

## 第一章：

### 名词解释：

**硬件：**是指可以看得见、摸得着的物理设备（部件）实体，一般讲硬件还应包括将各种硬件设备有机组织起来的体系结构。

**软件：**程序（代码）+ 数据 + 文档。由两部分组成，一是使计算机硬件能完成运算和控制功能的有关计算机指令和数据定义的组合，即机器可执行的程序及有关数据；二是机器不可执行的，与软件开发、过程管理、运行、维护、使用 and 培训等有关的文档资料。

**固件：**将软件写入只读存储器 ROM 中，称为**固化**。只读存储器及其写入的软件称为固件。固件是介于硬件和软件之间的一种形态，从物理形态上看是硬件，而从运行机制上看是软件。

**机器字长：**指处理机运算中一次能够完成二进制运算的位数，如 32 位机、64 位机。它决定了计算机的运算精度，通常与主存单元的位数一致。

**指令流：**在**取指周期**中从内存中读出的信息流称为指令流，它通过总线、CPU 内部数据通路**流向控制器**。

**数据流：**在**执行周期**中从内存中读出的信息流称为数据流，它通过总线、CPU 内部数据通路**流向运算器**。

### 填空：

**CPU 执行时间  $t_{cpu}$ ：**表示 CPU 执行一段程序所占用的 CPU 时间，可用下式计算

$$\text{CPU 时间} = \text{时钟周期数} \times \text{时钟周期长} = \text{总时间周期数} / \text{频率}$$

**CPI：**执行一条指令所需要的平均时钟周期数，可用下式计算

$$\text{CPI} = \text{程序所需的 CPU 时钟周期数} / \text{该程序包含的指令条数}$$

**MIPS：**平均每秒执行多少百万条定点指令数，用下式计算

$$\begin{aligned} \text{MIPS} &= \text{指令条数} / (\text{程序执行时间} \times 10^6) \\ &= \text{频率} / (\text{CPI} \times 10^6) \end{aligned}$$

**FLOPS：**平均每秒执行浮点操作的次数，用下式计算

$$\text{FLOPS} = \text{程序中的浮点操作次数} / \text{程序执行时间（秒）}$$

### 冯.诺依曼型计算机的主要设计思想：

（1）由运算器、控制器、存储器、输入设备、输出设备五大部分构成计算机硬件系统概念结构；

（2）采用二进制代码表示数据和指令；

（3）采用存储程序控制方式（指令驱动）。

## 第二章:

### 填空:

**并行加法器的并行进位方式:** 所有进位可以同时产生, 实际上只依赖于数位本身、来自最低位的进位  $C_0$ 。

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0) = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

$$C_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$$

### 加速比:

$$C_k = TL / T_k = (n \cdot k) / (k + (n-1))$$

当任务数很大时, 采用一个任务的完成时间相比

## 第三章:

### 名词解释:

**cache:** 高速缓冲存储器, 高速小容量半导体存储器, 强调快速存取指令和数据, 存取速度快但存储容量小

**虚拟存储器:** 通过硬件/操作系统, 实现主存-外存之间的信息部分调入调出, 为用户提供一个比实际物理内存容量大得多的存储器逻辑空间, 使之成为更大或更多的程序所使用。主存-外存之间的信息部分调入调出过程对用户透明。

### 填空:

**存储器带宽:** 单位时间内从存储器读出的二进制数信息量, 一般用 B/s (字节/秒) 表示,

$$\text{顺序存储器带宽} = (\text{字长} \times \text{模块数 bits}) / (\text{模块数} \times T)$$

$$\text{交叉存储器带宽} = (\text{字长} \times \text{模块数 bits}) / (T + (\text{模块数} - 1) \times t)$$

### Cache/主存系统平均访问时间:

$t_c$  命中 cache 访问时间、 $t_m$  未命中访问时间

$$\text{命中率 } h = N_c / (N_c + N_m)$$

$$\text{平均访问时间 } t_a = h \cdot t_c + (1-h) \cdot t_m$$

$$r = t_m / t_c$$

$$\text{访问效率 } e = 1 / (r + (1-r) \cdot h) = t_c / t_a$$

### 三级存储体系:

高速缓冲存储器 (cache)、主存储器、外存储器

### 大题:

### Cache 基本原理:

cache 是一种高速缓冲存储器, 为了解决 CPU 和主存之间速度不匹配而采用的一项重要技术。基于程序和数据的局部性访问原理, 通过 cache 和主存之间的动态数据块交换, 尽量争取 CPU 访存操作在 cache 命中, 从而总体提高访存速

度。

主存和 cache 均按照约定长度划分为若干块；

主存中一个数据块调入到 cache 中，则将数据块地址（块编号）存放到相联存储器 CAM 中，将数据块内容存放在 cache 中；

当 CPU 访问主存时，同时输出物理地址给主存、相联存储器 CAM，控制逻辑判断所访问的块是否在 cache 中：

若在，则命中，CPU 直接访问 cache。

若不在，则未命中，CPU 直接访问主存，并将该单元所在数据块交换到 cache 中。

#### 第四章：

##### 名词解释：

**指令系统：**一台计算机中所有机器指令的集合，称为这台计算机的指令系统。指令系统是表征一台计算机性能的重要因素，其格式与功能直接影响机器的硬件结构、软件、适用范围等。

**寻址方式：**告诉计算机如何获取指令和运算所需要的操作数。即如何提供将要执行的指令所在存储单元的物理地址；如何提供运算所需要的操作数所在存储单元的物理地址、或者操作数所在内部寄存器的编号。

##### **CISC: Complex Instruction Set Computer, 复杂指令系统计算机**

指令条数多、结构形式复杂多样、寻址方式种类繁多、功能复杂多样、翻译执行效率低、很多指令难得用到。CISC 使计算机的研制周期长，难以保证正确性，不易调试、维护，大量使用频率很低的复杂指令浪费了系统硬件资源。

##### **RISC: Reduced Instruction Set Computer, 精简指令集计算机**

选取使用频率最高的一些简单指令，指令条数少，复杂功能通过宏指令实现；指令长度、格式、结构形式、寻址方式种类少，翻译执行效率高；只有取数/存数指令访问存储器，其余指令的操作均在 CPU 内部寄存器之间进行。

RISC 可缩短计算机的研制周期、易于保证正确性、调试、维护，系统硬件资源使用效率高。

## 第五章：

### 名词解释：

**微操作：**部件接收微命令以后所完成的操作，微操作是执行部件中最基本的、不可再分解的操作。

**微指令：**一组实现一定操作功能的微命令的组合形式，称为微指令。由操作控制和顺序控制两大部分组成。

**指令流水线：**指指令执行步骤的并行。将指令流的处理过程划分为取指令、指令译码、执行、写结果等几个并行处理的过程段。

**算术流水线：**是指运算操作步骤的并行，例如流水乘法器、流水除法器、流水浮点运算器等。

**处理机流水线：**又称为宏流水线，是指程序步骤的并行。由一串级联的处理机构成流水线的各个过程段，每台处理机负责某一特定的任务。

### 填空：

**CPU 的功能：**（1）指令控制（2）操作控制（3）时间控制

**三级时序体制：**按指令周期、CPU 周期、时钟周期进行**分级控制**

**流水线的相关问题：**资源相关、数据相关、控制相关

### 大题：

#### 微程序控制器原理：

（1）设计层面：首先，根据 CPU 的数据通路结构、指令操作定义等，画出每条指令的指令周期流程图（具体到每个时钟周期、微操作、微命令）。然后，根据微指令格式、指令周期流程图编写每条指令的微程序。最后，把整个指令系统的微程序（其中取指令的微程序段是公用的）固化到控制存储器中。

（2）运行层面：首先，逐条执行取指令公用微程序段，控制取指令操作。然后，根据指令的操作码字段，经过变换，找到该指令所对应的特定微程序段，从控制存储器中逐条取出微指令，根据微操作控制字段，直接或经过译码产生微命令（控制信号），控制相关部件完成指定的微操作。一条微指令执行以后，根据微地址字段取下一条微指令。

## 第六章：

### 名词解释：

**总线仲裁：**当总线上的多个主设备（主方）同时竞争使用总线时，必须通过总线仲裁部件，以某种方式和策略选择其中一个主设备（主方），接管总线的控制权，传送信息。

**总线定时：**一次总线操作由若干的事件（基本操作）组成，而且这些事件间具有一定的时序关系，即一个事件什么时候开始、多长时间内完成、事件的先后顺序关系。

**大题：**

**多级总线结构：**

分层次设置三类总线，包括 HOST 总线、PCI 总线、LAGACY 总线，属于多总线结构，实现 I/O 设备按速度的分级管理。

（1）HOST 总线：连接多 CPU、cache、主存。

（2）PCI 总线：与处理器无关的高速外围总线，连接高速的 PCI 设备；

（3）LAGACY 总线：可以是 ISA、EISA、MCA 等传统总线；连接中、低速设备。

（4）南桥、北桥：连接两条总线，使彼此相互通信；总线协议转换；信号缓冲、电平转换、时序和控制逻辑转换等。

**第八章：**

**名词解释：**

**直接内存访问(DMA)方式：**是一种完全由硬件（称为 DMA 控制器）执行、在 I/O 设备和内存之间直接交换批量信息的工作方式，包括 DMA 请求、DMA 响应、DMA 传输、DMA 结束处理等步骤。在 DMA 传输过程中，DMA 控制器从 CPU 接管总线控制权，向内存发出地址和控制信号、修改地址、计数、以中断方式向 CPU 报告 DMA 传输结束，数据交换不经过 CPU，而直接在内存和 I/O 设备之间进行。

**通道：**是一个特殊功能的处理器（基于微处理器 CPU、单片机实现的），它有自己的指令和程序（通道指令、通道程序）专门负责数据输入/输出的传输控制，而 CPU 将“传输控制”的功能下放给通道后只负责“数据处理”功能。CPU 和通道分时使用系统总线和存储器，实现了 CPU 内部运算与 I/O 设备的并行工作。

**大题：**

**中断处理过程：**

（1）中断请求获取

CPU 在一条指令执行完毕后，转入公操作，查询是否有中断请求。

（2）决定是否响应中断请求、寻找中断源

优先级排队：中断优先级排队电路决定是否响应该级中断请求。

寻找中断源：中断响应，沿着指定优先级的菊花链，寻找中断源，并获取中

断向量。

### (3) 中断周期

状态寄存器、断点地址进入堆栈。

关中断，即修改中断屏蔽寄存器 IM（本级及以下的中断请求不予响应，开放本级以上的中断请求）。

查中断向量表，获取中断服务子程序入口地址。

### (4) 中断处理

保护现场。

设备服务。

恢复现场

开中断，即修改中断屏蔽寄存器 IM；

中断返回，状态寄存器、断点地址从堆栈中出栈，返回主程序继续执行。

## DMA 数据传送过程：

(1) 外围设备发出 DMA 请求；

(2) CPU 在指令执行公操作期间，查询是否有 DMA 请求，决定是否响应设备的 DMA 请求；若响应请求，把 CPU 工作改成 DMA 操作方式，CPU 初始化 DMA 控制器（内存起始地址、字个数），I/O 设备准备数据，CPU 继续获得总线控制权；

(3) I/O 设备每准备好发送/接收一个数据后，由 DMA 控制器向 CPU 申请获得一个总线周期的控制权，传输一个字数据，然后释放总线控制权交给 CPU，I/O 设备继续准备；DMA 控制器修改内存地址指针、字计数器；

(4) 数据块传输结束，字计数器溢出，以中断方式向 CPU 报告

(5) CPU 响应 DMA 传送结束的中断请求，善后处理，收回总线控制权，一次 DMA 传送操作结束。

## 第九章：

### 名词解释：

**对称多处理机 SMP：**结构与功能相同的多处理机通过总线或专用网络连接在一起，共享主存和 I/O 设施，由集中操作系统统一管理，提供各处理机的作业/任务调度、以及处理机间的同步，使得多个处理机的存在对用户是透明的，感觉就是一个处理机。

**机群系统：**是由一组完整的计算机（指离开机群系统仍能独立运行自己任务，有自己的操作系统、内存、I/O 设施，一般称为节点），通过标准通信协议、商品化网络（易于连接应用）连接在一起，在机群软件（中间件）统一管理调度下，可作为一个统一的计算资源协同工作，使用户感觉就是一个计算机系统。

填空：

### 体系结构的分类（格林分类法）

- （1）单指令流单数据流 **SISD**：传统单处理机系统
- （2）单指令流多数据流 **SIMD**：向量处理机
- （3）多指令流单数据流 **MISD**：没意义
- （4）多指令流多数据流 **MIMD**：多处理机系统、机群系统