

2007 University/College IC Design Contest

Cell-Based IC Category

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。每個參賽隊伍必須根據下一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 IC 設計競賽比賽時間為上午 08:30 到下午 20:30。當 IC 設計競賽結束後，CIC 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，各參賽隊伍應參考附錄 E 中所列的要求，附上評分所需要的檔案。

本題目之測試樣本置於 **`/usr/cad/icc2007/cb/icc2007cb.tar`**，請執行以下指令取得測試樣本：

```
> tar xvf /usr/cad/icc2007/cb/icc2007cb.tar
```

軟體環境及設計資料庫說明請參考附錄 G 與附錄 H。

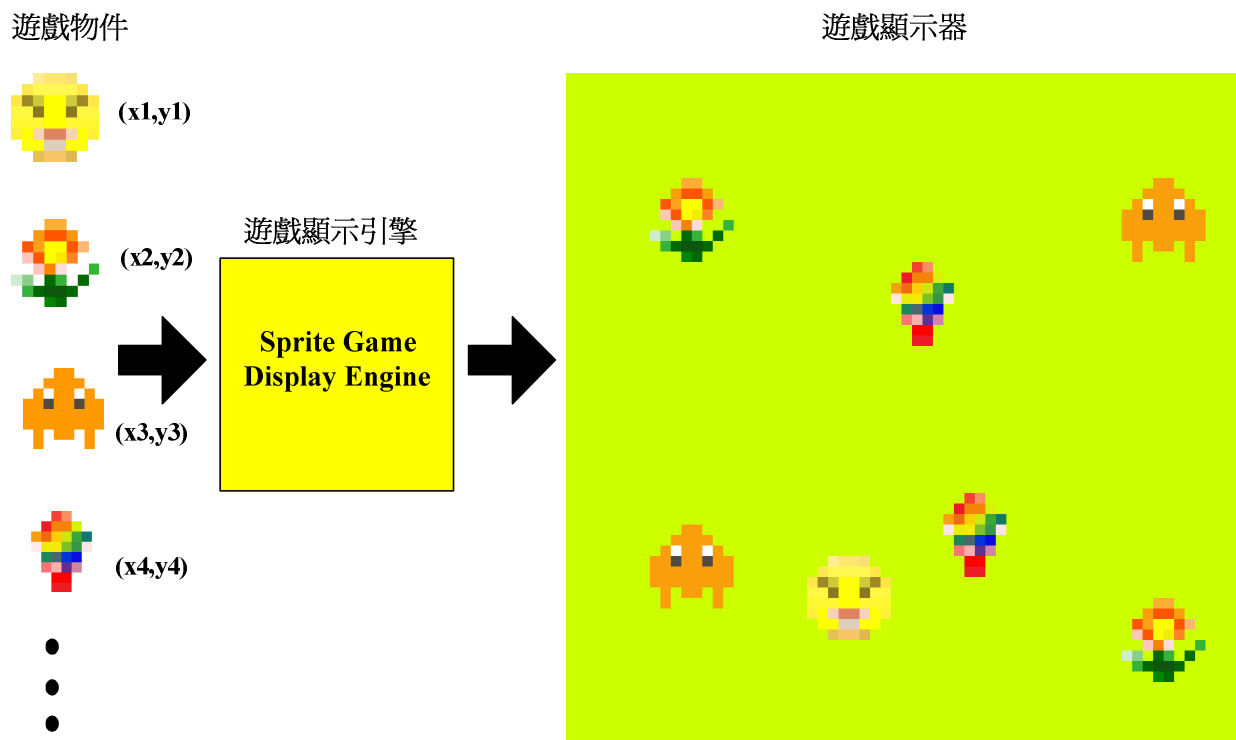


圖 1 系統示意圖

2.設計規格

2.1 系統方塊圖

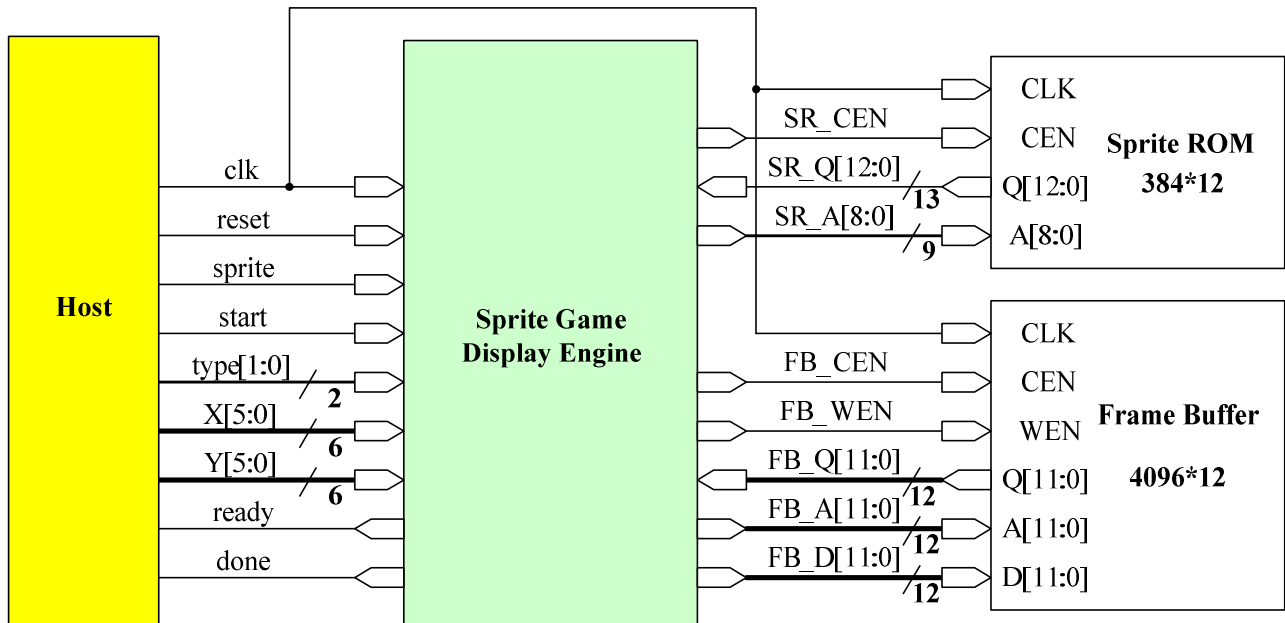


圖 2 系統方塊圖

2.2 輸入/輸出介面

表 1 -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)。

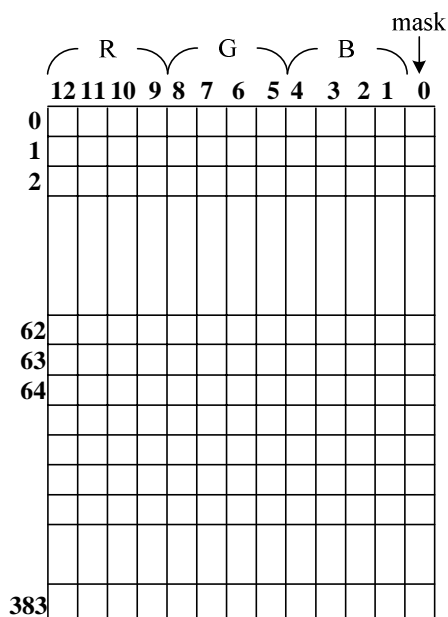


圖 3-1 Sprite ROM

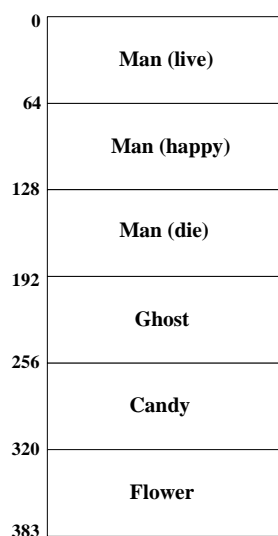


圖 3-2 Sprite ROM 儲存物件內容

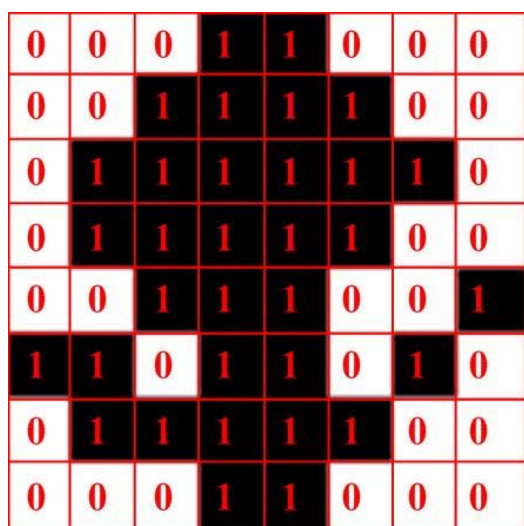


圖 4-1 mask 設定

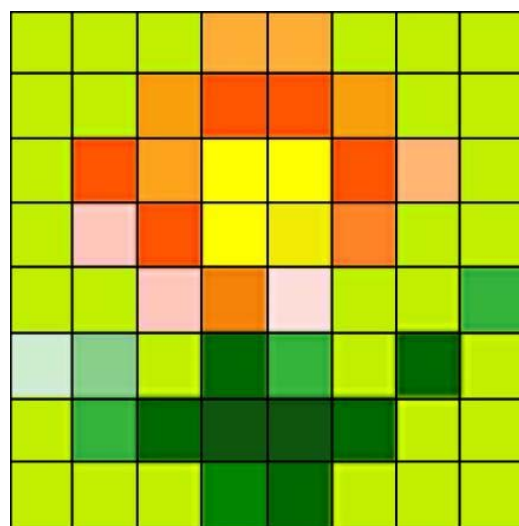


圖 4-2 Flower 實際顯示圖

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。

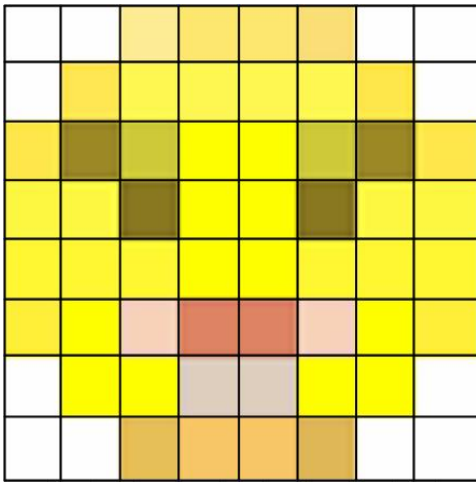


圖 5-1 live 狀態

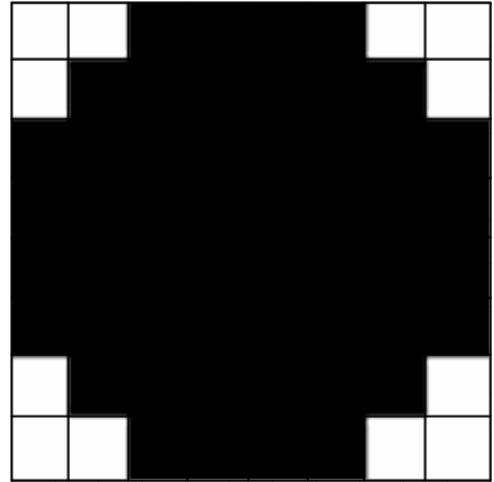


圖 5-2 live 狀態的 mask 圖示

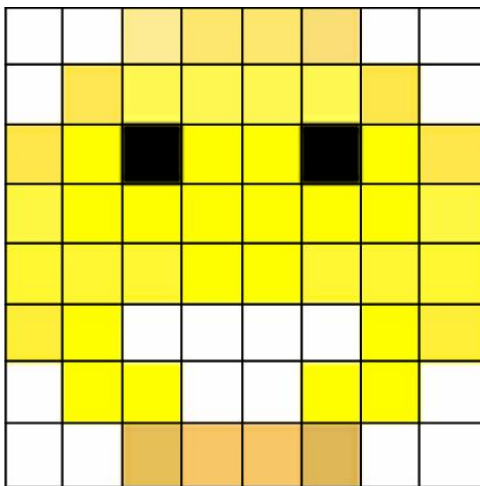


圖 6-1 happy 狀態

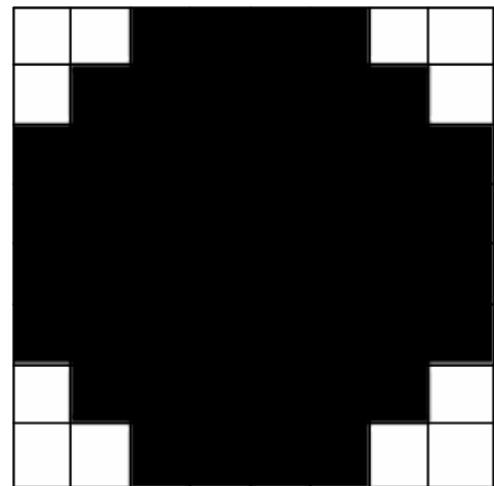


圖 6-2 happy 狀態的 mask 圖示

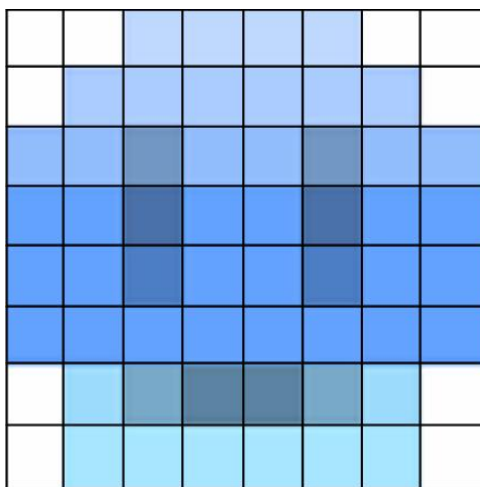


圖 7-1 die 狀態

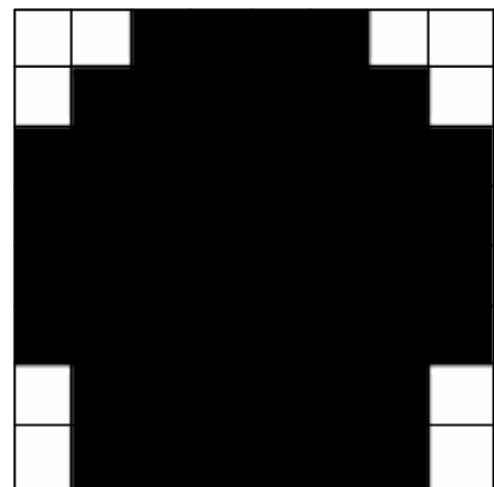


圖 7-1 die 狀態的 mask 圖示

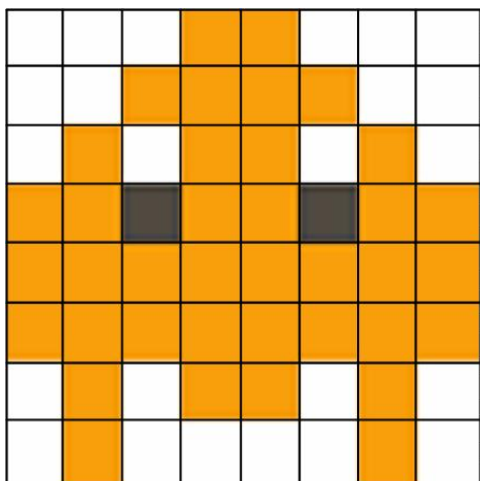


圖 8-1 Ghost 圖示

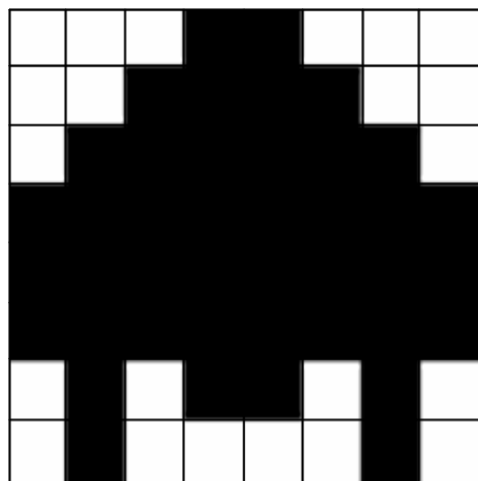


圖 8-2 Ghost 狀態的 mask 圖示

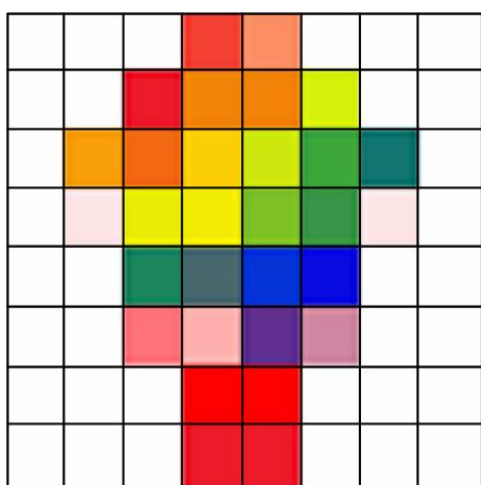


圖 9-1 Candy 圖示

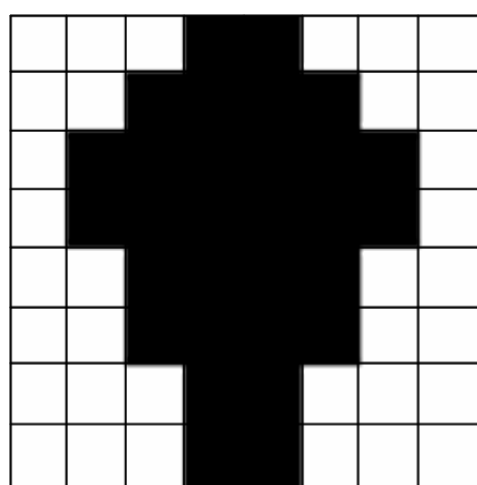


圖 9-2 Candy 狀態的 mask 圖示

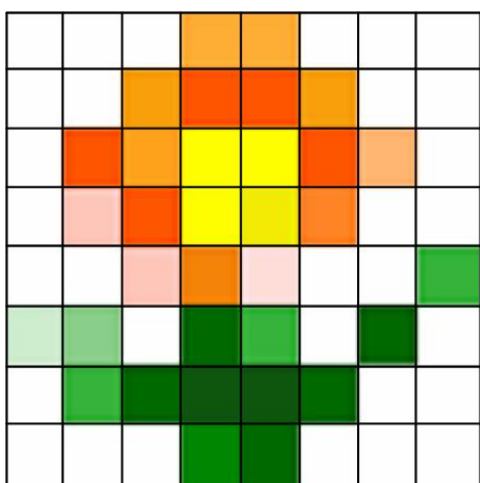


圖 10-1 Flower 圖示

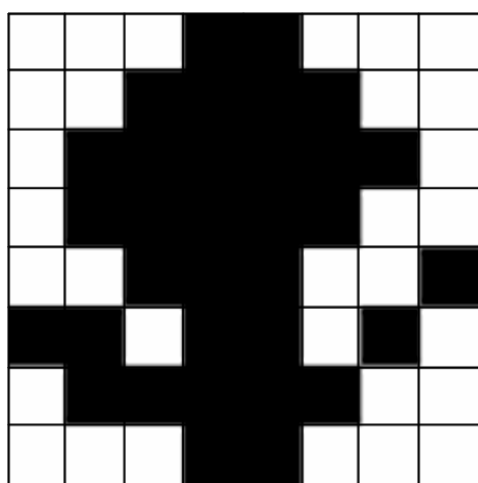


圖 10-2 Flower 狀態的 mask 圖示

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。

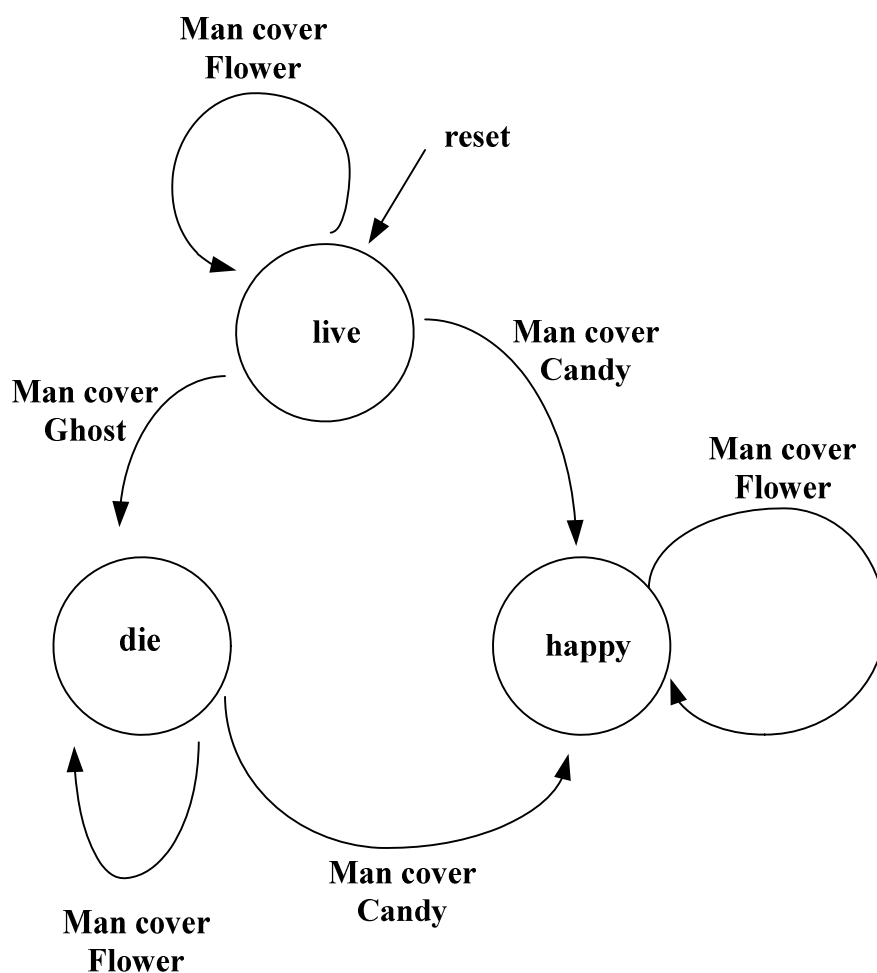


圖 12 狀態圖

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

[illegible]

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： $0 \leq X \leq 56$ ； $0 \leq Y \leq 56$ ，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。

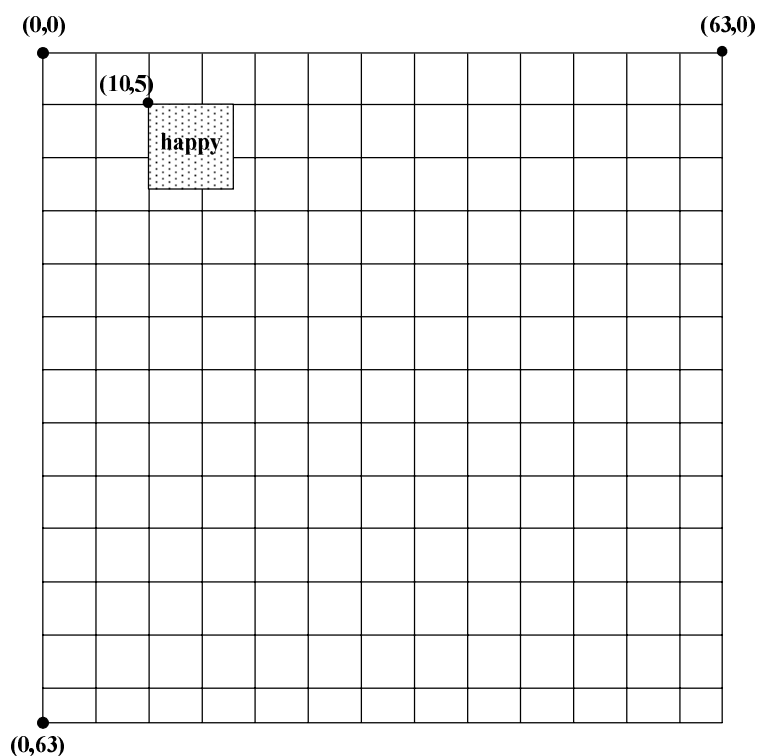


圖 18 實際座標圖

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

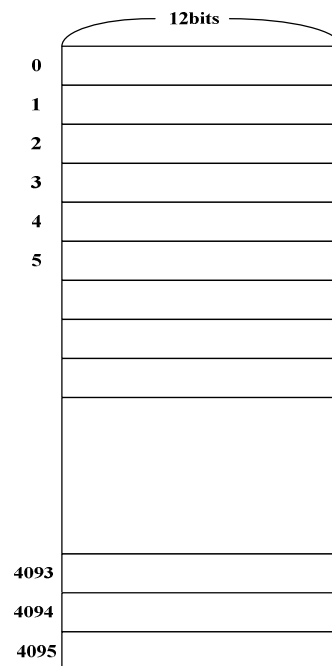
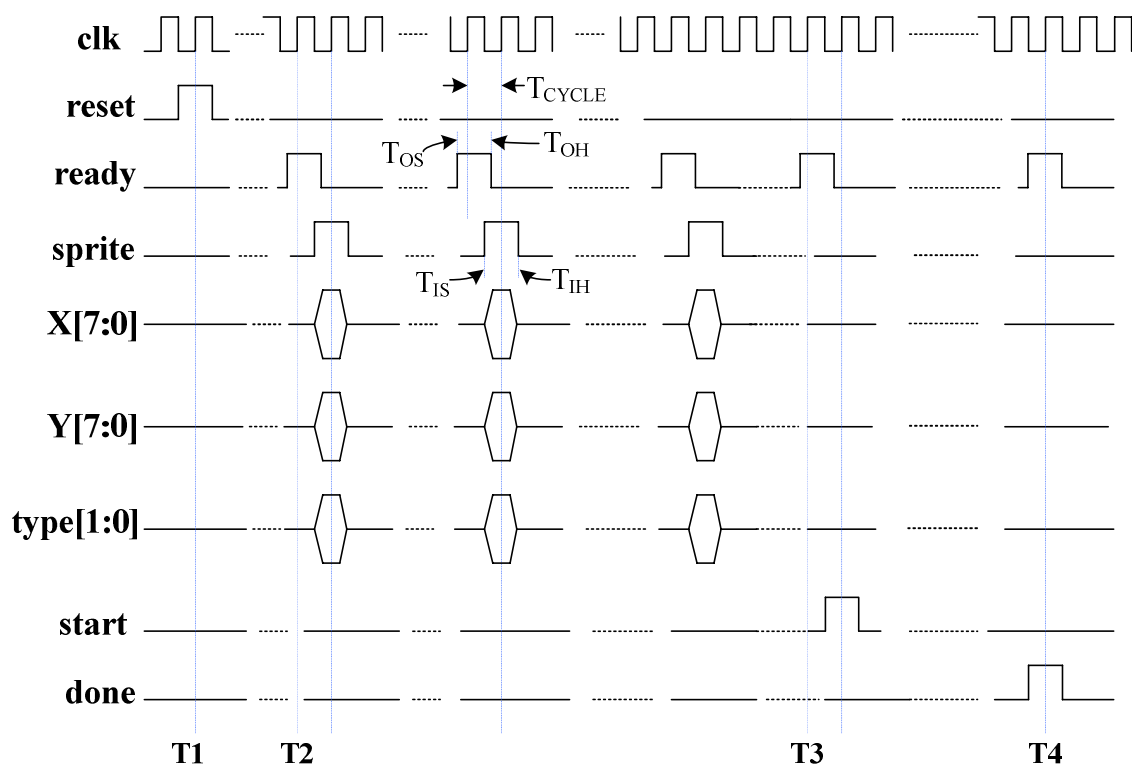


圖 20 Frame Buffer

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



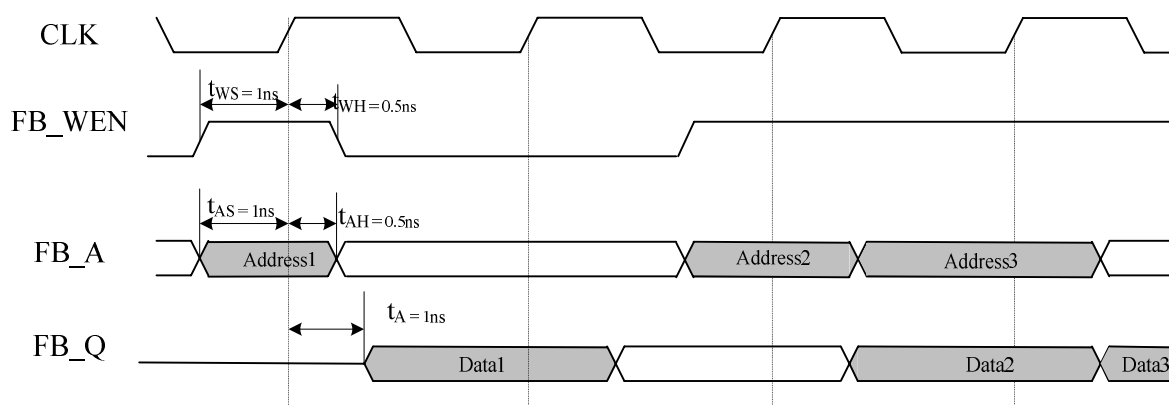
1. T1 時間點，系統啟動送出重置(reset)訊號，SGDE 進行前置工作準備，準備前置工作期間 ready 設為 low。
2. T2 時間點，Host 發現 ready 為 high，表示 SGDE 可以開始接受輸入，host 在 clock 負緣送出第一筆物件資料，包含物件種類(type)及物件座標(X,Y)，送出物件資料的同時，sprite 升為 high 表示此時物件資料有效。SGDE 若無法立刻處理此物件，即將 ready 設為 low。
3. T2 至 T3 之間，Host 以相同方式送出多筆物件資料
4. T3 時間點，物件資料都已送出，此時 ready 為 high，Host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

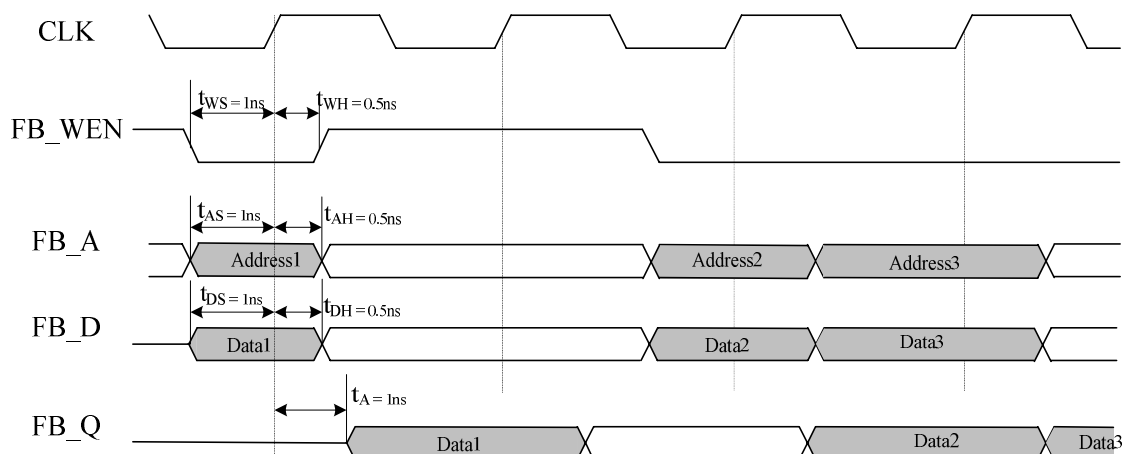
Symbol	Description	Value
T_{cycle}	clock period	user defined
T_{IS}	setup time (time period from valid input signal to positive edge of clock) for all inputs from host.	1/2 clock period
T_{IH}	hold time(time period from positive edge of clock to invalid input signal.) for all inputs from host	1/2 clock 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)



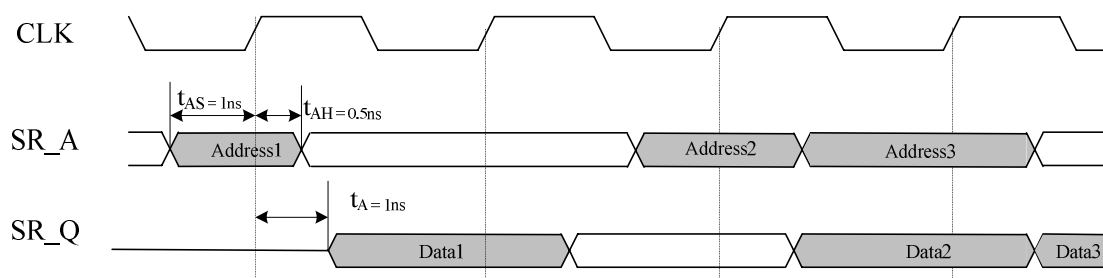
Frame Buffer Write : (FB_CEN=0)



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

評分人員將依照附錄 D 所列之驗證步驟驗證參賽者繳交之設計，通過驗證者以下列 ”完成設計” 所列之公式評分。但若通過完成設計的組數太少，CIC 亦將根據繳交資料依據 ”完成部分設計” 所列分析各組之完成度：

1. 完成設計：

評分標準如下：

$Cost = Area$ if $Time \leq 50000ns$

$Area * Time$ if $Time > 50000ns$

Area：P&R 之後，layout 實際面積大小(單位： um^2)

Time：模擬五個測試樣本後，取模擬時間最長者(單位：ns)

Cost 值越低，所得分數越高。

2. 完成部份設計：對於未通過所有模擬樣本者，CIC 將依據以下原則分析各組 RTL 之完成度。完成度的判定，分成以下幾個等級，等級越高，分數越高。

第一級：遊戲物件的座標位置錯誤。

第二級：遊戲物件的座標位置符合，但是物件本身的方向錯誤。

第三級：物件的座標位置符合，擺放方向正確但是 Man 的狀態及遊戲物件重疊皆錯誤。

第四級：物件的座標位置符合，擺放方向正確，但是 Man 的狀態錯誤或是遊戲物件重疊錯誤。

附錄

在附錄 A 說明主辦單位提供的遊戲物件的 RGB 資訊；附錄 B 為主辦單位所提供各參賽者的設計檔說明；附錄 C 為主辦單位提供的測試樣本；附錄 D 為設計驗證說明；附錄 E 為評分用檔案，亦即參賽者必須繳交的檔案資料；附錄 F 則為設計檔案壓縮整理步驟說明；附錄 G 中說明本次競賽之軟體環境；附錄 H 中說明本次競賽使用之設計資料庫。

附錄 A 遊戲物件 RGB 資訊

主辦單位提供的遊戲物件的 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	544	F90	F90	544	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. 下表為主辦單位所提供各參賽者的設計檔

表 2

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
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_DC.sdc	Design Compiler timing constraint
SGDE_SOCE.sdc	SOC Encounter timing constraint
SGDE_Astro.sdc	Astro timing 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. 比賽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. 請參賽隊伍使用主辦單位所提供的測試樣本(testfixture.v)，來進行設計之模擬驗證。主辦單位除了將利用本試題所提供的測試樣本進行各參賽隊伍之設計測試外，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)**，可由參賽者依需求自行修改。

附錄 C 測試樣本

比賽共提供五組測試樣本，為方便設計者除錯之用，主辦單位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

參賽者繳交資料前應完成 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：參賽者必須進行 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. **VDD 與 VSS 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**，以下分為 Astro 及 SOC Encounter 兩軟體說明 netlist 與 sdf 寫出步驟。
 - i. 使用 Synopsys Astro 者，執行步驟如下：
在 Astro 視窗底下點選

“Timing > SDF Out”

Specify Version	Version 2.1
Operation Mode	Normal SDF
File Name	SGDE.sdf

按 。

“Cell > Hierarchical Verilog Out”

Flattened Cell Name (.EXP .CEL)	SGDE.CEL
Enter File Name	SGDE_apr.v
No power/ground ports	Enable
No power/ground nets	Disable
Output bus as individual bits	Disable
No empty Cell Module Definitions	Enable
No Corner Pad Instances	Enable
No Pad Filler Cell Instances	Enable
No Core Filler Cell Instances	Enable
No Unconnected Cell Instances	Enable
No Unconnected Ports	Enable
Strip BackSlash Before Hierarchy Separator	Enable
No Diode Ports	Enable
Output Wire Declaration	Enable
Output 1'b1 for Power(VDD, vdd, ...) and 1'b0 for Ground(VSS, gnd, ...)	Enable
Generate macro definitions	Disable

按 。

- ii. 使用 Cadence SOC Encounter 者，執行步驟如下：

在 SOC Encounter 視窗下點選：

“Design → Save → Netlist...”

Netlist File	SGDE_apr.v
All other options	Default value

按 。

“Timing → Calculate Delay...”

存成 SGDE_apr.sdf，按 。

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**，以下分為 Astro 及 SOC Encounter 說明執行步驟。

- i. 使用 Synopsys Astro 者，驗證 DRC 與 LVS 步驟如下：

在 Astro 視窗底下點選

“Verify > DRC”

List Error Summary Immediately	Enable
All other options	Default value

按 。

將跳出來的 DRC report 存成 DRC.report 檔。

“Verify > LVS” Default 值，按 。

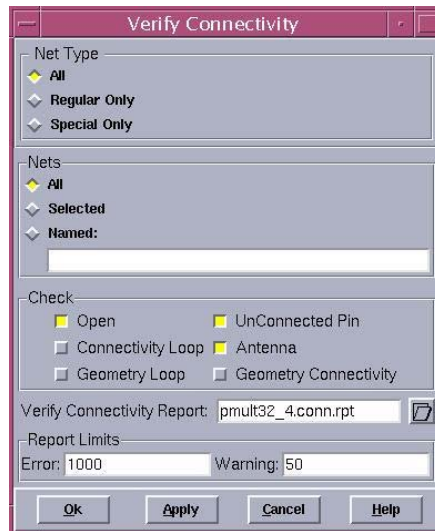
List Error Summary Immediately	Enable
All other options	Default value

將跳出來的 LVS report 存成 LVS.report 檔。

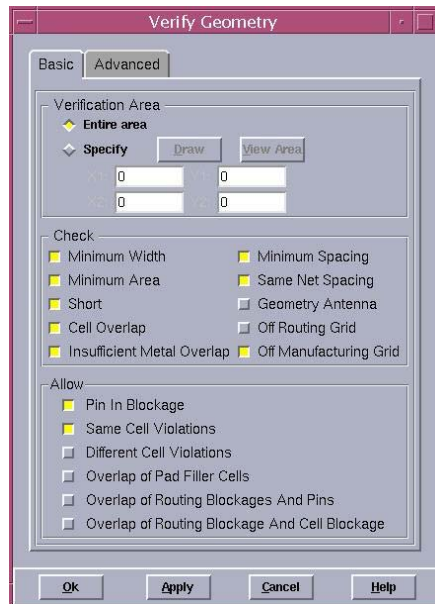
- ii. 使用 Cadence SOC Encounter 者，驗證 DRC 與 LVS 步驟如下：

在 SOC Encounter 視窗下點選

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



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



“Verify → Violation Browser...”

將 Verify 的結果存成 SGDE.viols.rpt

附錄 E 評分用檔案

評分所須檔案可以下幾個部份：(1)RTL design，即各參賽隊伍對該次競賽設計的 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，以免評審進行評分時，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，使用 **Synopsys Astro** 者，請記得將 **Astro** 整個相關的 **design library**，壓縮成一個檔案。使用 **Cadence SOC Encounter** 者，請將 **SOC Encounter** 相關的 **design library**，壓縮成一個檔案。壓縮的檔案格式如下：假設參賽者的 design library 目錄名為“your_lib”，請執行底下的 UNIX 指令，最後可以得到“your_name.tar”的檔案。


```
> tar cvf your_name.tar your_lib
```

在執行以上的指令之前，請確定將你使用的 P&R Tool 儲存後關閉，再執行以上的指令，否則在壓縮的過程會出現錯誤。

表 3

<i>RTL category</i>		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
N/A	N/A	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.vg	Verilog gate-level netlist generated by Synopsys Design Compiler
	*_syn.sdf	Pre-layout gate-level sdf
<i>Physical category</i>		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
P&R	*.tar	archive of the design library directory
	*.gds	GDSII layout
	DRC/LVS report	For Astro : DRC.report ; LVS.report For SOC Encounter : SGDE.viols.rpt
Post-layout Gate-level Simulation	*_pr.vg	Verilog gate-level netlist generated by Cadence SOC Encounter or Synopsys Astro
	*_pr.sdf	Post-layout gate-level sdf

附錄 F 檔案整理步驟

當所有的文件準備齊全如表 3 所列，請按照以下的步驟指令，提交相關設計檔案，需要壓縮於同一個資料夾下，步驟如下：

1. 建立一個新目錄，目錄名稱為組別帳號，如“cb001”。
2. 複製所有需要繳交的檔案放到這個新建立的目錄。
3. 執行以下的 UNIX 指令，當完成這個指令後，會得到“cb001.tar”的檔案。

```
> tar cvf cb001.tar cb001
```

 cb001.tar 表示要產生的檔案，最後的 cb001 表示步驟 1 建立的新目錄。
4. 在自己的 home directory 建立一個新目錄，名稱叫做“**result**” 例如：

```
> mkdir ~/result
```
5. 複製剛剛得到的檔案放到 result 這個目錄。

```
> cp cb001.tar ~/result/
```
6. 在 Design Report Form 中，填入所需的相關資訊。

附錄 G 軟體環境

1. 軟體環境設定檔： /usr/cad/cshrc/env.cshrc
2. 設定軟體環境，請在登入後，開啟 terminal 視窗並依以下步驟執行：

```
cp /usr/cad/cshrc/env.cshrc .cshrc  
source .cshrc
```

3. 此 cshrc 所設定好的軟體環境包括：

NC-Verilog
NC-VHDL
SOC Encounter
Debussy
Laker
ModelSim
Design Vision
Astro
VCS
joe
textedit
nedit
vim
gvim
xv

EDA 軟體所須使用的 license 皆已設定完成，不須額外設定，且每組限定**每個軟體只能使用一套 license**。

附錄 H 設計資料庫

設計資料庫位置： /usr/cad/icc2007/CBDK_IC_Contest_v1.0

目錄架構

Astro/	tsmc13gfsg_fram/ tsmc13_CIC.tf macro.map	Astro core library Astro technology layer mapping file
SOCE/	lef/ tsmc13fsg_8lm_cic.lef antenna_8.lef lib/ fast.lib slow.lib typical.lib streamOut.map	LEF for core cell LEF for antenna best case for core cell worst case for core cell typical case for core cell Layout map for GDSII out
SynopsysDC/	db/ fast.db slow.db typical.db lib/ fast.lib slow.lib typical.lib	Synthesis model (fast) Synthesis model (slow) Synthesis model (typical) timing and power model timing and power model timing and power model
Verilog/	tsmc13_neg.v	Verilog simulation model
VHDL/	tsmc13.vhd	VHDL simulation model

Design Report Form

隊號(Team number):		
<i>RTL category</i>		
<i>Design Stage</i>	<i>Description</i>	<i>File Name</i>
RTL Simulation	使用之 HDL 名稱 (請填入 Verilog 或 VHDL)	
RTL Simulation	RTL 檔案名稱 (RTL Netlist file name)	
<i>Gate-Level category</i>		
<i>Design Stage</i>	<i>Description</i>	<i>File Name</i>
Pre-layout Gate-level Simulation	Gate-Level 檔案名稱 (Gate-Level Netlist file name)	
	Pre-layout sdf 檔案名稱	
	Gate-Level simulation, 所使用最 小的 CYCLE Time	() ns
<i>Physical category</i>		
<i>Design Stage</i>	<i>Description</i>	<i>File Name or Value</i>
P&R	使用之 P&R Tool (請填入 Astro 或 SOC Encounter)	
	設計資料庫檔案名稱(Library name)	
	佈局檔檔案名稱(GDSII file name)	
	佈局面積(layout area)	() um X () um
	佈局座標點	左下角座標點(Lower-Left Coordinate) : XLB = YLB = 右上角座標點(Upper-Right Coordinate): XRT = YRT =
	DRC report file	
	LVS report file	
Post-layout Gate-level Simulation	Post-layout Gate-Level 檔案名稱	
	Post-layout sdf 檔案名稱	
	Gate Level simulation, 所使用最 小的 CYCLE Time	() ns
	Gate Level simulation, 完成所有 運算所需的時間	() ns
其他說明事項(Any other information you want to specify:(如設計特點 ...) 如寫不下可寫於背面		