95 年度大學院校 積體電路設計競賽

標準單元式(Cell-Based)競賽參考解答

研究所/大學組

一、 設計結果報告

二、 暫存器轉換階層(RTL level)設計結果

三、 模擬結果(post layout simulation)

四、 佈局與驗證

一、設計結果報告

Design Report Form

隊號(Team number):CIC			
RTL category			
Design Stage	*Description*	*File Name*	
RTL Simulation	使用之 HDL 名稱 (請填入 Verilog 或 VHDL)	Verilog	
RTL Simulation	RTL 檔案名稱 (RTL Netlist file name)	SGDE.v	
Gate-Level category			
Design Stage	*Description*	*File Name*	
Pre-layout Gate-level Simulation	Gate-Level 檔案名稱 (Gate-Level Netlist file name)	SGDE_syn.vg	
	Pre-layout sdf 檔案名稱	SGDE_syn.sdf	
	Gate-Level simulation, 所使用最小的 CYCLE Time	(7.4) ns	
Physical category			
Design Stage	*Descritpion*	*File Name or Value*	
P&R	使用之 P&R Tool (請填入 Astro 或 SOC Encounter)	SoC Encounter	
	設計資料庫檔案名稱(Library name)	SGDE	
	佈局檔檔案名稱(GDSII file name)	SGDE.gds	
	佈局面積(layout area)	(114.745) um X (109.96) um	
	佈局座標點	左下角座標點(Lower-Left Coordinate) : XLB = 0 YLB = 0 右上角座標點(Upper-Right Coordinate): XRT = 114.745 YRT = 109.96	
	DRC report file	SGDE.geom.rpt	
	LVS report file	SGDE.conn.rpt	
Post-layout Gate-level Simulation	Post-layout Gate-Level 檔案名稱	SGDE_pr.vg	
	Post-layout sdf 檔案名稱	SGDE_pr.sdf	
	Gate Level simulation, 所使用最小的 CYCLE Time	(7.4) ns	
	Gate Level simulation, 完成所有運算所需的時間	(49563) 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

1. 此題目計分方式著重於 area，所以只要將總運算時間限定在 50000ns 之內便可。因此，本電路直接對外部的 memory 來做讀寫，以減少使用 register，進而降低 area 的浪費。
2. 本電路主要是以一個核心的 Finite State Machine 來做全盤的控制，其動作如下：
   a. 進入 initial state(INIT)對 FB 做初始化，寫入底色(12'b1100_1111_0000)。
   b. 讀第一個座標(MAN 的座標)
   c. 讀第二個物件的座標，並把 MAN 的座標存在內部暫存器
   d. 進入 OPER state，此時主要動作是將 SR 的資料輸出來並存入 FB 之中(參考第 3 點)
   e. 當運算完一個物件，便會抓取下一筆座標，再回到 OPER state 進行物件的運算
   f. 當 start 爲 1，也就是其它物件都處理完後，便開始處理 MAN，此時就進入 LAST state
   g. 最後，將 MAN 擺放好後，就可以輸出 done
3. 運算說明：
   a. Candy 及 Ghost 直接由 SR 讀出來，便可以寫入 FB 之中
   b. 由於 Flower 擺放在最下面，所以需要判斷背景顏色是否爲 Candy 或 Ghost，一旦是這兩者，就不能直接寫入 FB
   c. 在運算的過程中，最重要的就是判斷 MAN 的 state，當每個物件從 SR 讀出來時，就要對每一點來做判斷，看看是否有與 MAN 重疊，如果有重疊再來判斷 MAN 的狀態是否需要改變。這邊判斷重疊的方法是直接把 MAN 的 boundary 定出來，只要有其它物件進入此範圍就算重疊
4. 由於本電路主要是靠 FSM，因此爲了避免轉態時產生 glitch，特別使用 gray code 來做編碼。

二、暫存器轉換階層(RTL level)設計結果 (Verilog HDL)

```
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;
// Signals
  // FSM
  reg   [2:0] cstat,
              nstat;
  // output register
  reg          ready,
               done;
  // record coordinates of MAN
  reg   [5:0] man_x,
              man_y;
  // MAN state
  reg   [1:0] mans;
  // 8x8 block counter, b_cnt[6:1] for counter 0~63
  // b_cnt[0] = 0 : for SR/FB read cycle
  // b_cnt[0] = 1 : for RB write cycle
  reg   [6:0] b_cnt;
  // Sprite ROM address
  reg   [8:0] SR_A;
  // Frame RAM address
  reg [11:0] fb_addr;
  // decide flower overlap
  reg         is_flw_n;
  // boundary coordinates of MAN
  wire [5:0] man_x1, man_x2, man_x3, man_x4, man_x5;
  wire [5:0] man_y1, man_y2, man_y3, man_y4, man_y5;
  // flower write mask, avoid flower overlap other object
  wire         flw_mask;
  // temp signal for LAST state
  wire [5:0] X_n, //buffer coordinates of MAN
             Y_n;
  wire [1:0] type_n;   //buffer type of MAN
// Parameters
  // for main state
  parameter INIT = 3'b000, //initial
            MAN1 = 3'b001, //read MAN
            MAN2 = 3'b011, //store MAN
            READ = 3'b010, //read first objec
            OPER = 3'b110, //compare, read/write ...
            NEXT = 3'b111, //read next objec
            LAST = 3'b101, //operater for MAN
            DONE = 3'b100; //finish, send done
  // Object types
  parameter MAN = 2'b00,
            GST = 2'b01,
            CND = 2'b10,
            FLW = 2'b11;
  // MAN state
  parameter LIVE = 2'b00,
            HAPY = 2'b01,
            DIED = 2'b11;
  // Boundary coordinates of MAN
  assign man_x1 = man_x+3'd7;
  assign man_x2 = man_x+3'd6;
  assign man_x3 = man_x+3'd5;
  assign man_x4 = man_x+3'd1;
  assign man_x5 = man_x+3'd2;
  assign man_y1 = man_y+3'd7;
  assign man_y2 = man_y+3'd6;
  assign man_y3 = man_y+3'd5;
  assign man_y4 = man_y+3'd1;
  assign man_y5 = man_y+3'd2;
  // Sprite ROM enable signal
  assign SR_CEN = ((cstat == OPER) || (cstat == LAST)) ? 1'b0 : 1'b1;
  // Frame RAM signals
  assign FB_WEN = ((SR_Q[0] && b_cnt[0] && flw_mask) || (cstat == INIT) || sprite) ? 1'b0 : 1'b1;
  assign FB_CEN = 1'b0;
  assign FB_D = (((cstat != OPER) && (cstat != LAST)) || sprite) ? 12'b1100_1111_0000 : SR_Q[12:1];
  assign FB_A = ((cstat == OPER) || (cstat == LAST)) ? {Y_n + b_cnt[6:4], X_n + b_cnt[3:1]} : fb_addr;
  // flower mask, when object is FLOWER & background pixel not belong to FLOWER
  assign flw_mask = ((type_n == FLW) && ~(is_flw_n)) ? ((FB_Q != 12'hf90) && (FB_Q != 12'hfff) && (FB_Q != 12'h544) && (FB_Q != 
12'hf86) && (FB_Q != 12'hf43) && (FB_Q != 12'he12) && (FB_Q != 12'hc8a) && (FB_Q != 12'hf00)) : 1'b1;
  // internal signal for MAN
  assign type_n = (cstat == LAST) ? MAN : type;
assign X_n      = (cstat == LAST) ? man_x : X;
assign Y_n      = (cstat == LAST) ? man_y : Y;
always @(*)
begin : FSM_Combination
   case (cstat)    //synopsys full_case
      INIT: begin
         ready = 1'b0;
         done   = 1'b0;
         if (FB_A == 12'hffe)
            nstat = MAN1;
         else
            nstat = INIT;
      end
      MAN1: begin
         ready = 1'b1;
         done   = 1'b0;
         nstat = MAN2;
      end
      MAN2: begin
         ready = 1'b1;
         done   = 1'b0;
         nstat = OPER;
      end
      READ: begin
         ready = 1'b0;
         done   = 1'b0;
         if (start)
            nstat = LAST;
         else
            nstat = OPER;
      end
      OPER: begin // compare & write
         ready = 1'b0;
         done   = 1'b0;
         if (b_cnt == 7'b111_1111)
            nstat = NEXT;
         else
            nstat = OPER;
      end
      NEXT: begin
         ready = 1'b1;
         done   = 1'b0;
            nstat = READ;
      end
      LAST: begin    // for MAN
         ready = 1'b0;
         done   = 1'b0;
         if (b_cnt == 7'b111_1111)
            nstat = DONE;
         else
            nstat = LAST;
      end
      DONE: begin
         ready = 1'b0;
         done   = 1'b1;
         nstat = DONE;
      end
      default: begin
         ready = 1'b0;
         done   = 1'b0;
         nstat = INIT;
      end
   endcase
end
always @(posedge clk or posedge reset)
begin : FSM
   if (reset)
      cstat <= INIT;
   else
      cstat <= nstat;
end
always @(posedge clk or posedge reset)
begin : MAN_coordinates
   if (reset) begin
      man_x <= 6'd0;
      man_y <= 6'd0;
   end else if (cstat == MAN2) begin
      man_x <= X;
      man_y <= Y;
   end else begin
      man_x <= man_x;
      man_y <= man_y;
   end
end
always @(posedge clk or posedge reset)
begin : block_counter
   if (reset)
```

```
         b_cnt <= 7'd0;
      else if ((cstat == OPER) || (cstat == LAST))
         b_cnt <= b_cnt + 1;
      else
         b_cnt <= b_cnt;
   end
   always @(*)
   begin : SR_address_generater
      if (reset)
         SR_A <= 9'd0;
      else begin
         case (type_n) //synopsys parallel_case
            MAN:
               if (mans == LIVE)
                  SR_A <= b_cnt[6:1];
               else if (mans == HAPY)
                  SR_A <= 9'h040 + b_cnt[6:1];
               else if (mans == DIED)
                  SR_A <= 9'h080 + b_cnt[6:1];
               else
                  SR_A <= b_cnt[6:1];
            GST:
               SR_A <= 9'h0c0 + b_cnt[6:1];
            CND:
               SR_A <= 9'h100 + b_cnt[6:1];
            FLW:
               SR_A <= 9'h140 + b_cnt[6:1];
         endcase
      end
   end
   always @(posedge clk or posedge reset)
   begin : FB_address_gen
      if (reset)
         fb_addr <= 12'hfff;
      else
         if (cstat == INIT)
            fb_addr <= fb_addr + 1'b1;
         else
            fb_addr <= fb_addr;
   end
   always @(posedge clk or posedge reset)
   begin : flower_overlap_flag
      if (reset)
         is_flw_n <= 1'b0;
      else if ((type == FLW) && (b_cnt[0]) && ((FB_Q == 12'hfa3) || (FB_Q == 12'hf50)))
         is_flw_n <= 1'b1;
      else if (type != FLW)
         is_flw_n <= 1'b0;
      else
         is_flw_n <= is_flw_n;
   end
   always @(posedge clk or posedge reset)
   begin : MAN_state
      if (reset)
         mans <= LIVE;
      else if (cstat == OPER)
         if ((SR_Q[0]) && b_cnt[0])
            if (((FB_A[5:0] >= man_x) && (FB_A[5:0] <= man_x1) && (FB_A[11:6] >= man_y5) && (FB_A[11:6] <= man_y3)) ||
((FB_A[5:0] >= man_x4) && (FB_A[5:0] <= man_x2) && (FB_A[11:6] >= man_y4) && (FB_A[11:6] <= man_y2)) || ((FB_A[5:0] >=
man_x5) && (FB_A[5:0] <= man_x3) && (FB_A[11:6] >= man_y) && (FB_A[11:6] <= man_y1)))
               case (mans)   //synopsys full_case
                  LIVE: begin
                     if (type == CND)
                        mans <= HAPY;
                     else if (type == GST)
                        mans <= DIED;
                     else if (type == FLW)
                        mans <= LIVE;
                  end
                  HAPY: begin
                     mans <= HAPY;
                  end
                  DIED: begin
                     if (type == CND)
                        mans <= HAPY;
                     else if (type == GST)
                        mans <= DIED;
                     else if (type == FLW)
                        mans <= DIED;
                  end
               endcase
      else
         mans <= mans;
   end
endmodule
```

三、模擬結果（Gate-level）

Pattern 1:

```
ncverilog: 06.10-p001: (c) Copyright 1995-2007 Cadence Design Systems, Inc.
TOOL:   ncverilog
ncverilog
        testfixture.v
        SGDE_pr.vg
        ../FB.v
        ../SR.v
        -v
        ../tsmc13.v
        +access+r
Recompiling... reason: file './SGDE_pr.sdf' is newer than expected.

                                            …

Loading snapshot worklib.test:v .................... Done
ncsim> source /usr/cad/cadence/IUS/cur/tools/inca/files/ncsimrc
ncsim> run

==================================================
=  Total Execution Time:            37057 ns  =
==================================================

---------------------------------------------

All data have been generated successfully!

-------------------PASS-------------------

---------------------------------------------

Simulation complete via $finish(1) at time 67377 NS + 0
./testfixture.v:358    $finish;
ncsim> exit
```

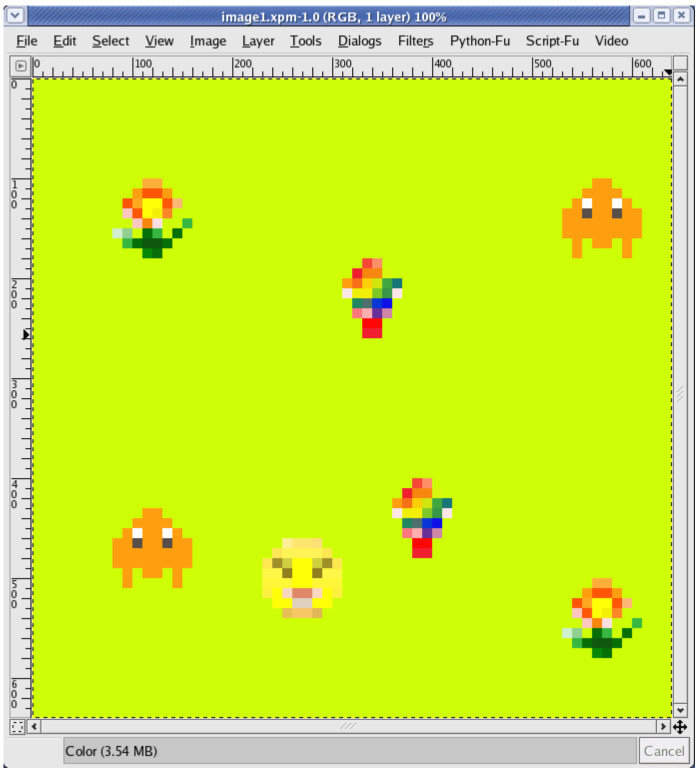


image1.xpm-1.0 (RGB, 1 layer) 100%
File Edit Select View Image Layer Tools Dialogs Filters Python-Fu Script-Fu Video
Color (3.54 MB)
Cancel

Pattern 2:

```
ncverilog: 06.10-p001: (c) Copyright 1995-2007 Cadence Design Systems, Inc.
TOOL:   ncverilog
ncverilog
        testfixture.v
        SGDE_pr.vg
        ../FB.v
        ../SR.v
        -v
        ../tsmc13.v
        +access+r
Recompiling... reason: file './SGDE_pr.sdf' is newer than expected.

                                          …

Loading snapshot worklib.test:v .................... Done
ncsim> source /usr/cad/cadence/IUS/cur/tools/inca/files/ncsimrc
ncsim> run

==================================================
=  Total Execution Time:            38019 ns  =
==================================================

---------------------------------------------

All data have been generated successfully!

-------------------PASS-------------------

---------------------------------------------

Simulation complete via $finish(1) at time 68339 NS + 0
./testfixture.v:358    $finish;
ncsim> exit
```

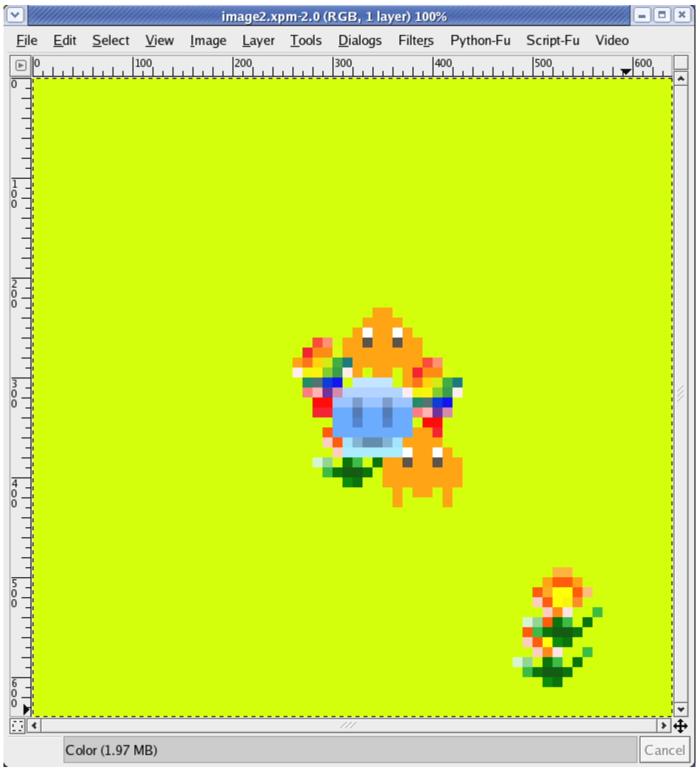


image2.xpm-2.0 (RGB, 1 layer) 100%
File Edit Select View Image Layer Tools Dialogs Filters Python-Fu Script-Fu Video
Color (1.97 MB)
Cancel

Pattern 3:

```
ncverilog: 06.10-p001: (c) Copyright 1995-2007 Cadence Design Systems, Inc.
TOOL:   ncverilog
ncverilog
        testfixture.v
        SGDE_pr.vg
        ../FB.v
        ../SR.v
        -v
        ../tsmc13.v
        +access+r
Recompiling... reason: file './SGDE_pr.sdf' is newer than expected.

                                                …

Loading snapshot worklib.test:v .................... Done
ncsim> source /usr/cad/cadence/IUS/cur/tools/inca/files/ncsimrc
ncsim> run

==================================================
=  Total Execution Time:            39943 ns  =
==================================================

---------------------------------------------

All data have been generated successfully!

-------------------PASS-------------------

---------------------------------------------

Simulation complete via $finish(1) at time 70263 NS + 0
./testfixture.v:358    $finish;
ncsim> exit
```

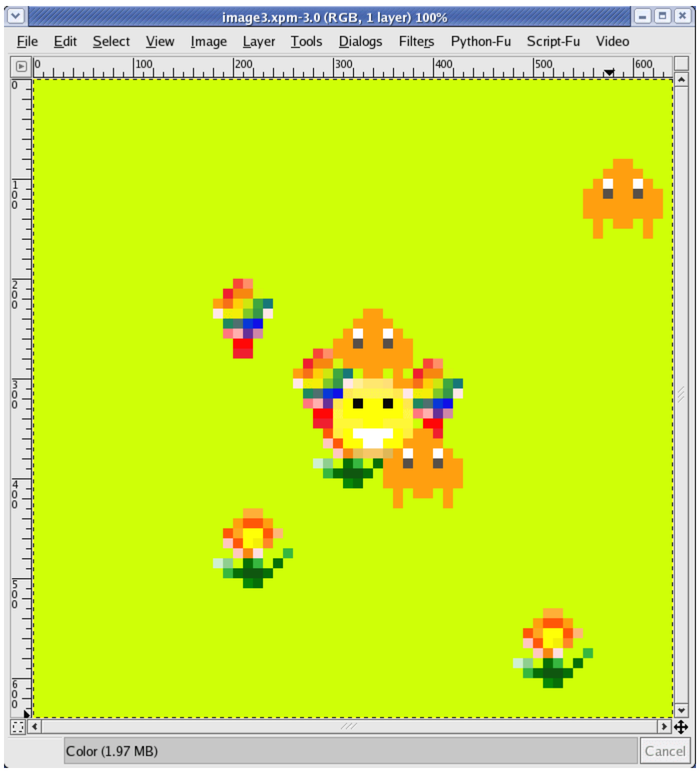


image3.xpm-3.0 (RGB, 1 layer) 100%
File Edit Select View Image Layer Tools Dialogs Filters Python-Fu Script-Fu Video
Color (1.97 MB)
Cancel

Pattern 4:

```
ncverilog: 06.10-p001: (c) Copyright 1995-2007 Cadence Design Systems, Inc.
TOOL:   ncverilog
ncverilog
        testfixture.v
        SGDE_pr.vg
        ../FB.v
        ../SR.v
        -v
        ../tsmc13.v
        +access+r
Recompiling... reason: file './SGDE_pr.sdf' is newer than expected.

                                            …

Loading snapshot worklib.test:v .................... Done
ncsim> source /usr/cad/cadence/IUS/cur/tools/inca/files/ncsimrc
ncsim> run

==================================================
=  Total Execution Time:            49563 ns  =
==================================================

---------------------------------------------

All data have been generated successfully!

-------------------PASS-------------------

---------------------------------------------

Simulation complete via $finish(1) at time 79883 NS + 0
./testfixture.v:358    $finish;
ncsim> exit
```

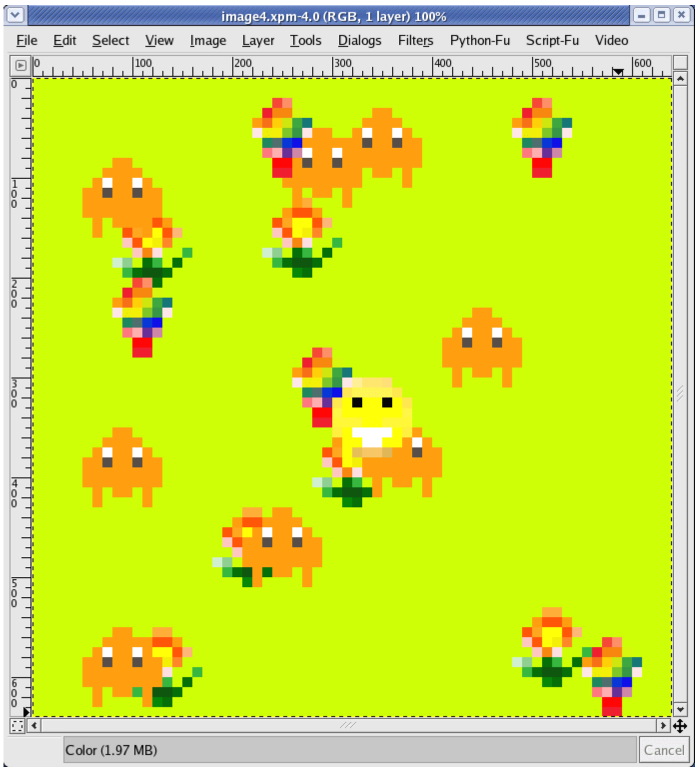


image4.xpm-4.0 (RGB, 1 layer) 100%
File Edit Select View Image Layer Tools Dialogs Filters Python-Fu Script-Fu Video
Color (1.97 MB)
Cancel

Pattern 5:
```
ncverilog: 06.10-p001: (c) Copyright 1995-2007 Cadence Design Systems, Inc.
TOOL:   ncverilog
ncverilog
        testfixture.v
        SGDE_pr.vg
        ../FB.v
        ../SR.v
        -v
        ../tsmc13.v
        +access+r
Recompiling... reason: file './SGDE_pr.sdf' is newer than expected.

                                              …

Loading snapshot worklib.test:v .................... Done
ncsim> source /usr/cad/cadence/IUS/cur/tools/inca/files/ncsimrc
ncsim> run

=================================================
=  Total Execution Time:            49563 ns  =
=================================================

--------------------------------------------

All data have been generated successfully!

-------------------PASS-------------------

--------------------------------------------

Simulation complete via $finish(1) at time 79883 NS + 0
./testfixture.v:358    $finish;
ncsim> exit
```

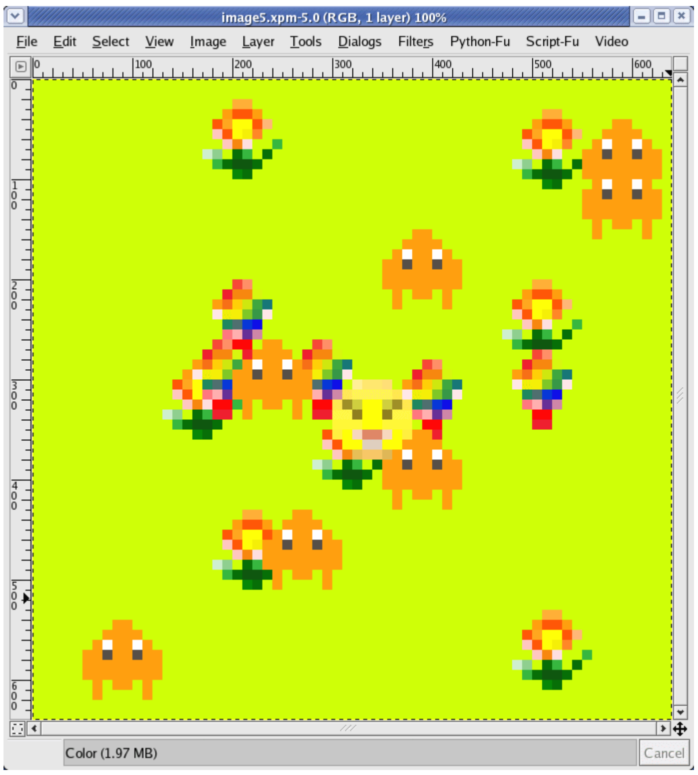
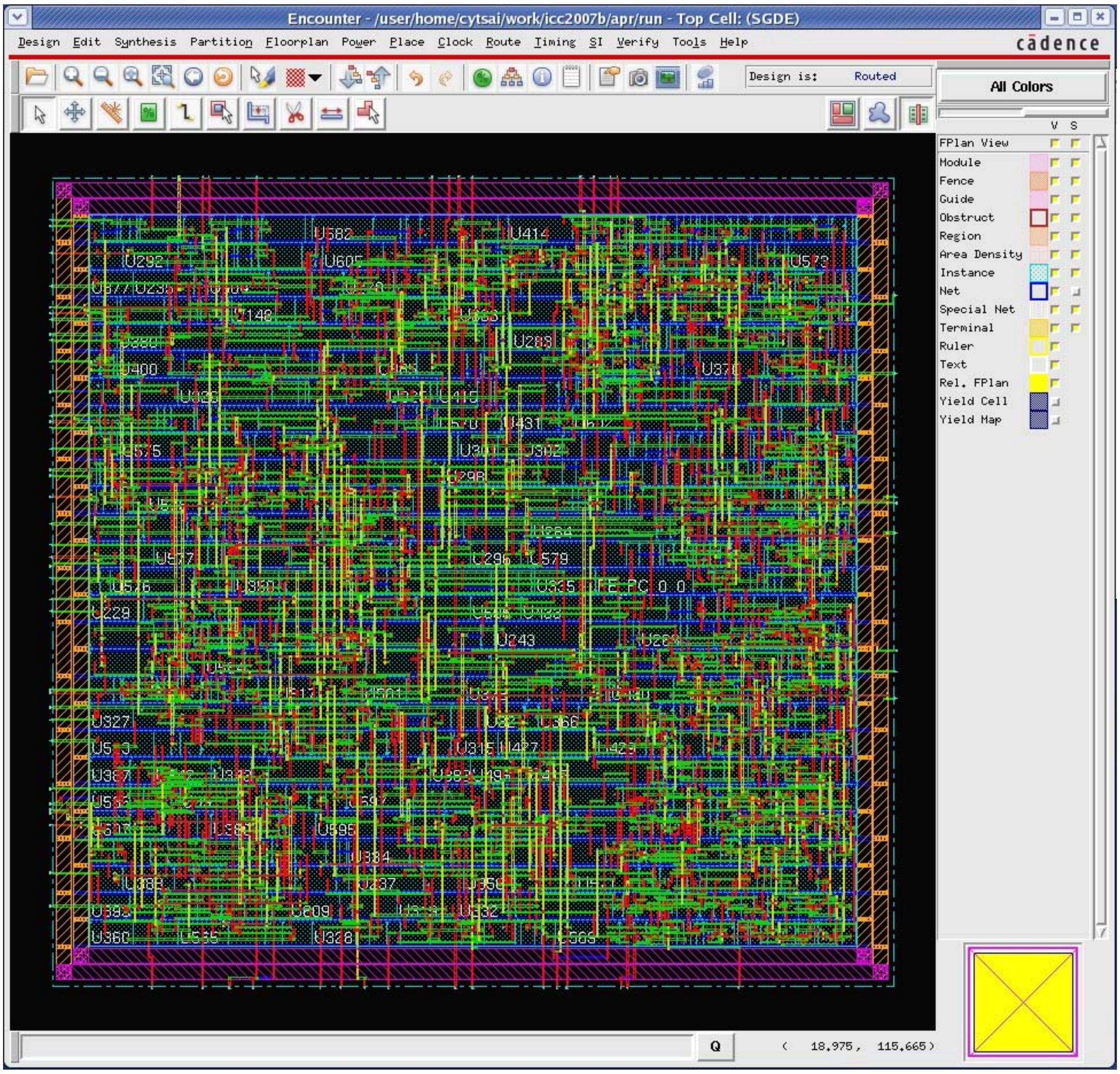


image5.xpm-5.0 (RGB, 1 layer) 100%
File Edit Select View Image Layer Tools Dialogs Filters Python-Fu Script-Fu Video
Color (1.97 MB)
Cancel

四、 佈局與驗證

(a) 佈局圖


Encounter - /user/home/cytsai/work/icc2007b/apr/run - Top Cell: (SGDE)
Design Edit Synthesis Partition Floorplan Power Place Clock Route Timing SI Verify Tools Help
cādence
Design is: Routed
All Colors
FPlan View
Module
Fence
Guide
Obstruct
Region
Area Density
Instance
Net
Special Net
Terminal
Ruler
Text
Rel. FPlan
Yield Cell
Yield Map
Q
(18.975 , 115.665)

(b) on-line DRC/LVS (SoC-Encounter)驗證

DRC 驗證結果：

```
Verify Geometry report created on Fri Aug 10 15:06:27 2007


Begin Summary ...
  Cells      : 0
  SameNet   : 0
  Wiring     : 0
  Antenna    : 0
  Short      : 0
  Overlap    : 0
End Summary

No DRC violations were found
```

LVS 驗證結果：

```
Verify Connectivity report created on Fri Aug 10 15:06:39 2007



Begin Summary
  Found no problems or warnings.
End Summary
```