

## 第四章-中央处理器

### (一)CPU 的功能和组成

CPU 的功能：指令顺序控制、操作控制、时间控制、数据加工。

CPU 的组成：运算器、控制器、数据通路、Cache。

数据通路是指各部件之间通过数据线的相互连接。CPU 执行一条指令，实际上就是由控制器对计算机的部件发操作控制信号、并对数据通路进行设置来实现。

指令执行的三个基本步骤：取址、译码、执行。

### (二)关于模型机

#### 1. 模型机

MIPS 结构的一种简单实现。

#### 2. 所包含的指令

算术逻辑运算指令（R 类型）：add, sub, and, or, slt. Op=0

存储器访问指令（I 类型）：lw(load word, Op=35), sw(store word, Op=43)

等于 0 分支（I 类型格式）：beqz, Op=63

#### 3. 指令字长：4 Bytes (32 bits)

#### 4. 指令格式：

##### (1) R 类型

操作码=0 (6 位)	rs(5 位)	rt(5 位)	rd(5 位)	shamt(5 位)	funct(6 位)
-------------	---------	---------	---------	------------	------------

Op: 操作码字段。

rs: 第一源操作数字段。

rt: 第二源操作数字段。

rd: 目标操作数字段（结果字段）。

shamt: 无用。

funct: ALU 指令的运算函数码字段。

##### (2) I 类型

操作码(6 位)	rs(5 位)	rt(5 位)	adr(16 位)
----------	---------	---------	-----------

rs: 基址寄存器字段。

adr: 偏移量字段。

rt: 对 load 指令，rt 指出的寄存器是存放所取的数据，对 store 指令，是存放要写入存储器的数据，对 beqz 指令来说，是存放被检测的数据。

#### 5. 从计算机组成的角度来看，CPU 的设计步骤：

- 根据各指令的执行步骤来简并数据通路；
- 定义各个部件的控制信号；
- 确定时钟周期；
- 完成控制器的设计。

### (三)逻辑设计的约定和定时方法

#### 1. 逻辑设计约定：

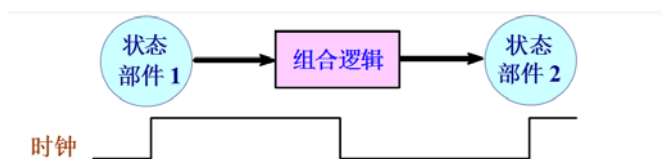
中央处理器设计中应有两种逻辑部件：组合逻辑电路（操作数值）、时序电路（包含状态）。

状态部件包含时钟输入和另外至少一个输入，还有至少一个输出。（2 输入，1 输出）

#### 2. 定时方法：确定什么时候可以进行读或写

采用边缘触发的定时方法。

传输时间 T: 信号从状态部件 1 出发, 经过组合逻辑, 再到达状态部件 2 所需的时间。



➤ T 的最大值确定了时钟周期的大小。

如果一个状态部件每个周期都会进行写操作, 那么它就不需要写信号, 可用时钟信号替代。

#### (四) 实现 MIPS 的一个基本方案

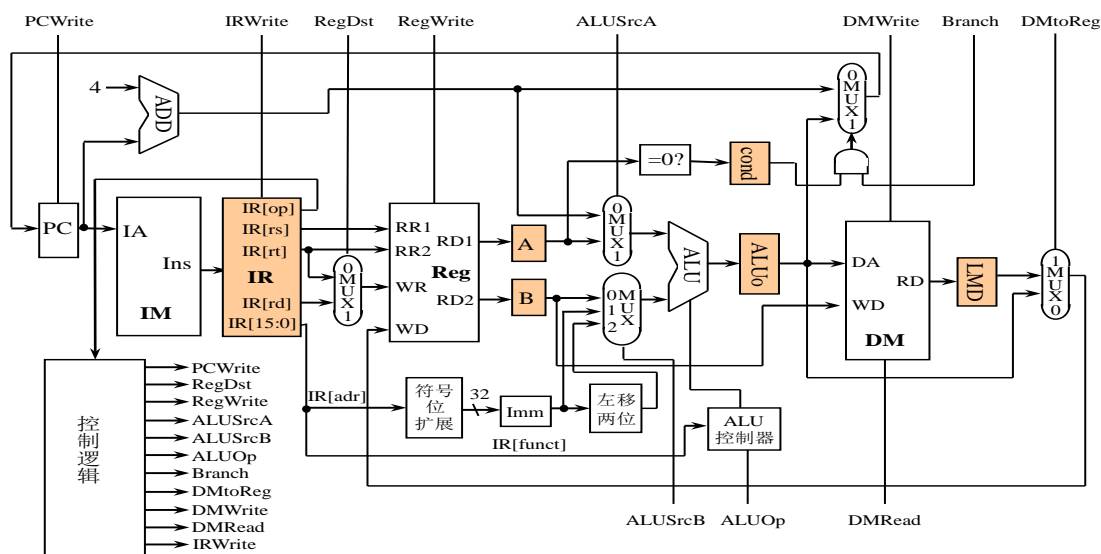
##### 1. 两种数据通路组织方式

- 基于总线的结构
- 直接连接 (✓)

##### 2. 构建数据通路要用到的一些基本构建

- **程序计数器 PC:** 用于指出当前正在执行的指令的地址, 执行一次后加 4 (一条指令 4 Bytes)。
- **指令存储器 IM:** 地址加载到其输入端 IA, 就能在其输出端 Ins 得到相应的指令码。
- **数据存储区 DM:** 有两个输入端 DA(Data Address, 用于给出要写入或者读出的存储单元的地址) 和 WD(Write Data, 用于给出要写入 DM 的数据), 一个输出端 RD, 两个信号 DMRead 和 DMWrite (二者最多只能有一个有效)。
- **通用寄存器组 Reg:** 有两个读端口, 一个写端口, 能同时进行两个读操作和一个写操作。RR1 和 RR2 为两个读操作的地址(5 位), WR 为给出写操作的地址(5 位), WD 为给出要写入的数据(32 位)。两个输出端 RD1 和 RD2 分别给出地址是 RR1 和 RR2 的寄存器的数据(32 位)。RegWrite 为 Reg 的控制信号, 为高时表示对寄存器组进行写操作。
- **ALU:** 输入为两个 32 位的数据, 输出 ALUo 为两个数据进行运算的结果(32 位)。有一个 4 位控制信号 ALUCtrl 确定进行什么操作。
- **加法器 ADD:** 把两个输入的数据相加, 把结果放到输出端 SUM。
- **符号位拓展部件:** 用于把 16 位的数据按符号拓展为 32 位的数据。
- **判 0 部件(=0?):** 输入一个 32 位数据 X, 若  $X == 0$ , 则输出 1(1 位), 否则输出 0(1 位)。

#### (五) 多周期实现方案【暴力背下来】



PC: 程序计数器      IM: 指令存储器      IR: 指令寄存器      Reg: 通用寄存器组  
DM: 数据存储区      A、B、ALUo、LMD、Imm、cond: 临时寄存器

## (六)控制器的设计

### 1. 控制器的一般组成:

**指令部件:** 用于取指令和分析指令。例如程序计数器 PC、指令寄存器 IR、指令译码器 ID 以及地址形成部件。

**时序控制部件:** 用来产生一系列时序信号, 为各个微操作定时, 以保证微操作的顺序执行。例如时钟 CP、时序信号发生器 TSG。

**微操作控制信号形成部件:** 根据指令部件提供的操作控制电位、时序控制部件所提供的各种时序信号, 以及有关的状态条件, 产生计算机所需要的各种微操作的控制信号。

**中断控制逻辑:** 称为中断机构, 用于异常或突发情况的处理。

**程序状态寄存器 PSR:** 存放程序状态字 PSW。

### 2. 控制方式与时序系统

控制方式: **同步控制** (有时间浪费但是控制方便便于调试)、**异步控制** (没有时间上的浪费但是设计难度大而且可靠性不易保证)。

时序系统: **时钟**、**节拍**和**节拍电位**构成了计算机的时序系统。

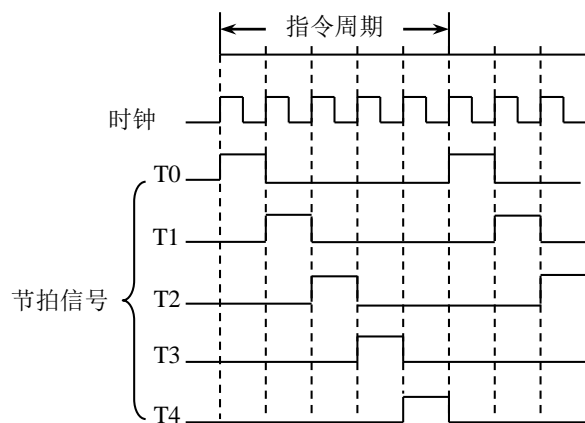
指令周期: 从取指令、分析指令到执行完该指令所需的全部时间。一般由若干个时钟周期组成。

节拍: 一般用具有一定宽度的电位信号表示, 称为节拍电位。

脉冲: 用于寄存器的复位或打入脉冲等。

**指令周期、节拍、脉冲之间的关系:**

每个指令周期包含 5 个节拍 T0~T4, 每个节拍有一个脉冲。如下图所示:



### 3. 模型机控制器的设计

【懒得写了, 看课件 4.6.3 去】