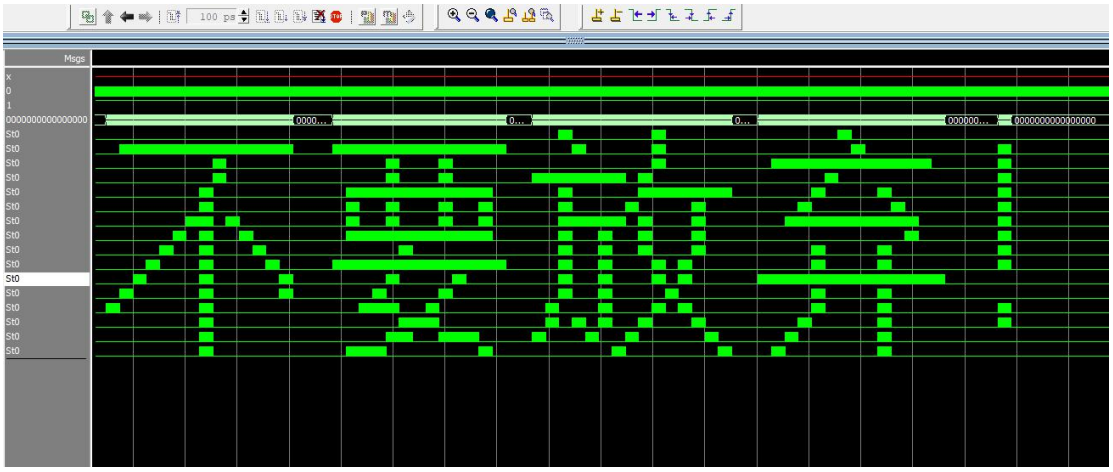
这个小工程利用 modelsim 仿真波形图案来显示字符。效果如下，送给正在复习备考的同学，坚持就是胜利：



用到的工具如下：

PCtoLCD2002：产生显示用的字符点阵

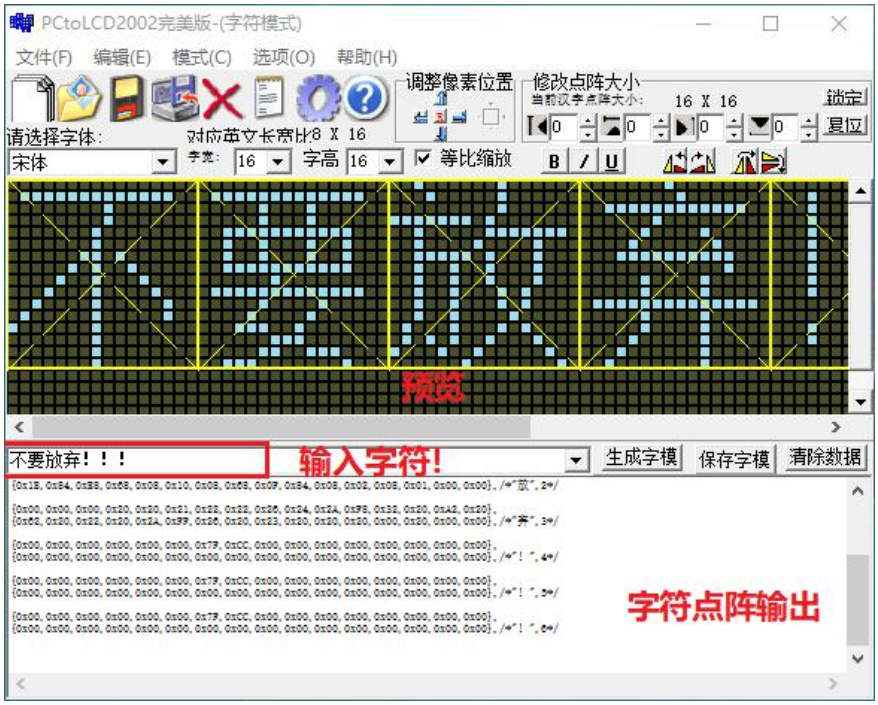
Quartus：代码编写，仿真文件生成。

Modelsim：生成要显示的字符图案。

1，字符点阵产生

PCtoLCD2002 端设置如下：

生成字符点阵：



保持字模后，生成的字符点阵文件如下：

```
 不(0) 要(1) 放(2) 弃(3) ！(4) ！(5) ！(6)

{0x00,0x08,0x40,0x10,0x40,0x20,0x40,0x40,0x40,0x80,0x41,0x00,0x42,0x00,0x4F,0xFF},
{0x70,0x00,0x42,0x00,0x41,0x00,0x40,0x80,0x40,0x40,0x40,0x30,0x00,0x00,0x00,0x00},/*"不",0*/

{0x00,0x00,0x40,0x40,0x4F,0x41,0x49,0x49,0x49,0x59,0x7F,0x6A,0x49,0xC6,0x49,0x44},
{0x49,0x4C,0x7F,0x52,0x49,0x62,0x49,0x42,0x4F,0x41,0x40,0x40,0x00,0x00,0x00,0x00},/*"要",1*/

{0x10,0x02,0x10,0x0C,0x9F,0xF0,0x52,0x04,0x12,0x02,0x13,0xFC,0x12,0x01,0x04,0x02},
{0x1B,0x84,0xE8,0x68,0x08,0x10,0x08,0x68,0x0F,0x84,0x08,0x02,0x08,0x01,0x00,0x00},/*"放",2*/

{0x00,0x00,0x00,0x20,0x20,0x21,0x22,0x22,0x26,0x24,0x2A,0xF8,0x32,0x20,0xA2,0x20},
{0x62,0x20,0x22,0x20,0x2A,0xFF,0x26,0x20,0x23,0x20,0x20,0x20,0x00,0x20,0x00,0x00},/*"弃",3*/

{0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0xCC,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/*"！",4*/

{0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0xCC,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/*"！",5*/

{0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0xCC,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},/*"！",6*/
```

2，字符点阵数据处理

将字符点阵文件中的提示信息，备注，标点符号等全部删除，只留下点阵的编码。并将相邻奇偶两行的数据调整到一行。效果如下：

```
0008
4010
4020
4040
4080
4100
4200
4FFF
7000
4200
4100
4080
4040
4030
0000
0000
0000
4040
4F41
4949
4959
7F6A
49C6
4944
494C
7F52
4962
4942
4F41
4040
0000
0000
1002
100C
9FF0
```

将处理后的字符文件重命名为　char_code.hex

3，代码编辑

（1,）字符点阵数据储存

利用 BRAM 模块来储存字符点阵数据。利用 readmemh 函数加载字符点阵数据。

```
// Memory Array
reg  [DATAWIDTH-1:0] memory[0:(2**MEMWIDTH-1)];

initial
begin
    $readmemh("char_code.hex", memory);
end
```

（2）波形生成

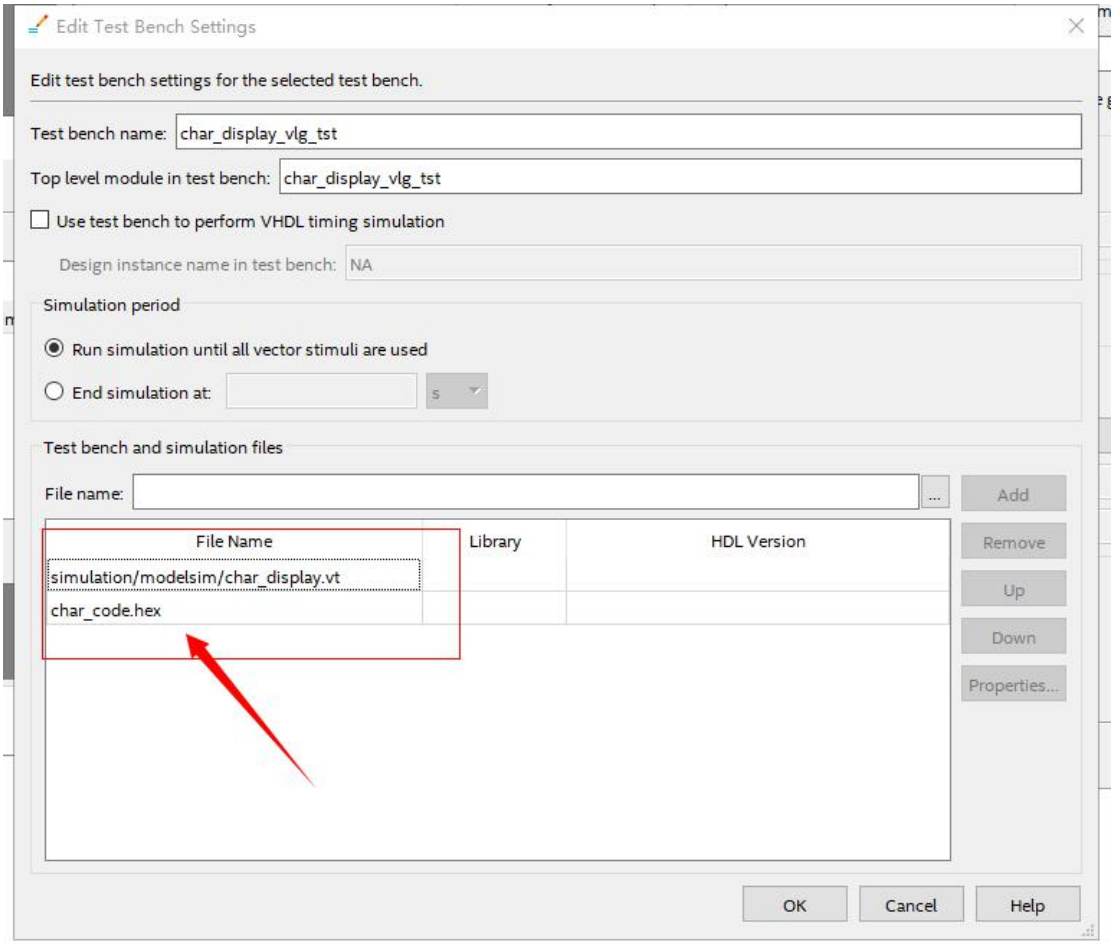
将从 BRAM 模块中读出的数据与时钟信号按位相与即可。

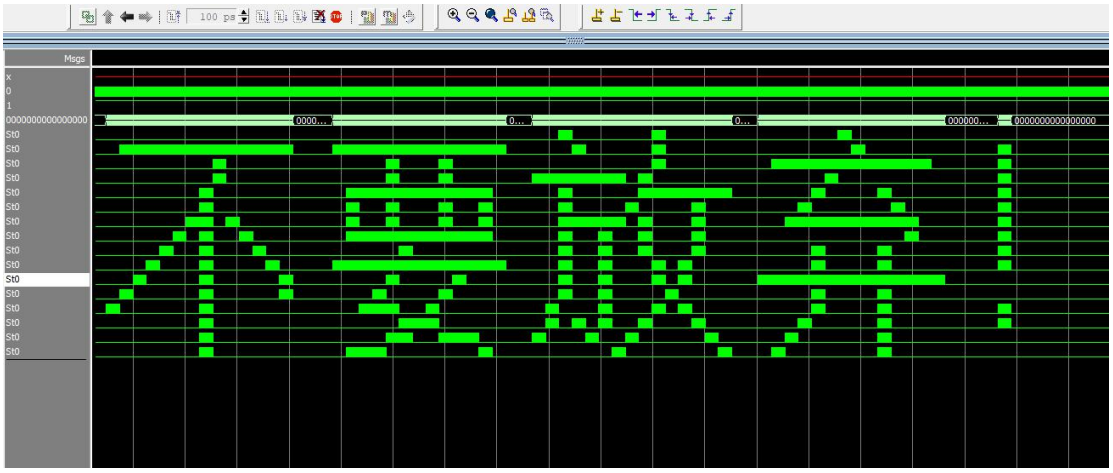
```
wire    [15:0]  data_out;
assign  out =   {16{clk}} & data_out;
```

4，仿真

编写仿真文件，运行仿真，展开 out 信号，缩放到合适比例，即可看到字符显示。效果即同文章开头所示。

【注】本工程是利用 quartus 设置的仿真环境，在仿真设置时，最好将 char_code.hex 也添加进仿真文件列表中，仿真仿真软件找不到该文件。





更多细节请查看源代码！

源代码链接 https://github.com/WayneGong/char_display