

字符显示实验

黑金动力社区 2020-03-13

1 实验简介

在 VGA 测试实验中讲解了 VGA 显示原理和显示方式，本实验介绍如何使用 FPGA 实现字符显示，通过这个实验更加深入的了解 VGA 的显示方式。

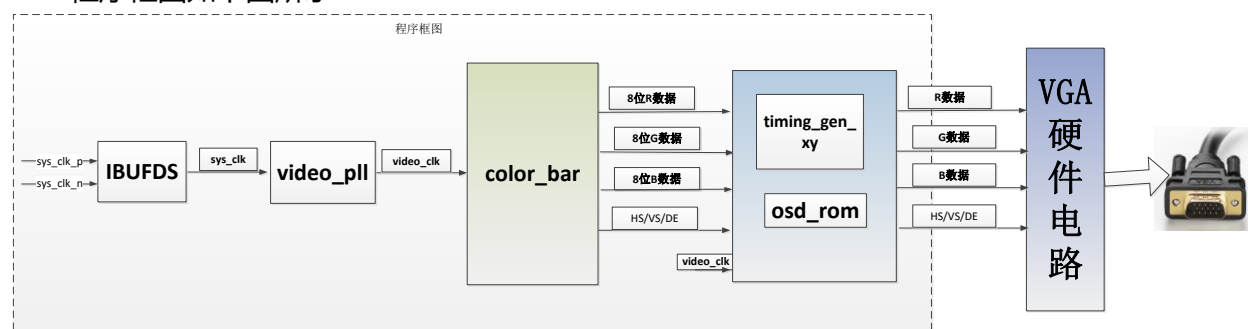
2 实验原理

实验通过字符转换工具将字符转换为 16 进制 coe 文件存放单端口的 ROM IP 核中，再从 ROM 中把转换后的数据读取出来显示到 VGA 上。

3 程序设计

字符显示例程是在 VGA 显示的基础上增加了一个 osd_display 的模块，“osd_display”模块是用来读取存储在 Rom ip 核里转换后的字符信息，并在指定区域显示。

程序框图如下图所示：

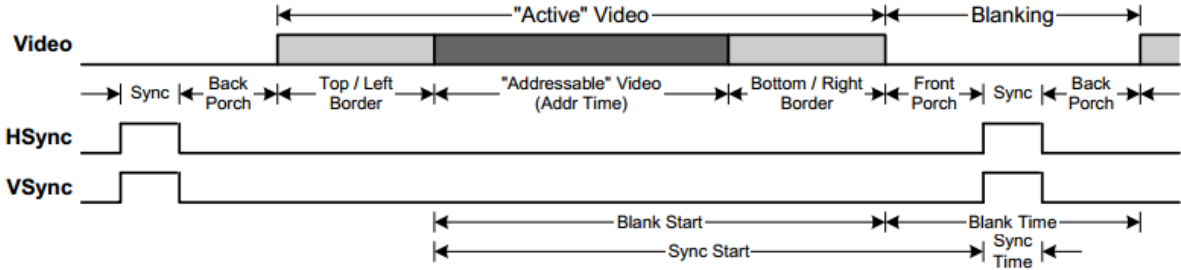


osd_display 模块包含 timing_gen_xy 模块和 osd_rom 模块。Osd_rom 里存储的字符数据，如果数据为 1，OSD 的区域显示 ROM 中的前景红色（显示 ALINX 芯驿），如果数据是 0，OSD 的区域显示数据为背景色（彩条）。

在“timing_gen_xy”模块是根据 VGA 时序标准定义了“x_cnt”和“y_cnt”两个计数器并由这两个计数器产生了 VGA 显示的“x”坐标和“y”坐标。程序中用“vs_edge”和“de_falling”分

别表示场同步开始信号和数据有效结束信号。其原理如下图所示：

Definition of Terms

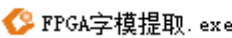


信号名称	方向	说明
rst_n	in	异步复位输入,低复位
clk	in	外部时钟输入
i_hs	in	行同步信号
i_vs	in	场同步信号
i_de	in	数据有效信号
i_data	in	color_bar 数据
o_hs	out	输出行同步信号
o_vs	out	输出场同步信号
o_de	out	输出数据有效信号
o_data	out	输出数据
x	out	生成 X 坐标
y	out	生成 Y 坐标

timing_gen_xy 模块端口

下面介绍如何存储文字信息的 ROMIP，首先需要生成能够被 XILINX FPGA 识别的.coe 文件。

1. 在软件工具及驱动文件夹下找到“FPGA 字模提取”工具。



2. 双击.exe 文件打开工具



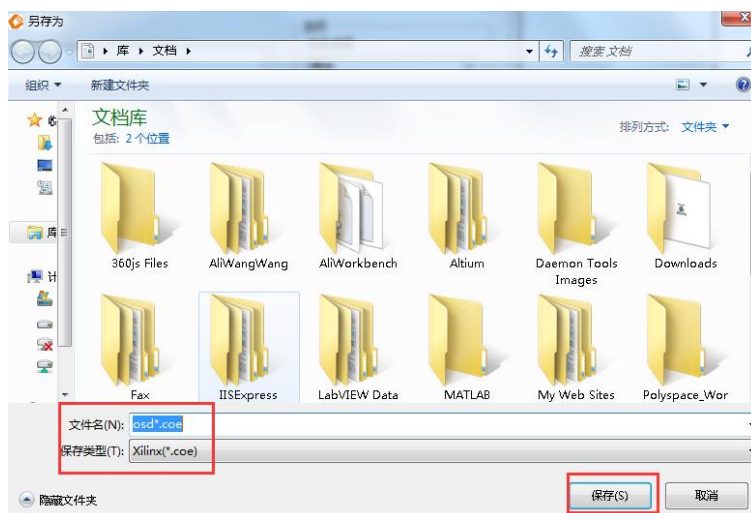
3. 在提取工具的“字符输入”框中输入需要显示的字符，字体和字符高度可以自定义选择。设置完成后点击“转换”按钮，在界面左下角可以看到转换后的字符点阵大小，点阵的宽和高在程序中是需要用到的



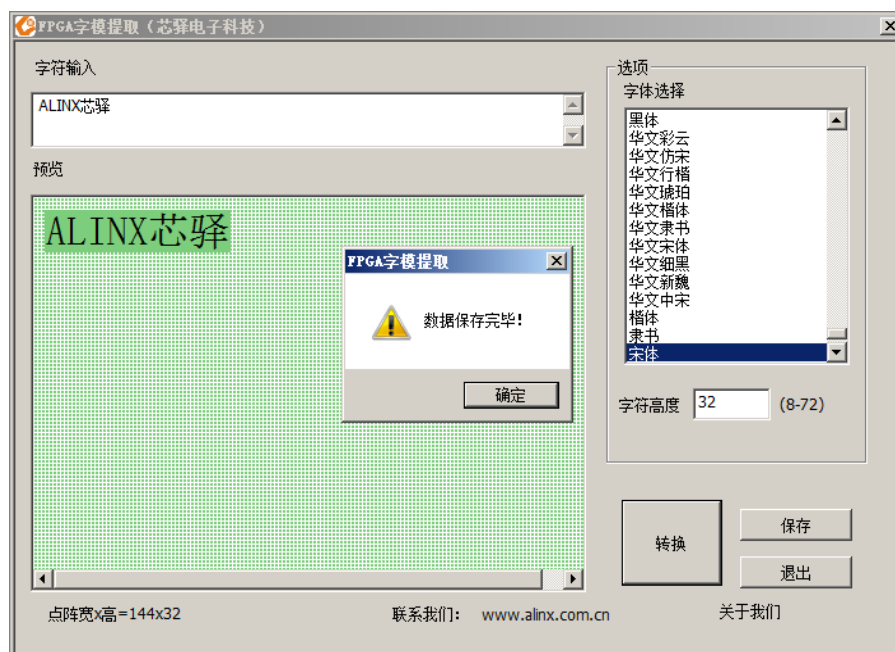
点阵的宽和高这里位 144x32,需要跟 osd_display 程序中定义的一致:

```
parameter OSD_WIDTH   = 12'd144;  
parameter OSD_HEIGHT  = 12'd32;
```

4. 点击“保存”按钮，将文件保存到本例程源文件目录下，需要注意的是在保存类型下应该选择 Xilinx (*.coe) ,点击“保存”按钮。



5. 回到字符提取工具界面出现如下对话框表示保存完成，点击确定，退出即可



另外，在前面已经介绍将字符转换后是成点阵的方式存储到 Rom IP 核里，找到生成的.coe 文件打开后可以看到如下：

```

1 MEMORY_INITIALIZATION_RADIX=16;
2 MEMORY_INITIALIZATION_VECTOR=
3 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
4 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
5 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,04,
6 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
7 00,1C,38,00,00,00,00,03,00,00,00,00,00,00,00,00,
8 00,00,00,0C,38,00,00,E6,FF,07,80,01,00,00,00,00,
9 00,00,00,00,00,0C,18,08,FC,0F,01,03,C0,01,7E,00,
10 F8,1F,0F,FC,3E,7E,00,0C,18,1C,00,06,82,01,C0,01,
11 18,00,80,01,1C,10,1C,18,FC,FF,FF,3F,30,06,C6,00,
12 C0,01,18,00,80,01,1C,10,18,08,00,0C,18,00,30,06,
13 EC,00,C0,03,18,00,80,01,3C,10,18,0C,00,0C,18,00,
14 10,06,78,00,60,03,18,00,80,01,34,10,30,04,00,0C,
15 18,00,18,02,38,00,20,03,18,00,80,01,74,10,30,06,
16 00,04,08,00,18,03,FC,00,20,03,18,00,80,01,64,10,
17 60,02,00,60,00,00,18,03,C6,03,20,03,18,00,80,01,
18 E4,10,60,03,00,C0,00,00,18,83,01,3F,30,06,18,00,
19 80,01,C4,10,C0,01,00,82,03,00,18,63,38,0C,10,06,

```

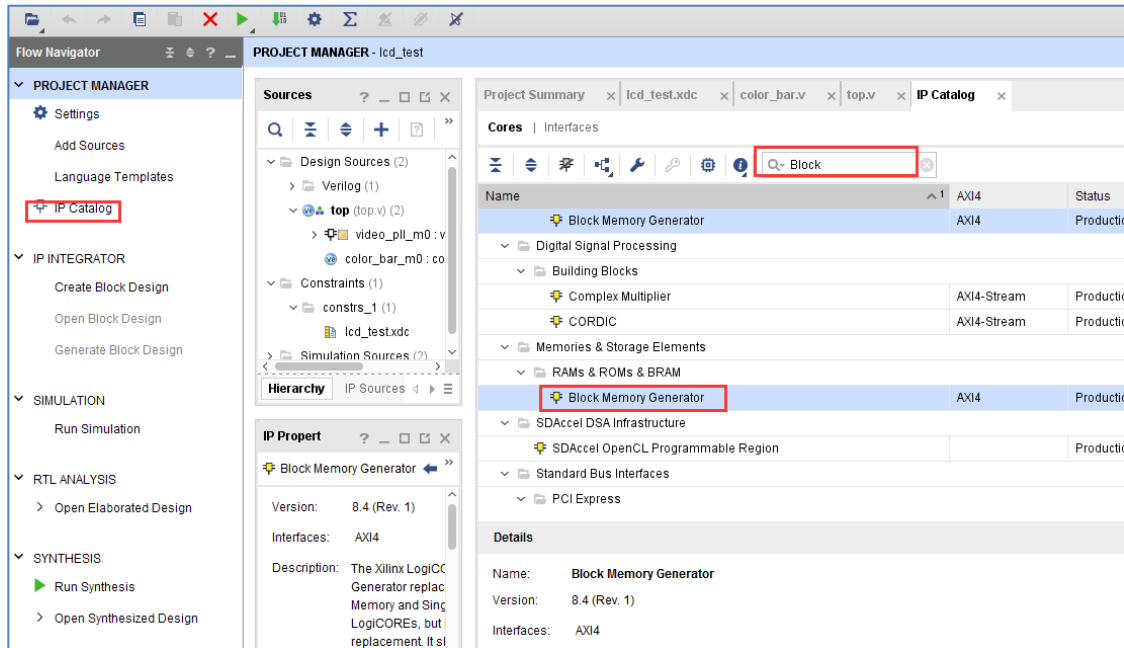
转换后的字符已经成了 8bit 的数值文件，一个字节代表 8 个时钟的像素，所以在从 Rom IP 核中读取 coe 文件的值，需要判断每一位的值，如果值为 1，显示红色前景色，否则显示背景色：

```

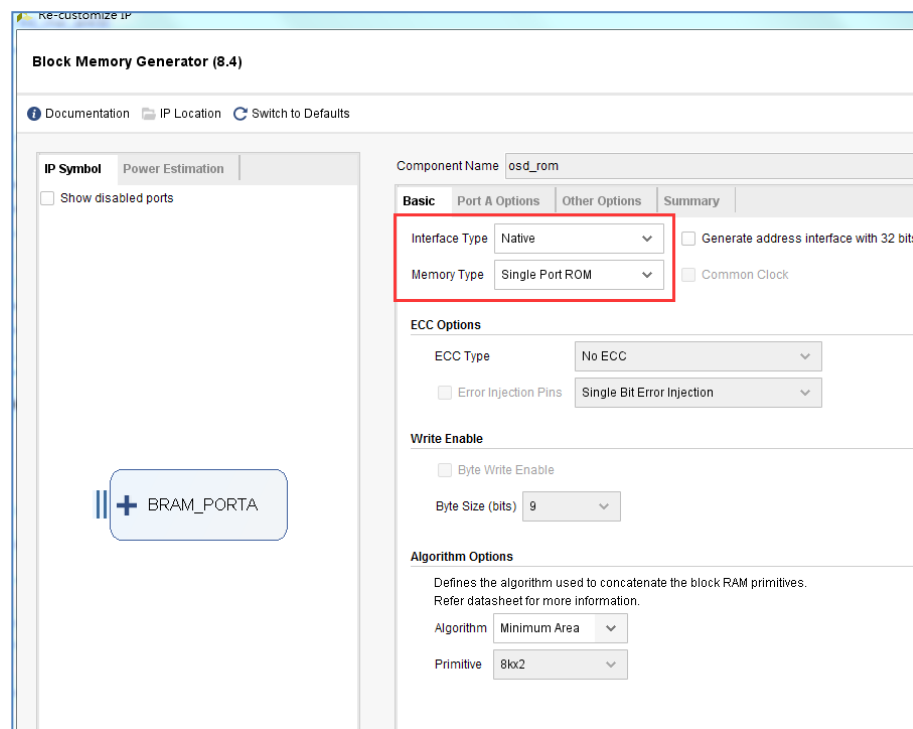
always@(posedge pclk)
begin
    if(region_active_d0 == 1'b1)
        if(q[osd_x[2:0]] == 1'b1)
            v_data <= 24'hff0000;
        else
            v_data <= pos_data;
    else
        v_data <= pos_data;
end

```

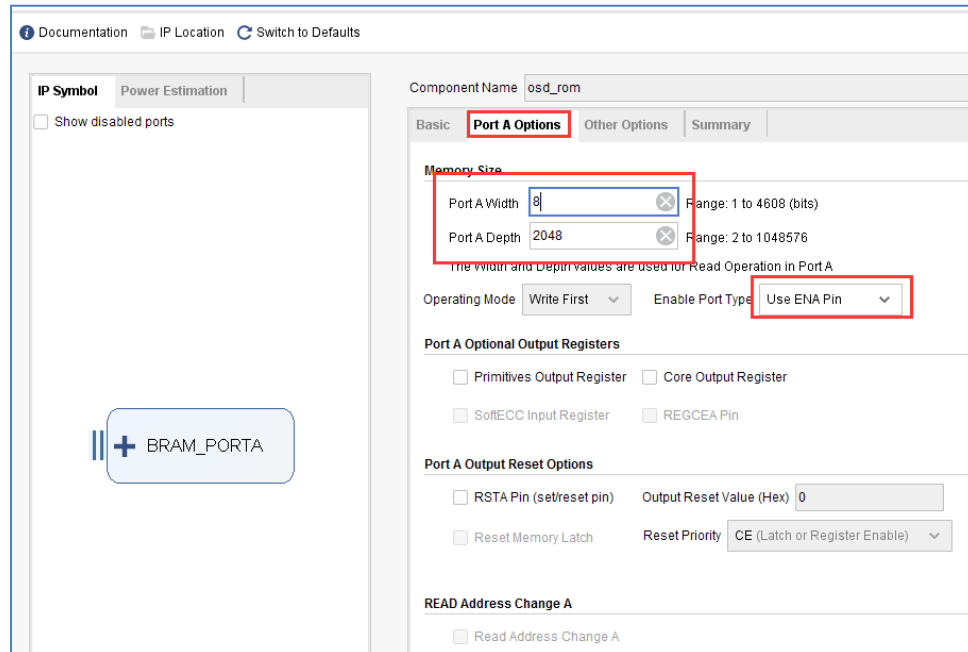
6. 调用单端口 Rom IP 核的过程和调用其他 IP 核一样，打开 “IP Catalog” 搜索“Block”:



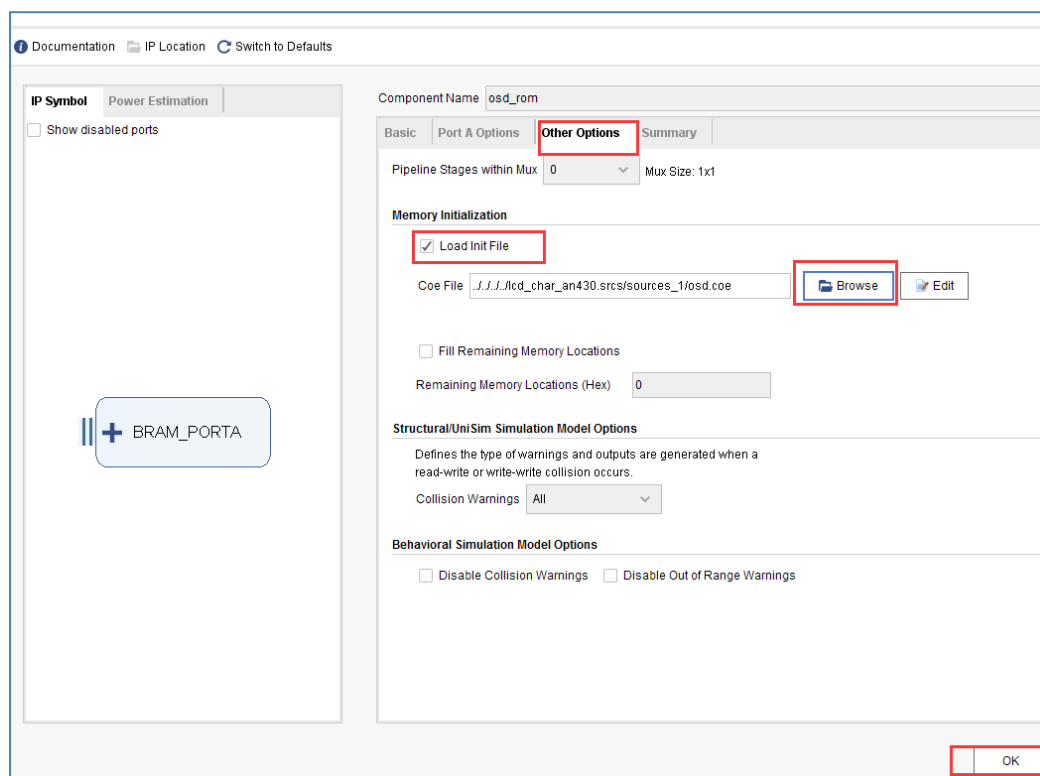
再按下图在 Basic 栏中设置为 ROM:



在 PortA Options 栏中设置如下：



按如下图添加 osd.coe 文件（找到前面生成的 coe 文件），完成后点击“OK”按钮：



Rom IP 核在“osd_display”模块中例化如下：

```
osd_rom osd_rom_m0
```

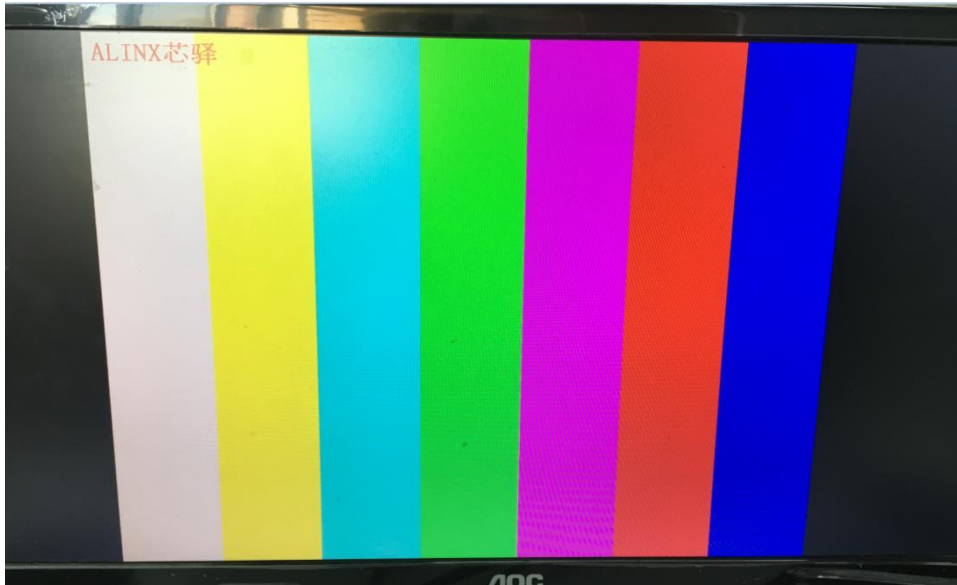
```
(  
    .address(osd_ram_addr[15:3]),  
    .clock(pclk),  
    .q(q)  
);
```

信号名称	方向	说明
rst_n	in	异步复位输入,低复位
pclk	in	外部时钟输入
i_hs	in	行同步信号
i_vs	in	场同步信号
i_de	in	数据有效信号
i_data	in	color_bar 数据
o_hs	out	输出行同步信号
o_vs	out	输出场同步信号
o_de	out	输出数据有效信号
o_data	out	输出数据

osd_display 模块端口

4 试验现象

连接好开发板和显示器，连接方式参考《VGA 测试实验》教程，需要注意，开发板的各个连接器不要带电热插拔，下载好试验程序，可以看到显示器显示以彩条为背景的字符。开发板做为 VGA 输出设备，只能通过 VGA 显示设备来显示，不要试图通过笔记本电脑的 VGA 接口来显示，因为笔记本也是输出设备。



默认字符显示的位置在 x 坐标和 y 坐标都是 9 的地方显示，另外用户可以修改下面的 pos_y 和 pos_x 的判断条件将字符显示在显示屏的任意位置：

```
73 always@(posedge pclk)
74 begin
75     if(pos_y >= 12'd9 && pos_y <= 12'd9 + OSD_HEIGHT - 12'd1 && pos_x >= 12'd9 && pos_x <= 12'd9 + OSD_WIDTH - 12'd1)
76         region_active <= 1'b1;
77     else
78         region_active <= 1'b0;
79 end
```