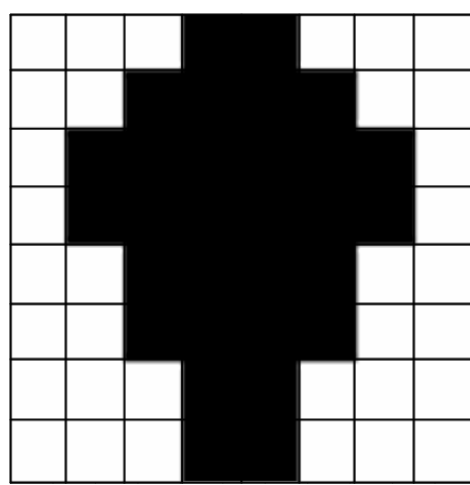


圖 9-2 Candy 狀態的 mask 圖示

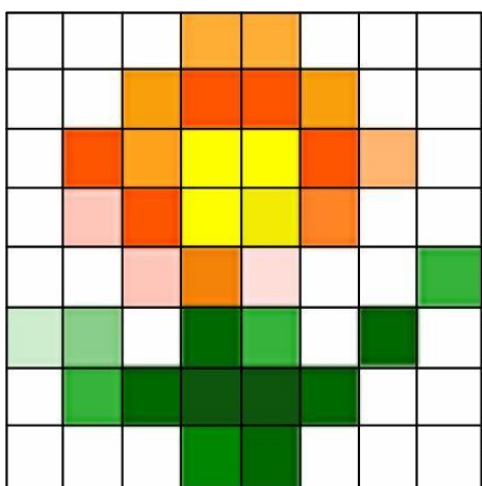


圖 10-1 Flower 圖示

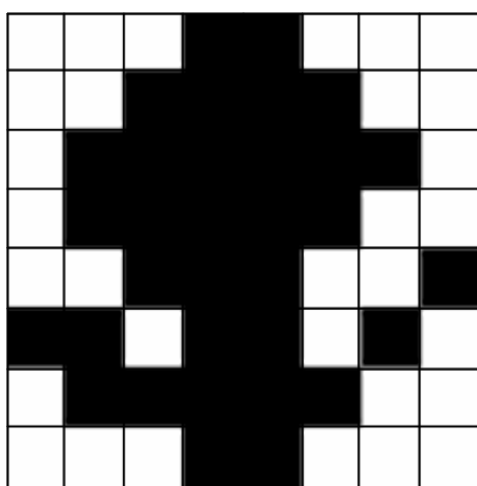


圖 10-2 Flower 狀態的 mask 圖示

每個遊戲物件的圖片資料在 Sprite ROM 內的儲存位置先後順序如圖 11 所示，從位置 0 開始依序儲存到位置 63。各物件儲存位置如圖 3-2 所示。Man(live)的儲存位置在位址為 0~63，Man(happy)的儲存位址為 64~127，Man(die)的儲存位址為 128~191，Ghost 的儲存位置在位址為 192~255，Candy 的儲存位址為 256~319，Flower 的儲存位址為 320~383。

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

圖11 物件圖片儲存順序

2.3.2 遊戲畫面底色定義

所謂底色表示遊戲畫面最底層的顏色，整個遊戲畫面中，除了遊戲物件 mask=1 的區域以外，其他部分都應填入底色。

畫面的底色定義為12'b 110011110000，即RGB 三顏色分別為 4'b1100，4'b1111 與 4'b0000。

2.3.3 遊戲規則定義

本遊戲之物件總共分為四大類型：Man、Ghost、Candy 與 Flower。其中 Man 的狀態會受其他物件重疊而有所影響，其規則如下：

- ◆ Man 沒有和其他物件重疊，Man 狀態為 live。
- ◆ Man 與 Candy 重疊的時候，Man 狀態變為 happy。
- ◆ Man 與 Ghost 重疊的時候，Man 狀態變為 die。
- ◆ Man 同時與 Candy 和 Ghost 重疊的時候，Man 狀態變為 happy。
- ◆ Man 不管在何種狀態，與 Flower 重疊的時候，Man 仍維持原來狀態。

綜合以上的規則，可以定義出如圖 12 所示的狀態圖。

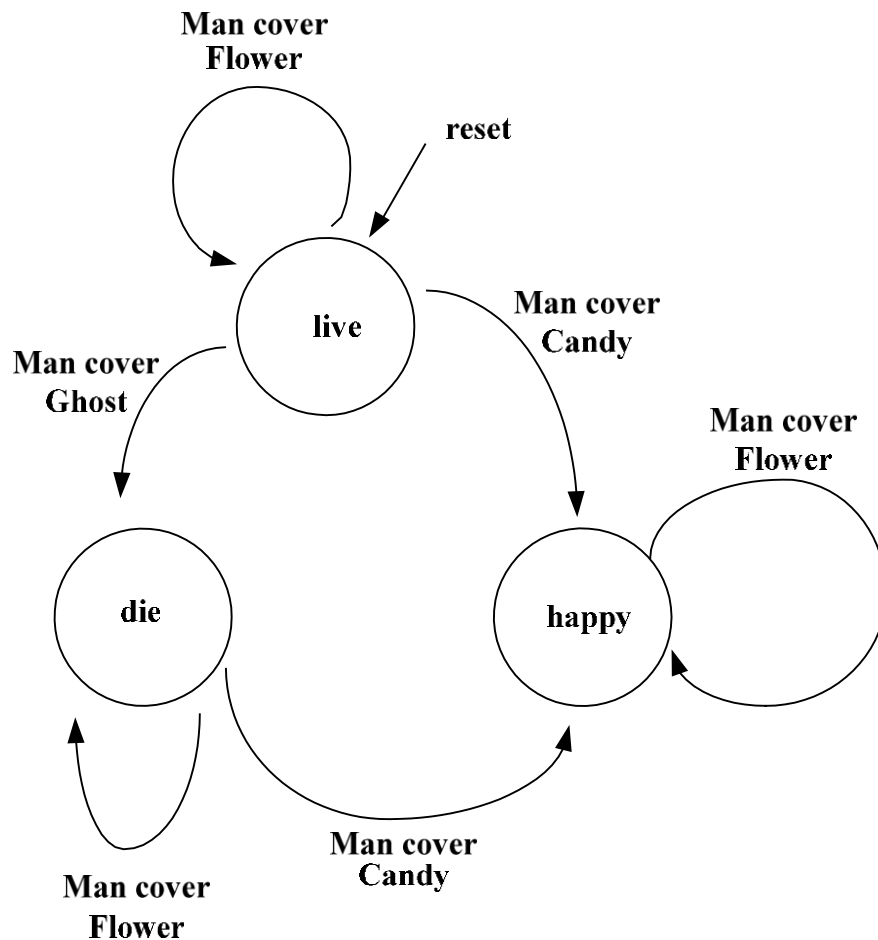


圖12 狀態圖

2.3.4 圖片資料重疊有效定義

假設兩個 8*8 圖片資料座標重疊在一起，並非只要重疊在一起就稱為有效。利用Man 的 mask 圖示與 Candy 的 mask 圖示說明，已經將兩張圖片mask 值標上，綠色代表Man 的 mask 值，紅色代表 Candy 的 mask 值，如圖 13 與圖 14。

將圖 13 與圖 14 的兩張圖片重疊，如圖 15 所示，很明顯的看到，雖然兩張圖片重疊在一起，但是兩者之間 mask 值 1 的部份並沒有重疊，所以判定重疊無效。

利用相同的例子，如圖 16 所示，調整兩張照片重疊的部份，可以看到兩者之間 mask 值 1 的部份重疊了，所以判定這兩個物件重疊有效。

Man 和其他物件重疊與否判斷，以 Man (live)的 mask 為判斷依據。

0	0	1	1	1	1	0	0
0	1	1	1	1	1	1	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
0	1	1	1	1	1	1	0
0	0	1	1	1	1	0	0

圖 13 Man 的 mask 值

0	0	0	1	1	0	0	0
0	0	1	1	1	1	0	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0

圖 14 Candy 的 mask 值

0	0	1	1	1	10	00	01
0	1	1	1	1	10	10	00
1	1	1	1	1	10	10	10
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	0
0	0	1	1	1	1	0	0

圖 15 遊戲物件重疊無效判斷

再舉一個例子說明，如輸入的順序為 Man、Ghost1、Ghost2、Candy1、Flower1、Ghost3、Flower2、Candy2、Flower3。則重疊時的顯示由上而下為：Man、Candy2、Ghost3、Candy1、Ghost2、Ghost1、Flower3、Flower2、Flower1。

2.3.5 顯示器及物件座標定義

SGDE 顯示器為一64*64 陣列，其**原點(0,0)定義於左上角**，向右X 軸座標增加，向下Y 軸座標增加。**Host 端輸入的物件座標代表該物件左上角對應於顯示器上的座標。**

Host 端輸入之物件座標(X, Y)範圍： $0 \leq X \leq 63$ $0 \leq Y \leq 63$ ，即 SGDE 不需要考慮物件超出顯示器的狀況。

假設提供一個位於 X、Y 座標，(10,5)的 happy 圖片資料，在64*64 的 Sprite Game Display Engine 顯示器中實際的位置如圖 18 所示。

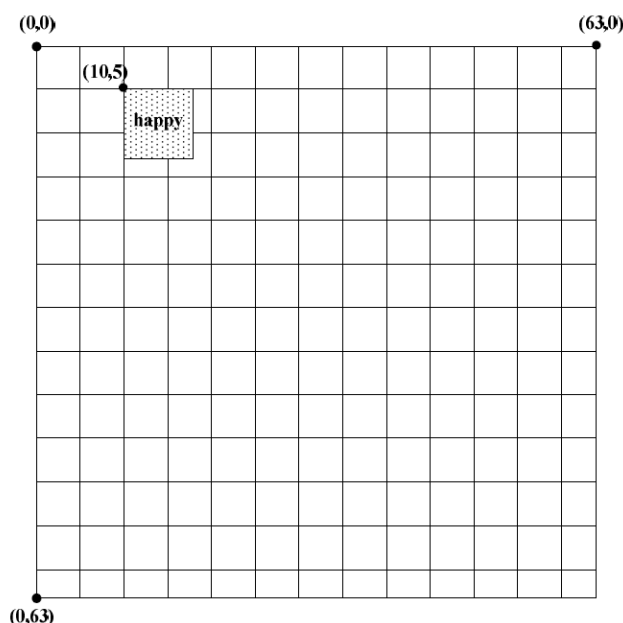


圖 18 實際座標圖

2.3.6 顯示器及 Frame Buffer 位址定義

Sprite Game Display Engine 顯示器為一64*64 陣列，總共有4096 個pixel 值，如圖 19 所示，每個pixel 12bits 記錄著 RGB 的資訊。顯示器和 Frame Buffer 的位址對應如圖 19 和圖 20 所示。圖 19 的位置 0 對應到圖 20 Frame Buffer 位置 0 的地方，位置 1 對應到 Frame Buffer 位置 1 的地方，其他位置照此規則。Frame Buffer 的每個位址共 12bits，代表該 pixel 的RGB 值。位置11~8bit 為 R 顏色；7~4bit 為 G 顏色；3~0bit 為 B 顏色。

0	1	2	3	4			59	60	61	62	63
64	65										127
128	129										191
3968	3969	3970					4027	4028	4029	4030	4031
4032										4094	4095

圖 19 4096 pixel 位址

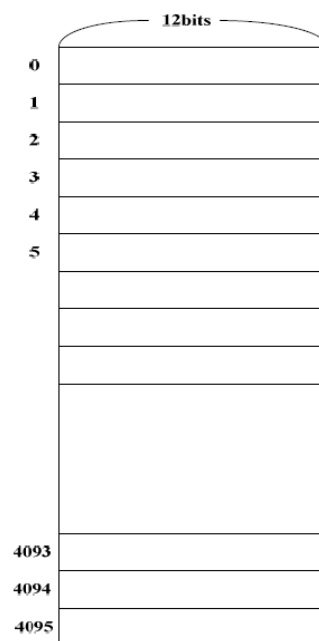
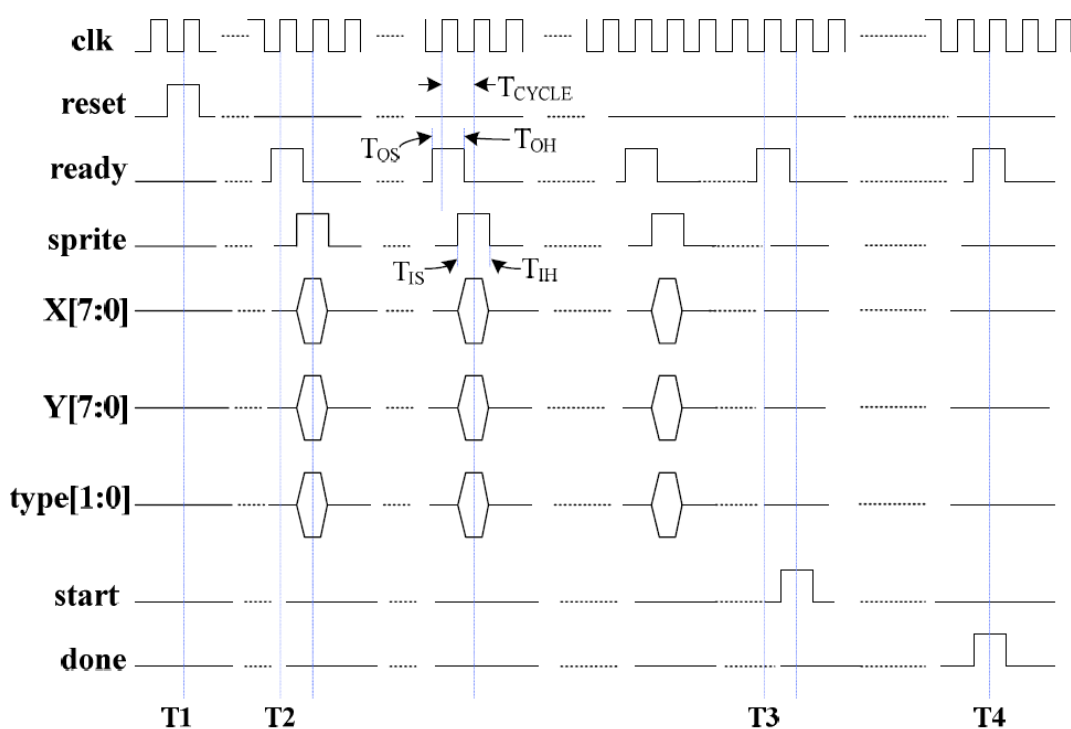


圖 20 Frame Buffer

本顯示引擎的任務是將所有的 4096 pixel 的RGB 資訊儲存在 Frame Buffer 中。

2.4 時序規格

2.4.1 Host 端時序圖



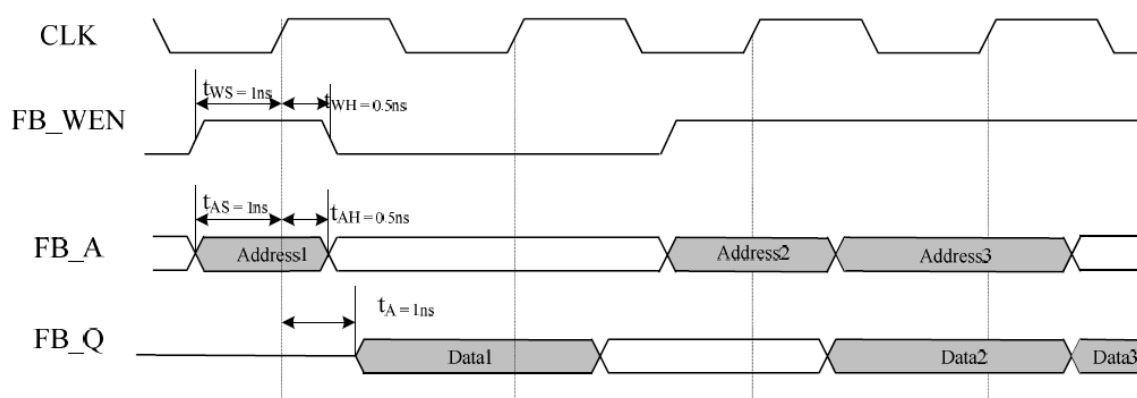
1. T1 時間點，系統啟動送出重置(reset)訊號，SGDE 進行前置工作準備，準備前置工作期間ready 設為low。
2. T2 時間點，Host 發現 ready 為 high，表示 SGDE 可以開始接受輸入，host 在 clock 負緣送出第一筆物件資料，包含物件種類(type)及物件座標(X,Y)，送出物件資料的同時，sprite 升為 high 表示此時物件資料有效。SGDE 若無法立刻處理此物件，即將 ready 設為 low。
3. T2 至T3 之間，Host 以相同方式送出多筆物件資料
4. T3 時間點，物件資料都已送出，此時 ready 為 high，Host 送出一個 cycle 的 start 訊號。
5. SGDE 收到 start 後，進行畫面最後繪製(寫入 Frame Buffer)，完成後送出 done 訊號(T4)。

2.4.2 Host 端時序規格

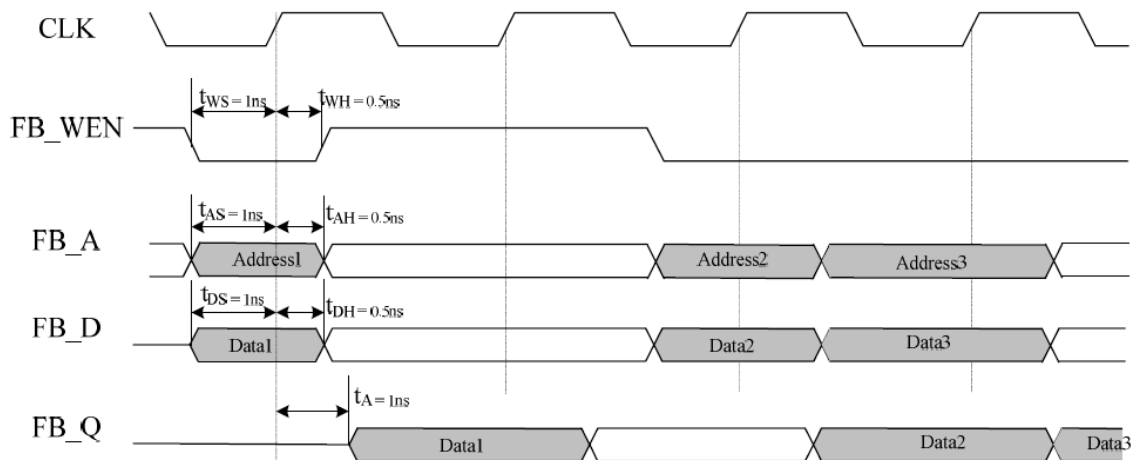
Symbol	Description	Value
T_{cycle}	clock period	user defined
T_{IS}	setup time (time period from valid input signal to positive edge of clock) for all inputs from host.	1/2 clock period
T_{IH}	hold time(time period from positive edge of clock to invalid input signal.) for all inputs from host	1/2 clock period
T_{OS}	setup time (from the time at which output is valid to the clock rising edge) for all outputs to the host	1/4 clock period
T_{OH}	hold time (from the clock rising edge to the time at which output is invalid) for all outputs to the host	0.1 ns

2.4.3 Frame Buffer 時序圖

Frame Buffer Read : (FB_CEN=0)



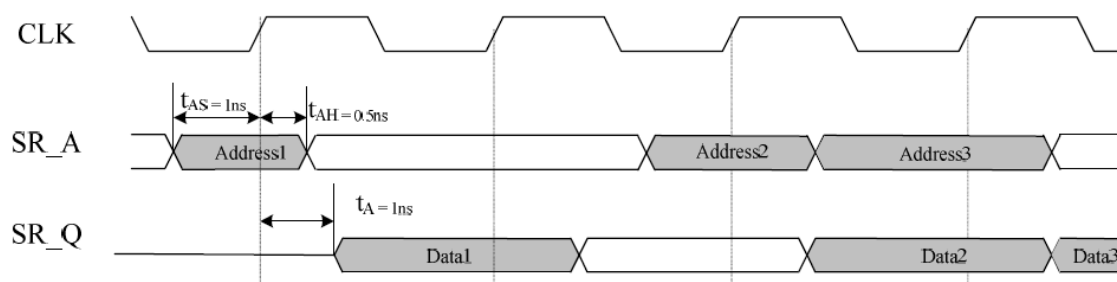
Frame Buffer Write : (FB_CEN=0)



Symbol	Description	Delay(ns)	Notes
t_{AS}	Address setup time	1	Min
t_{AH}	Address hold time	0.5	Min
t_{WS}	Write Enable Bar(FB_WEN) setup time	1	Min
t_{WH}	Write Enable Bar(FB_WEN) hold time	0.5	Min
t_A	Access time	1	Max
t_{DS}	Data input setup time	1	Min
t_{DH}	Data input hold time	0.5	Min

2.4.4 Sprite Rom 時序圖

Sprite ROM Read : (SR_CEN=0)



Symbol	Description	Delay(ns)	Notes
t_{AS}	Address setup time	1	Min
t_{AH}	Address hold time	0.5	Min
t_A	Access time	1	Max

3. 評分標準

課程助教將依照各組同學提供之系統時脈進行 RTL simulation 與 gate-level simulation，以驗證設計正確性。各組同學應於本專案定義的系統時脈下，確保輸出結果無設置與保持時間(setup/hold time)的問題，並完全符合課程助教所提供的標準設計結果為準。

表三為本次專案的評分標準，一旦設計經課程助教驗證後，且同學的設計結果正確(post-layout)將可得到基本分80分，而其次課程助教將依據 Performance Metric的表現，來給予而外的分數，而 Performance Metric的計算方式如下所示，其數值越低表示設計的電路效能越好。

$$\bullet \text{ Performance Metric} = \text{Time Complexity} \times \text{Area Complexity}$$

Time Complexity: 當你完成 Post layout 的 gate level netlist 的 ncverilog的驗證時，你可以從“ncverilog.log”的檔案中得到你的 Time Complexity結果，如圖十一所示。這個時間代表整個電路完成工作所需的時間，與電路運作頻率(operating frequency)及電路效能(operations per cycle)有關。

Area Complexity: Area Complexity 可以從 SOC encounter 的 Layout圖透過尺規量得，如圖 21 所示。

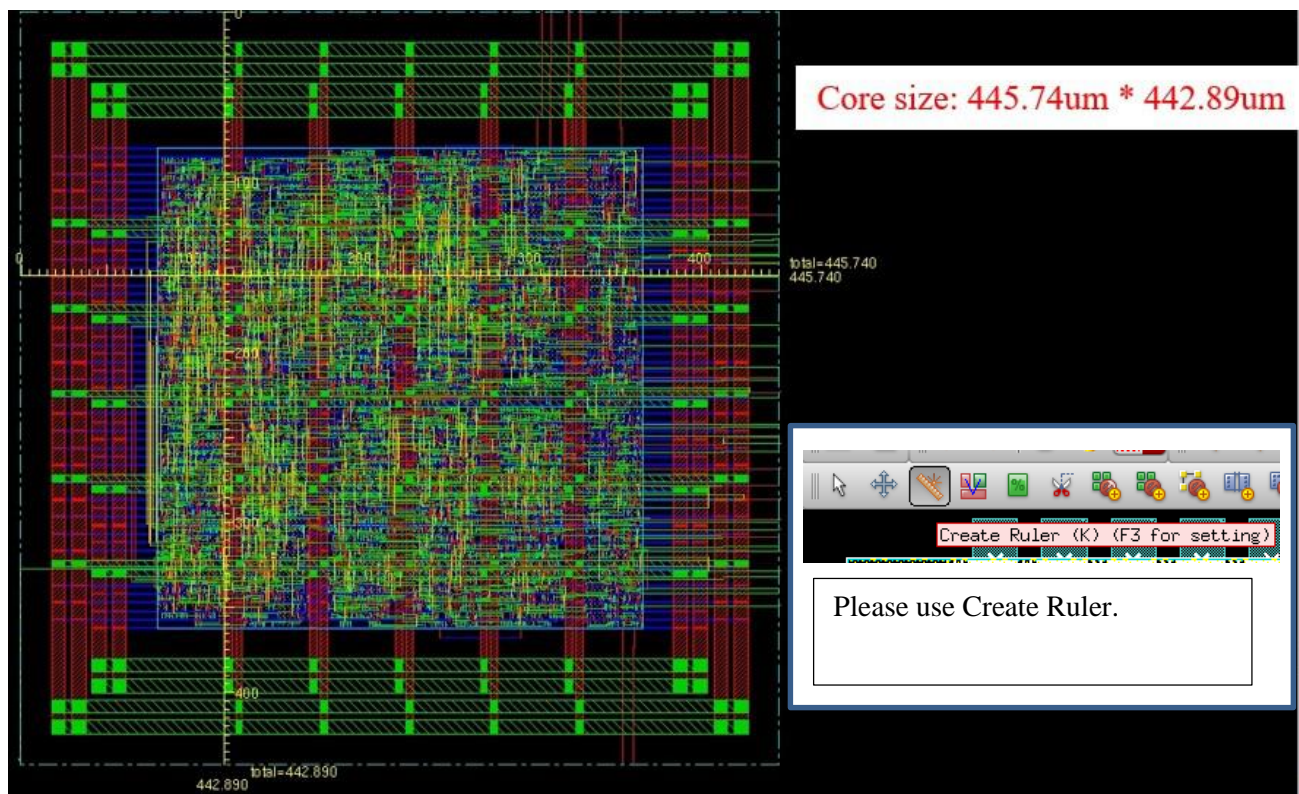


圖 21 Layout 尺規量測圖

附錄

在附錄 A 說明課程助教提供的遊戲物件的 RGB 資訊；附錄 B 為課程助教所提供各組同學的設計檔說明；附錄 C 為課程助教提供的測試樣本；附錄 D 為設計驗證說明；附錄 E 為評分用檔案，亦即各組同學必須繳交的檔案資料；附錄 F 為檔案上傳相關說明。

附錄 A 遊戲物件 RGB 資訊

課程助教提供的遊戲物件的 RGB 資訊紀錄如下(以 16 進位表示)。

1. Man_live

FFF	FFF	FE9	FE6	FE6	FD7	FFF	FFF
FFF	FE5	FF5	FF5	FF5	FF5	FE4	FFF
FE4	982	CC3	FF0	FF0	CC3	982	FE4
FF4	FF4	871	FF0	FF0	871	FF4	FF4
FF3	FF3	FF3	FF0	FF0	FF3	FF3	FF3
FE3	FF0	FDB	D86	D86	FDB	FF0	FE3
FFF	FF0	FF0	DCB	DCB	FF0	FF0	FFF
FFF	FFF	EB5	FC6	FC6	DB5	FFF	FFF

2. Man_happy

FFF	FFF	FE9	FE6	FE6	FD7	FFF	FFF
FFF	FE5	FF5	FF5	FF5	FF5	FE4	FFF
FE4	FF0	000	FF0	FF0	000	FF0	FE4
FF4	FF0	FF0	FF0	FF0	FF0	FF0	FF4
FF3	FF3	FF3	FF0	FF0	FF3	FF3	FF3
FE3	FF0	FFF	FFF	FFF	FFF	FF0	FE3
FFF	FF0	FF0	FFF	FFF	FF0	FF0	FFF
FFF	FFF	EB5	FC6	FC6	DB5	FFF	FFF

3. Man_die

FFF	FFF	BDF	BDF	BDF	BDF	FFF	FFF
FFF	ACF	ACF	ACF	ACF	ACF	ACF	FFF
9BF	9BF	79C	9BF	9BF	79C	9BF	9BF
6AF	6AF	47A	6AF	6AF	47A	6AF	6AF
6AF	6AF	47C	6AF	6AF	47C	6AF	6AF
6AF	6AF	6AF	6AF	6AF	6AF	6AF	6AF
FFF	9DF	7AC	58A	58A	7AC	9DF	FFF
FFF	AEF	AEF	AEF	AEF	AEF	AEF	FFF

4. Ghost

FFF	FFF	FFF	F90	F90	FFF	FFF	FFF
FFF	FFF	F90	F90	F90	F90	FFF	FFF
FFF	F90	FFF	F90	F90	FFF	F90	FFF
F90	F90	S44	F90	F90	S44	F90	F90
F90	F90	F90	F90	F90	F90	F90	F90
F90	F90	F90	F90	F90	F90	F90	F90
FFF	F90	FFF	F90	F90	FFF	F90	FFF
FFF	F90	FFF	FFF	FFF	FFF	F90	FFF

5. Candy

FFF	FFF	FFF	F43	F86	FFF	FFF	FFF
FFF	FFF	E12	F80	F80	DF0	FFF	FFF
FFF	F90	F61	FC0	CE0	3A3	177	FFF
FFF	FEE	EE0	FE0	7C2	394	FEE	FFF
FFF	FFF	185	466	03D	00E	FFF	FFF
FFF	FFF	F77	FAA	629	C8A	FFF	FFF
FFF	FFF	FFF	F00	F00	FFF	FFF	FFF
FFF	FFF	FFF	E12	E12	FFF	FFF	FFF

6. Flower

FFF	FFF	FFF	FA3	FA3	FFF	FFF	FFF
FFF	FFF	F90	F50	F50	F90	FFF	FFF
FFF	F50	FA1	FF0	FF0	F50	FB7	FFF
FFF	FCB	F50	FF0	FE0	F82	FFF	FFF
FFF	FFF	FCB	F80	FDD	FFF	FFF	3B3
CEC	8C8	FFF	060	3B3	FFF	060	FFF
FFF	3B3	060	050	050	060	FFF	FFF
FFF	FFF	FFF	080	060	FFF	FFF	FFF

附錄 B 設計檔(For verilog or VHDL)

1. 下表為課程助教所提供各組同學的設計檔

表二

檔名	說明
testfixture.v	測試樣本檔。此測試樣本檔定義了時脈週期與測試樣本之輸入信號
Encounter_example	SOC Encounter 的範例檔(包含 mmmc, LEF, GDS 生成指令
SGDE.v	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告
FB.v	Frame Buffer 的verilog 檔
SR.v	Sprite ROM 的 verilog 檔
SR_verilog.rcf	Sprite ROM 資料
synopsys_dc.setup	Design Compiler 設定檔
SGDE.tcl	The script for Design Compiler constraint
input1.dat	第一組測試樣本之指令
golden_image1.xpm	第一組測試樣本使用之影像檔
golden1.dat	第一組測試樣本之正確結果
input2.dat	第二組測試樣本之指令
golden_image2.xpm	第二組測試樣本使用之影像檔
golden2.dat	第二組測試樣本之正確結果
input3.dat	第三組測試樣本之指令
golden_image3.xpm	第三組測試樣本使用之影像檔
golden3.dat	第三組測試樣本之正確結果
input4.dat	第四組測試樣本之指令
golden_image4.xpm	第四組測試樣本使用之影像檔
golden4.dat	第四組測試樣本之正確結果
input5.dat	第五組測試樣本之指令
golden_image5.xpm	第五組測試樣本使用之影像檔
golden5.dat	第五組測試樣本之正確結果

2. 請使用 **SGDE.v**，進行遊戲引擎控制器之設計。其模組名稱、輸出/入埠宣告如下所示：

```
module SGDE ( ready, done, clk, reset, sprite, start, type, X, Y, SR_CEN, SR_A, SR_Q,
FB_CEN, FB_WEN, FB_A, FB_D, FB_Q);
```

```
input clk, reset, sprite, start;
```

```
input [1:0] type;
```

```

input [5:0] X, Y;
input [12:0] SR_Q;
input [11:0] FB_Q;
output ready, done;
output SR_CEN, FB_CEN, FB_WEN;
output [8:0] SR_A;
output [11:0] FB_A, FB_D;
endmodule

```

3. 此專案共提供五組測試樣本，請自行修改 testfixture.v 內容，以使用其它組測試樣本，修改方法如下(以第二組為例):

```

`define INDATAFILE "./input1.dat"
`define GOLDENDATA "./golden1.dat"
`define OUTIMAGE "./image1.xpm"

```

將以上三行修改為:

```

`define INDATAFILE "./input2.dat"
`define GOLDENDATA "./golden2.dat"
`define OUTIMAGE "./image2.xpm"

```

4. 使用 verilog 模擬之後，會產生一個影像檔 image1.xpm (OUTIMAGE 定義之檔名)。此影像檔為模擬遊戲顯示器之輸出結果，請執行底下的UNIX 指令以觀看此影像：

> **xv image1.xpm &**

5. 請各組同學使用課程助教所提供的測試樣本(testfixture.v)，來進行設計之模擬驗證。各組同學除了將利用本試題所提供的測試樣本進行各組同學之設計測試外，我們亦準備另一份測試樣本，進行設計之第二次驗證。唯通過二次驗證且均能符合輸出結果，始能稱為無誤之設計。

測試樣本(testfixture.v) 包含 *clk*, *reset*, *sprite*, *start*, 與 *type* 測試信號。而於**測試樣本內所定義的時脈週期參數(CYCLE)**，可由同學們依需求自行修改。

附錄 C 測試樣本

此專案共提供五組測試樣本，為方便設計者除錯之用，課程助教提供的遊戲物件座標以及遊戲物件型式詳列如下:

Man(type=0)、Ghost(type=1)、Candy(type=2)、Flower(type=3)

1. 測試樣本一：input1.dat

順序	1	2	3	4	5	6	7
type	0	1	2	1	3	2	3
X	23	8	30	53	53	35	8
Y	46	43	18	10	50	40	10

2. 測試樣本二：input2.dat

順序	1	2	3	4	5	6	7	8
type	0	3	1	2	2	3	1	3
X	30	48	31	25	36	49	35	28
Y	30	53	23	26	28	49	35	33

3. 測試樣本三：input3.dat

順序	1	2	3	4	5	6	7	8	9	10
type	0	1	3	2	1	2	3	2	1	3
X	30	55	48	17	30	25	28	36	35	18
Y	30	8	53	20	23	27	33	28	35	43

4. 測試樣本四：input4.dat

順序	1	2	3	4	5	6	7	8	9	10
type	0	1	3	3	2	3	1	1	3	1
X	30	5	23	18	7	8	25	41	48	33
Y	30	8	12	43	20	13	5	23	53	34
順序	11	12	13	14	15	16	17	18	19	20
type	2	2	3	1	1	2	2	1	3	1
X	21	25	28	5	31	47	54	5	9	21
Y	2	27	35	35	3	2	56	55	55	43

5. 測試樣本五：input5.dat

順序	1	2	3	4	5	6	7	8	9	10
type	0	3	1	2	2	3	2	1	1	3
X	30	48	55	17	25	13	36	35	23	18
Y	30	53	8	20	26	28	28	15	43	43
順序	11	12	13	14	15	16	17	18	19	20
type	3	1	3	2	3	2	1	1	3	1
X	48	55	17	15	28	47	35	5	47	20
Y	3	4	2	26	33	27	35	54	20	26

附錄 D 設計驗證說明

各組同學繳交資料前應完成 RTL，Gate-Level 與Physical 三種階段驗證，以確保設計正確性。

- RTL 與Gate-Level 階段：同學們必須進行RTL simulation 及Gate-Level simulation，模擬結果必須於參賽者自行定義的系統時脈下，**輸出結果正確且無setup/hold time**的問題。

- Physical 階段，包含三項驗證重點：

1.完成最後layout

i. Marco layout，不含IO Pad。

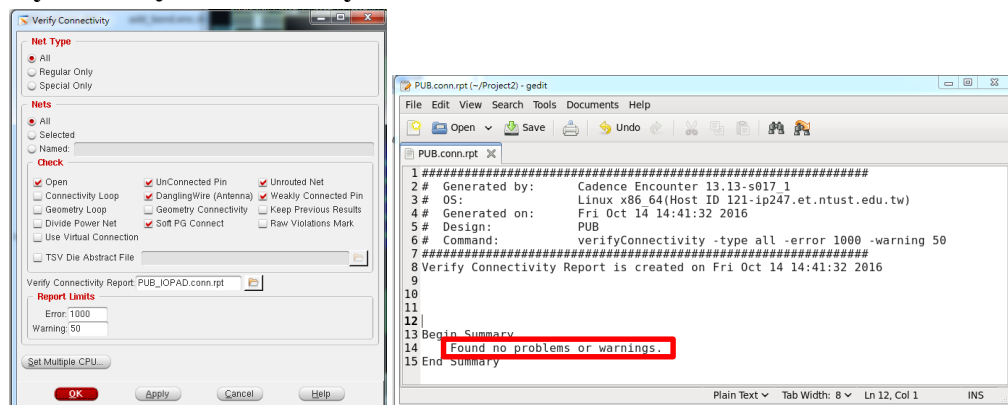
ii. VCC 與GND power ring 請各設定為2um。

2.完成post-layout simulation：參賽者必須使用P&R軟體寫出之 netlist 檔與 sdf 檔完成 post-layout gate-level simulation

3.完成DRC 與LVS 驗證：參賽者必須以其所使用之P&R 軟體內含之 DRC 與 LVS 驗證功能完成 DRC 與 LVS 驗證

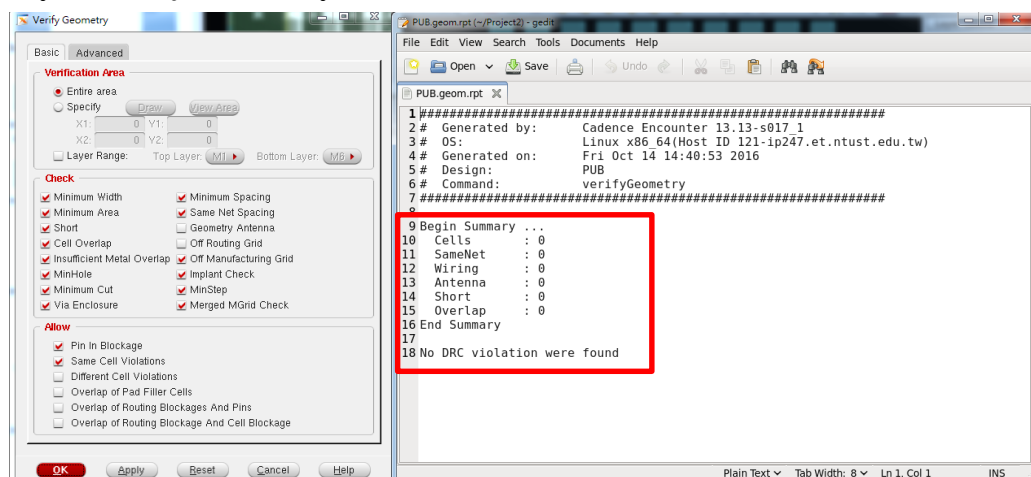
- i 驗證 DRC 與 LVS 步驟如下：在SOC Encounter 視窗下點選

“Verify → Verify Connectivity...” Default 值，按 **OK**。



在此步驟完成後會產生 *.conn.rpt 檔，如驗證結果正確，會顯示如右圖紅框處之文字。

“Verify → Verify Geometry...” Default 值，按 **OK**。



在此步驟完成後會產生 *.geom.rpt 檔，如驗證結果正確，會顯示如右圖紅框處之文字。

附錄 E 評分用檔案

評分所須檔案可以分為以下幾個部份：(1)RTL design，即各組同學對該次專案設計的 RTL code，若設計採模組化而有多個設計檔，請務必將合成所要用的各 module 檔放進來，以免助教進行評分時，無法進行模擬；(2)Gate-Level design，即由合成軟體所產生的 gate-level netlist，以及對應的SDF 檔；(3)Physical design，即由 **Cadence SOC Encounter** 所產生的 GDS Layout 檔，以及 DRC/LVS report 檔等。

表三

<i>RTL category</i>		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
N/A	*.doc	Design Report Form
RTL Simulation	*.v or *.vhd	Verilog (or VHDL) synthesizable RTL code
<i>Gate-Level category</i>		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
Pre-layout Gate-level Simulation	*_syn.v	Verilog gate-level netlist generated by Synopsys Design Compiler
	*_syn.sdf	Pre-layout gate-level sdf
	*_syn.vcd	Wave information generated by Ncverilog
	ncverilog.log	The log file generated by Ncverilog
<i>Physical category</i>		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
P&R	*.gds	GDSII layout
	*.conn.rpt *.geom.rpt	DRC/LVS report generated by Cadence SOC Encounter
Post-layout Gate-level Simulation	*_pr.v	Verilog gate-level netlist generated by Cadence SOC Encounter
	*_pr.sdf	Post-layout gate-level sdf
	*_pr.vcd	Wave information generated by Ncverilog
	ncverilog.log	The log file generated by Ncverilog

附錄 F 檔案上傳

所有包含於如附錄 E 中表格所示的檔案，均需要提交至課程助教指定的資料夾內。步驟如下：

1. 參考附錄 E 評分用檔案，將除了 Design Report Form以外的檔案繳交檔案複製到 (140.118.121.247/Computer/File System/Project/Group_X) 資料夾中，其中X表示組別，請務必確保檔案放入正確的資料夾內。
2. Design Report Form的 WORD 檔，請於填寫完後上傳到 Moodle，檔名請依各組組別命名(ex: Group_1)。

若有任何問題，請聯絡課程助教：

陳偉哲 s2017091@gmail.com ; 辦公室: IB714-1

曾昱豪 M10402146@mail.ntust.edu.tw 辦公室: IB714-1