

浙江大学

本科实验报告

RISC-V 微处理器设计

课程名称： 计算机组成与设计

姓 名： 姚桂涛

学 院： 信息与工程学院

专 业： 信息工程

学 号： 3190105597

指导老师： 屈民军、唐奕

2021 年 12 月 24 日

浙江大学实验报告

专业： 信息工程
姓名： 姚桂涛
学号： 3190105597
日期： 2021 年 12 月 24 日
地点： 教 11-400

课程名称： 计算机组成与设计 指导老师： 屈民军、唐奕 成绩： _____
实验名称： RISC-V 微处理器设计 实验类型： 设计 同组学生姓名： _____

一、 实验目的

二、 实验任务

1. 基本要求

设计一个流水线 RISC-V 微处理器，具体要求如下所述。

(1) 至少运行下列 RV32I 核心指令。

算术运算指令：add、sub、addi

逻辑运算指令：and、or、xor、slt、sltu、di、ori、xori、slti、sltiu

移位指令：sll、srl、sra、slli、srli、srai

条件分支指令：beq、bne、blt、bge、bltu、eu

无条件跳转指令：jal、jalr

数据传送指令：lw、sw、lui、auipc

空指令：nop

(2) 采用 5 级流水线技术，对数据冒险实现转发或阻塞功能。

(3) 在 Nexys Video 开发系统中实现 RISC-V 微处理器，要求 CPU 的运行速度大于 25MHz。

2. 扩展要求

(1) 要求设计的微处理器还能运行 lb、lh、ld、lbu、lhu、lwu、sb、sh 或 sd 等字节、半字和双字数据传送指令。

(2) 要求设计的 CPU 增加异常 (exception)、自陷 (trap)、中断 (interrupt) 等处理方案。

三、 实验原理与模块设计

1. 总体设计

流水线是数字系统中一种提高系统稳定性和工作速度的方法，广泛应用于高档 CPU 的架构中。根据 RISC-V 处理器指令的特点，将指令整体的处理过程分为取指令 (IF)、指令译码 (ID)、执行 (EX)、存储器访问 (MEM) 和寄存器回写 (WB) 五级。如图 30.2 示，一个指令的执行需要 5 个时钟周期，每个时钟周期的上升沿来临时，此指令所代表的一系列数据和控制信息将转移到下一级处理。

图所示为符合设计要求的流水线 RISC-V 微处理器的原理框图，采用五级流水线。由于在流水线中，数据和控制信息将在时钟周期的上升沿转移到下一级，所以规定流水线转移的变量命名遵守如下格式：名称 _ 流水

线级名称。如, 在 ID 级指令译码电路 (decode) 产生的寄存器写允许信号 RegWrite 在 ID 级、EX 级、MEM 级和 WB 级上的命名分别为 RegWrite_id、RegWrite_ex、RegWrite_mem 和 RegWrite_wb。在顶层文件中, 类似的变量名称有近百个, 这样的命名方式起到了很好的识别作用。

2. 流水线 RISC-V 微处理器的设计

根据流水线不同阶段, 将系统划分为 IF、ID、EX 和 MEM 四大模块, WB 部分功能电路非常简单, 可直接在顶层文件中设计。另外, 系统还包含 IF/ID、ID/EX、EX/MEM、MEM/WB 四个流水线寄存器。

2.1 取指令级模块 (IF) 的设计

IF 模块由指令指针寄存器 (PC)、指令存储器子模块 (Instruction ROM)、指令指针选择器 (MUX) 和一个 32 位加法器组成, IF 模块接口信息如表所示。

核心代码如下:

其中

2.2 指令译码模块 (ID) 的设计

指令译码模块的主要作用是从机器码中解析出指令, 并根据解析结果输出各种控制信号。ID 模块主要由指令译码 (Decode)、寄存器堆 (Registers)、冒险检测、分支检测和加法器等组成。ID 模块的接口信息如表所示。

(1) 寄存器堆 (Register) 子模块的设计

寄存器堆由 32 个 32 位寄存器组成, 这些寄存器通过寄存器号进行读写存取。寄存器堆的原理框图如图 30.8 所示。因为读取寄存器不会更改其内容, 故只需提供寄存器号即可读出该寄存器内容。读取端口采用数据选择器即可实现读取功能。

寄存器堆设计还应解决三阶数据相关的数据转发问题。当满足三阶数据相关条件时, 寄存器具有 Read After Write 特性。

RBW 子模块核心代码如下:

实现 Read After Write 寄存堆代码如下:

(2) 指令译码 (包含立即数产生电路) 子模块的设计

该子模块主要作用是根据指令确定各个控制信号的值, 同时产生立即数 Imm 和偏移量 offset。该模块是一个组合电路。

根据操作数的来源和立即数的不同, 将指令细分为不同类型, 并对于不同类型产生不同的控制信号。同时产生立即数和偏移量。

核心代码如下:

(3) 分支检测电路的设计

分支检测电路主要用于判断分支条件是否成立, 在 Verilog HDL 可以用比较运算符 “>”、“==” 和 “<” 描述, 但要注意符号数和无符号数的处理方法不同。在这里, 我们用加法器来实现。

用一个 32 位加法器完成 $rs1Data + (rs2Data) + 1$ (即 $rs1Data - rs2Data$), 设结果为 $sum[31:0]$ 。

确定比较运算的结果。对于比较运算来说, 如果最高位不同, 即 $rs1Data[31] \neq rs2Data[31]$, 可根据 $rs1Data[31]$ 、 $rs2Data[31]$ 决定比较结果, 但是应注意符号数、无符号数的最高位 $rs1Data[31]$ 、 $rs2Data[31]$ 代表意义不同。若两数最高位相同, 则两数之差不会溢出, 所以比较运算结果可由两个操作数之差的符号位 $sum[31]$ 决定。

最终得到符号数与无符号数比较运算的结果:

$isLT = rs1Data[31] \&\& (rs2Data[31]) || (rs1Data[31] \wedge rs2Data[31]) \&\& sum[31]$

isLTU = (rs1Data[31])&&rs2Data[31]||((rs1Data[31]^rs2Data[31])&&sum[31])

最后用数据选择器完成下式即完成分支检测。

$$Branch = \begin{cases} (|sum[31:0]); & SB_type \&\&(funct3 == beq_funct3) \\ |sum[31:0]; & SB_type \&\&(funct3 = bne_funct3) \\ isLT; & SB_type \&\&(funct3 = blt_funct3) \\ isLT; & SB_type \&\&(funct3 = bge_funct3) \\ isLTU; & SB_type \&\&(funct3 = bltu_funct3) \\ isLTU; & SB_type \&\&(funct3 = bgeu_funct3) \\ 0 & others \end{cases}$$

2.3 执行模块 (EX) 的设

2.4 数据存储器模块 (DataRAM) 的设计

2.5 流水线寄存器的设计

2.6 顶层文件的设计

3. 实验设备

- (1) 装有 Vivado 和 ModelSim SE 软件的计算机。
- (2) Nexys Video 开发板一套。
- (3) 带有 HDMI 接口的显示器一台。

四、 实验结果