

2024 考研 408

# 计算机组成原理

## 复习笔记

### 【考查目标】

1. 理解单处理器计算机系统中主要部件的工作原理、组成结构以及相互连接方式。
2. 掌握指令集体系结构的基本知识和基本实现方法，对计算机硬件相关问题进行分析，并能够对相关部件进行设计。
3. 理解计算机系统的整机概念，能够综合运用计算机组成的基本原理和基本方法，对高级编程语言 (C 语言) 程序中的相关问题进行分析，具备软硬件协同分析和设计能力。

2023 年 10 月 28 日

# 目 录

<b>1 计算机系统概述</b>	<b>2</b>
1.1 计算机系统层次结构	2
1.2 计算机性能指标	2
<b>2 数据的表示和运算</b>	<b>3</b>
2.1 数制与编码	3
2.2 运算方法和运算电路	3
2.3 整数的表示和运算	3
2.4 浮点数的表示和运算	3
<b>3 存储器层次结构</b>	<b>4</b>
3.1 存储器的分类	4
3.2 层次化存储器的基本结构	4
3.3 半导体随机存取存储器	4
3.4 主存储器	4
3.5 外部存储器	4
3.6 高速缓冲存储器 (Cache)	4
3.7 虚拟存储器	5
<b>4 指令系统</b>	<b>6</b>
4.1 指令系统的基本概念	6
4.2 指令格式	6
4.3 寻址方式	6
4.4 数据的对齐和大/小端存放方式	6
4.5 CISC 和 RISC 的基本概念	6
4.6 高级语言程序与机器级代码之间的对应	6
<b>5 中央处理器 (CPU)</b>	<b>7</b>
5.1 CPU 的功能和基本结构	7
5.2 指令执行过程	7
5.3 数据通路的功能和基本结构	7
5.4 控制器的功能和工作原理	7
5.5 异常和中断机制	8
5.6 指令流水线	8
5.7 多处理器基本概念	8
<b>6 总线和输入/输出系统</b>	<b>9</b>
6.1 总线	9
6.2 I/O 接口 I/O 控制器	9
6.3 I/O 方式	9

# 1 计算机系统概述

## 1.1 计算机系统层次结构

1. 计算机系统的基本组成
2. 计算机硬件的基本组成
3. 计算机软件和硬件的关系
4. 计算机系统的工作原理

“存储程序”工作方式，高级语言程序与机器语言程序之间的转换，程序和指令的执行过程

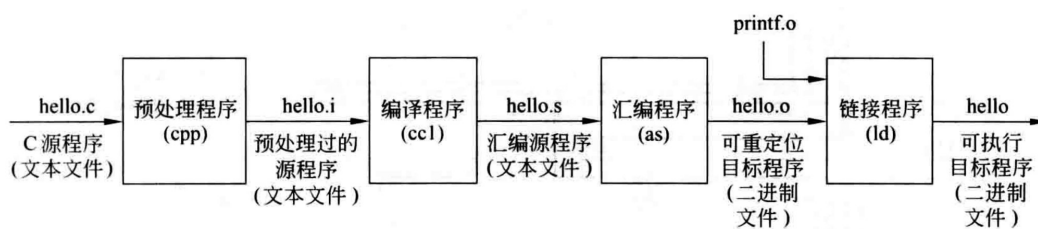


图 1: hello.c 源程序文件到可执行目标文件的转换过程

## 1.2 计算机性能指标

吞吐量、响应时间；

CPU 时钟周期、主频、CPI、CPU 执行时间；

MIPS、MFLOPS、GFLOPS、TFLOPS、PFLOPS、EFLOPS、ZFLOPS

## 2 数据的表示和运算

### 2.1 数制与编码

1. 进位计数制及其数据之间的相互转换
2. 定点数的编码表示

### 2.2 运算方法和运算电路

1. 基本运算部件：加法器，算术逻辑部件 (ALU)
2. 加减运算：补码加/减运算器，标志位的生成

1. 补码运算：Sub = 1 时，减法， $X + \bar{Y} + 1 = [x]_{\text{补}} + [-y]_{\text{补}}$ ，Sub = 0 时，加法， $X + Y = [x]_{\text{补}} + [y]_{\text{补}}$ 。
2. 标志位的生成：
  - 1) 溢出标志 OF：OF=1 表示带符号数运算发生溢出，无符号数运算没有意义。OF =  $C_n \oplus C_{n-1}$ 。
  - 2) 符号标志 SF：结果的符号，无符号数没有意义。SF =  $F_{n-1}$ 。
  - 3) 零标志 ZF：ZF=1 表示结果为 0，无符号/带符号整数都有意义。ZF =  $F = 0$ 。
  - 4) 进位/借位标志 CF：加法时，CF=1 表示无符号数加法溢出，减法时，CF=1 表示有借位，不够减，带符号整数没有意义。CF =  $C_{\text{out}} \oplus C_{\text{in}}$ 。

3. 乘除运算：乘/除法运算的基本原理，乘法电路和除法电路的基本结构

### 2.3 整数的表示和运算

1. 无符号整数的表示和运算
2. 带符号整数的表示和运算

### 2.4 浮点数的表示和运算

1. 浮点数的表示：IEEE754 标准

表 1: IEEE754 浮点数的范围

格式	最小值	最大值
单精度	$(-1)^s \times 1.0 \times 2^{1-127} = (-1)^s \times 2^{-126}$	$(-1)^s \times 1.11 \dots 1 \times 2^{254-127} = (-1)^s \times 2^{127} \times (2 - 2^{-23})$
双精度	$(-1)^s \times 1.0 \times 2^{1-1023} = (-1)^s \times 2^{-1022}$	$(-1)^s \times 1.11 \dots 1 \times 2^{2046-1023} = (-1)^s \times 2^{1023} \times (2 - 2^{-52})$

IEEE 754 格式的浮点数，阶码全 0 或全 1 时，有特别的解释：

- 1) 全 0 阶码全 0 尾数，+0/-0，零的符号取决于数符 S。
- 2) 全 1 阶码全 0 尾数， $+\infty / -\infty$ ，引入无穷大数是为了计算过程出现异常的情况下程序能继续进行。

2. 浮点数的加减运算

溢出判断：

- 1) 右规和尾数舍入都有可能引起阶码上溢。
- 2) 左规可能引起阶码下溢。
- 3) 尾数溢出结果不一定溢出。

### 3 存储器层次结构

#### 3.1 存储器的分类

1. 随机存取存储器 (RAM)：按地址访问，SRAM (Cache)，DRAM (主存)
2. 相联存储器 (TLB)：按内容访问

#### 3.2 层次化存储器的基本结构

#### 3.3 半导体随机存取存储器

1. SRAM 存储器
2. DRAM 存储器
3. Flash 存储器

#### 3.4 主存储器

1. DRAM 芯片和内存条
2. 多模块存储器
3. 主存和 CPU 之间的连接

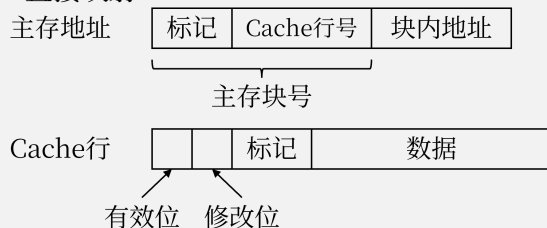
#### 3.5 外部存储器

1. 磁盘存储器
2. 固态硬盘 (SSD)

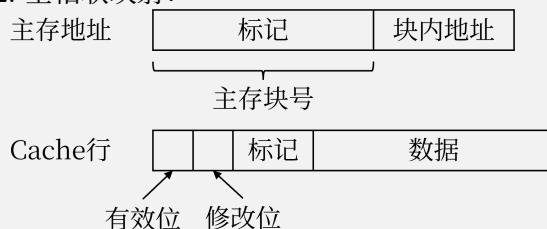
#### 3.6 高速缓冲存储器 (Cache)

1. Cache 的基本原理
2. Cache 和主存之间的映射方式

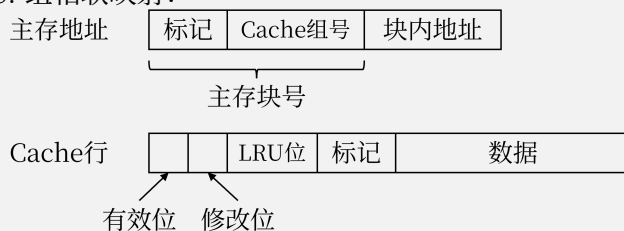
##### 1. 直接映射：



##### 2. 全相联映射：



##### 3. 组相联映射：



3. Cache 中主存块的替换算法
4. Cache 写策略

### 3.7 虚拟存储器

1. 虚拟存储器的基本概念
2. 页式虚拟存储器：基本原理，页表，地址转换，TLB（快表）

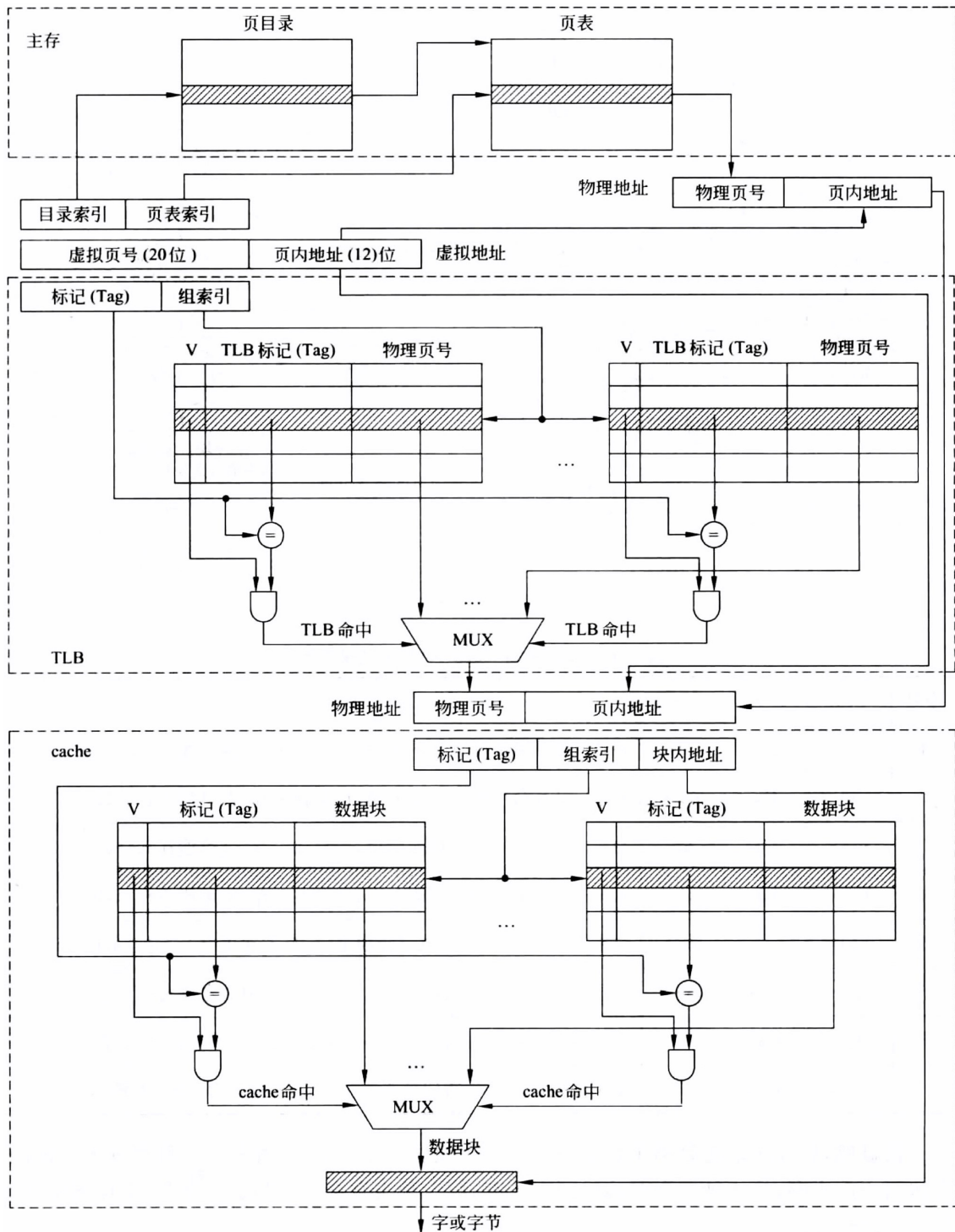


图 2: TLB 和 Cache 的访问过程

3. 段式虚拟存储器
4. 段页式虚拟存储器

## 4 指令系统

### 4.1 指令系统的基本概念

### 4.2 指令格式

循环移位：

- 1) 小循环左移：最高位移入进位标志位，同时也移入最低位。
- 2) 小循环右移：最低位移入进位标志位，同时也移入最高位。
- 3) 大循环左移：最高位移入进位标志位，而进位标志位移入最低位。
- 4) 大循环右移：最低位移入进位标志位，而进位标志位移入最高位。

表 2: 条件转移指令中标志信息

	条件	标志位
无符号整数	$A > B$	$CF = 0 \text{ AND } ZF = 0$
	$A \geq B$	$CF = 0 \text{ OR } ZF = 1$
	$A < B$	$CF = 1 \text{ AND } ZF = 0$
	$A \leq B$	$CF = 1 \text{ OR } ZF = 1$
有符号整数	$A > B$	$SF = OF \text{ AND } ZF = 0$
	$A \geq B$	$SF = OF \text{ OR } ZF = 1$
	$A < B$	$SF \neq OF \text{ AND } ZF = 0$
	$A \leq B$	$SF \neq OF \text{ OR } ZF = 1$

### 4.3 寻址方式

表 3: 寻址方式、有效地址及访存次数

寻址方式	有效地址	访存次数
隐含寻址	程序指定	0
立即寻址	A 即是操作数	0
直接寻址	$EA = A$	1
一次间接寻址	$EA = (A)$	2
寄存器寻址	$EA = R_i$	0
寄存器间接一次寻址	$EA = (R_i)$	1
相对寻址	$EA = (PC) + A$	1
基址寻址	$EA = (BR) + A$	1
变址寻址	$EA = (IX) + A$	1

### 4.4 数据的对齐和大/小端存放方式

### 4.5 CISC 和 RISC 的基本概念

### 4.6 高级语言程序与机器级代码之间的对应

1. 编译器，汇编器和链路器的基本概念
2. 选择结构语句的机器级表示 3. 循环结构语句的机器级表示 4. 过程（函数）调用对应的机器级表示

## 5 中央处理器 (CPU)

### 5.1 CPU 的功能和基本结构

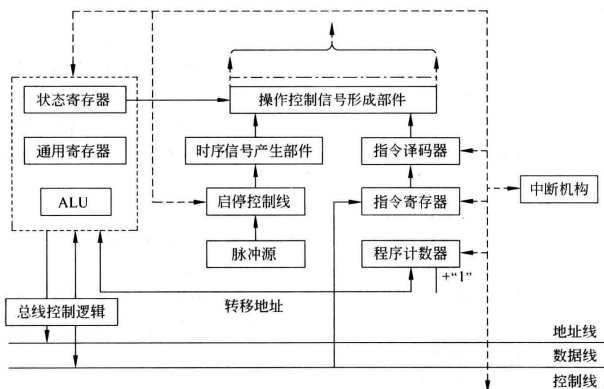


图 3: CPU 基本组成原理图

### 5.2 指令执行过程

### 5.3 数据通路的功能和基本结构

数据通路：指令执行过程中数据所经过的路径，包括路径上的部件称为数据通路。ALU、通用寄存器、状态寄存器、cache、MMU、浮点运算逻辑、异常和中断处理逻辑等都是指令执行过程中数据流经的部件。

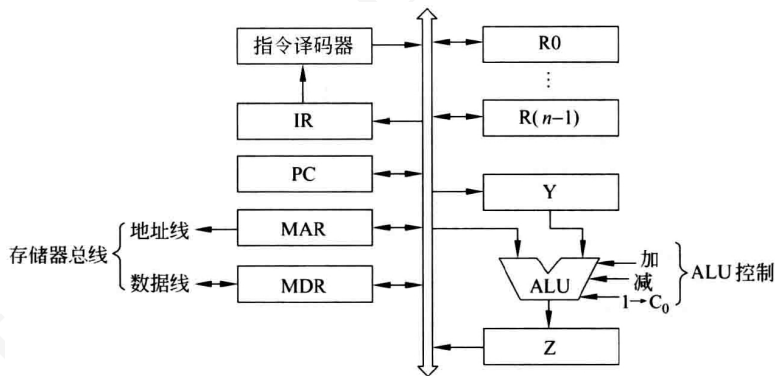


图 4: 单总线数据通路

### 5.4 控制器的功能和工作原理

1. 每条机器指令对应一个微程序，每个微程序包含若干微指令，每条微指令对应一个或几个微操作命令。
2. 微命令是微操作的控制信号，微操作是微命令的执行过程。
3. 控制存储器：在 CPU 内部，由 ROM 构成，存放微程序，按地址访问。
4. 微指令的编码方式：字段直接编码：相容性微命令分在不同段
5. 微指令的格式：
  - 1) 水平型微指令：微程序短，执行速度快；微指令长，编写微程序麻烦
  - 2) 垂直型微指令：微指令短，便于编写；微程序长，执行速度慢，工作效率低



## 5.5 异常和中断机制

1. 异常和中断的基本概念
2. 异常和中断的分类
3. 异常和中断的检测与响应

1. 保护断点和程序状态（PC 和 PSW 寄存器）
2. 识别异常事件并转异常处理

## 5.6 指令流水线

1. 指令流水线的基本概念

理想情况下，每个时钟周期都有一条指令进入流水线，每个时钟周期都有一条指令完成，每条指令的时钟周期数（即 CPI）都为 1。

2. 指令流水线的基本实现
3. 结构冒险、数据冒险和控制冒险的处理

1. 结构冒险：采用数据 cache 和代码 cache 分离的方式
2. 数据冒险：
  - 1) 插入空操作指令：在软件上采取措施，使相关指令延迟执行
  - 2) 插入气泡：在硬件上采取措施，使相关指令延迟执行
  - 3) 采用转发技术（数据旁路）：将数据通路中生成的中间数据直接转发到 ALU 的输入端
3. 控制冒险：
  - 1) 对转移指令进行分支预测，尽早生成转移目标地址
  - 2) 预取转移成功和不成功两个控制流方向上的目标指令

4. 超标量和动态流水线的基本概念

1. 超标量流水线技术：每个时钟周期内可并发多条独立指令，需配置多个功能部件。多数超标量 CPU 都结合动态流水线调度技术，通过动态分支预测等手段提高指令并行性。
2. 超流水线技术：通过提高流水线主频的方式提升流水线性能。

## 5.7 多处理器基本概念

1. SISD、SIMD、MIMD、向量处理器的基本概念
2. 硬件多线程的基本概念
3. 多核处理器 (multi-core) 的基本概念
4. 共享内存多处理器 (SMP) 的基本概念

## 6 总线和输入/输出系统

### 6.1 总线

#### 1. 总线的基本概念 2. 总线的组成及性能指标

1. 数据线用来承载在源部件和目的部件之间传输的数据、命令或地址（数据线和地址线复用）。
2. 地址线用来给出源数据或目的数据所在的主存单元或 I/O 端口的地址，**地址线是单向的**。
3. 控制线用来控制对数据线和地址线的访问和使用，传输定时信号和命令信息。除地址线和数据线以外的通信线都称为控制线。
4. 同步总线采用公共的时钟信号进行定时，适合于存取时间相差不大的多个功能部件之间的通信，同步总线不能过长，否则将会降低总线传输效率。
5. 越来越多的总线采用异步串行方式进行传输。因为串行总线每次在一根信号线上传送数据位，**传输速率可以比并行总线高得多**。每个位各自传输，传输时延的细微变化不会影响其他数据位的传送。通过多个数据通道的组合，可以实现比传统并行总线高得多的数据传输带宽。
6. **总线带宽（总线的最大数据传输率）= 总线宽度（总线上同时能够传送的数据位数）× 总线频率**

#### 3. 总线事务和定时

### 6.2 I/O 接口 I/O 控制器

#### 1. I/O 接口的功能和基本结构 2. I/O 端口及其编址

1. I/O 端口：可被 CPU 直接访问的寄存器，CPU 对数据端口可读可写，状态端口只读，控制端口只写。
2. 编址方式：**统一编址**：I/O 端口当作存储器的单元进行地址分配，使用统一的访存指令访问 I/O 端口；**独立编址**：设置专门的 I/O 指令来访问 I/O 端口。

### 6.3 I/O 方式

1. 程序查询方式
2. 程序中断方式：中断的基本概念；中断响应过程；中断处理过程；多重中断和中断屏蔽的概念

1. 中断的基本概念：
  - 1) “缺页”或“溢出”等异常事件是由特定指令在执行过程中产生的；中断相对于指令的执行则是异步的，中断不和任何指令相关联。CPU 只需要在开始一个新指令之前检测是否有外部发来的中断请求。
  - 2) 异常的发生和异常事件的类型是由 CPU 自身发现和识别的，不必通过外部的某个信号通知 CPU，而 CPU 必须通过对外部中断请求线进行采样，才能获知哪个设备发生了何种中断。
2. 中断响应过程（硬件实现）：
  - 1) 关中断：屏蔽掉所有可屏蔽中断请求。
  - 2) 保护断点：将 PC 和 PSW 送入栈或特殊寄存器。
  - 3) 识别中断源并转中断服务程序：取优先级最高中断源的中断服务程序首地址和初始 PSW，并分别送 PC 和 PSWR。
3. 中断处理过程（软件实现）：
 保存现场和屏蔽字，开中断，执行中断服务程序，关中断，恢复现场和屏蔽字，开中断

#### 3. DMA 方式：DMA 控制器的组成，DMA 传送过程