

VGA controller

```
// VGA controller
module vga_sync(CLK25, reset, in_r, in_g, in_b,
               px, py, vga_r, vga_g, vga_b, hsync, vsync, vga_sync, vga_blank, vga_clk);
    input CLK25; // 25MHz clock
    input reset;
    input [9:0] in_r, in_g, in_b;
    output reg [9:0] px, py; // pixel_x, pixel_y location
    output [9:0] vga_r, vga_g, vga_b;
    output reg hsync, vsync;
    output vga_sync, vga_blank;
    output vga_clk;

    reg video_on;
    reg [9:0] hcount, vcount;
```

```
// Horizontal Sync -----
// hcount      0          640          640      20      659      95      755      45      799
// width

// hcount : mod-800 counter
always @(posedge CLK25 or posedge reset) begin
    if(reset) hcount <= 0;
    else begin
        if(hcount==799) hcount <= 0;
        else hcount <= hcount + 1;
    end
end

// hsync
always @(posedge CLK25) begin
    if((hcount >= 659) && (hcount <= 755)) hsync <= 0;
    else hsync <= 1;
end
```

```
// Vertical Sync -----
// vcount      0          480          480      13      493      2      494      30      524
// width

// vcount : mod-525 counter
always @(posedge CLK25 or posedge reset) begin
    if(reset)
        vcount <= 0;
    else if (hcount==799) begin
        if(vcount == 524) vcount <= 0;
        else vcount <= vcount + 1;
    end
end

// vsync
always @(posedge CLK25) begin
    if((vcount >= 493) && (vcount <= 494)) vsync <= 0;
    else vsync <= 1;
end
```

```
// video_on_h
always @(posedge CLK25) begin
    video_on <= (hcount <= 639) && (vcount <= 479);
    px <= hcount;
    py <= vcount;
end

assign vga_clk = ~CLK25;
assign vga_blank = hsync & vsync;
assign vga_sync = 1'b0;

assign vga_r = video_on ? in_r : 10'h000;
assign vga_g = video_on ? in_g : 10'h000;
assign vga_b = video_on ? in_b : 10'h000;

endmodule // End of vga_sync module
```

Top module

```
assign reset = ~iKEY[3];

always @(posedge iCLK_50 or posedge reset) /* generate 25MHz clock */
    if (reset) CLK25 <= 0;
    else CLK25 <= ~CLK25;

vga_sync u1 (CLK25, reset, in_r, in_g, in_b,
            px, py, vga_r, vga_g, vga_b, hsync, vsync, vga_sync, vga_blank, vga_clock);

always @* begin
    oVGA_CLOCK = vga_clock;
    {oVGA_R, oVGA_G, oVGA_B} = {vga_r, vga_g, vga_b};
    {oVGA_HS, oVGA_VS} = {hsync, vsync};
    oVGA_SYNC_N = vga_sync;
    oVGA_BLANK_N = vga_blank;
end
```

```
/* DEMO 1 - color selected by three switches */
assign {sw_r, sw_g, sw_b} = iSW[2:0]; // R, G, B input switch
assign in_r = (sw_r) ? 10'h3ff : 10'h0;
assign in_g = (sw_g) ? 10'h3ff : 10'h0;
assign in_b = (sw_b) ? 10'h3ff : 10'h0;

/* DEMO 2 - three color horizontal bar */
assign in_r = (py < 160) ? 10'h3ff : 10'h0;
assign in_g = (py >= 160 && py < 320) ? 10'h3ff : 10'h0;
assign in_b = (py >= 320) ? 10'h3ff : 10'h0;
```