Introduction to VGA on FPGA

Professor：Terng-Yin Hsu

TA：Yen-Fu Liu

Outline

- VGA
  - Introduction
  - Working Principle
  - Picture Drawing

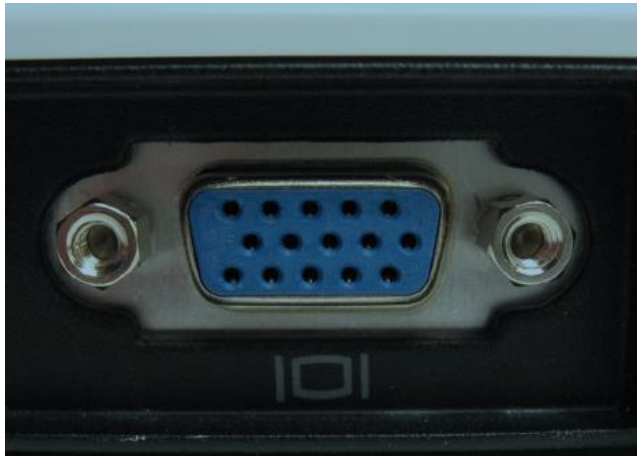
Outline

- VGA
  - Introduction
  - Working Principle
  - Picture Drawing

Introduction

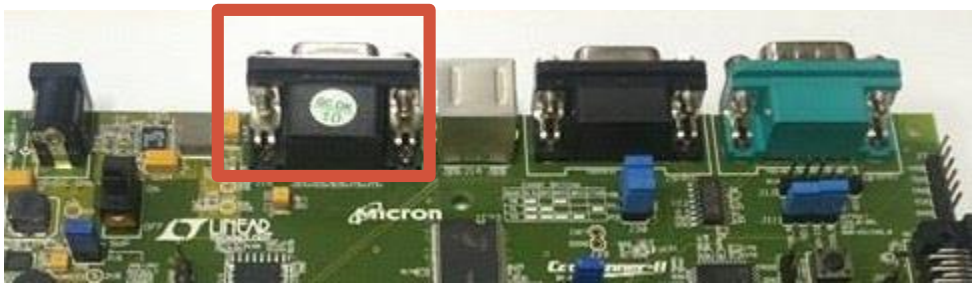
- What is VGA ?
  - VGA (Video Graphics Array)
  - Analog computer display standard marketed in 1987 by IBM

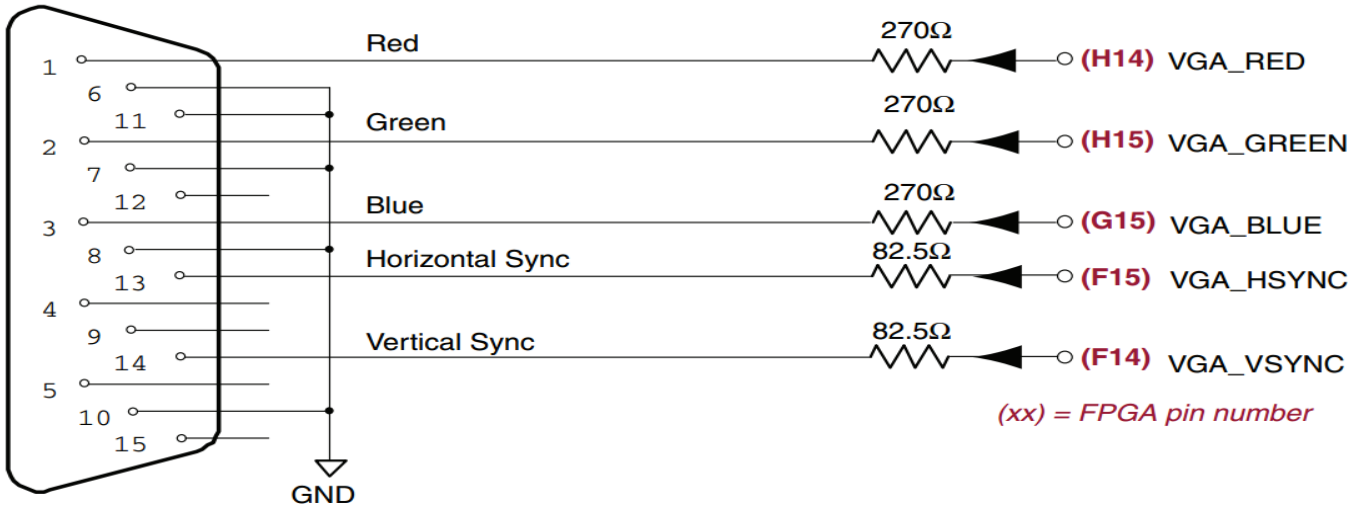
VGA port



VGA plug

VGA Port



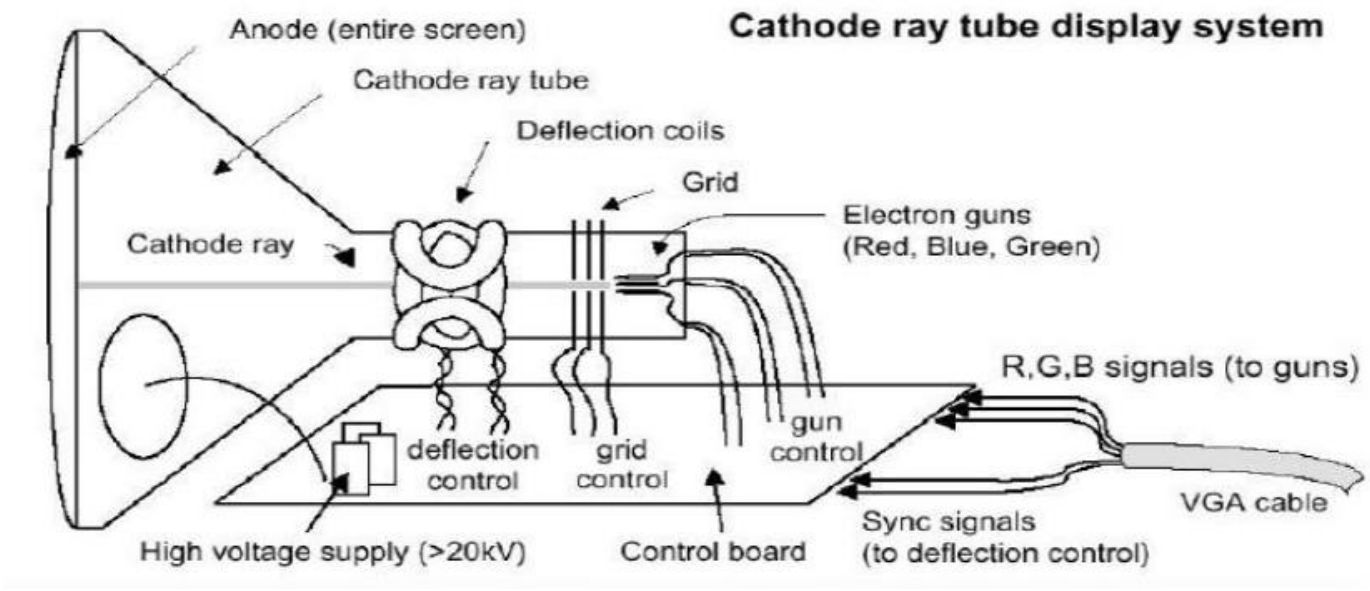

1
6
11
2
7
12
3
8
13
4
9
14
5
10
15
GND
Red
Green
Blue
Horizontal Sync
Vertical Sync
270Ω
270Ω
270Ω
82.5Ω
82.5Ω
(H14) VGA_RED
(H15) VGA_GREEN
(G15) VGA_BLUE
(F15) VGA_HSYNC
(F14) VGA_VSYNC
(xx) = FPGA pin number

Outline

- VGA
  - Introduction
  - Working Principle
  - Picture Drawing

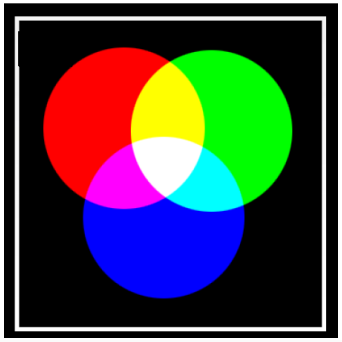
Overview of CRTs

- Cathode Ray Tube
  - Electron gun
  - Deflection coils
  - Anode



Overview of CRTs

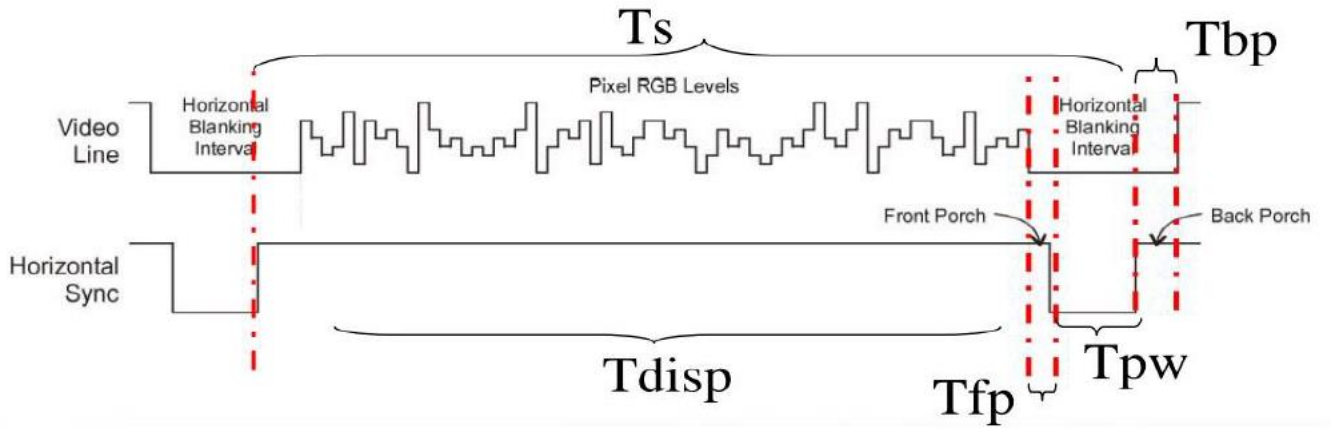
- Illustrates an image on the screen by following
- Signal
  - hsync for horizontal
  - vsync for vertical
  - Red for color red
  - Blue for color blue
  - Green for color green



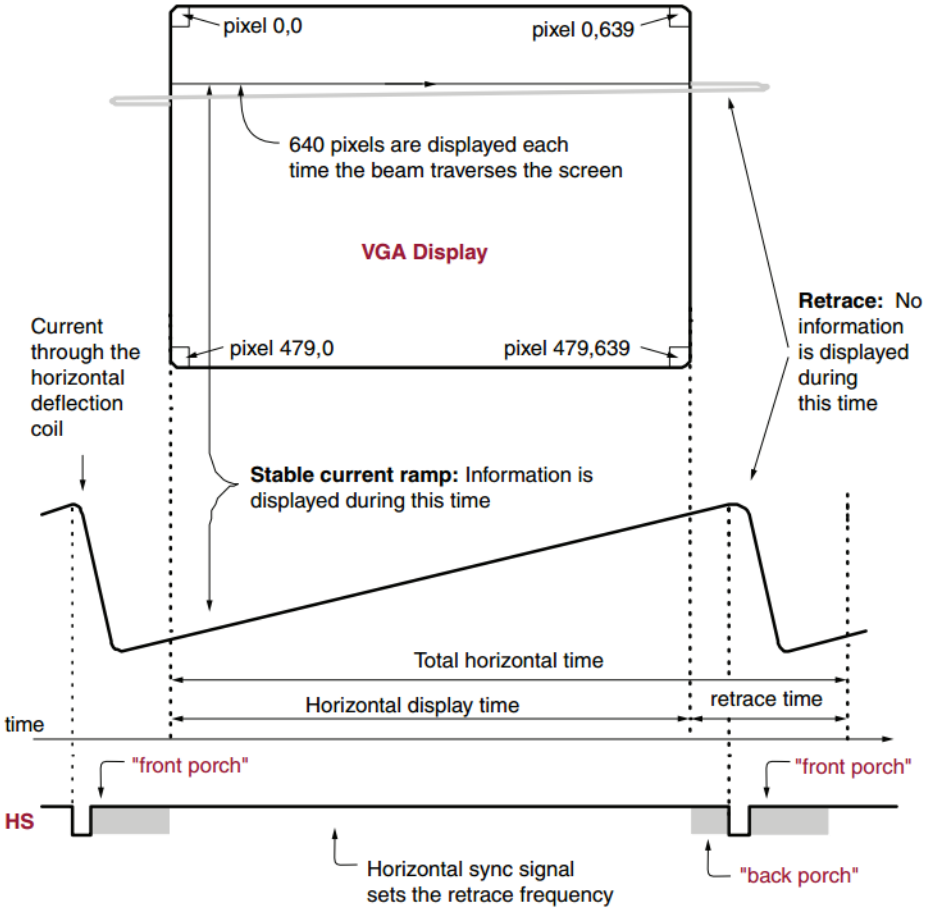
VGA_RED	VGA_GREEN	VGA_BLUE	Resulting Color
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

Signal Timing of VGA

- Two synchronization signals :
  - hsync & vsync
  - Normally active “high” , but at regular times emit “low” synchronization pulse.
  - When horizontal blanking interval no video data transferred
  - Horizontal blanking interval used by CRT monitor to realign the electron gun in preparation for the next scan line.



CRT Display Timing Example



Code

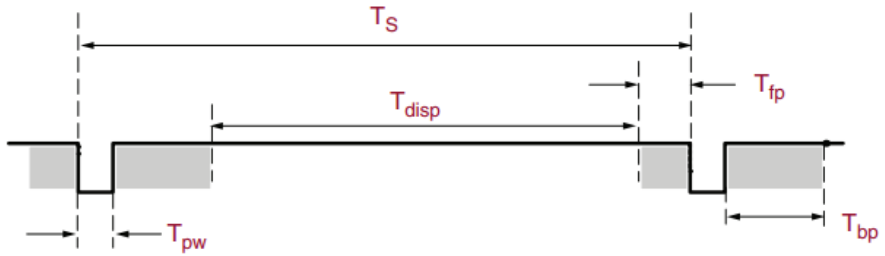
```
always @ ( posedge CLK ) begin                          //Column
    if ( RESET ) pixel_col <= 0;
    else if ( pixel_col == 1039 ) pixel_col <= 0;
    else pixel_col <= pixel_col + 1;
end

always @ ( posedge CLK) begin                           //Row
    if ( RESET ) pixel_row <= 0;
    else if ( pixel_row == 665 ) pixel_row <= 0;
    else if (  pixel_col == 1039 ) pixel_row <= pixel_row + 1;
    else pixel_row <= pixel_row;
end

assign hsync    = ~( (pixel_col >= 919) & (pixel_col < 1039) );
assign vsync    = ~( (pixel_row >= 659) & (pixel_row < 665) );
assign visible  =  ( (pixel_col >= 104) & (pixel_col < 904) & (pixel_row >= 23) & (pixel_row < 623));
```

VGA Signal Timing

- Synchronization period for 800x600



Horizontal Synchronization

Symbol	Name	Number of 50Mhz clock periods	Reference index
Tbp	Back Porch	104	103
Tdisp	Display Time	800	903
Tfp	Front Porch	16	919
Tpw	Pulse Width	120	1039
Ts	Sync pulse time	1040	1039

Vertical Synchronization

Symbol	Name	Number of 50Mhz clock periods	Reference index
Tbp	Back Porch	23*1040	22
Tdisp	Display Time	600*1040	622
Tfp	Front Porch	37*1040	659
Tpw	Pulse Width	6*1040	665
Ts	Sync pulse time	666*1040	665

Outline

- VGA
  - Introduction
  - Working Principle
  - Picture Drawing

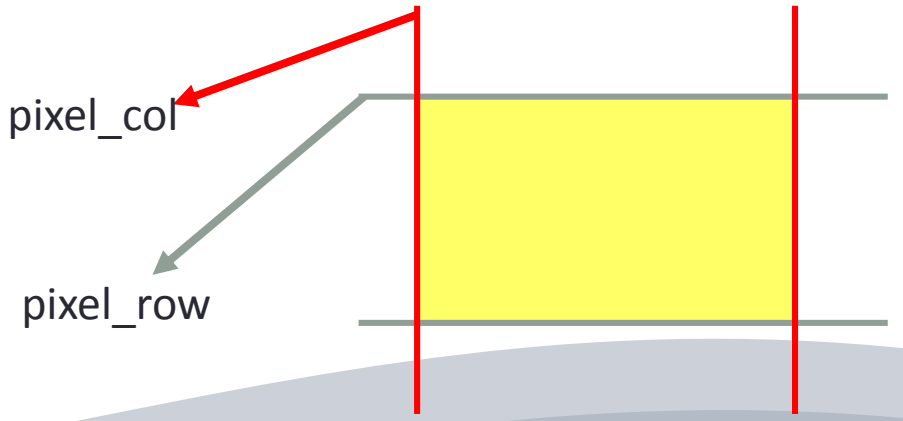
Picture Drawing

- Mathematic Function Description
- Picture Map

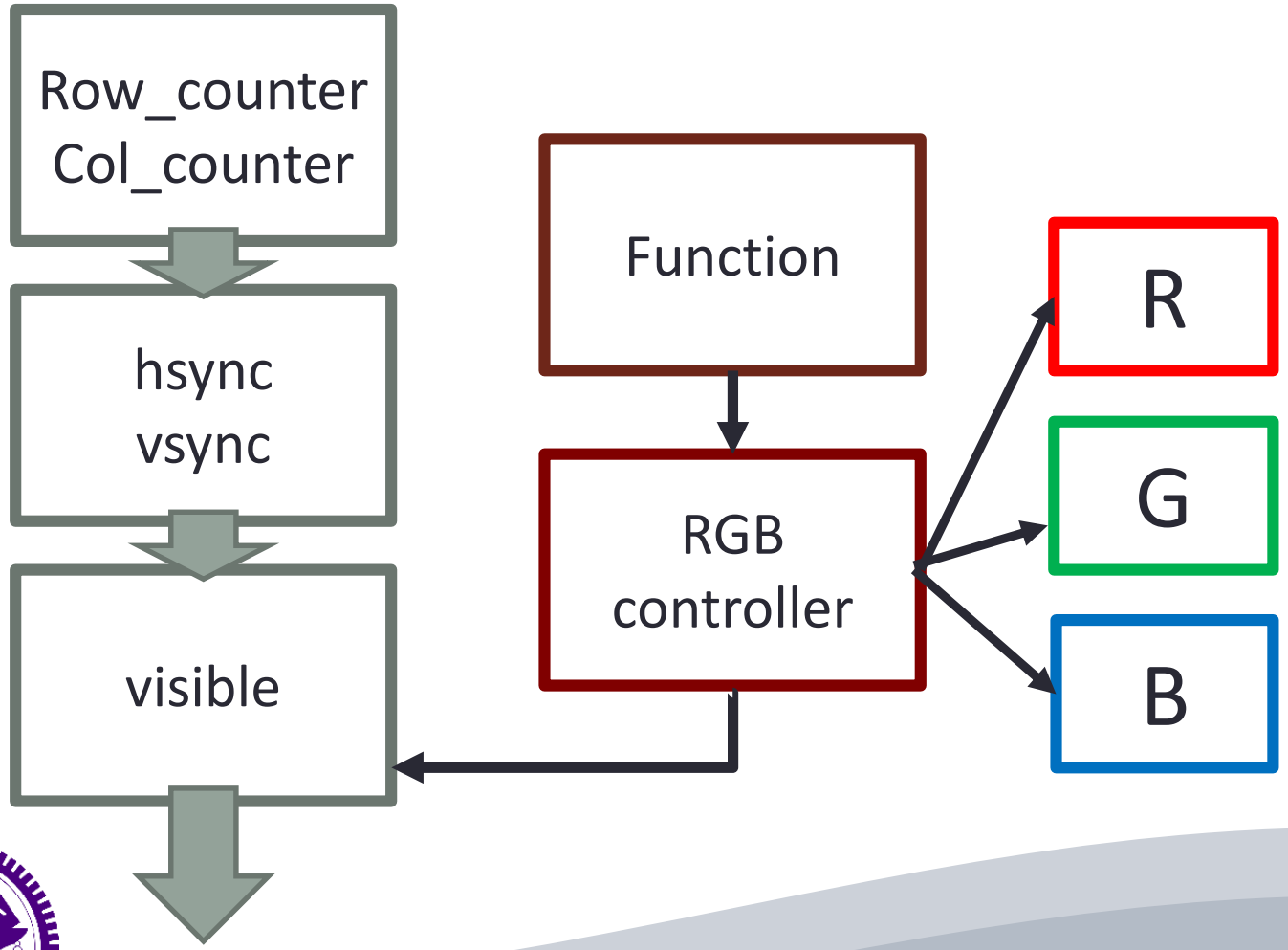
Mathematic Function Description

- You can use the mathematics function
- For example : Rectangle

```
always@(posedge CLK) begin
    if(RESET)
        rectangle<=0;
    else if(
        (pixel_col-center_x>=center_x) & (pixel_col<center_x+100) &
        (pixel_row>=center_y) & (pixel_row<center_y+100)
    )
        rectangle<=1;
    else
        rectangle<=0;
end
```



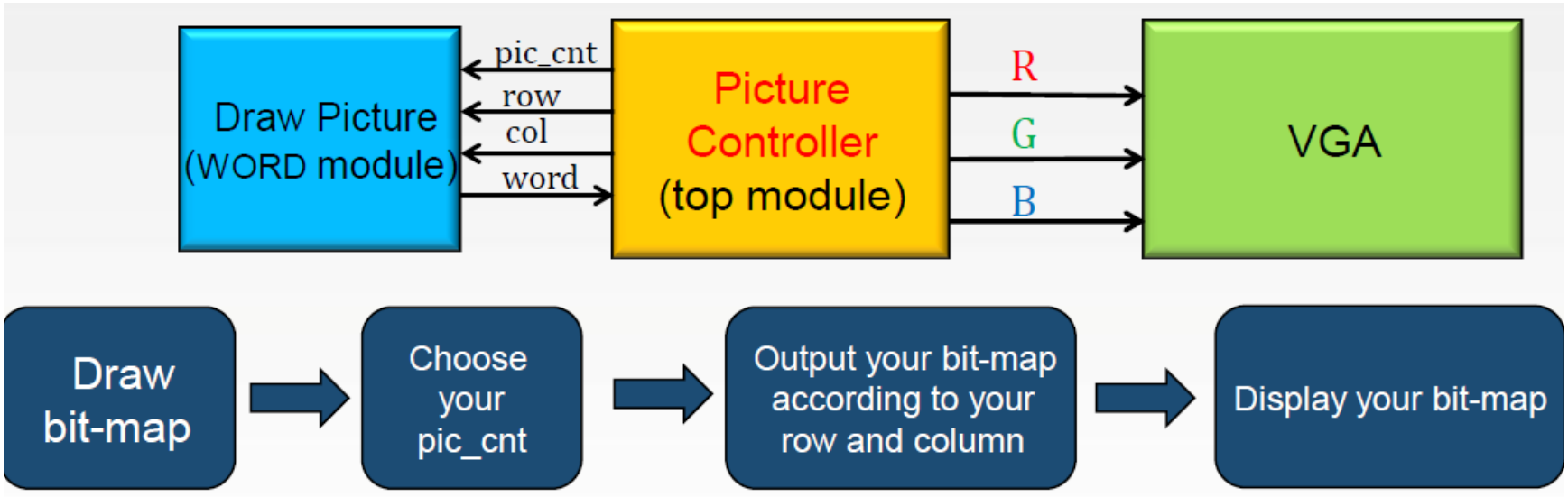
Architecture



Picture Map Architecture

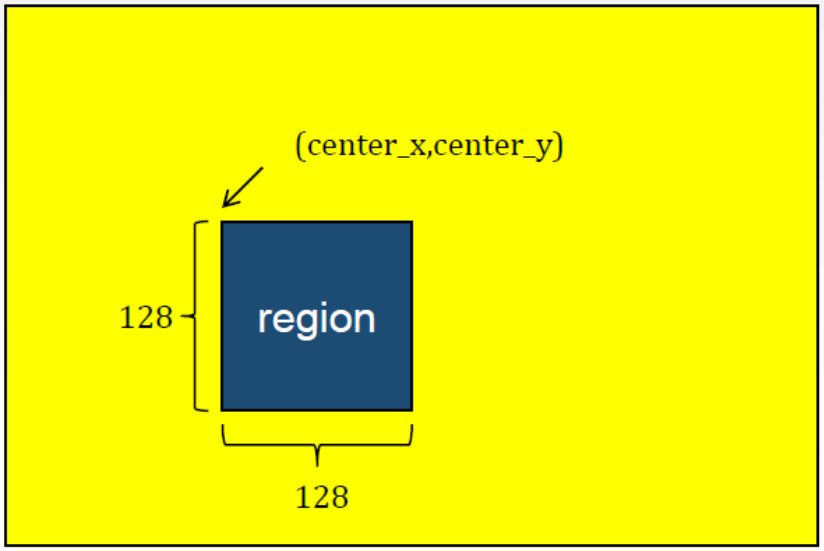
- Using bit-map to draw a picture .
- Animation is composed of many pictures .

Architecture



Region

- You must to set your region .
- If you don’t set region, your screen would be recognized as a region and your bit-map would be duplicated to fill out the full screen.




```
assign region = ( pixel_col >= center_x ) & ( pixel_col < center_x + 128 ) &
                ( pixel_row >= center_y ) & ( pixel_row < center_y + 128 );
```

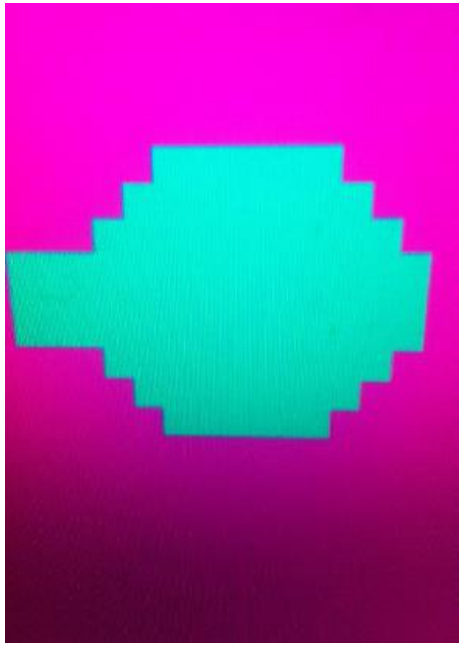
Picture

- Draw a picture .

```
always@( pic_cnt1 )begin
    case( pic_cnt1 )
        0: begin
            line_a <= 16'b0000000000000000;
            line_b <= 16'b0000000000000000;
            line_c <= 16'b0000000000000000;
            line_d <= 16'b0000000000000000;
            line_e <= 16'b0000011111100000;
            line_f <= 16'b0000111111110000;
            line_g <= 16'b0001111111111000;
            line_h <= 16'b1111111111111100;
            line_i <= 16'b1111111111111100;
            line_j <= 16'b1111111111111100;
            line_k <= 16'b0001111111111000;
            line_l <= 16'b0000111111110000;
            line_m <= 16'b0000011111100000;
            line_n <= 16'b0000000000000000;
            line_o <= 16'b0000000000000000;
            line_p <= 16'b0000000000000000;
           end
    endcase
end
```

```
case( { row, col } )
    8'b0000_0000: word <= line_a[15];
    8'b0000_0001: word <= line_a[14];
    8'b0000_0010: word <= line_a[13];
    8'b0000_0011: word <= line_a[12];
    8'b0000_0100: word <= line_a[11];
    8'b0000_0101: word <= line_a[10];
    8'b0000_0110: word <= line_a[9];
    8'b0000_0111: word <= line_a[8];
    8'b0000_1000: word <= line_a[7];
    8'b0000_1001: word <= line_a[6];
    8'b0000_1010: word <= line_a[5];
    8'b0000_1011: word <= line_a[4];
    8'b0000_1100: word <= line_a[3];
    8'b0000_1101: word <= line_a[2];
    8'b0000_1110: word <= line_a[1];
    8'b0000_1111: word <= line_a[0];

    8'b0001_0000: word <= line_b[15];
    8'b0001_0001: word <= line_b[14];
    8'b0001_0010: word <= line_b[13];
    8'b0001_0011: word <= line_b[12];
    8'b0001_0100: word <= line_b[11];
    8'b0001_0101: word <= line_b[10];
    8'b0001_0110: word <= line_b[9];
    8'b0001_0111: word <= line_b[8];
    8'b0001_1000: word <= line_b[7];
    8'b0001_1001: word <= line_b[6];
    8'b0001_1010: word <= line_b[5];
    8'b0001_1011: word <= line_b[4];
    8'b0001_1100: word <= line_b[3];
    8'b0001_1101: word <= line_b[2];
    8'b0001_1110: word <= line_b[1];
    8'b0001_1111: word <= line_b[0];
```



Thank for your attention