

```
timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2016/03/28 22:19:17
// Design Name:
// Module Name: vga
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////

(* use_dsp48 = "yes" *) module vga(
    input clk,
    input clk1,
    input rst,
    output reg hs,
```

```

output reg vs,
output reg [3:0] r,
output reg[3:0] g,
output reg[3:0] b
);

//maximum value for the horizontal pixel counter
parameter HMAX=10'b1100100000;    // 800
//maximum value for the vertical pixel counter
parameter VMAX=10'b1000001101;    // 525
//total number of visible columns
parameter HLines=10'b1010000000; // 640
// value for the horizontal counter where front porch ends
parameter HFP=10'b1010010000;    // 648
//value for the horizontal counter where the synch pulse ends
parameter HSP=10'b1011110000;    // 744
//total number of visible lines
parameter VLines=10'b0111100000; // 480
// value for the vertical counter where the front porch ends
parameter VFP=10'b0111101010;    // 482
//value for the vertical counter where the synch pulse ends
parameter VSP=10'b0111101100;    //484
//polarity of the horizontal and vertical synch pulse
// only one polarity used, because for this resolution they
coincide.

parameter SPP=1'b0;
parameter pix_width=240;
parameter pix_hight=320;
// horizontal and vertical counters
reg [9:0] hcounter=10'b0000000000;

```

```

reg [9:0] vcounter=10'b0000000000;

// active when inside visible screen area.
wire video_enable ;
reg vidon;
wire [9:0] x_center;
(*rom_style="block"*) reg [11:0] pix_memory
[0:pix_width*pix_hight-1];
initial
begin
    $readmemh("pic240x320.txt", pix_memory);
end
vga_center Inst_vga_center(.clk(clk1),.x_center(x_center));
// increment horizontal counter at pixel_clk rate
// until HMAX is reached, then reset and keep counting
always @(posedge clk or posedge rst)
begin
    if(rst)
        hcounter<=10'b0000000000;
    else
        if(hcounter == HMAX)
            hcounter<=10'b0000000000;
        else
            hcounter <= hcounter+1;
end

//increment vertical counter when one line is finished
//(horizontal counter reached HMAX)
//until VMAX is reached, then reset and keep counting

```

```

always @(posedge clk or posedge rst)
begin
if(rst)
    vcounter<=10'b00000000000;
else
    if(hcounter == HMAX)
        if(vcounter == VMAX)
            vcounter<=10'b00000000000;
        else
            vcounter <= vcounter + 1;
end

//The HS is active (with polarity SPP) for a total of 96 pixels.
always@(posedge clk)
begin
    if((hcounter >= HFP) && (hcounter < HSP))
        hs<=SPP;
    else
        hs<=~SPP;
end

// The VS is active (with polarity SPP) for a total of 2 video
lines
// = 2*HMAX = 1600 pixels.
always@(posedge clk)
begin
    if((vcounter >= VFP) && (vcounter < VSP))
        vs <=SPP;
    else
        vs <=~SPP;
end

```

```
end
```

```
//enable video output when pixel is in visible area
```

```
assign video_enable=((hcounter < H_LINES) && (vcounter < V_LINES))?
```

```
1'b1: 1'b0;
```

```
always@(posedge clk)
```

```
begin
```

```
    vidon <= ~video_enable;
```

```
end
```

```
/*always @(hcounter)
```

```
begin
```

```
    if(hcounter<=x_center)
```

```
        x_zone=x_center-hcounter;
```

```
    else
```

```
        x_zone=hcounter-x_center;
```

```
end
```

```
always@(vcounter)
```

```
begin
```

```
    if(vcounter<=y_center)
```

```
        y_zone=y_center-vcounter;
```

```
    else
```

```
        y_zone=vcounter-y_center;
```

```
end
```

```
always @(*)
```

```
begin
```

```
    if(vidon==1'b0)
```

```

if((x_zone*x_zone+y_zone*y_zone)<=10000)
    color=12'b1100000000000;
else
    color=12'b0001110000000;
else
    color=12'b0000000000000;
end*/
//assign rom_address=(vcounter-100)*240+hcounter;
always @(*)
begin

if(vcounter>=0 && vcounter<320 && hcounter>=x_center &&
hcounter<x_center+240)
    begin
        r=pix_memory[vcounter*240+hcounter-x_center][11:8];
        g=pix_memory[vcounter*240+hcounter-x_center][7:4];
        b=pix_memory[vcounter*240+hcounter-x_center][3:0];
    end
else
    begin
        r=4'b0000;
        g=4'b0000;
        b=4'b0000;
    end
end
endmodule

```