

（深圳）

实验报告

开课学期： 2024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： 设计型

实验学时： 4 地点：

学生班级： 10

学生学号： 220111016

学生姓名： 杨绵绵

作业成绩：

实验与创新实践教育中心制

2024年5月

|  |
| --- |
| 1、系统功能详细设计 |
| 要求：描述系统主要功能，绘制硬件模块框图，并结合模块框图描述各模块之间的相互关系。*\*若完成了附加题，则需要绘制Booth乘法器硬件模块框图，并结合模块框图描述各模块之间的相互关系。* |
| 除法器：  如图，clk、rst、start信号连接至控制逻辑模块，产生busy信号、count\_reg计数信号这两个内部信号用于控制运算；输入x、y连至运算模块（移位+加法器），产生divisor\_reg、dividend\_reg、divisor\_abs，同时通过异或xy的符号位产生quotient\_sign商符号位，通过x的符号位产生remainder\_sign余数符号位；  运算模块部分（移位模块+加法器模块）：移位模块负责dividend\_reg、quotient\_reg、remainder\_reg的移位；移位模块将remainder\_reg信号输出给加法器模块，加法器根据remainder\_reg信号的符号位、通过组合逻辑对remainder\_reg进行加或减y\*补码，产生remainder\_reg\_add信号再传输给移位模块；    如图，clk、rst、start信号连接至控制逻辑模块，产生busy信号、count\_reg计数信号这两个内部信号用于控制运算；输入x、y连至运算模块（移位+加法器），产生multiplier、x\_data，并将x求补后得到x\_data\_2；  运算模块部分（移位模块+加法器模块）：产生一位初始为0的multiplier\_plus，作为Booth乘法的新增一位乘数；移位模块负责multiplier、part\_product的移位；移位模块将part\_product信号输出给加法器模块，加法器根据multiplier[0]和multiplier\_plus的值、通过组合逻辑对part\_product进行加或减multiplier\*补码，产生part\_product\_add信号再传输给移位模块；  最后输出由part\_product与multiplier组成。 |
| 2、除法器算法流程 |
| 要求：绘制除法器算法流程图，并用文字详细描述算法执行过程。*\*若完成了附加题，则需要绘制乘法器算法流程图，并配以文字描述其执行过程。* |
| 如图所示， **初始化**：   * 商（Quotient）寄存器初始化为0。 * 余数（Remainder）寄存器初始化为被除数。 * 除数寄存器保存除数。 * 设定计数器为除数的位数。    **循环操作**：   * 移位：将商寄存器和余数寄存器整体左移一位。 * 比较和减法：如果余数寄存器的值大于或等于除数，则进行减法操作，并在商寄存器中记录一次减法操作（设置对应位为1）。否则，记录一次加法操作（商寄存器对应位为0）。    **结束条件**：   * 重复上述步骤直到计数器减到0。 * 商寄存器中的值即为最终的商，余数寄存器中的值即为最终的余数。   Booth乘法：     **初始化**：   * 将乘数（Multiplier）和被乘数（Multiplicand）存入寄存器。 * 商寄存器初始化为乘数。 * 被乘数寄存器初始化为被乘数。 * 在商寄存器的最低位右边添加一位额外的0。 * 设定计数器为乘数的位数。    **循环操作**：   * **判断商寄存器的最低两位（Q0 和 Q-1）**：   + 如果是 01，则将被乘数加到部分积寄存器中。   + 如果是 10，则从部分积寄存器中减去被乘数。 * **算术右移**：将商寄存器、部分积寄存器和附加位整体右移一位（保留符号位）。 * 计数器减1。    **结束条件**：   * 重复上述步骤直到计数器减到0。 * 最终部分积寄存器和商寄存器中的值即为乘积。 |
| 3、调试报告 |
| 要求：至少分析2个不同的测试用例，且必须包含完整的仿真波形截图及详细的时序分析。*\*若完成了附加题，则需要再分析2个不同的乘法测试用例，且必须包含完整的仿真波形截图及详细的时序分析。* |
| 除法器测试用例1：    dividend\_reg用于记录运算中的被除数，divisor\_reg用于记录运算中的除数；quotient\_reg为商；remainder\_reg为余数，remainder\_reg\_add用于记录运算过程中对remainder\_reg进行加或减divisor之后得到的结果。  如上图，start\_div信号拉高，下一时钟上升沿div\_busy被拉高；div\_busy为高电平期间共经历32个时钟周期，恰为移位次数；div\_busy被拉低后，remainder\_reg\_add作为余数输出，quotient\_reg作为商输出。  最终结果为 remainder\_reg\_add：000000bb; quotient\_reg：002de4c0  与预期结果相吻合。  详细的波形数据如下图：      由波形图和手算数据可以验算得知每一步的操作是正确的。  除法测试用例2：  完整波形图如下：    如上图，start\_div信号拉高，下一时钟上升沿div\_busy被拉高；div\_busy为高电平期间共经历32个时钟周期，恰为移位次数；div\_busy被拉低后，remainder\_reg\_add作为余数输出，quotient\_reg作为商输出。    最终结果为 remainder\_reg\_add：00000001; quotient\_reg：00000005  与预期结果相吻合。  由波形图和手算数据可以验算得知每一步的操作是正确的。  乘法测试用例1：    part\_product用于记录部分积，part\_product\_add用于记录运算过程中对部分积进行加或减被乘数补码之后得到的结果；multiplier为乘数，multiplier\_plus为Booth乘法添加的乘数的额外一位（初始为0）  如上图，start\_mul信号拉高，下一时钟上升沿mul\_busy被拉高；mul\_busy为高电平期间共经历32个时钟周期，恰为移位次数；mul\_busy被拉低后，part\_product与multiplier共同组成结果积；  详细的波形数据如下图：      由波形图和手算数据可以验算得知每一步的操作是正确的。  乘法测试用例2：  完整波形图如下：    如上图，start\_mul信号拉高，下一时钟上升沿mul\_busy被拉高；mul\_busy为高电平期间共经历32个时钟周期，恰为移位次数；mul\_busy被拉低后，part\_product与multiplier共同组成结果积；  详细的波形数据如下图： |
| 由波形图和手算数据可以验算得知每一步的操作是正确的。 |