

# 数字逻辑与处理器基础

## 多周期处理器

cvxbzn

2021 年 6 月 4 日

## 1 数字通路设计

### 1.1 寄存器与多路选择器及其功能

- 寄存器
  - PC: 输入下一状态 PC, 输出当前 PC, 使能信号为 1 时在时钟上升沿写入。
  - 指令寄存器 (IR): 存放从存储器中取出的指令, IRWrite 为 1 时在时钟上升沿写入
  - 数据寄存器 (MDR): 输入存储器的输出 MemData。
  - 临时寄存器 A: 输入为寄存器堆中的 ReadData1
  - 临时寄存器 B: 输入为寄存器堆中的 ReadData2
  - ALU 寄存器 ALUOut: 存放 ALU 输出 ALUOut
- 多路选择器
  - IorD: Instruction or data, 选择读取存储器的数据为指令 (0) 或者数据 (1)。
  - RegDst: 选择写回寄存器堆中的寄存器位置, rt(00),rd(01) 或者 \$ra(10), 其中 \$ra 在执行 jal 指令时使用。
  - MemToReg: 选择写回寄存器堆的数据来源。ALUOut(00), 内存数据 (01), PC(10)。
  - ALUSrcA: 选择 ALU 操作数 1 的数据来源, PC(00), 临时寄存器 A(01), 以及在移位时用到的 Shamt(10)
  - ALUSrcB: 选择 ALU 操作数 2 的数据来源, 临时寄存器 B(00), 常数 4(01), 立即数 (10) 以及移位后的立即数 (11)
  - PCSrc: 选择更新 PC 时 PC 的来源, PC+4(00), 分支指令计算结果 (01) 以及伪直接寻址 (10)

具体情况如下示意图所示, 代码见附件。

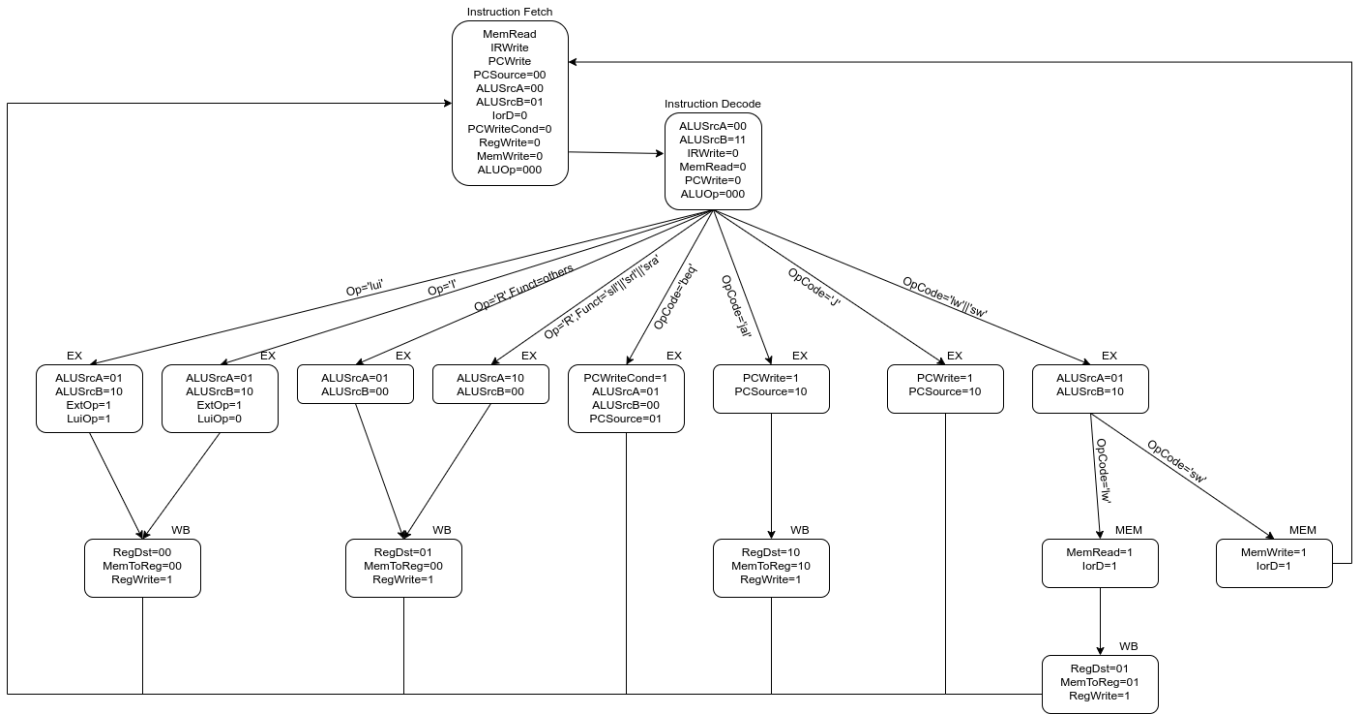
## 1.2 示意图

## 2 控制信号分析与有限状态机实现

### 2.1 控制信号及具体功能

- PCWrite: PC 寄存器的写使能信号; 0-不能写 PC; 1-允许写 PC
- PCWriteCond: 分支指令 PC 写使能信号, 0-无分支指令; 1-有分支指令
- IorD: 选择读取存储器的数据信号, 0-指令, 1-数据
- MemWrite: 内存的写使能信号; 0-不能写内存; 1-允许写内存
- MemRead: 内存的读使能信号; 0-不能读内存; 1-允许读内存
- IRWrite: 指令寄存器的写使能信号; 0-不能写; 1-允许写
- MemToReg: 写回寄存器堆数据来源选择信号; 00-ALUOut, 01-内存数据, 10-PC
- RegDst: 写回寄存器堆的寄存器位置选择信号, 00-rt, 01-rd 或者 10-\$ra
- RegWrite: 寄存器堆的写使能信号; 0-不能写; 1-允许写
- ExtOp: 符号扩展信号, 0-逻辑扩展; 1-算术扩展
- LuiOp: Lui 控制信号, 0-不左移 16 位; 1-左移 16 位
- ALUSrcA: ALU 操作数 1 数据来源选择信号; 00-PC; 01-临时寄存器 A; 10-Shamt
- ALUSrcB: ALU 操作数 2 数据来源选择信号; 00-临时寄存器 B; 01-常数 4; 10-立即数; 11-移位后的立即数
- ALUOp: ALU 控制信号; 000-ADD; 001-SUB; 100-AND; 101-SLT; 010-取决于 Funct
- PCSrc: PC 更新来源选择信号; 00-PC+4; 01-分支指令计算结果; 10-伪直接寻址

## 2.2 状态转移图



## 3 ALU 功能拓展

### 3.1 setsub 类型和机器码字段内容

setsub rd rs rt : {6'h0,rs[4:0],rt[4:0],rd[4:0],5'h0,6'h28}

### 3.2 ALU verilog 代码修改

### 3.3 仿真结果

## 4 汇编程序分析-1

### 4.1 计算寄存器值

### 4.2 仿真结果

## 5 汇编程序分析-2

### 5.1 程序功能以及代码注释

### 5.2 将这段汇编翻译成机器码并写出

### 5.3 \$a0,\$v0 值

### 5.4 观察、描述并解释 PC、\$a0、\$v0、\$sp、\$ra 如何变化

## 6 异常处理