实验名称：实验十一 字符输入界面

姓名：张涵之

学号：191220154

班级：周一5-6

邮箱：[191220154@smail.nju.edu.cn](mailto:191220154@smail.nju.edu.cn)

实验时间：2020/11/30

11.3 实验内容

实现一个可以用键盘输入，并在VGA显示器上回显的交互界面。

基本要求：支持所有小写英文字母和数字、符号输入；

一直按压某个键时，重复输出该字符；

输入至行尾后自动换行，输入回车也换行；

扩展要求：支持BackSpace键删除字符；

删除至本行开始后，再按删除回车键，停留在上一行末尾的非空字符后；

支持Shift键、Caps键以及大小写字符输入。

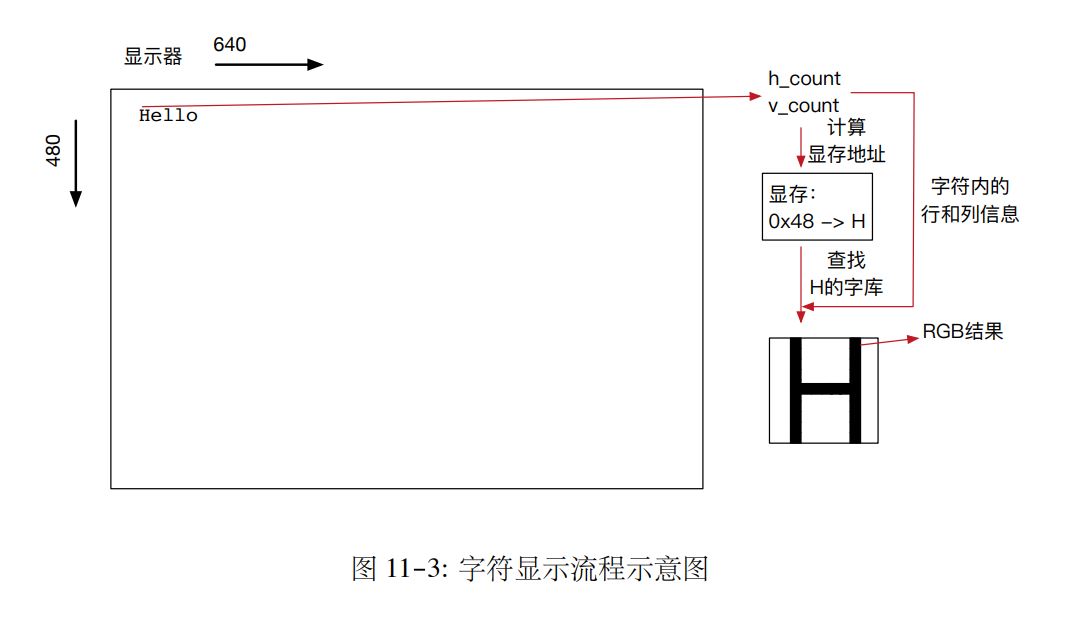
实验目的：实现一个可以用键盘输入，并在VGA显示器上回显的交互界面。

实验原理：

1）字符显示：ASCII字符用7bit表示，共128个字符。大部分情况下用8bit来表示单个字符，系统预留256个字符。系统中预先存储这256个字符的字模点阵。

这里每个字符高为16个点，宽为9个点。单个字符可以用16个9bit数来表示，每个数代表字符的一行，对应的点为“1”时显示白色，为“0”时显示黑色。只需要256 × 16 × 9 ≈ 37kbit的空间即可存储整个点阵。在显示时，根据当前屏幕位置，确定应该显示哪个字符，再查找对应的字符点阵即可完成显示。对于640×480的屏幕，可以显示30行（30×16=480），70 列（70 × 9=630）。系统的显存需要30 × 70大小，每单元存储8bit的ASCII字符。

2）扫描显示：VGA 控制模块可以输出当前扫描到的行和列的位置信息，稍加改动即可让其输出当前扫描的位置对应30×70 字符阵列的坐标 (0 ≤ x ≤ 69, 0 ≤ y ≤ 29)。利用坐标查询字符显存，获取对应字符的ASCII编码。利用ASCII编码查询对应的点阵ROM，根据扫描线的行和列信息，知道当前扫描的是字符内哪个点，根据该点对应的bit选择输出颜色。



显示的过程总结如下：

1. 根据当前扫描位置获取对应字符的x，y坐标，以及扫描单个字符点阵内的行列信息；

2. 根据字符的x，y坐标查询字符显存，获取对应的ASCII编码；

3. 根据ASCII编码和字符内的行信息查询点阵ROM，获取对应行的9bit数据；

4. 根据字符内的列信息取出对应的bit，并根据该bit设置颜色。

3）显存读写：对于键盘输入可以复用之前实现的键盘控制器。在键盘有输入的时候对字符显存进行改写，将按键对应的ASCII码写入显存的合适位置，将输入直接反馈到屏幕上。

程序代码或流程图：

ps2\_keyboard —\

ascii\_ram ———> key2ascii —\

vga\_ctr ——\

/———> vga\_ram ———> control

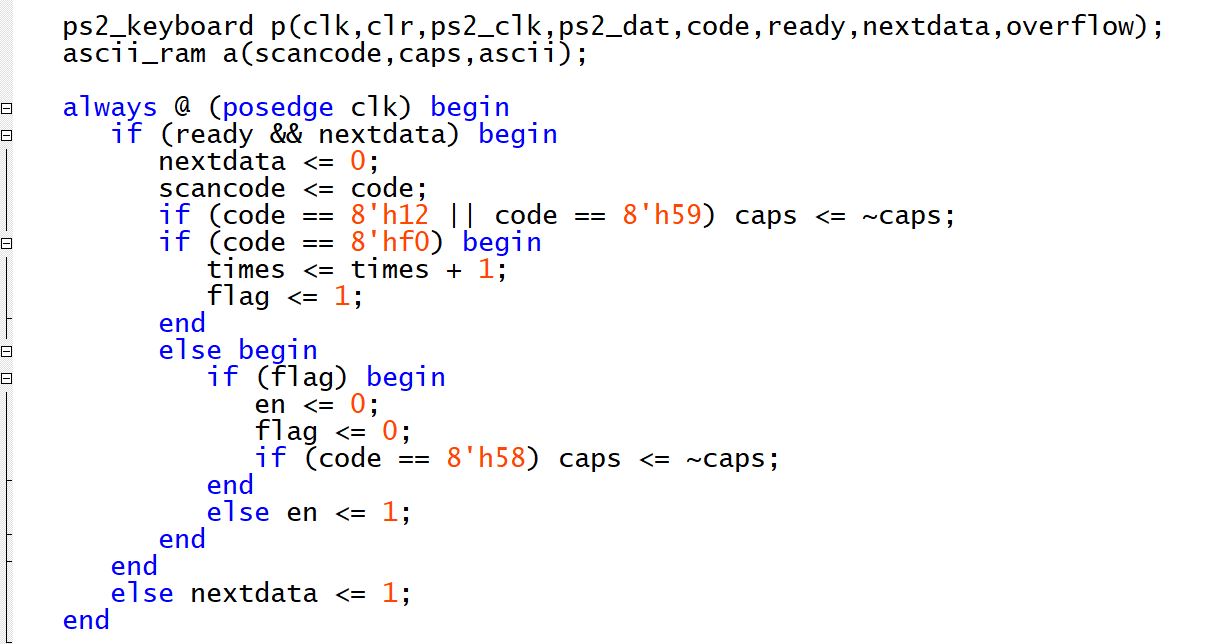
clkgen /———> font\_rom ——/ \——> hex

ps2\_keyboard //键盘控制器获取键码

ascii\_ram //读文件将键码转化为对应的ascii码

\*以上两个模块基本复用exp08中代码

key2ascii //综合键盘控制器和ram并判断shift和caps键



其中shift按下和弹起时caps各取反一次，caps按下弹起一次caps才取反

vga\_ctrl //显示器控制器提供接口

clkgen //生成特定频率的时钟

\*以上两个模块基本复用exp09中代码

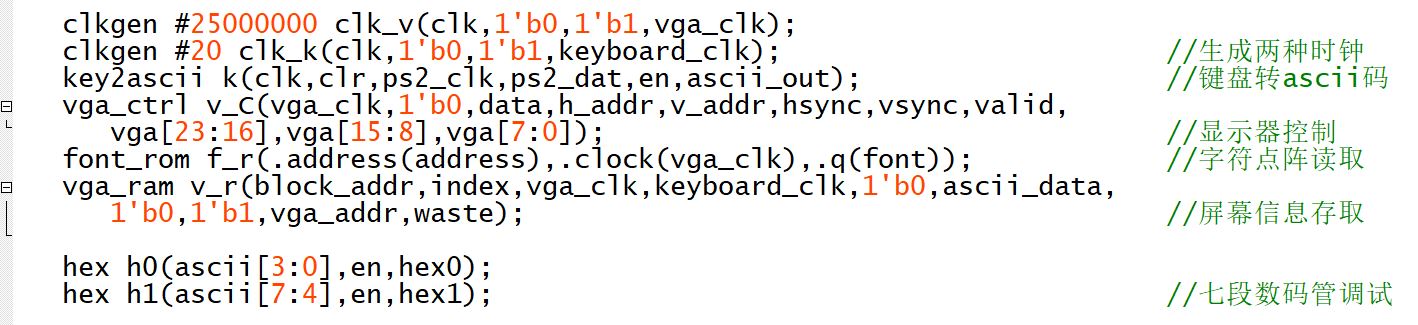
vga\_ram //存储和读写屏幕上70 × 30的内容

font\_rom //存储ascii码黑白点阵图

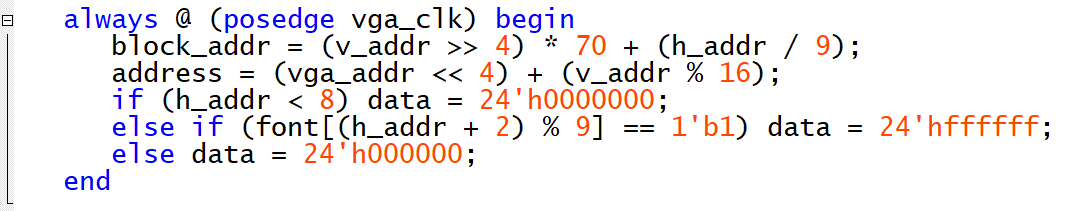
hex //将ascii码显示在七段数码管上便于调试

control //综合键盘和显示器的顶层控制模块

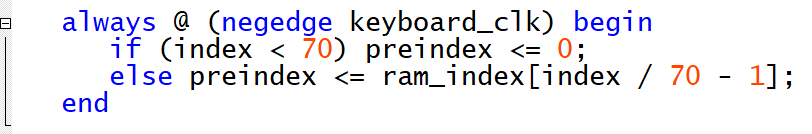




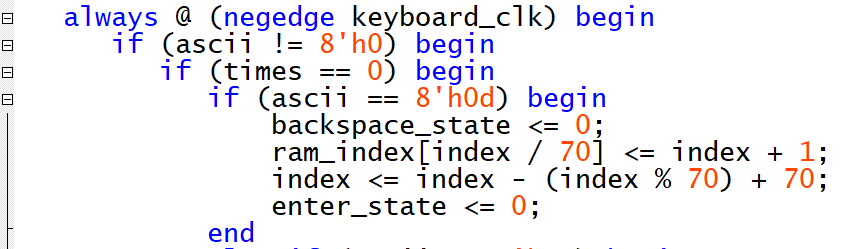
//计算两个存储器中读取的坐标，并根据当前点阵设置颜色



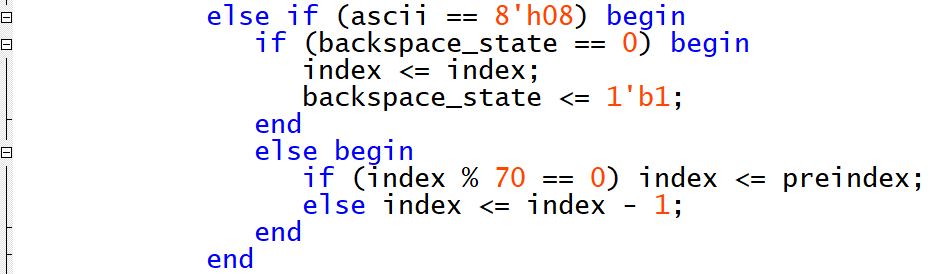
//记录光标上一次的位置，用于删除时退回



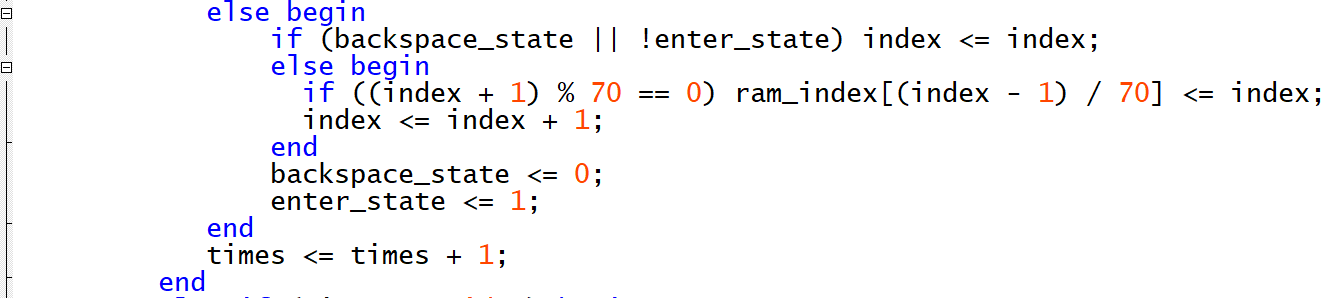
//按键时：进入退格键单个删除模式



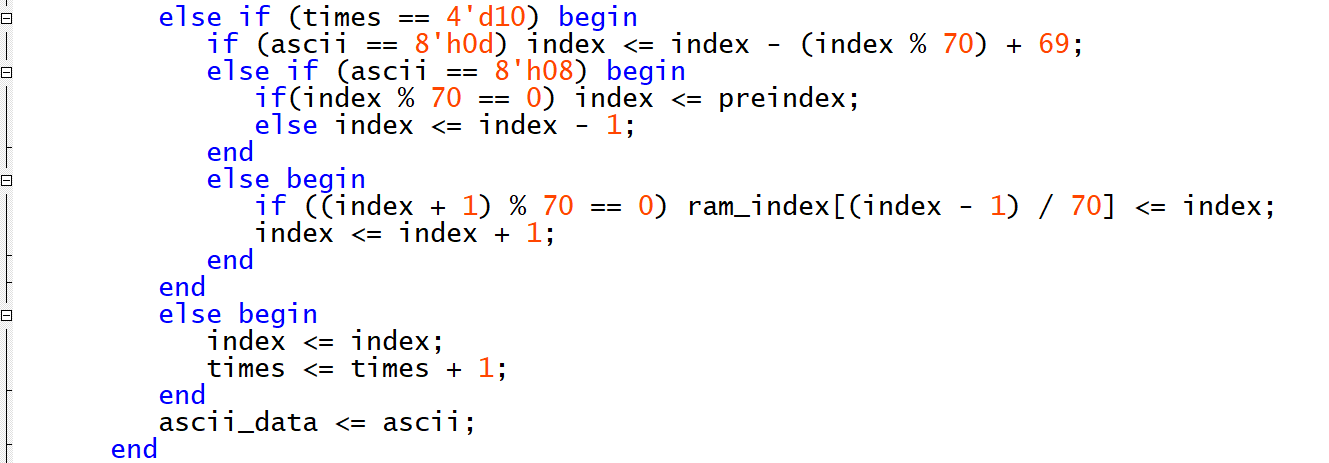
//进入回车键单个换行模式



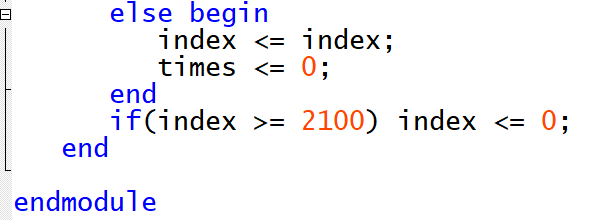
//进入其他按键单个输入模式



//按键超过十个周期，进入连续删除/换行/输入模式



//没有按键时：光标保持，计数器清零，超出屏幕范围归零



实验环境/器材：实验箱一个，笔记本电脑一台，键盘一个，显示器一个。

实验步骤/过程：

分开编写各模块的代码，用七段数码管显示参数，分别进行调试。

将提供的ascii码点阵文本文档编成mif文件，用IP核生成单口rom存储。

用IP核生成双口ram读写（存取）屏幕上的坐标信息。

对键盘和显示器进行调试，观察显示器的输出，对代码进行修改。

测试方法：按下键盘，在显示器上显示字符。

实验结果：字符的显示符合预期。

实验中遇到的问题及解决办法：

时序配合不当，键盘输入在七段数码管上可以正确显示，但显示器没有输出。

修改时序逻辑，增加不同的时钟分别用于键盘和显示器存储的读写。

屏幕显示的字符可以看出符合输入，但看起来边缘总有1-2个像素的错位。

每行最前面输入第一个字符时，会显示两次，其余字符正常。

思考应该是各模块时序配合仍然不够好，然而这种程度的小错误似乎并不需要也不值得对整体框架进行重构，于是决定人为地对下标进行判断，将每行最前面的重复字符的第一个置为空白，加上一个通过观察和调试得出的常数来修正相位差，得到了看似正常的显示效果。

实验得到的启示：无。

意见和建议：无。