

（深圳）

实验作业

开课学期： 2021春季

课程名称：计算机组成原理（实验）

实验名称： 原码除法器设计

实验性质： 综合设计型

实验学时： 4 地点： T2615

学生班级： 1901105

学生学号： 190110509

学生姓名： 王铭

作业成绩：

实验与创新实践教育中心制

2021年4月

|  |
| --- |
| 1. **系统功能详细设计** |
| 设计模块：top，divider\_1m，div\_rr，div\_as，display  整体框架图：    主要功能：top为顶层模块，模块内部分别调用div\_rr和div\_as模块计算x除以y的余数和商得到z1,z2,r1,r2，调用divider\_1m模块产生分频时钟，最终将产生的时钟和计算得到的结果作为参数调用display模块将计算的结果z1,z2,r1,r2显示到数码管上。  **div\_rr模块：**    输入：时钟信号clk，复位信号rst\_n，被除数x和除数y，开始信号start  输出：加减交替法计算x/y的商z1和余数r1.  重要变量：flag,Syn\_res,div\_reg  div\_reg寄存器用于保留计算过程中的余数和商，Syn\_res表示商的符号,flag为1表示模块开始运算。  在时钟上升沿或复位信号有效时，先判断复位信号是否有效，有效则将flag,div\_reg,Syn\_res均置0。否则判断start信号是否有效，若start信号有效，置flag为1,div\_reg置为x的真值，Syn\_res置为x,y符号位的异或。当flag为1且start为0时，对div\_reg左移一位后，取div\_reg的高8位加，若结果符号位为1，则div\_reg的前8位加,div\_reg[0]置0；若符号位为0，div\_reg置1。此过程由cnt控制完成7次。cnt为0时，判断余数的符号位div\_reg的最高位是否为1决定是否恢复余数，将Syn\_res与div\_reg的[13:7]位组合形成r1，将x的最高位与div\_reg的低7位[6:0]组合形成z1。整个过程中busy1的值与flag相同。  **div\_as模块：**    输入：时钟信号clk，复位信号rst\_n，被除数x和除数y，开始信号start  输出：加减交替法计算x/y的商z2和余数r2.  重要变量：plus\_reg，Syn\_reg,bflag  在时钟上升沿或复位信号有效时，先判断复位信号是否有效，有效则将flag,plus\_reg,Syn\_reg均置0。否则判断start信号是否有效，若start信号有效，置bflag为1,plus\_reg置为x的真值，Syn\_reg置为x,y符号位的异或。当bflag为1且start为0时，对plus\_reg的最高位判断，若符号位为1，则对plus\_reg左移一位后,plus\_reg的前8位加,若符号位为0，则对plus\_reg左移一位后，plus\_reg的前8位加,最终判断plus\_reg的最高位，为1则置plus\_reg[0]为0否则置1。此过程由cnt控制完成7次。cnt为0时，判断余数的符号位plus\_reg的最高位是否为1决定是否恢复余数，将Syn\_reg与plus\_reg的[13:7]位组合形成r2，将x的最高位与plus\_reg的低7位[6:0]组合形成z2。整个过程中busy2的值与bflag相同。  **divider\_1m模块:**    输入:时钟信号clk  输出:1ms的时钟信号clk\_i  设置计数器counter的值，控制生成时钟的周期为1ms  **display模块:**    输入：时钟信号clk，恢复余数法和加减交替法计算的商和余数z1,r1,z2,r2，复位信号rst\_n，busy  输出：led0\_en,led1\_en,led2\_en,led3\_en,led4\_en,led5\_en,led6\_en,led7\_en,led\_ca,led\_cb,led\_cc,led\_cd,led\_ce,led\_cf,led\_cg,led\_dp  display相当于一个存储器，根据输入的z1,r1,z2,r2，找到其相应的16进制数在7段数码管上的显示。  **top模块:**    输入：时钟信号clk，复位信号rst\_n，被除数x和除数y，开始信号start  输出：7段数码管的控制信号led0\_en,led1\_en,led2\_en,led3\_en,led4\_en,led5\_en,led6\_en,led7\_en,led\_ca,led\_cb,led\_cc,led\_cd,led\_ce,led\_cf,led\_cg,led\_dp  顶层模块，调用div\_as,div\_rr,divider\_1m,display模块。 |
| 1. **除法器算法流程图** |
| **恢复余数法：**    **加减交替法：** |
| 1. **调试报告** |
| **仿真核心代码：**  #10 rst\_n = 1'b1;  #20 begin x = 8'b1001\_0101;y = 8'b0000\_0100; end //x为-21 y为4 输出85 81  #50 start = 1'b1;  #20 start = 1'b0;    #100 begin x = 8'b0001\_1111;y = 8'b1000\_0111; end //x为31,y为-7 输出84 03  #50 start = 1'b1;  #20 start = 1'b0;    #100 rst\_n = 1'b0;  #50 rst\_n = 1'b1;  #100 begin x = 8'b0000\_1101;y=8'b0000\_0011;end //x是13,y是3 输出04 01  #50 start = 1'b1;  #20 start = 1'b0;  #100 begin x = 8'b1111\_1111;y=8'b1001\_0100;end//x是-127,y是-20 输出06 87  #50 start = 1'b1;  #20 start = 1'b0;  **出现的问题：**  **1.不能重复计算多组测试数据。**  仿真截图：    原因：在未按复位键的条件下，程序按下start键后，未复位计数器的值为7，导致算法无法正确运行，z2,r2输出错误。  解决方案：在start==1的情况下，设置计数器cnt的值为7  **2.部分测试用例余数符号出现错误。**  仿真截图：      原因：余数的符号，与被除数相同而非x和y符号位的异或。  解决方案：将余数的符号r[7]直接表示为x[7]  解决上述问题后最终仿真截图：      时序分析：start为1且时钟处于上升沿时，立即输出busy信号为1，start为0时开始计算，cnt初始化为7，每一个时钟上升沿减一，7个周期后输出答案85,81(即商为-5,余数为-1)，期间busy信号保持为1。最终结果的输出时间可一直维持到下一次计算结束。    第二组测试数据，同样start为1且时钟处于上升沿时，输出busy为1，start为0后的7个周期后，输出结果84,03（即商为-4，余数为3），期间busy信号保持为1    遇复位信号低电平有效，全部复位，清零。    第三组测试数据，同样start为1且时钟处于上升沿时，输出busy为1，start为0后的7个周期后，输出结果04,01（即商为4，余数为1），期间busy信号保持为1    第四组测试数据，同样start为1且时钟处于上升沿时，输出busy为1，start为0后的7个周期后，输出结果06,87（即商为6，余数为-7），期间busy信号保持为1 |