

附1: RISC设计技术简介

CISC — Complex Instruction Set Computer

RISC — Reduced Instruction Set Computer

一、RISC起源

传统处理器设计的难题:

速度、复杂性、设计周期等矛盾

对指令系统进行的研究, 统计表明:

- 软件中大部分指令为简单指令(约80%), 复杂指令只占少数 (约20%);
- 软件中的简单指令约占总运行时间的20%, 复杂指令约占总运行时间的80%;
- 造成控制电路复杂的主要原因复杂指令的存在;

结论: 从指令集中去掉复杂指令, 复杂指令功能由软件实现, 以简化电路设计; 去掉微程序, 采用硬连控制方法, 提高处理器速度。

二、RISC处理器主要特征

1. 简单固定的指令格式

- 指令长度固定: 指令长度无需译码, 简化了电路并节省了长度译码时间; 指令长度一般设定在总线宽度以内, 保证取指令码在一个总线周期完成, 避免了多周期取指造成的流水线阻塞;
- 指令字段位置固定
使得指令译码与取源操作数并行;
- 指令意义简单: 功能单一, 简化硬件逻辑

2. 减少寻址方式和指令数量
作为简化硬件逻辑的措施之一。

3. 流水线(或超级流水线)

尽量使指令在单周期执行完成;

RISC的设计思想更利于指令按流水线方式的运行。

4. 大容量高速缓存

节省的芯片面积有利于集成大容量高速缓存;
缓存更多的指令和数据,减少访存次数。

5. 大量寄存器

减少访存,提高执行速度;上下文切换尽可能在寄存器中完成。

6. 硬连控制(去掉微程序)

以简化的指令集为基础, 提高指令执行速度。

7. 采用存取式体系结构(Load/Store结构)

仅专门的访存指令才允许访存。

避免执行周期访存造成的流水线阻塞。

8. 哈佛(Harvard)总线结构

采用相互分离的指令cache和数据cache, 使数据存取和指令预取可以并行。

9. 优化编译技术

在编译时, 合理调整指令顺序, 使CPU最大限度地让流水线并行执行。

附2: 指令流水线

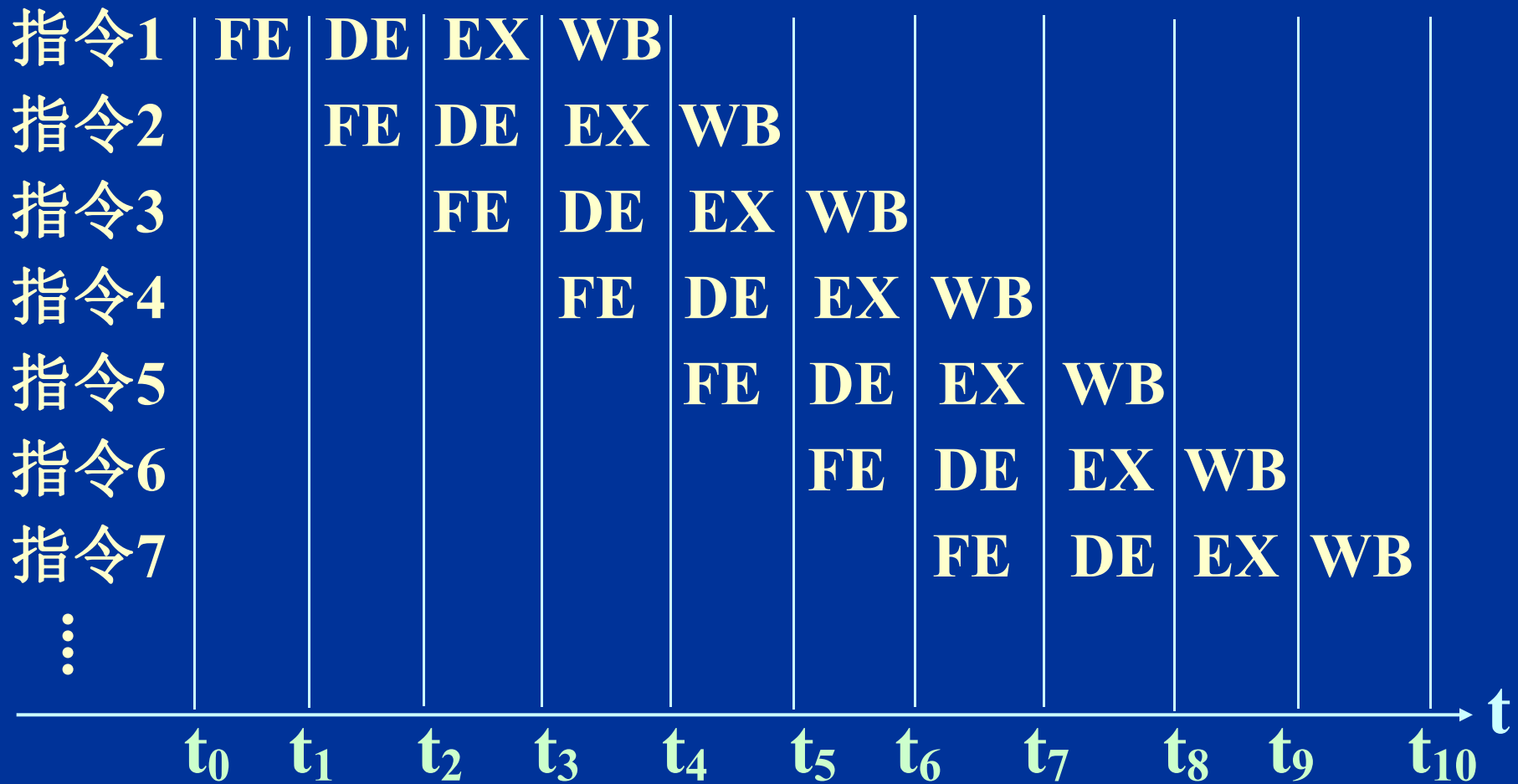
指令分解为多个独立的执行阶段, 每一个阶段所需硬件相对独立。一种基本的分段方式是:

<u>取指</u>	<u>译码</u>	<u>执行</u>	<u>写结果</u>
FE	DE	EX	WB

每一执行阶段的时间长度:

一个总线周期(或者更短)

此时, CPU执行指令的过程如下图所示:



按流水线执行方式的指令执行时间T：

即，假设指令执行分为 m 个阶段(级数)， n 条指令执行完成时间。

如果每一阶段的时间均为 Δt (按照上图, $\Delta t = t_j - t_i$), 则有: $T = m \Delta t + (n-1) \Delta t$

m 是一个常数, 随着 n 不断增大, 当 $n \gg m$, 则有:

$$T \approx (n-1) \Delta t$$

吞吐率: 单位时间内流水线完成的指令或输出的结果数量。

最大吞吐率: 流水线连续流动达到稳定状态后 (流水线中各个阶段都处于工作状态后) 所获得的吞吐率。 $T_{\max} = 1 / \Delta t$

实际吞吐率: 流水线完成 n 条指令的实际吞吐率。

$$T_p = n / T = n / (m \Delta t + (n-1) \Delta t)$$

流水线在实际情况下不能按理想方式执行的主要原因:

- ① 指令码太长 (CISC结构)
- ② 执行阶段时间开销太大 (CISC结构)
- ③ 数据相关 (CISC和RISC结构)

比如: `MOV AX, 8`

PF1	D1	D2	EX	WB
-----	----	----	----	----

`MOV BX, AX`

PF	D1	D2		EX	WB
----	----	----	---	----	----



- ④ 资源冲突(CISC和RISC结构) 阻塞一个周期

- ⑤ 无条件 and 条件转移指令(CISC和RISC结构)

本章小结

1、建立CPU整机概念 组成和结构 工作机制

(1) 逻辑组成

寄存器、ALU设置, 数据通路结构

(2) 工作机制

指令的执行过程:

{ 寄存器传送级: 各类指令的流程
微操作控制级: 微命令序列

● 拟定流程的关键:

熟练掌握数据通路结构

熟练掌握模型机寻址方式

2、基于模型机的控制器设计

(1) 组合逻辑设计

以下面四个方面为基础:

- 寄存器、ALU设置、数据通路结构
- 时序安排
- 各类指令的寻址方式和执行流程
- 微操作控制级—微命令序列

综合微操作信号、写出产生每一微操作信息的逻辑表达式, 简化表达式并用逻辑电路实现。

(2) 微程序设计

以下面四个方面为基础:

- 寄存器、ALU设置, 数据通路结构
- 时序安排
- 各类指令的寻址方式和执行流程
- 微操作控制级—微命令序列

- 拟定微命令的格式以及编码方式;
- 确定后续微地址形成方式;
- 按指令类型和所有指令执行过程所需微操作进行编码(包括后续地址或后继地址形成方式), 形成微指令。

3、有关的基本概念

(1) 微命令的产生方式

- 组合逻辑控制方式:
基本思想、优缺点、应用场合
- 微程序控制方式:
基本思想、优缺点、应用场合

(2) 时序控制方式

同步控制方式: 定义、特点、应用场合

异步控制方式: 定义、特点、应用场合

同步控制的变形(半同步方式):

定义、特点、应用场合

(3) 主机与外设的信息传送控制方式

- 中断方式: 定义、流程、应用场合
- DMA方式: 定义、流程、应用场合

第三章 复习提纲

1. CPU组成(重点是数据通路结构)。
2. 组合逻辑设计: 指令流程(寄存器传送级), 操作时间表(微命令序列)。
3. 微指令设计方法: 分段原则、各段功能、编码方法、微地址形成方法)。
4. 基本概念如: 同步控制与异步控制(含义、应用场合), 组合逻辑控制与微程序控制(基本思想、优缺点、应用场合), I/O传送控制方式(定义、应用场合), 控制字, 状态字, 程序状态字, 主设备, 从设备等。