wen-cpu:基于RISC-V的CPU设计

wenyu

[dingzhiwen18@gmail.com](mailto:dingzhiwen18@gmail.com)

2023/9/2

目录

[1前言 3](#_Toc144634831)

[2 硬件设计 3](#_Toc144634832)

[2.1基于RISC-V的单周期CPU实现 3](#_Toc144634833)

[2.1.1 设计框图 3](#_Toc144634834)

[2.1.2 数据通路 4](#_Toc144634835)

[2.1.3 控制通路 5](#_Toc144634836)

[2.1.4 CPU工作方式 5](#_Toc144634837)

1前言

该CPU设计参考了计算机组成与设计硬件/软件接口（原书第五版），简单实现五级流水线RISC-V架构CPU。

2 硬件设计

## 2.1基于RISC-V的单周期CPU实现

### 2.1.1 设计框图



### 2.1.2 数据通路

**数据通路单元：**一个用来操作或保存处理器中数据的单元。在RISC-V实现中，数据通路单元包括指令存储器、数据存储器、寄存器堆、ALU和加法器。

**PC（程序计数器）：**用于保存当前指令的地址,每个时钟周期通过加法器自增4以获得下一条指令的地址（4字节为32位）。

**指令存储器：**用于保存指令的存储单元，指令内存地址线宽64位，数据位宽32位。

**寄存器组：**包括32个64位寄存器，用来存储常数0、参数、PC值、子程序入口等。

**ALU（算数处理单元）：**是CPU的核心，负责算术运算,如I型指令中的加减法，异或，或，与操作。它读取两个64位输出并产生一个64位输出和zero信号指示结果是否为0。

|  |  |
| --- | --- |
| ALU控制线 | 功能 |
| 0000 | AND |
| 0001 | OR |
| 0010 | Add |
| 0110 | Sub |

**立即数产生单元：**用于将指令中的12位偏移量符号扩展为64位有符号数，将其传输给ALU或PC偏移值。

**数据内存：**用于存储大量数据的存储单元，决定了计算机最大处理数据量。如一个数据内存的位宽是1字节，地址宽度为32，最大数据量为232B=4GiB（1字节（B）=8bit），数据内存宽度为64（8字节）。

### 2.1.3 控制通路

控制通路是根据指令译码的结果来控制数据通路之间数据传输的单元。

|  |  |  |
| --- | --- | --- |
| 信号名 | 无效时效果（置0） | 有效时效果（置1） |
| Branch | 无 | 与zero决定是否进行分支操作 |
| MemRead | 无 | 读内存 |
| MemToReg | 寄存器写数据来自ALU | 寄存器写数据来自内存 |
| MemWrite | 无 | 写内存 |
| ALUSrc | 第二个ALU操作数来自第二个寄存器堆的输出 | 第二个ALU操作数来自指令的低12位复合扩展 |
| RegWrite | 无 | 写数据到寄存器 |
| RegSrc |  |  |

### 2.1.4 CPU工作方式

**取指：**要执行任意一条指令，首先要从指令存储器中取出指令，在RISC-V中指令都是32比特

**译码：**分析指令，控制单元产生控制信号，立即数产生单元生效

**执行：**ALU根据ALU控制单元执行算术运算，分支操作生效。

**访存：**ALU运算结果被送往数据内存的地址端口，仅在该阶段内存被访问，读写操作均由译码阶段的控制单元来决定。

**写回：**所有与内存相关的数据必须在寄存器中进行操作，将load指令从内存读出的数据或ALU运算结果写回寄存器。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 名称 | 字段 | | | | | | 备注 |
| 7位 | 5位 | 5位 | 3位 | 5位 | 7位 |
| R型 | funct7 | rs2 | rs1 | funct3 | rd | opcode | 算数指令 |
| I型 | Immediate[11:0] | | rd | 加载、立即数 |
| S型 | Im [11:5] | rs2 | Im[4:0] | 存储 |
| SB型 | Im [12,10:5] | rs2 | Im[4:1,11] | 条件分支 |
| UJ型 | Immediate[20,10:1,11,19:12] | | | | rd | 无条件跳转 |
| U型 | Immediate[31:12] | | | | rd | 大立即数 |