****

*计算机硬件系统设计*

实验二 字符显示与数字钟显示

姓 名：肖子原

学 号：1120151869

组 员：朱威浦，沈翰文，肖子原，廖汉龙

学 院：计算机学院

邮 箱：[liamliaohl@gmail.com](mailto:liamliaohl@gmail.com)

2018年4月14日 星期六

目录

[第一章 字符显示 4](#_Toc511906391)

[一、实验目的 4](#_Toc511906392)

[二、实验内容 4](#_Toc511906393)

[三、实验原理与步骤 4](#_Toc511906394)

[3.1 实验环境配置 4](#_Toc511906395)

[3.2 实验逻辑与原理 4](#_Toc511906396)

[3.3实验过程 5](#_Toc511906397)

[四、实验结果与功能拓展 10](#_Toc511906398)

[五、问题与解决思路 10](#_Toc511906399)

[第二章 LCD、VGA 显示数字钟实现 11](#_Toc511906400)

[一、实验目的 11](#_Toc511906401)

[二、实验内容 11](#_Toc511906402)

[三、实验原理与配置 11](#_Toc511906403)

[3.1 实验环境配置 11](#_Toc511906404)

[3.2实验原理 11](#_Toc511906405)

[3.3 实验步骤 11](#_Toc511906406)

[四、实验结果及拓展功能 13](#_Toc511906407)

[五、问题及解决思路 14](#_Toc511906408)

[第三章 数码管、LCD、VGA显示数字钟实验 14](#_Toc511906409)

[一、实验目的 14](#_Toc511906410)

[二、实验内容 14](#_Toc511906411)

[三、实验原理与步骤 15](#_Toc511906412)

[3.1 实验环境配置 15](#_Toc511906413)

[3.2实验原理 15](#_Toc511906414)

[3.3实验步骤 15](#_Toc511906415)

[四、实验结果与功能拓展 17](#_Toc511906416)

[五、问题与解决思路 18](#_Toc511906417)

[5.1文件重复，模块重定义问题 18](#_Toc511906418)

[第四章 个人工作与心得体会 19](#_Toc511906419)

[1. 个人工作与学习过程 19](#_Toc511906420)

[2 心得体会 19](#_Toc511906421)

# 第一章 字符显示

## 一、实验目的

1.通过VGA字符显示实验，了解FPGA字符显示的内在逻辑与基本原理

2.进一步熟悉Verilog 硬件描述语言以及对于实验硬件，QuartusⅡ软件的熟练掌握

3.了解VGA的rgb成像的原理

4.了解如何建立.mif文件并通过IP核建立Rom

## 二、实验内容

1.使用字符取模软件，生成字符点整信息，并将点整信息写入.mif文件中,可以自己实现一些特定的字符。

2.通过IP核管理界面生成生成定制的Rom逻辑，并实现访问点整信息

3.在生成的Rom逻辑上建立字符显示文件，可以改变字符的大小，格式等信息。

4.建立RGB信号转化文件

## 三、实验原理与步骤

### 3.1 实验环境配置

|  |  |
| --- | --- |
| 计算机操作系统 | Windows10 家庭版 |
| FPGA 试验板 | Dei2-150 |
| RAM | 8G |
| CPU | Core-i5 8250 |
| 编译软件 | Quartus Ⅱ(13.0-64bit Edition) |

### 3.2 实验逻辑与原理

.mif文件（Memory Initalization File）时Rom 的初始化配置文件，字符显示的原理如下图所示：

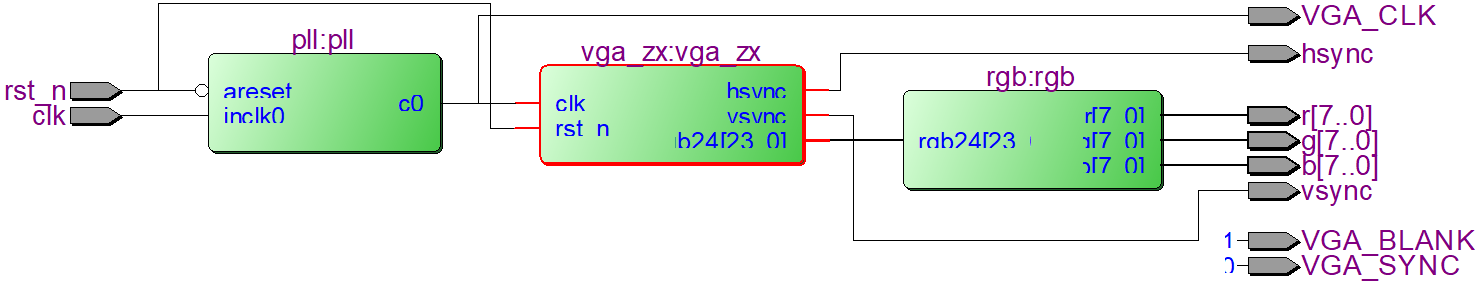


图 1-3-1

上述的逻辑表示PLL为vga\_xz模块提供固定频率的时钟脉冲，vga\_xz模块用于显示字符，这个逻辑下面还有一个逻辑，即IP核生成的Rom, 其中的mif文件包含有字符的点整信息。Rgb模块将24位的rgb信号转化为单独的rgb三种信号，输出至VGA对应的开发板上。

下面简单说明.mif文件的生成原理。

在实验过程中的重要步骤是通过.mif文件导入rom中，显示点整信息。、

在配置.mif文件时需要配置以下内容：

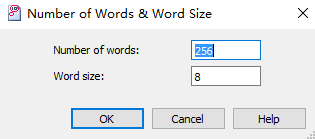
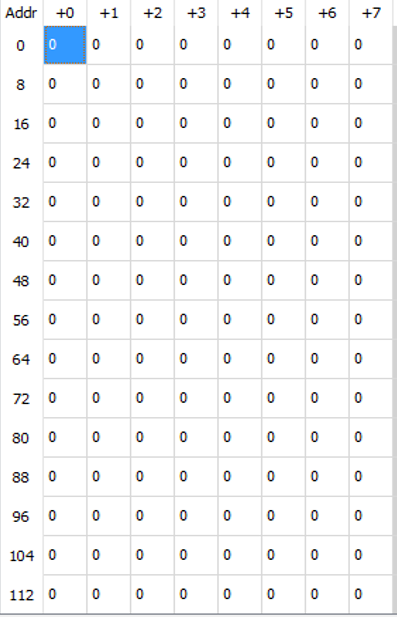
 

图 1-3-2

一项配置是Rom的大小和字长度，在此即指每一行的位数。在配置完成以后，就会出现配置每一位数据的表格，通过软件写入的点整信息，再通过字符显示文件即可在显示器中显示出来。

IP core还可用于配置RAM, FIFO等文件。

整个逻辑实现过程：

1.将点整信息放入.mif文件中

2.使用IP核生成Rom逻辑，访问点整信息。使用IP核管理界面生成了定制的Rom逻辑，包含了前述的.mif文件，可以访问点整信息。

3.建立.v文件，用于字符显示，此文件实现对于第2步生成的Rom逻辑的访问。

### 3.3实验过程

VGA字符实验是在VGA纯色实验的基础上进行的，在VGA纯色显示的实验工程中，新建输出字符的模块和内存初始化文件。所以模块主要分两部分，根据要显示的内容，建立相应的.mif文件，新建ROM模块并完成top.v配置。

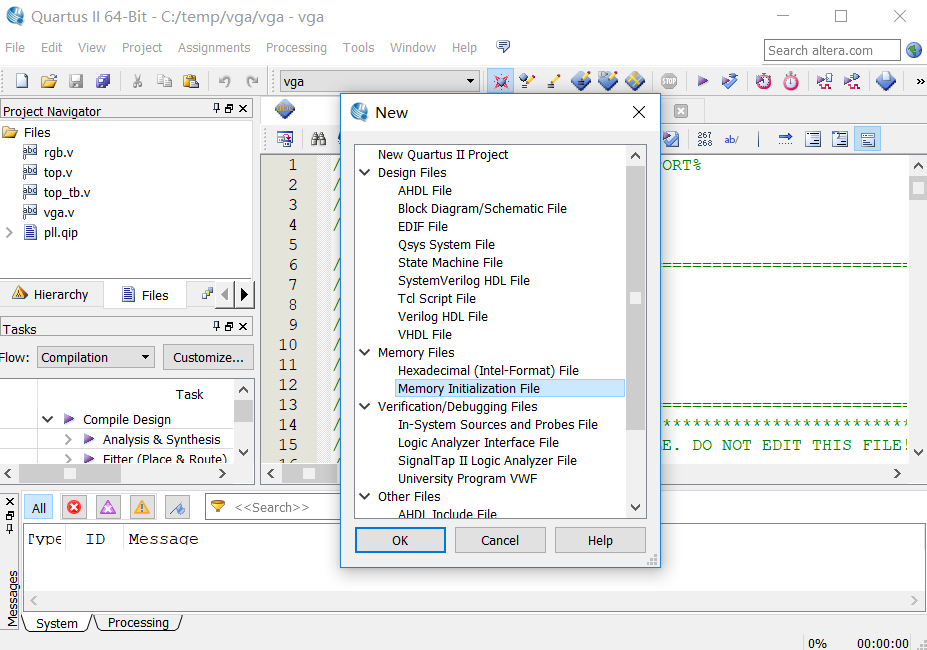


图 1-3-3

（1）新建.mif文件，供ROM模块使用

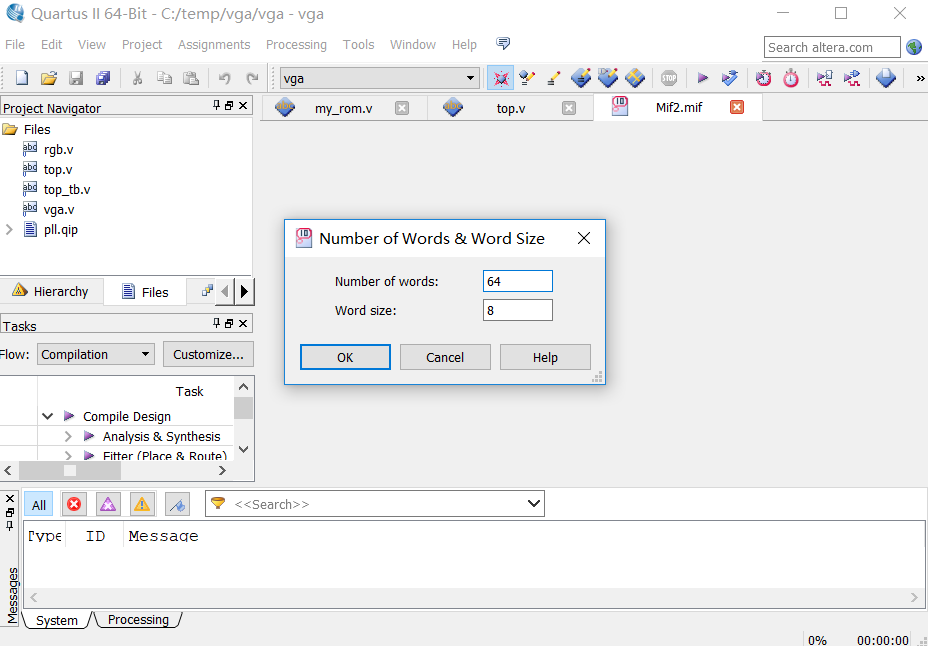


图 1-3-4

（2）确定大小，这里的number of words与PC to LCD软件的输出尺寸相对应。在这一步的时候我们遇到了一些问题，在尝试了多种方法，并最终解决。

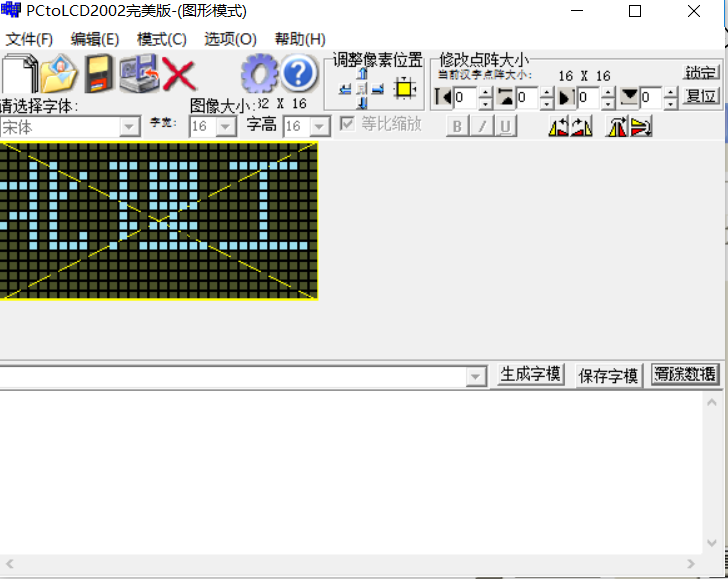


图 1-3-5

（3）键入需要显示的字模。PC to LCD软件可以将确定的字模转化成二进制信息的格式，供mif文件使用。我们按照教程确定了显示区域的大小为32\*16，并在输入框中输入“北理工”三个汉字。

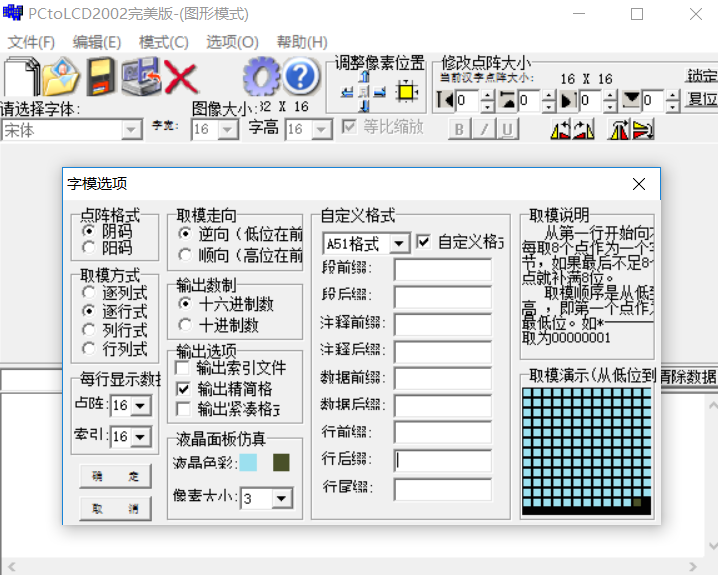


图 1-3-6

4）更改PC to LCD软件的输出格式。为了适应mif文件的格式，我们将输出进制改为16进制，点阵和索引都改为16，但输出格式一栏需要特别注意，为了方便下一步工作，我将所有的前缀和后缀都去除了。

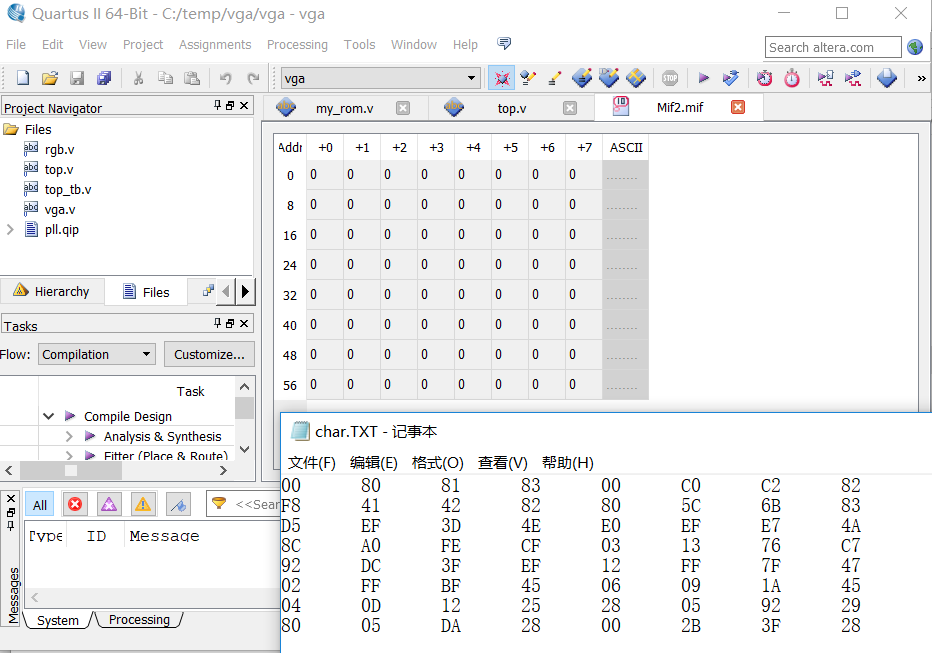


图 1-3-7

（5）将字模输入到mif文件。在这一步我们最初始终不能正确将16进制数键入到mif框中，后来经过尝试多种方法，最终解决了这个问题，解决过程后有详细阐述。

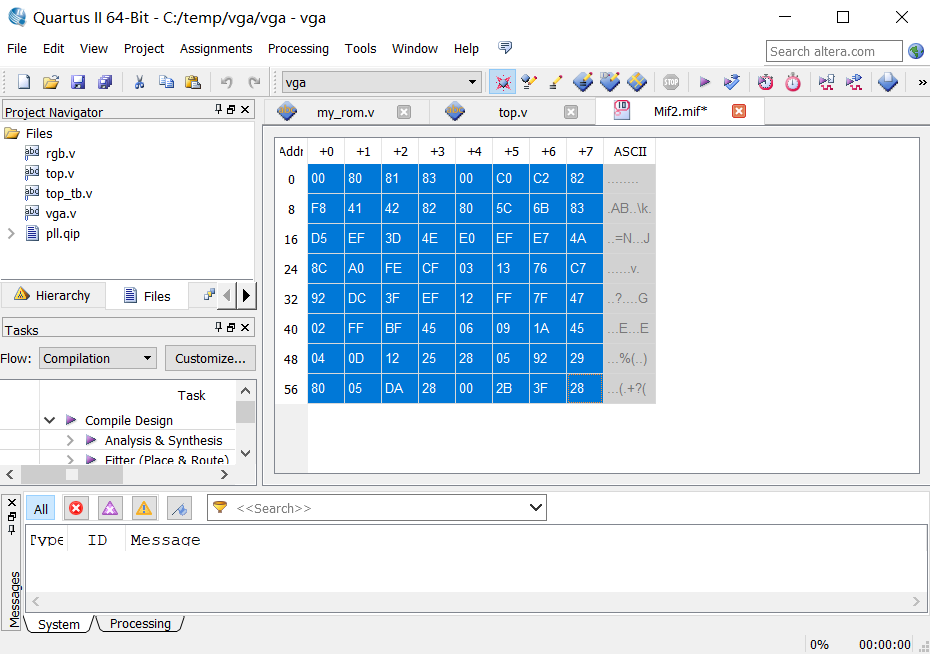


图 1-3-8

6）完成了mif文件的键入，将其保存，以备后续的ROM配置模块使用。

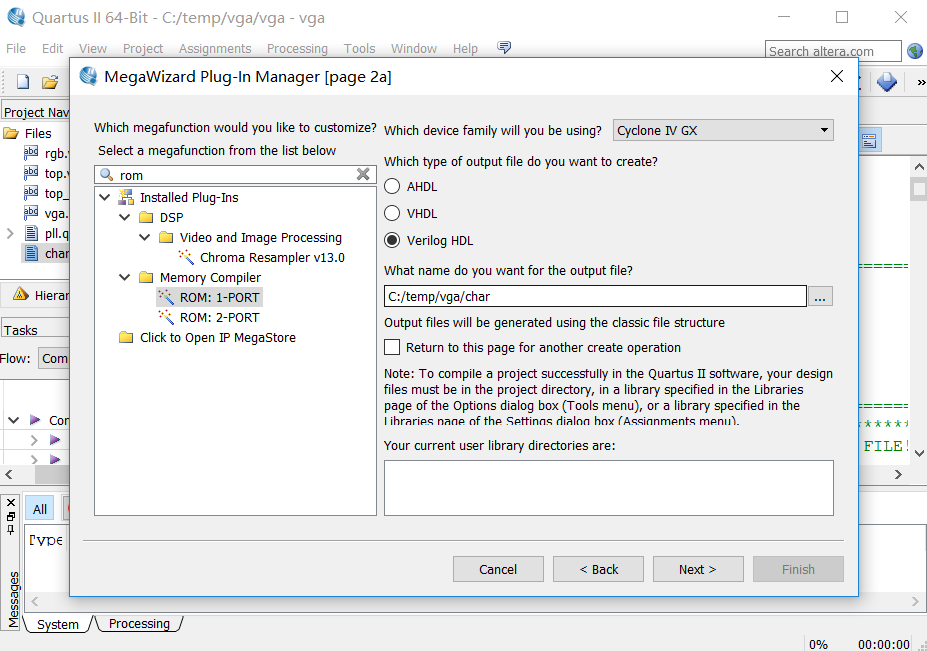


图 1-3-9

（7）IP核管理界面。

在左上方搜索框内输入rom，点击下方出现的ROM: 1-PORT，右侧矩形框内输入输出文件的名称，点击Next。

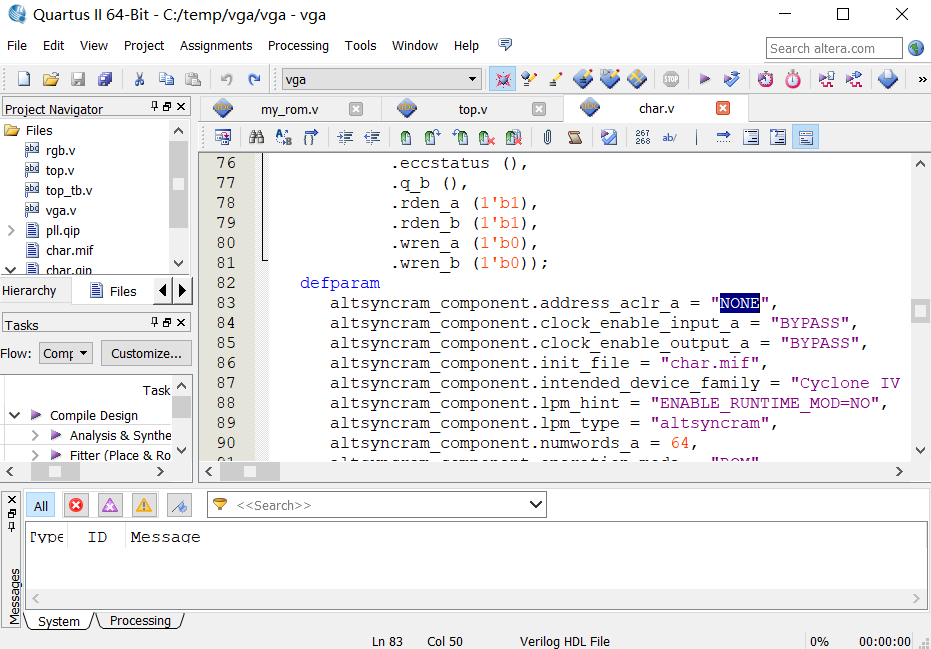


图 1-3-10

8）修改char.v文件。参照老师给出的实例项目，在自动生成的char.v文件之后，还需要进行一些修改才能够正确下板运行。

将address\_aclr\_a原来的“NONE”修改为“BYPASS”。

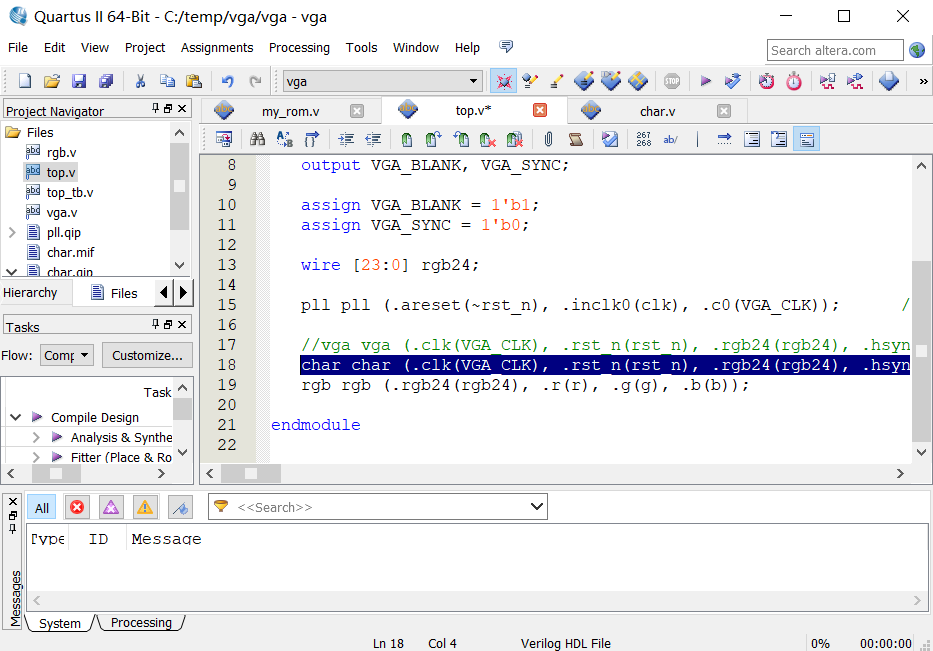


图 1-3-11

（9）配置top.v文件。在top文件中调用刚刚新建的字符ROM模块char，并将原来的vga模块注释，就可以将ROM配置部分换成新建的char模块，编译完成后下板运行即可。

## 四、实验结果与功能拓展

实验结果如图所示：

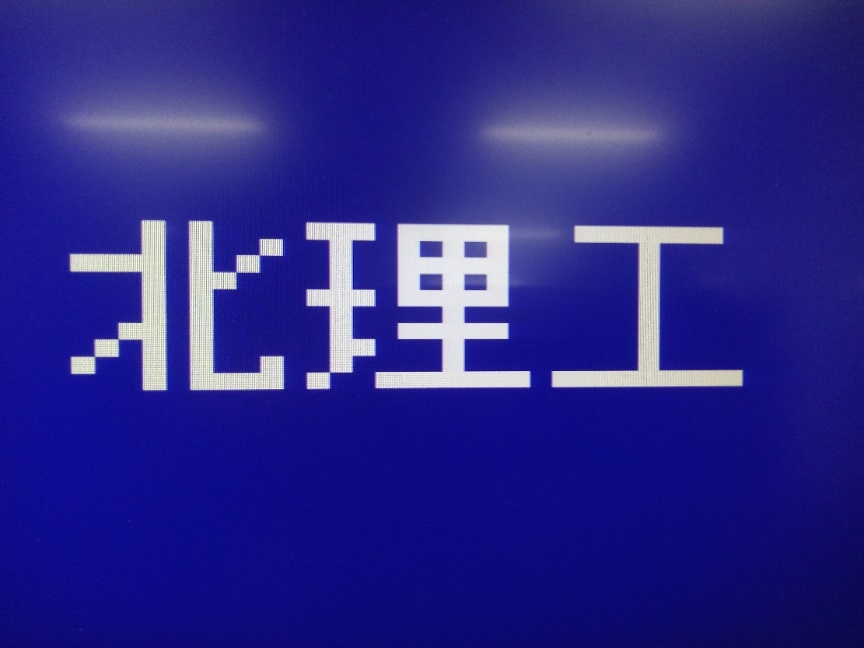


图 1-4-1

## 五、问题与解决思路

在VGA字符实验新建.mif文件时，number of words的大小无法确定。

我们最初不清楚这个选项的含义，在新建.mif文件的过程中就直接使用默认值256，这样就造成了后面为其赋值时，始终无法填满。

后来我们发现，这个number of words要与PC to LCD软件的输出相匹配，我们在PC to LCD软件的输出是32\*16，一共512 bit，等于64个字节，所以我们的number of words就确定为64。

# **第二章 LCD、数码管显示数字钟实现**

## 一、实验目的

了解数码管和LCD作为DE2i-150板上的显示部件的工作原理以及如何利用FPGA驱动它们并且实现简单的数字钟。

## 二、实验内容

实现LCD、数码管的数字钟显示。

## 三、实验原理与配置

### 3.1 实验环境配置

|  |  |
| --- | --- |
| 计算机操作系统 | Windows10 家庭版 |
| FPGA 试验板 | De2i-150 |
| RAM | 8G |
| CPU | Core-i5 8250 |
| 编译软件 | Quartus Ⅱ(13.0-64bit Edition) |

### 3.2实验原理

实验中用到的数码管是七段数码管，以及其分段排序示意图如图2-3-1。

七段数码管的每一段都可以单独进行控制其点亮与否，最终实现一定的图案，此处即是指相应的数字。

由于DE2i-150板子有足够多的引脚，我们只需给不同的引脚与相应的数码管的某一段搭配即可实现控制功能。

图 2-3-1

### 3.3 实验步骤

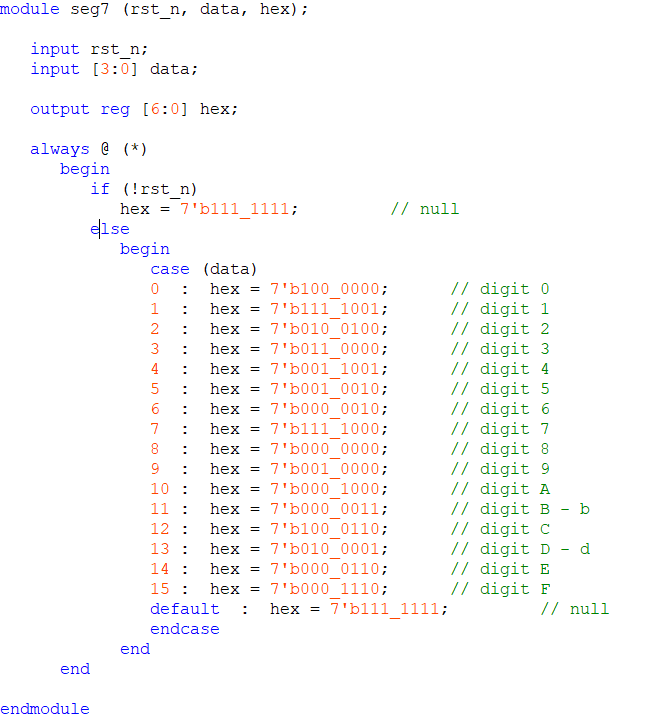
#### 3.3.1 新建一个工程

新建一个Quartus工程，并依据实验指导手册的流程对工程文件进行配置。

#### 3.3.2对单个数码管建模

单个数码管显示从0到F的16个数字，对应的输入data是0到F（此处需要转成十进制），编写seg7.v模块，显然对于不同的输入，对应要输出不同的电平控制信号。

其大致代码如下图：



#### 3.3.3 对8个数码管建模

单个数码管建模完成了，下面就要研究对8个数码管的建模，其系统架构图如图2-3-2：可以想见由于显示时钟的格式形如23:59:59.99，则八个数码管应被分为两两一组。

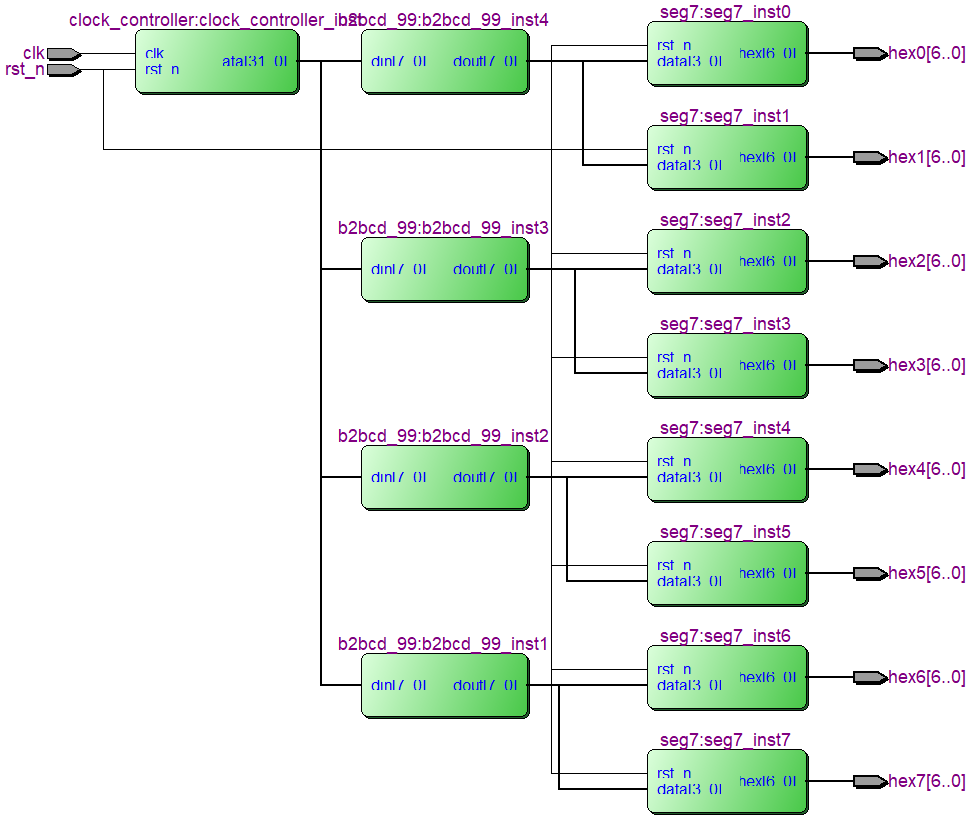


图2-3-2

接着再编写时钟控制器模块和二进制转BCD模块，最后编写顶层模块top.v，将以上三个模块实例化。

#### 3.3.4修改引脚文件

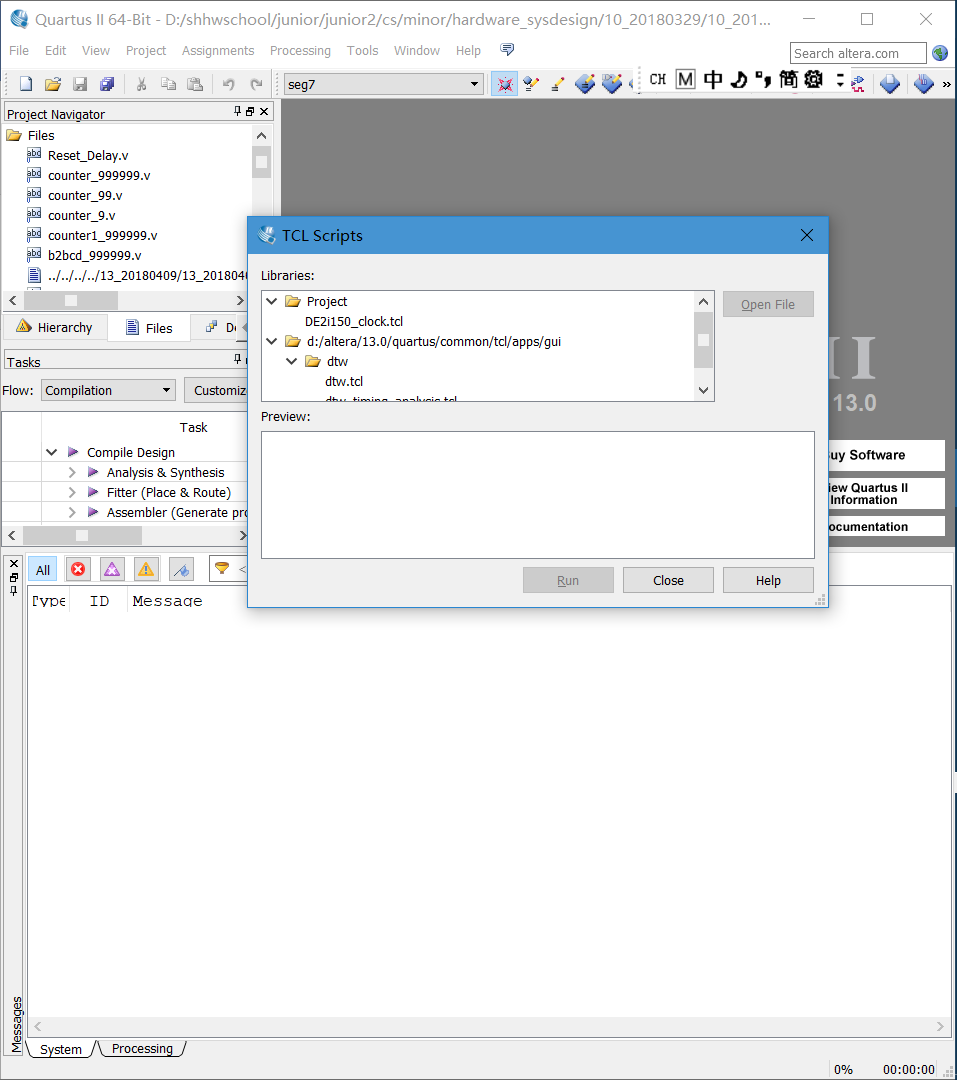


图 2-3-3

在Tools菜单下点击TCL Scripts，然后选择DE2i150\_clock.tcl文件进行运行，配置FPGA引脚。进行全编译，实现数码管的显示功能，效果图见后。

#### 3.3.5 LCD部分

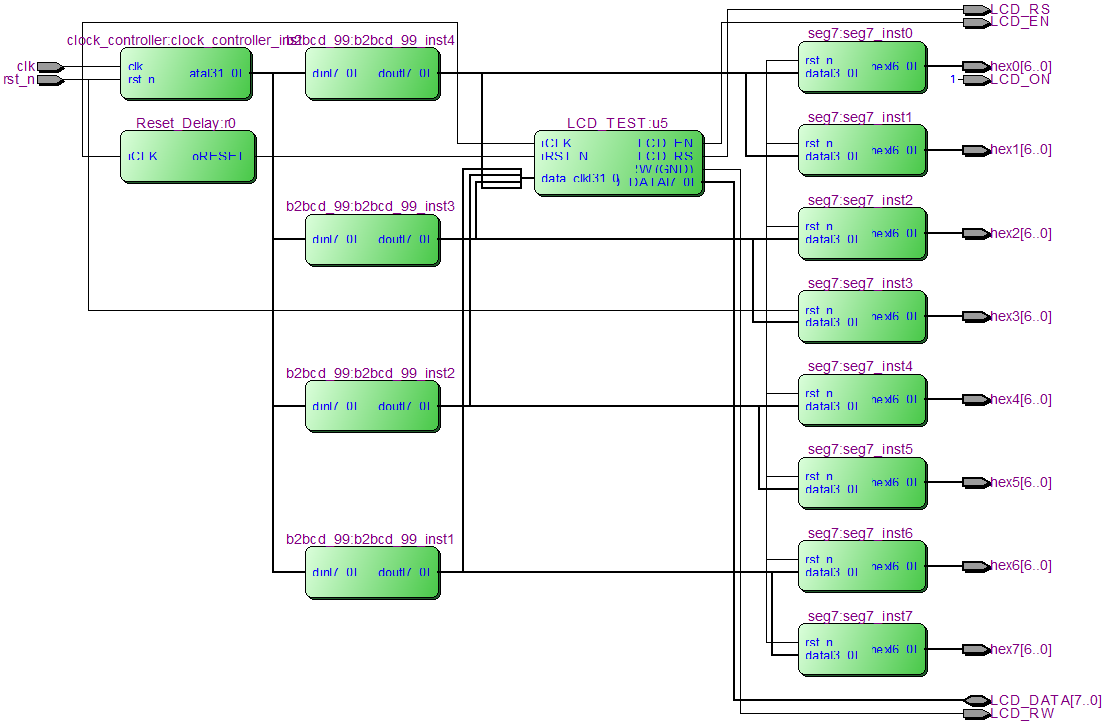


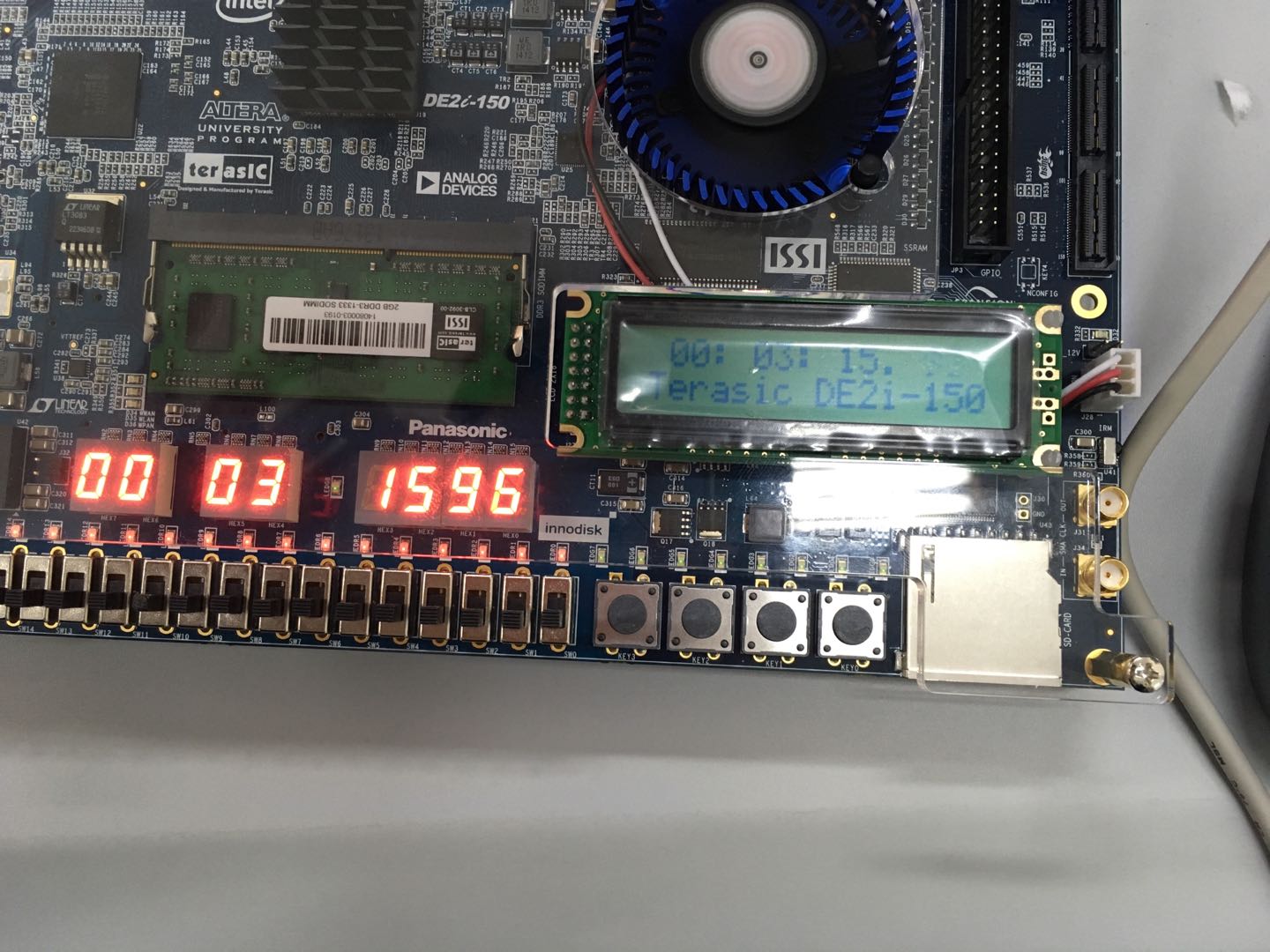
图 2-3-4

上图是LCD显示的系统构架图，相比原来的系统架构图，增加了延时模块Rest\_Delay.v和LCD驱动模块LCD\_TEST.v。

再修改顶层的top.v文件，加入LCD显示的有关参数。

## 四、实验结果及拓展功能

将sof文件下到DE2i-150板上，数码管显示出时钟信号，同时LCD也显示出数字时钟。如图：



## 五、问题及解决思路

在实验时，下板的时候一开始数码管和LCD无显示，我们后来发现，是由引脚文件没有运行造成的，运行后可以正常显示了。

# **第三章 数码管、LCD、VGA显示数字钟实验**

## 一、实验目的

综合运用数码管、LCD、VGA设计的相关知识，通过将数码管、LCD、VGA显示功能集中展示在一个工程项目， 加深对SOPC中外设驱动实现原理的了解。

## 二、实验内容

1.基于前述字符VGA显示实验，在本实验中实现数字钟VGA显示。

2.完成LCD数字钟显示。

3.完成数码管数字钟显示。

4.将数码管，LCD, VGA同时在De2i-150实验板同时同步实现。

## 三、实验原理与步骤

### 3.1 实验环境配置

|  |  |
| --- | --- |
| 计算机操作系统 | Windows10 家庭版 |
| FPGA 试验板 | Dei2-150 |
| RAM | 8G |
| DPU | Core-i5 8250 |
| 编译软件 | Quartus Ⅱ(13.0-64bit Edition) |

### 3.2实验原理

可编程片上系统（SOPC）是一种特殊的嵌入式系统，由单个芯片完成整个系统的主要逻辑功能，具有可裁减、可扩充、可升级和软硬件在系统可编程功能。

在VGA显示数字钟实验的基础上，增加数码管、LCD模块至总体框架中，将相应的输入接口配置到数码管和LCD模块，最后添加与设备相对应的引脚，即可实现三个模块同时显示数字钟功能。

### 3.3实验步骤

#### 3.3.1 新建VGA工程

新建名为vga\_clock的Quartus工程，并依据实验指导手册的流程对工程文件进行配置。

#### 3.3.2 添加数码管模块

将seg7\_clock实验中建立的seg7.v文件加入工程中，作为数码管的驱动模块，其输入接口信号为VGA中的b2bcd\_99模块输出的四位bcd码信号。

如图3-3-1：

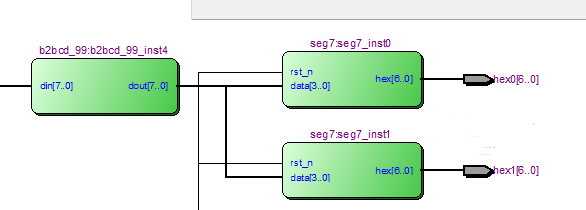


图3-3-1

#### 3.3.3 添加LCD模块

将seg7\_clock实验中建立的LCD\_TEST.v和LCE\_Controller.v文件加入工程中，作为LCD的驱动模块，其输入接口信号为VGA中的四个b2bcd\_99模块输出的32位bcd码信号。如图3-2：

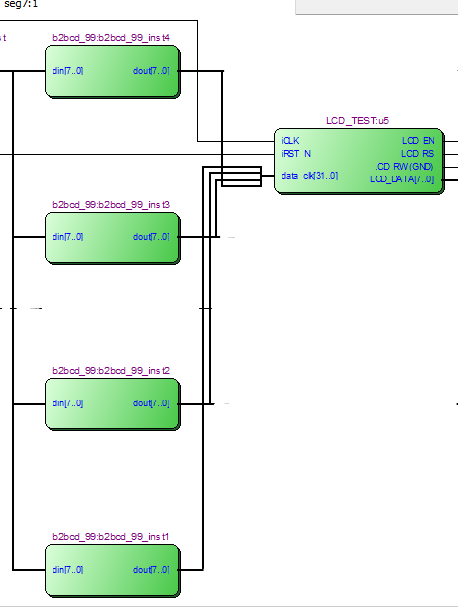


图 3-3-2

#### 3.3.4 修改vga\_top

在vga\_top.v模块的接口中加入8个数码管输出接口hex0, hex1, hex2, hex3, hex4, hex5, hex6, hex7和LCD输出接口LCD\_ON, LCD\_RW, LCD\_EN, LCD\_RS, LCD\_DATA。

将seg7.v模块例化为8个数码管，分别命名为seg7\_insti(i = 0～7)，data\_clk由低到高每四位作为该数码管的输入信号。

将LCE\_TEST.v模块例化为LCE驱动，将data\_clk作为其输入信号。

如图3-3：

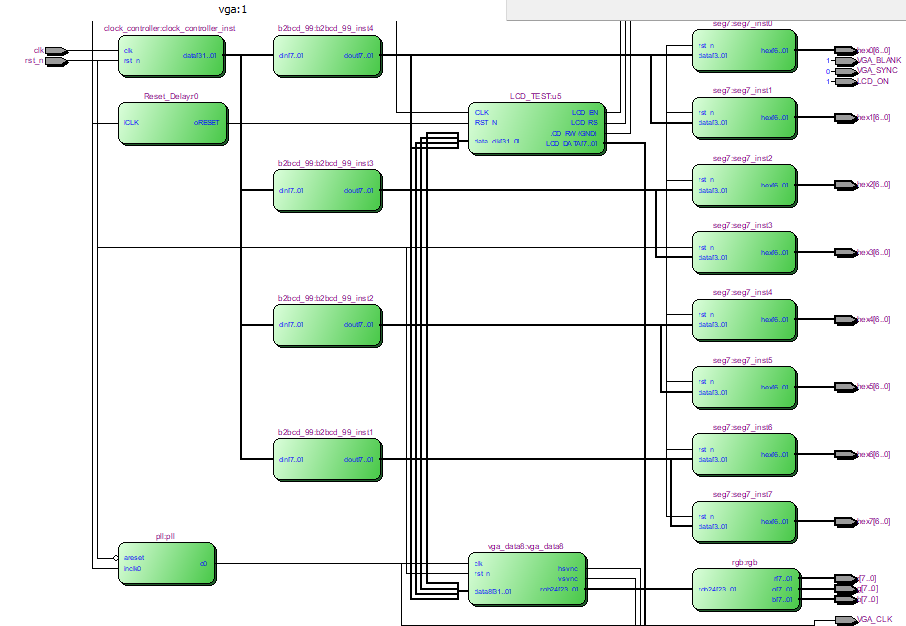


图 3-3-3

#### 3.3.5 修改引脚文件

在vga\_clock工程下的DE2i150\_clock\_VGA.tcl文件中加入数码管和LCD的引脚。

以数码管第一段为例，其配置代码如下：

|  |
| --- |
| **set\_location\_assignment PIN\_E15 -to hex0[0]**  **set\_location\_assignment PIN\_E12 -to hex0[1]**  **set\_location\_assignment PIN\_G11 -to hex0[2]**  **set\_location\_assignment PIN\_F11 -to hex0[3]**  **set\_location\_assignment PIN\_F16 -to hex0[4]**  **set\_location\_assignment PIN\_D16 -to hex0[5]**  **set\_location\_assignment PIN\_F14 -to hex0[6]**  **#============================================================**  **# hex0**  **#============================================================**  **set\_instance\_assignment -name IO\_STANDARD "2.5 V" -to hex0[0]**  **set\_instance\_assignment -name IO\_STANDARD "2.5 V" -to hex0[1]**  **set\_instance\_assignment -name IO\_STANDARD "2.5 V" -to hex0[2]**  **set\_instance\_assignment -name IO\_STANDARD "2.5 V" -to hex0[3]**  **set\_instance\_assignment -name IO\_STANDARD "2.5 V" -to hex0[4]**  **set\_instance\_assignment -name IO\_STANDARD "2.5 V" -to hex0[5]**  **set\_instance\_assignment -name IO\_STANDARD "2.5 V" -to hex0[6]** |

## 四、实验结果与功能拓展

数码管、LCD、VGA同时显示时钟信号，如图：



图 3-4-1



图3-4-2

## 五、问题与解决思路

### 5.1文件重复，模块重定义问题

实验初始时，我们想要通过暴力地直接将vga\_clock和seg7\_clock两个工程的文件合并在一起，以实现同时显示的功能，但是两个工程中所包含的文件有许多相同的部分，其次，两个工程的top文件也包含相同的接口及模块，导致编译时出现redefinaiton问题。

我们的解决思路是，首先观察两个工程中的公有模块，若vga\_clock工程中已经包含相应的文件，则不再将seg7\_clock工程中的文件拷贝过来。其次，直接在vga\_top.v自行定义8个数码管输出接口hex0,hex1,hex2,hex3,hex4,hex5,hex6,hex7和LCD输出接口LCD\_ON, LCD\_RW, LCD\_EN, LCD\_RS, LCD\_DATA。

将seg7.v模块例化为8个数码管，分别命名为seg7\_insti(i = 0～7)，data\_clk由低到高每四位作为该数码管的输入信号。

将LCE\_TEST.v模块例化为LCE驱动，将data\_clk作为其输入信号。

# **第四章 个人工作与心得体会**

## 1. 个人工作与学习过程

在实验过程中，我主要和廖汉龙在DE2i-150开发板下完成了VGA的字符显示的功能，通过自己编写，显示了汉字“北理工”，并且参数了数字钟的数码管，LCD, 以及VGA的显示的实现。和廖汉龙主要撰写了第一，第二部分的实验报告的撰写。

## 2 心得体会

通过这个实验，我对Verilog语言的工程文件的组织结构有了进一步的了解，通过阅读几个工程文件的代码以及结合相应的开发文档，我对于通过数码管，LCD，VGA的数字钟显示的原理有了理解。数字钟是一个比较小的工程，但是很能够说明问题，可以照着写一遍，立刻可以对整个过程有一个理解，有立竿见影的效果。后来我们还根据要求对几个工程文件通过编写一个顶层文件就机械地合起来，但是失败了，究其原因还是对于整个开发过程不够理解，因为我们并没有考虑到引脚的配置。实现了相应的功能后，我希望在研究生阶段从事计算机体系结构，CPU方向的进一步研究。