

# 实验 2-2 报告

学号：2016K8009929060

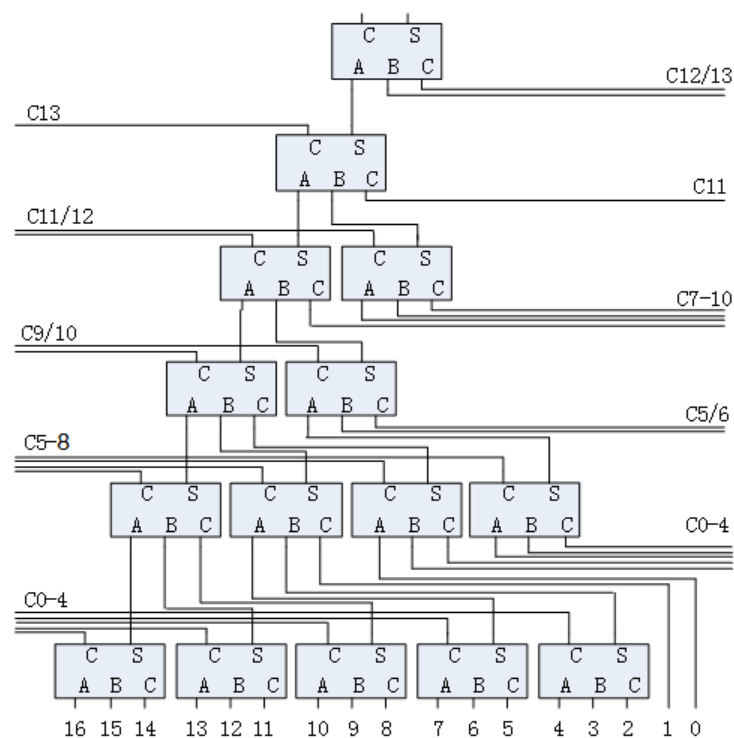
姓名：王晨赳

## 一、实验任务（10%）

在之前实验基础上继续增加指令，并解决数据相关问题。数据相关用前递技术解决。用华莱士树实现乘法，用迭代算法进行除法运算。先对乘法和除法进行模块级检验，验证通过后再添加到实验代码中。之后再进行 CPU 的仿真验证，仿真通过后进行上板验证。

## 二、实验设计（30%）

在先前的指令基础上继续增加指令。新增指令除乘除法和数据移动指令外只需在译码阶段添加相应信号即可。对于乘法指令需要调用专门的乘法器模块实现。乘法器模块 mul 调用 booth 模块及 wallace\_tree 模块，其中 booth 模块负责产生所有部分积，用 booth 算法产生他们；wallace\_tree 模块则负责将这些部分积相加，相加的过程用华莱士树实现。mul 模块有输入时钟信号 mul\_clk，复位信号 resetn，输入的 32 位被乘数 x，32 位乘数 y，控制是否为有符号乘法的输入信号 mul\_signed，以及 64 位输出 result。booth 模块有 3 位的输入 y 用以控制部分积的产生，64 位输入 x，是扩展后的被乘数，输出 64 位部分积 p，以及是否需要+1 的信号 c。wallace\_tree 模块有来自低位的 14 位输入进位信号 Cin，部分积的 17 位输入 Pin，向高位输出的 14 进位信号 Cout，输出最终加法器加数的某一位 S 和 C。下图是 17 个数相加的 1 位华莱士树：



乘法器结构:

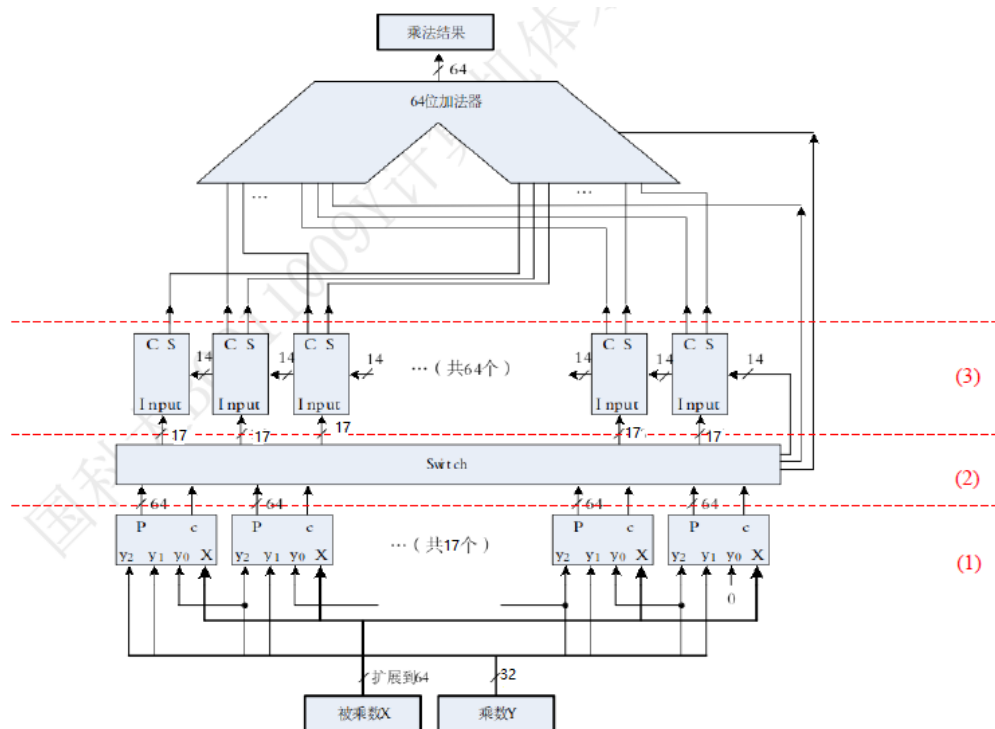


图 2：乘法器结构

一次乘法分为两拍完成，故在乘法指令流水线的的 EX 阶段需要阻塞 1 拍。

除法器模块 div 采用迭代算法，也就是通过一次次试商确定商和余数。div 包含输入时钟信号 div\_clk 和复位信号 resetn。输入信号 div 控制除法器是否工作，输入 div\_signed 控制是否为有符号除法。32 位输入被除数 x，32 位输入除数 y，32 位输出 s，32 位输出 r，以及输出标志除法完成的信号 complete。一次除法从取得操作数到输出结果到 hi 和 lo 寄存器需要 34 拍，故在除法指令流水线的 EX 阶段需要阻塞 33 拍。

对于数据移动指令需要声明两个 32 位寄存器 hi 和 lo，向其写入数据即可。

为了解决数据相关采用前递技术，在当前流水线的 EX 阶段检查源寄存器号是否和 MEM 阶段或 WB 阶段的写回寄存器号一致，若相同且非零，则将 MEM 阶段或 WB 阶段的需写回寄存器的数据前递给 EX 阶段，并使 MEM 阶段的数据优先于 WB 阶段。支持条件转移、存数、mtlo、mthi 指令的数据相关。

### 三、实验过程（60%）

#### （一）实验流水账

- 10.4 20:00~22:00: 看资料，构思
- 10.6 20:00~23:00: 构思，写代码
- 10.7 18:00~24:00: 写代码，调试
- 10.8 14:00~17:00: 调试；18:00~24:00: 调试

#### （二）错误记录

具体描述实验过程中的错误，环境问题、仿真阶段、上板阶段的都可以记录。

##### 1、错误 1

### (1) 错误现象

控制台报错，写回寄存器号不对。

### (2) 分析定位过程

观察波形，是与数据相关有关的指令，检查前递的相关逻辑。

### (3) 错误原因

判断数据相关时未考虑需要 MEM 或 WB 阶段的寄存器写使能信号为高，MEM 或 WB 阶段不需写回的指令不造成数据相关。

### (4) 修正效果

更改前递控制信号的赋值逻辑，问题解决。

## 2、错误 2

### (1) 错误现象

控制台报错，写回寄存器数据不对。

### (2) 分析定位过程

观察波形，是与数据相关有关的指令，检查前递的相关逻辑。

### (3) 错误原因

根据前递控制信号选择需要前递的数据不对。

### (4) 修正效果

更改前递数据赋值逻辑，问题解决。

## 3、错误 3

### (1) 错误现象

控制台报错，写回寄存器数据不对。

### (2) 分析定位过程

观察波形，是与数据相关有关的指令，检查前递的相关逻辑。无误后检查写回寄存器的代码，以及相应波形。

### (3) 错误原因

根据 WB 阶段的 RegWrite 控制寄存器写入，但数据在下一拍才会写入寄存器。同样地，根据 MEM 阶段的 data\_sram\_en 控制 data\_sram\_en 的读写，但数据在下一拍才会被读出或写入。

### (4) 修正效果

MEM 阶段的 data\_sram\_en 在 EX 阶段即可产生，于是可在 EX 阶段便准备好 data\_sram\_addr 及 data\_sram\_wdata（若必要），在 MEM 阶段时要求读写的数据已经读出或写入。同样 WB 阶段的 RegWrite 在 MEM 阶段便可用，于是在 MEM 阶段就准备好写寄存器号及写数据，在 WB 阶段就已将数据写入。修改相关代码，问题解决。

## 4、错误 4

### (1) 错误现象

控制台报错，写回寄存器数据不对。

### (2) 分析定位过程

观察波形，是与数据相关有关的指令，是与 lw 指令有关的数据相关。

### (3) 错误原因

之前在 MEM 阶段开始从 SRAM 读数据，并在 WB 阶段才将数据读出，因此若遇到的数据相关时 lw 指令造成的，需要在 EX 阶段空等一拍，等待 lw 的数据读出后再将该数据前递。但后来更改代码后再 MEM 阶段就可将数据读出，故无需空等一拍。

#### (4) 修正效果

更改前递控制信号和流水线阻塞控制的代码，问题解决。

### 5、错误 5

#### (1) 错误现象

控制台报错，debug\_wb\_pc 不对。

#### (2) 分析定位过程

观察波形，发现此时 MEM 到 WB 阶段是被阻塞的。

#### (3) 错误原因

被阻塞时，MEM 阶段的 RegWrite 不会往下传，于是 WB 阶段的 RegWrite 就保持原样。但 debug 信号所用的 RegWrite 信号是 WB 阶段的，该值未及时更新导致出错。

#### (4) 修正效果

阻塞状态把 WB 阶段的 RegWrite 置 0，问题解决。

### 6、错误 6

#### (1) 错误现象

控制台报错，写回寄存器数据不对。

#### (2) 分析定位过程

观察波形，是移位指令 srav，检查该指令的译码。

#### (3) 错误原因

译码产生的 ALU 控制信号出错。

#### (4) 修正效果

更改 srav 指令的译码，问题解决。

### 7、错误 7

#### (1) 错误现象

控制台报错，写回级 PC 不对。

#### (2) 分析定位过程

观察波形，是除法指令，需要在 EX 阶段阻塞，检查该指令之后指令的执行过程。

#### (3) 错误原因

EX 阶段阻塞时，后一条指令的 ID 结果不能往下送，再后一条指令取得的 instruction 也不能往下送。错误就是没有阻塞再后一条指令取得的 instruction 往 ID 送的数据通路。

#### (4) 修正效果

阻塞该通路，使得 ID 阶段使用的指令不被覆盖，问题解决。

### 8、错误 8

#### (1) 错误现象

控制台报错，写回寄存器数据不对。

## (2) 分析定位过程

观察波形，是数据移动指令，检查该指令及其前后指令的执行。

## (3) 错误原因

mthi 指令遇到了数据相关，但前递未将其考虑进去。

## (4) 修正效果

更改前递数据赋值逻辑，问题解决。