**设计名称：**

**组号：9**

**学生1姓名： 廖月伟 实现内容：数据处理、VGA屏显示**

**学生2姓名： 倪圆皓 实现内容：UART传输**

**一、**设计的功能和指标

**1.按键输入模块**

设计按键布局在4x4矩阵按键上，包括0~9，‘+’，‘-’，‘\*’，‘/’，‘（’，‘）’。监测矩阵按键输入，实现按键消抖，并将输入的字符编码为其ascii码。每当按键按下，延迟20ms后将传输给上层的输入标志拉高，并向上层模块传输数据。上层模块用一个寄存器组暂时储存用户输入的计算式数据。

**2.计算模块**

当某个拨码开关开启后，计算模块开始运行。将储存在寄存器组中的用户计算式数据转存到计算模块中。处理计算式字符串，识别数字与字符（+-\*/）并标记；同时进行数字处理，将计算式字符串中数字字符处理为数字：‘4’‘5’=‘45‘。计算处理好的计算式。将结果输出给上层模块。有限状态机（FSM），在不同状态下执行不同的功能，有空闲状态、copy状态（将上层模块中储存的输入计算式复制到此计算模块中（较稳定））、传输状态（处理计算式以及数字处理，将中缀表达式转化为前缀表达式）、计算状态（计算上一步生成的前缀表达式）

**3.VGA屏显示模块**

储存16个字符的点阵数据。将储存在寄存器组中的用户计算式数据实时显示到vga平面上。黑底白字。

**4.UART传输**

将用户输入计算式的计算结果发送到从机。从机uart接收数据并显示到从机的数码管上。实现主机端uart的连续发送，以及从机端的uart连续接收。

二、设计步骤

1.先根据课题要求设计实现不同功能的模块，并分别验证

2.在不同模块设计好后例化到顶层模块

3.仿真检验

4.生成bitstream文件传输到板子上测试

三、设计基本原理

图示

描述已自动生成

总设计RTL

按键输入模块：

4x4矩阵按键及对应字符0~9，+-\*/（），布局设计如下。

黑色的键盘

低可信度描述已自动生成

计算模块：先设计一个有限状态机，一共有5种不同的状态，分别为：空闲、复制、处理、计算、发送结果状态。当拨码开关1开启后，计算模块开始运行，状态机从空闲状态转到复制状态。在复制状态下，计算模块将上层模块中寄存器组储存的计算式复制到计算模块中，保证计算的稳定性。当复制结束，状态机转到处理状态。处理状态中，模块处理计算式字符串，识别数字与字符（+-\*/）并标记；同时进行数字处理，将计算式字符串中数字字符处理为数字：‘4’‘5’=‘45‘。然后新建两个寄存器组作为两个栈，将处理后的中缀表达式转化为前缀表达式。转化为前缀表达式后，状态机转到计算状态。在此状态中，计算前缀表达式的结果。转到结果状态，计算出用户输入计算式结果后，程序拉高计算完成flag，并将计算结果发送给上层模块。

VGA屏显示模块：在通常使用的连接方法里面，15个管脚里面的5个是最重要的，他们包括3个基本红，绿，蓝三条基本色彩线和水平与垂直两条控制线



在VGA视频传输标准中，视频图像被分解为红、绿、蓝三原色信号，经过数模转换之后，在行同步（HSYNC）和场同步（VSYNC）信号的同步下分别在三个独立通道传输。VGA在传输过程中的同步时序分为行时序和场时序。

vga设计两个模块，一个vga屏幕驱动模块，另一个display模块。vga驱动模块直接连接屏幕，提供三条基本色彩线的数据以及行同步和场同步信号。同时驱动模块向显示模块发送此时扫描屏幕的xy轴坐标，并请求此点的颜色数据。显示模块储存32\*16的字符点阵数据。将整个vga屏幕分成32\*16的小方块。将驱动模块输入的xy轴坐标取模（x%32，y%16）得到此时方块应该显示的字符数据。xy轴坐标取模（x%32，y%16）后的余数是该方块中的xy轴坐标，然后就能得到该点的颜色数据。

UART传输：UART采用的是一种起止式的异步的通信协议，所谓起止式就是传输一个字符总是以起始位开始，以停止位结束，特点是一个字符接着一个字符地传输，字符之间没有固定的时间间隔要求。这样做的好处是每一个字符收发双方都可以重新建立同步，哪怕双方的时钟频率略有偏差也没关系，以此保证异步通信的可靠性。

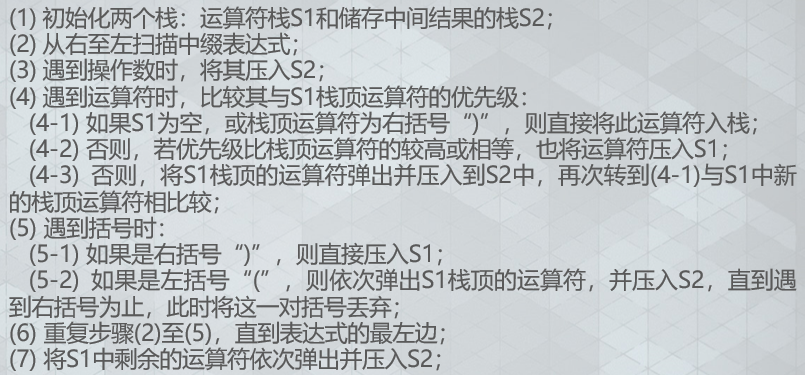
接收过程 —— 接收控制器在允许接收的条件下接收数据，然后进行奇偶校验，无论有无错误都将数据缓存进“接收寄存器”并通知控制电路产生中断，有错误则将“错误寄存器”设置成1，CPU接收到中断，经查询是接收中断，并同时确认了“错误寄存器”有无错误，之后CPU读取“接收寄存器”的值，如果没有错误则该值有效，如果有错误则舍弃掉这个值。控制电路确认CPU读走数据之后告知接收控制器可以继续接收下一个值。

发送过程 —— CPU将一个值写入“发送寄存器”中，然后再向“命令寄存器”写一个命令通知控制电路，表示可以发送数据了，控制电路这时就告知发送控制器进行数据发送，发送控制器计算出奇偶校验位之后将数据发送出去，发送完成之后则通知控制电路产生中断，CPU接收到中断，经查询是发送中断，表示本次发送已经完成，可以发送下一个数据。

除了上面两个基本过程，UART也可以工作在不同的波特率下（这里的波特率表示每秒钟传输的比特的个数，反映到具体设计中就是传输数据的时钟频率），所以在设计时也要考虑到CPU配置特定寄存器控制传输波特率的这个过程。

四、设计关键核心代码分析

计算模块：按照中缀表达式转化为后缀表达式的规则编写代码。



一些文字和图案

中度可信度描述已自动生成

核心代码如下：

如果扫描到计算式字符串的字符为数字

图形用户界面, 文本, 应用程序

描述已自动生成

将数字字符串处理为数字：data\_reg = data\_reg\*10 + data

文本

描述已自动生成

如果扫描到计算式字符串的字符为符号

图形用户界面, 文本, 应用程序

描述已自动生成

如果扫描到计算式字符串的字符为‘（‘’）‘

文本

描述已自动生成

vga显示模块：

图片包含 图示

描述已自动生成

vga设计两个模块，一个vga屏幕驱动模块，另一个display模块。vga驱动模块直接连接屏幕，提供三条基本色彩线的数据以及行同步和场同步信号。同时驱动模块向显示模块发送此时扫描屏幕的xy轴坐标，并请求此点的颜色数据。显示模块储存32\*16的字符点阵数据。图形用户界面, 文本, 应用程序

中度可信度描述已自动生成

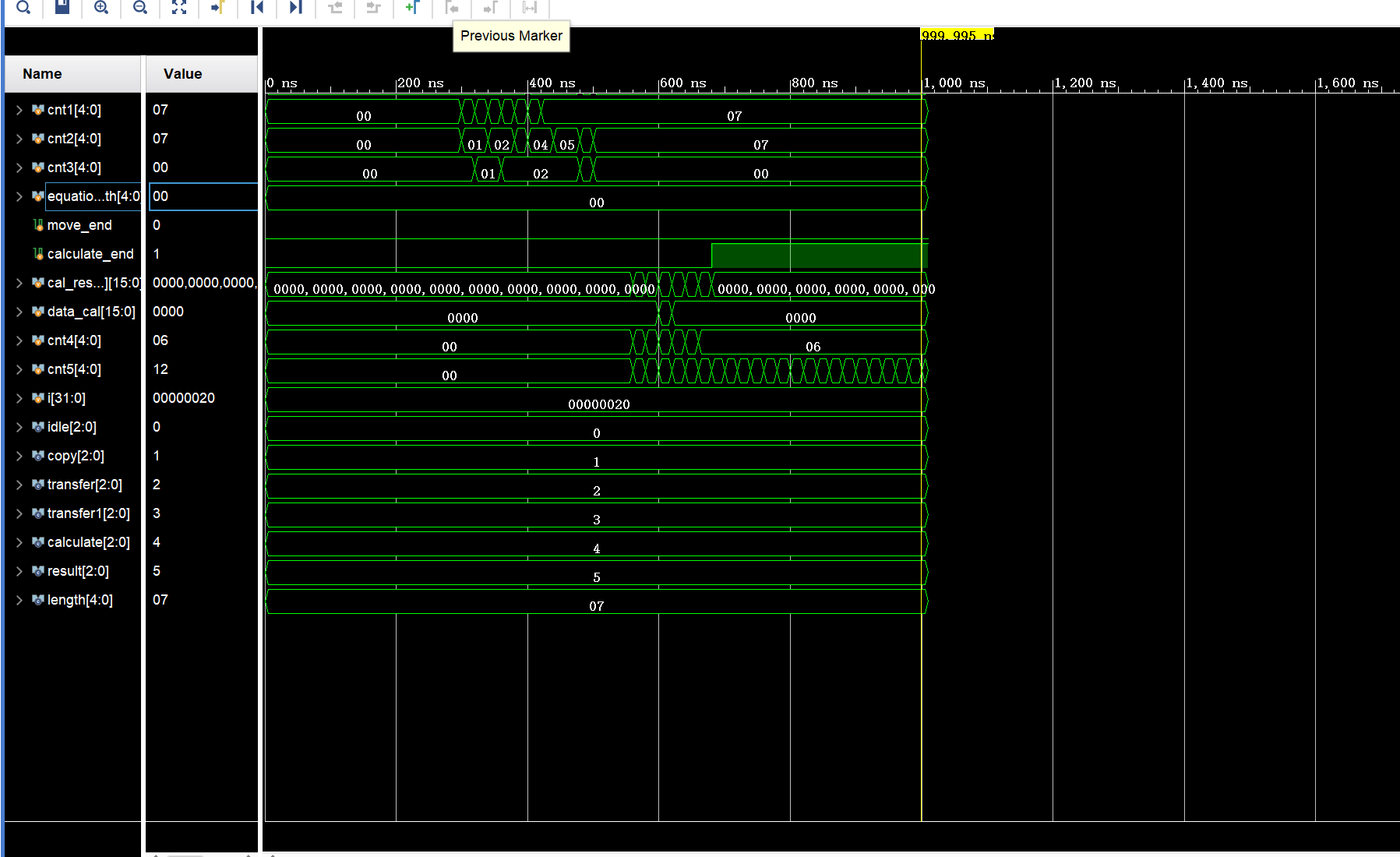
将整个vga屏幕分成32\*16的小方块。将驱动模块输入的xy轴坐标取模（x%32，y%16）得到此时方块应该显示的字符数据。xy轴坐标取模（x%32，y%16）后的余数是该方块中的xy轴坐标，然后就能得到该点的颜色数据。

uart传输模块：

五、仿真（如有）和实现结果

1.仿真结果

计算模块仿真：



电视游戏的萤幕截图

描述已自动生成

仿真输入用户数据为：1+(1+23)\*(45+67)-89.模块输出数据为2600，计算无误。

uart仿真：

2、下载到实验板查看结果

六、心得体会及问题