

## Basys3 VGA 显示控制原理及参考代码

### 实验原理

VGA 控制器是一个控制视频显示的模块，主要由五个信号来控制，分别是 R、G、B、HS 和 VS。其中 R、G、B 分别用来驱动显示器三个基色的显示，即红、绿和蓝，HS 是行同步信号，VS 是场同步信号。

图像的显示从屏幕的左上角开始，并从左到右、上到下依次逐行扫描显示，最终抵达屏幕的右下角（即为一帧）。每扫完一行，电子束回到左边下一行的开始位置。期间对电子束进行行消隐。并在每行结束时，用行同步信号对行进行同步，扫描完所有行后，再用场同步信号对场进行同步，并使电子束回到屏幕的左上方，同时对场进行消隐，并预备下一次扫描。VGA 时序控制如图 1 所示，FPGA 只要能按标准准确发送这些信号到 VGA 接口，就可以实现对 VGA 图像显示的控制。

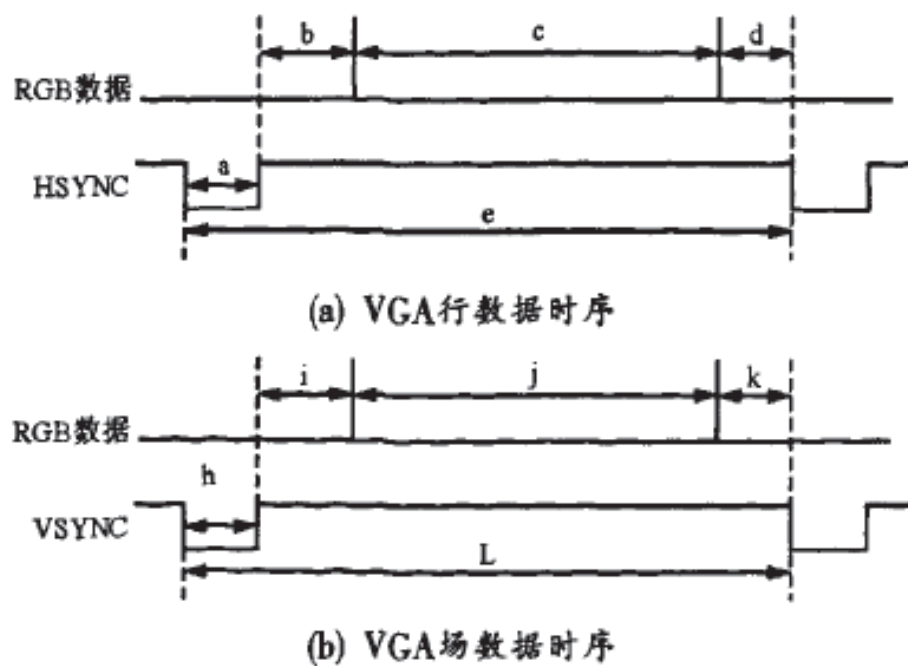


图 1 VGA 时序图

产生行同步脉冲信号 HSYNC 的周期  $e=a+b+c+d$ 。其中  $a$  为同步信号时间， $b$  为行消隐后肩时间， $c$  为 RGB 数据显示有效时间， $d$  为行消隐前肩。从图 1(b) 可以看出，场同步时序与行同步时序类似。也是显示一帧数据的时序。这里以行为单位，场同步脉冲信号 VSYNC 的周期  $L=h+i+j+k$ 。其中  $h$  为同步信号时间， $i$  为场消隐后肩时间， $j$  为数据有效时间， $k$  为场消隐前肩时间。

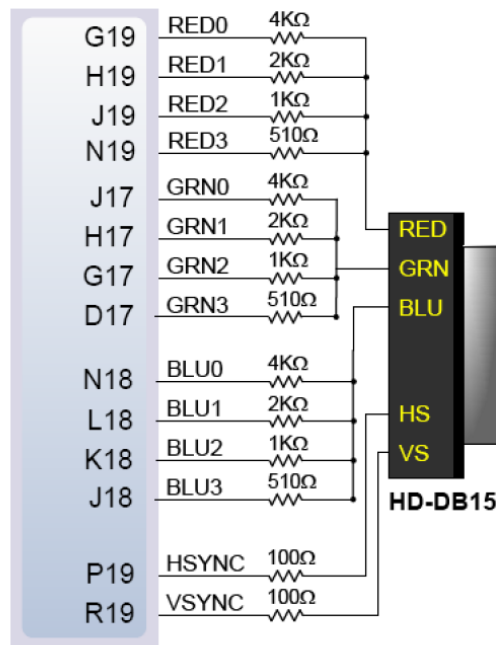
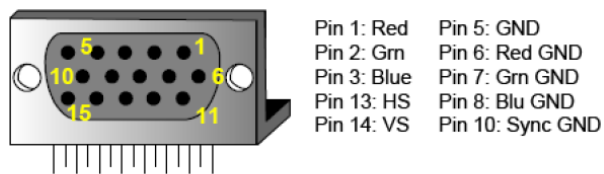
根据分辨率不同，VGA 可分为 VGA(640\*480)、SVGA(800x600)、XGA(1024x768)等。不同的分辨率下行同步和场同步信号的周期是不同的，时序上的时间也不一样。表 1 给出了不同分辨率的 VGA 显示刷新频率时序参数。

表 1 常用刷新频率时序参数表

显示模式	时钟 (Mhz)	行时序 (像素数)					场时序 (行数)				
		a	b	c	d	e	h	i	j	k	L
640*480@60	25.2	96	48	640	16	800	2	33	480	10	525
640*480@75	31.5	64	120	640	16	840	3	16	480	1	500
800*600@60	40.0	128	88	800	40	1056	4	23	600	1	628
800*600@75	49.5	80	160	800	16	1056	3	21	600	1	625
1024*768@60	65.0	136	160	1024	24	1344	6	29	768	3	806
1024*768@75	78.8	176	176	1024	16	1312	3	28	768	1	800

以 640×480 且刷新频率为 60Hz 为例，显示器每秒扫描 60 场（帧）。VGA 在实际工作时并不是每行扫描 640 个点，每场扫 480 行。实际是每行 800 个像素。每场 525 行。每行 800 个像素中包括行消隐前肩 d(16 个点)、行同步信号 a(96 个点)、行消隐后肩 b(48 个点)和有效点数 c(640 个)。每场 525 行中包括场消隐前肩 k(10 行)、场同步信号 h(2 行)、场消隐后肩 i(33 行)和有效行数 j(480 行)。所以，点像素的时钟频率为  $800 \times 525 \times 60 = 25.2\text{MHz}$ 。

输入到显示器的 R、G 和 B 信号是模拟信号。然而，FPGA 的输出信号是数字信号，即每一个颜色由 4 位二进制表示（如图 1），所以需要 D/A 转换器把它转变为模拟信号。开发板使用一个简单的电路实现了将 4 位的数字信号转换为 16 电平的模拟信号。开发板支持 12 位的 VGA 彩色显示：4 位红基色、4 位绿基色、4 位蓝基色，这将产生 4096 种不同的颜色。Basys 3 开发板的 VGA 接口及管脚信息如图 2 所示。



Artix-7

图 2 Basys 3 开发板 VGA 接口

实现图像在屏幕上移动即改变图像在屏幕上显示的起始坐标。

### 参考代码

**640x480@60Hz 标准的 VGA 显示时序控制模块 vga\_640x480:**

```
module vga_640x480(
input wire clk,
input wire clr,
output reg hsync,
output reg vsync,
output reg [9:0]hc,
output reg [9:0]vc,
output reg vidon
);
parameter hpixels=800;//行像素点
parameter vlines=521;//行数
parameter hbp=144;//行显示后延（96+48）
parameter hfp=784;//行显示前沿（96+48+16+640）
parameter vbp=31;//场显示后沿（2+29）
parameter vfp=511;//场显示前沿（2+29+480）
```

```

reg vsenable;//使能 vc;
//行同步信号计数器
always@(posedge clk or posedge clr)
    begin
        if(clr==1)
            hc<=0;
        else
            begin
                if(hc==hpixels-1)
                    begin
                        hc<=0;
                        vsenable<=1;
                    end
                else
                    begin
                        hc<=hc+1;
                        vsenable<=0;
                    end
            end
        end
    end
//产生 hsync 脉冲
always@(*)
    begin
        if(hc<96)
            hsync=0;
        else
            hsync=1;
        end
    end
//场同步信号计数器
always@(posedge clk or posedge clr)
    begin
        if(clr==1)
            vc<=0;
        else
            if(vsenable==1)
                begin
                    if(vc==vlines-1)
                        vc<=0;
                    else
                        vc<=vc+1;
                end
            end
        end
    end
//产生 vsync 脉冲
always@(*)

```

```

        begin
            if(vc<2)
                vsync=0;
            else
                vsync=1;
            end
        //使能显示
        always@(*)
        begin
            if((hc<hfp)&&(hc>=hbp)&&(vc<vfp)&&(vc>=vbp))
                vidon=1;
            else
                vidon=0;
        end
    endmodule

```

实现像素颜色控制模块，实现图像在显示屏上的红、绿、蓝显示：

```

module vga_stripes_top(
input wire clk_100M,
input wire [3:3]btn,
output wire hsync,vsync,
output wire [3:0]red,green,blue
);
    wire clk_25M,clr,vidon;
    wire [9:0]hc,vc;
    assign clr=btn[3];
    clk_div U1(clk_100M,clk_25M,clr);
    vga_640x480 U2(clk_25M,clr,hsync,vsync,hc,vc,vidon);
    vga_stripes U3(vidon,hc,vc,red,green,blue);
endmodule

```

```

module vga_stripes(
input wire vidon,
input wire [9:0]hc,vc,
output reg[3:0]red,green,blue
);
    always@(*)
    begin
        red=0;
        green=0;
        blue=0;
        if(vidon==1)
            begin

```

```
        red={vc[4],vc[4],vc[4],vc[4]};  
        green=~{vc[4],vc[4],vc[4],vc[4]};  
    end  
end  
endmodule
```