

# 第一章 计算机系统概论

## 1、基本概念

**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、使用和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。

**计算机系统的层次结构：**现代计算机系统是由硬件、软件有机结合的十分复杂的整体。在了解、分析、设计计算机系统时，人们往往采用分层（分级）的方法，即将一个复杂的系统划分为若干个层次，即计算机系统的层次结构。最常见的是从计算机编程语言的角度划分的计算机系统层次结构。

**虚拟计算机：**是指通过配置软件扩充物理机（硬件/固件实现）功能以后所形成的一台计算机，而物理机并不具备这种功能。虚拟机概念是计算机分析设计中的一个重要策略，它将提供给用户的功能抽象出来，使用户摆脱具体物理机细节的束缚。

## 2、计算机的性能指标。

**1 吞吐量：**表征一台计算机在某一时间间隔内能够处理的信息量，用 bps 度量。

**2 响应时间：**表征从输入有效到系统产生响应之间的时间度量，用时间单位来度量。

**3 利用率：**在给定的时间间隔内，系统被实际使用的时间所在的比率，用百分比表示。

**4 处理机字长：**常称机器字长，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。

**5 总线宽度：**一般指 CPU 从运算器与存储器之间进行互连的内部总线一次操作可传输的二进制位数。

**6 存储器容量：**存储器中所有存储单元（通常是字节）的总数目，通常用 KB、MB、GB、TB 来表示。

**7 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。

**8 主频/时钟周期：**CPU 的工作节拍受主时钟控制，按照规定在某个时间段做什么（从什么时候开始、多长时间完成），主时钟不断产生固定频率的时钟信号。主频（主时钟的频率）度量单位是 MHz、GHz；时钟周期（主频的倒数）度量单位是微秒、纳秒。

**9 CPU 执行时间：**表示 CPU 执行一段程序所占用的 CPU 时间，可用下式计算

CPU 时间 = CPU 时钟周期数 × CPU 时钟周期长。

**10 CPI：**执行一条指令所需要的平均时钟周期数，可用下式计算

CPI = 执行某段程序所需的 CPU 时钟周期数 / 该程序包含的指令条数。

**11 MIPS：**平均每秒执行多少百万条定点指令数，用下式计算

MIPS = 指令条数 / (程序执行时间 × 10<sup>6</sup>)

**12 FLOPS：**平均每秒执行浮点操作的次数，用来衡量机器浮点操作的性能，用下式计算

FLOPS = 程序中的浮点操作次数 / 程序执行时间（秒）

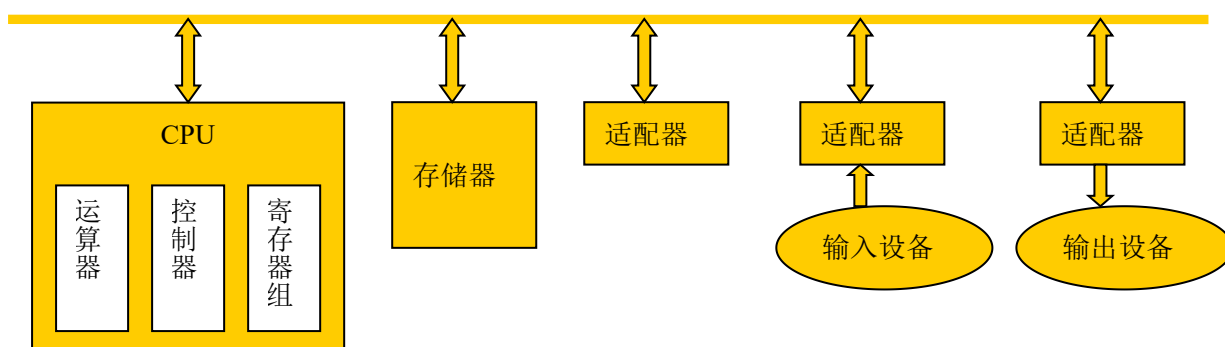
### 3、计算机硬件系统的概念性结构，各个部分的作用。

（1）计算机硬件系统由运算器、控制器、内存储器、输入设备、输出设备五大部分构成，一般还要包括它们之间的连接结构（总线结构）；

（2）将运算器、控制器、若干的寄存器集成在一个硅片上，称为中央处理器 CPU；

（3）由于输入设备、输出设备与 CPU、内存的处理速度差异，所以输入、输出设备通过适配器与总线、CPU、内存连接；

（4）概念性结构如下图所示；



(5) 控制器的作用有三个方面的：存储程序控制方式（指令控制）、操作控制、时间（时序）控制。基本任务如下：

从内存储器中按一定顺序取指令

译码（翻译）、产生控制信号

控制取操作数（源操作数、目的操作数）

控制执行（运算）

控制保存结果

形成下条指令地址（顺序、转移）

(6) 运算器的作用：由算术逻辑运算部件 ALU、寄存器、数据通路组成。实现数据的加工和处理（算术运算、逻辑运算、移位运算、关系（比较）运算、位运算）；

(7) 存储器的作用：存储程序和数据，记忆部件；

(8) 适配器的作用：在主机与 I/O 设备之间起数据缓冲、地址识别、信号转换等；

(9) 总线的作用：多个部件分时共享的信息传送通路，用来连接多个部件并为之提供信息传输交换服务。

**（注：后续章节还会逐步扩充）**

#### 4、指令流、数据流？计算机如何区分指令和数据？

**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。

从**时间**上来说，取指令事件发生在取指周期（取指令阶段），取数据事件发生在执行周期（执行指令阶段）；

从**空间（处理部件）**上来说，指令一定送给控制器，数据一定送给运算器。

#### 5、冯·诺依曼计算机的技术特点

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；

采用二进制代码表示数据和指令；

采用存储程序控制方式（指令驱动）。

# 第二章 运算方法和运算器

## 1、原码、补码、反码、移码的求法及表示范围。

- (1) 首先应明确机器字长；
- (2) 原码、补码、反码、移码的求法；
- (3) 表示范围；

	机器字长=8		机器字长=16	
	定点小数	定点整数	定点小数	定点整数
原码	$-(1-2^{-7})$ 至 $+(1-2^{-7})$	-127 至 +127	$-(1-2^{-15})$ 至 $+(1-2^{-15})$	-32767 至 +32767
补码	-1 至 $+(1-2^{-7})$	-128 至 +127	-1 至 $+(1-2^{-15})$	-32768 至 +32767
反码	$-(1-2^{-7})$ 至 $+(1-2^{-7})$	-127 至 +127	$-(1-2^{-15})$ 至 $+(1-2^{-15})$	-32767 至 +32767
移码	-1 至 $+(1-2^{-7})$	-128 至 +127	-1 至 $+(1-2^{-15})$	-32768 至 +32767

## 2、补码加减法运算，加法运算溢出检测。

- (1) 补码加法运算规则
- (2) 补码减法运算规则
- (3) 变形补码表示法

00 表示正数

11 表示负数

(4) 变形补码运算：规则同补码加减法运算规则，双符号位数值化、参加运算。

(5) 加法运算溢出检测

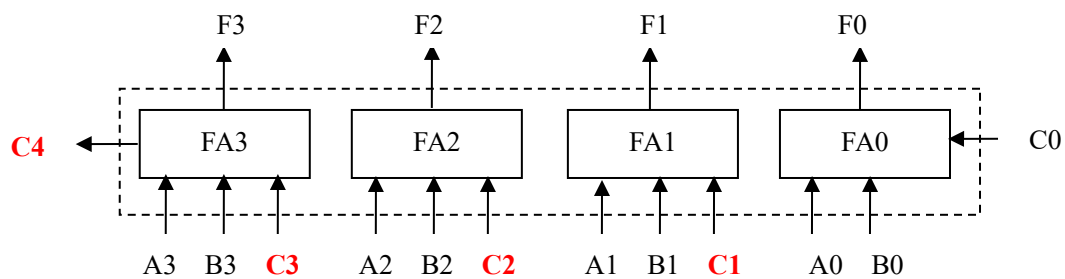
1) 单符号位法

2) 双符号位法

参见例题、习题

### 3、并行加法器的进位方法及逻辑表达式

(1) 直接从全加器的进位公式推导。



(2) 串行进位：某位的运算必须等到下一位的进位传递来以后，才能开始。也就是进位从最低位向最高位逐级传递，速度慢。

$$C1 = G0 + P0C0$$

$$C2 = G1 + P1C1$$

$$C3 = G2 + P2C2$$

$$C4 = G3 + P3C3$$

(3) 并行进位：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 C0。

$$C1 = G0 + P0C0$$

$$C2 = G1 + P1G0 + P1P0C0$$

$$C3 = G2 + P1G1 + P2P1G0 + P2P1P0C0$$

$$C4 = G3 + P3G2 + P3P2G1 + P3P2P1G0 + P3P2P1P0C0$$

其中：

$$G0 = A0B0 \quad P0 = A0 + B0$$

$$G1 = A1B1 \quad P1 = A1 + B1$$

$$G2 = A2B2 \quad P2 = A2 + B2$$

$$G3 = A3B3 \quad P3 = A3 + B3$$

$G_i$ : 进位产生函数, 表示两个数位都为 1

$P_i$ : 进位传递函数, 表示某位上的两个数位有一个为 1, 如果来自低位的进位为 1, 则肯定会产生进位。

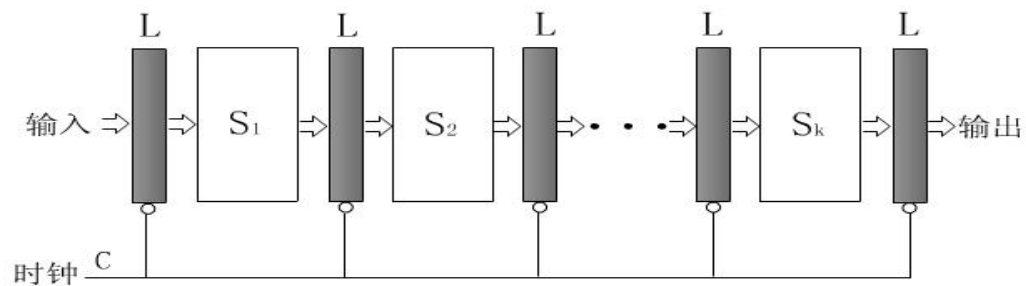
#### 4、浮点加减法运算方法。

- 比较阶码大小、对阶
- 尾数加减法运算
- 规格化处理
- 尾数舍入处理
- 溢出判断

参见例题、习题

#### 5、流水线原理、时钟周期确定、时间公式、加速比、时空图

(1) 把一个任务分割为一系列的子任务, 使各子任务在流水线中时间重叠、并行执行。



过程段  $S_i$  之间重叠执行。

(2) 时钟周期的确定

所有  $S_i$  中执行时间最大者, 参见例 2.32。

(3) 时间公式 (理想)

$$(K+(n-1))T$$

(4) 加速比

- $C_k = T_L / T_k = (n \cdot k) / (k + (n-1))$
- 当任务数很大时, 采用一个任务的完成时间相比, 参见例 2.32。

(5) 流水线时空图

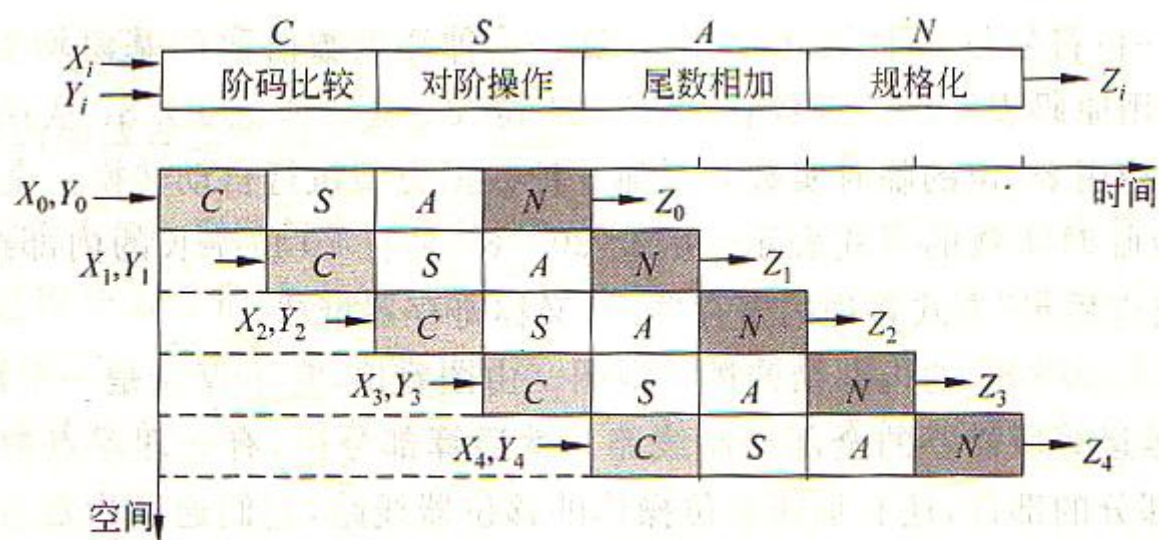
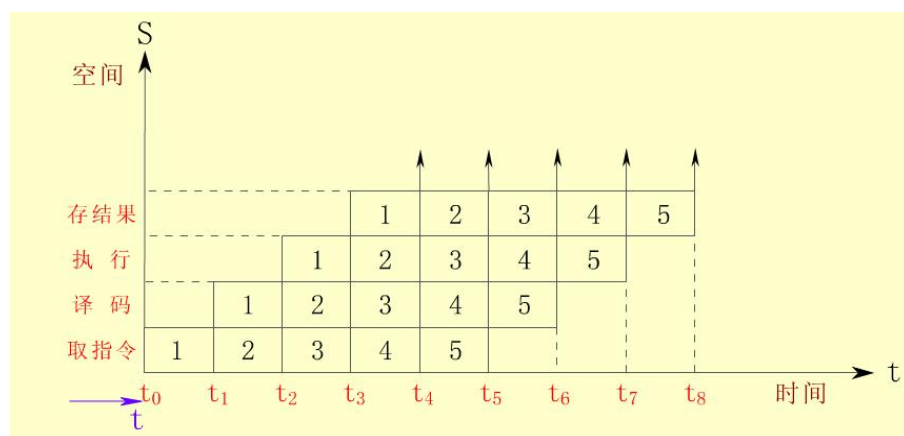


图 2.21 向量加法计算的流水时空图

# 第三章 存储系统

## 1、基本概念

**存储容量：**指一个存储器中可以容纳的存储单元总数。典型的存储单元存放一个字节，因此通常用字节数来表示，KB、MB、GB、TB。

**存取时间：**读操作时间指一次读操作命令发出到该操作完成、数据读出到数据总线上所经历的时间。通常写操作时间等于读操作时间，故称为存取时间。

**存取周期：**也称读写周期，指连续启动两次读/写操作所需间隔的最小时间。通常存储周期略大于存取时间，因为数据读出到总线上，还需要经过数据总线、CPU 内部数据通路传递给控制器/运算器。

**虚拟存储器：**是通过硬件/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，使之能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。

## 2、存储器的分级结构。

对存储器的要求是容量大、速度快、成本低，但是在一个存储器中要求同时兼顾这三个方面的要求是困难的。为了解决这方面的矛盾，目前在计算机系统中通常采用多级存储器体系结构，即高速缓冲存储器、主存储器和外存储器。

CPU 能直接访问高速缓冲存储器 cache 和内存；外存信息必须调入内存后才能为 CPU 进行处理。

(1) **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；

(2) **主存储器：**介于 cache 与外存储器之间，用来存放计算机运行期间的大量程序和数据。要求选取适当的存储容量和存取周期，使它能容纳系统的核心软件和较多的用户程序；

(3) **外存储器：**大容量辅助存储器，强调大的存储容量，以满足计算机的大容量存储要求，用来存放系统程序、应用程序、数据文件、数据库等。

## 3、主存储器的逻辑设计。

**第一步：**根据设计容量、提供的芯片容量构建地址空间分布图（类似搭积木），可能需要字、位扩展；

**第二步：**用二进制写出连续的地址空间范围；

**第三步：**写出各片组的片选逻辑表达式。

**第四步：**按三总线分析 CPU 和选用存储器芯片的数据线、地址线、控制线，以便设计 CPU 与存储器的连接。

**第五步：**设计 CPU 与存储器连接的逻辑结构图。参见例题、习题

## 4、顺序存储器和交叉存储器的定量分析。

顺序存储器： $mT$

交叉存储器：可以使用流水线存取， $T+(m-1)\tau$  参见例题、习题



## 5、高速缓冲存储器 cache 的基本原理，cache 命中率相关计算

Cache 的基本原理：

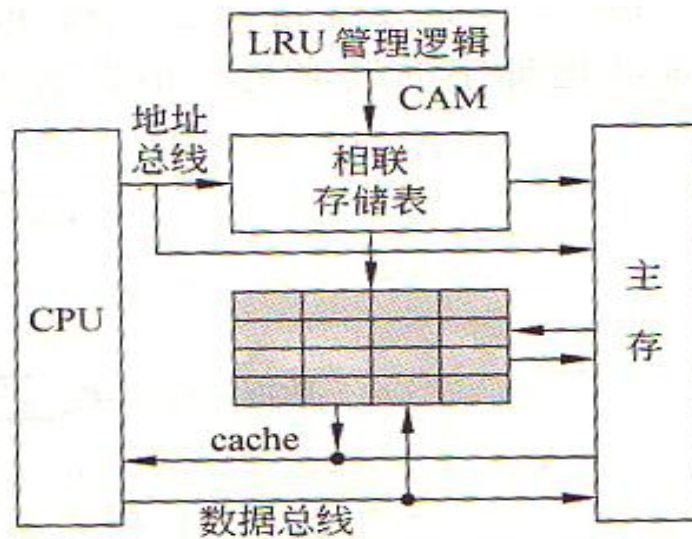


图 3.32 cache 原理图

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。

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。

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。

cache 命中率相关计算：

命中率

主存/cache 系统平均访问时间

访问效率

参加例题、习题。

# 第四章 指令系统

## 1、基本概念

**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：**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：**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。

## 2、指令格式及寻址方式辨析

操作码

地址码

指令字长度

操作数寻址方式

参见例题、习题

# 第五章 中央处理器

## 1、基本概念

**微命令：**控制器通过控制线向部件发送的各种控制信号/操作命令。

**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。

**微程序：**一条机器指令的功能是由多条微指令组成的序列来实现的，这个微指令序列称为微程序

**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。

## 2、CPU 的功能。

### （1）指令控制

控制程序的执行顺序；

由于程序是一个指令序列，这些指令的相互顺序不能任意颠倒，必须严格按照程序规定的顺序进行。

### （2）操作控制

控制器产生取指令、执行指令的所需要的全部操作控制信号，并依序送往相应的部件，从而控制这些部件按指令的要求完成规定的动作。

### （3）时间控制

对各种操作实施时间上的定时；

在计算机中，各种指令的操作信号和整个执行过程均受到时间的严格定时和事件先后顺序控制（应在规定的时间点开始，在规定的时间内结束），以保证计算机有条不紊地自动工作。

### （4）数据加工

完成指令规定的运算操作。

## 3、熟悉典型模型机结构、指令和数据的传输通路

## 4、指令周期流程

（1）根据模型机和数据通路结构，分析指令周期流程。

（2）指令周期流程实际上是一个指令流、数据流在数据通路上的流动过程。

## 5、指令周期、机器周期、时钟周期三级时序

**指令周期：**取出一条指令并执行这条指令所需要的时间。

**CPU 周期：**常称为机器周期，每条指令的执行过程可划分为若干个阶段，如取指令、取源操作数、取目的操作数、执行运算、保存结果等阶段，每个阶段所对应的时间。

**时钟周期：**每个阶段由若干有序的、不可再分的基本操作过程构成，如从数据 cache 中取一个源操作数送到 R0，包括地址→AR、AR→M、M→DR、DR→R0 等基本操作过程，每个操作过程所对应的时间。

显然，指令周期对应一条指令，CPU 周期对应一个阶段，时钟周期对应一个基本操作过程。

因此，时间应按指令周期、CPU 周期、时钟周期进行**分级控制**，这就是三级时序体制。

## 6、微程序控制器的原理及组成框图。

### （1）基本原理

**设计阶段：**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。

**运行阶段：**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

## (2) 构成框图

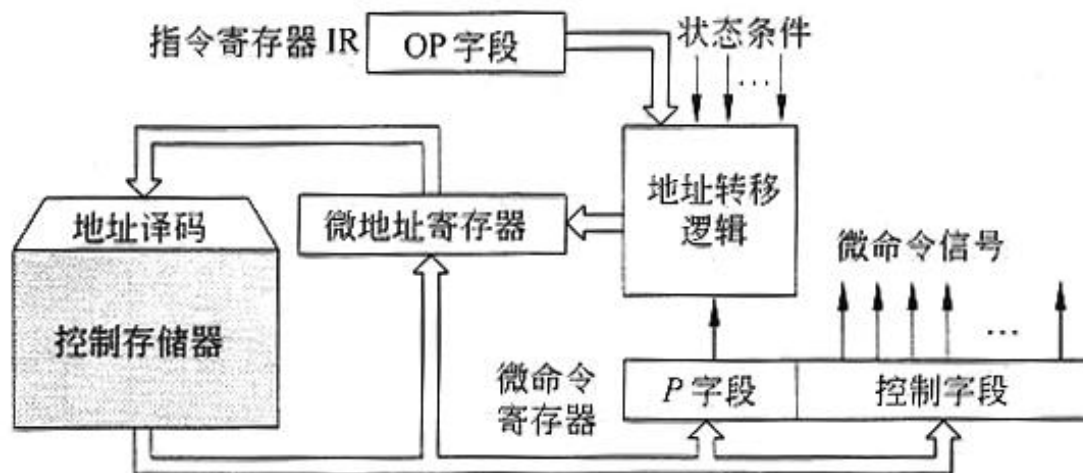


图 5.23 微程序控制器原理框图

**控制存储器 ROM:** 存放全部指令系统的微程序;

**微地址寄存器 uPC:** 具有自动增量功能, 给出顺序执行的下条微指令地址;

**微命令寄存器 uIR:** 存放由控制存储器读出的一条微指令。

**地址转移逻辑:** ①根据指令寄存器 IR 的操作码, 定位到该指令对应的微程序段, uPC 初值; ②如果判断条件 P/状态条件=FALSE, 则  $uPC = uPC + 1$ , 顺序执行; ③如果判断条件 P/状态条件=TRUE, 则 uPC=根据策略形成新的微指令地址, 程序转移。

## 7、流水线中资源相关、数据相关、控制相关问题。

**资源相关:** 是指多条指令进入流水线后, 在同一流水线时钟周期内争用同一个功能部件所发生的冲突。

**数据相关:** 在一个程序中, 如果必须等前一条指令执行完毕以后, 才能执行后一条指令 (即后一条指令需要引用前一条指令的结果数据), 那么这两条指令就是数据相关的。

**控制相关:** 控制相关冲突是由转移类指令引起的。当执行转移类指令时, 可能为顺序取下条指令; 也可能转移到新的目标地址取指令。如果流水线顺序取指令, 而程序却需要转移时, 进入流水线的指令并不是将要执行的指令, 或者转移的目标指令可能还没有进入流水线, 从而使流水线发生断流。

# 第六章 总线系统

## 1、基本概念。

**总线:** 总线是一组能为多个部件分时共享的信息传送线, 用来连接多个部件并为这些部件提供信息交换服务。

**总线带宽:** 总线在单位时间内可以传输的数据总量, 即总线的数据传输速率, 单位是兆字节每秒 (MB/s)

**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。

## 2、总线接口的逻辑结构、功能

### 逻辑结构：

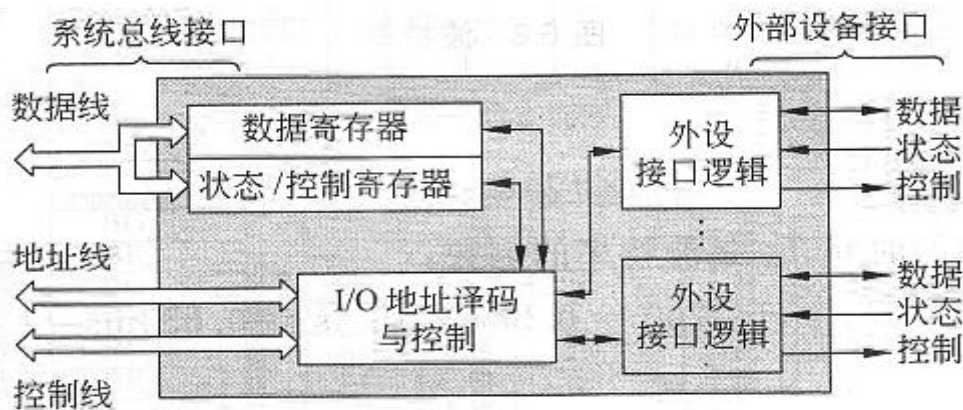


图 6.7 I/O 接口模块框图

**地址信息：** CPU→I/O 设备，寻址目标 I/O 设备。

**控制信息：** CPU→I/O 设备，通知 I/O 设备准备完成什么操作。

**状态信息：** I/O 设备→CPU，反馈设备的状态，如忙/闲、准备好/未准备好。

**数据信息：** 有效数据信息。

### 功能：

#### （1）控制

接口依据 CPU 的指令信息控制外围设备的动作，如启动、关闭设备等。

#### （2）缓冲

在为部设备和计算机系统其它部件之间用作为一个缓冲器，以补偿各种设备在速度上的差异。

#### （3）状态

接口监视外部设备的工作状态并保存状态信息，状态信息包括“准备就绪”、“忙”、“错误”等，供 CPU 询问外部设备时进行分析之用。

#### （4）转换

可以完成任何要求的数据转换，以确保数据能在为部设备和 CPU 之间正确地传送，如数据格式转换、并-串转换等。

#### （5）整理

可以完成一些特别的功能，如在批量数据传输时自动修改字计数器、当前内存地址寄存器。

(6) 程序中断

每当外围设备向 CPU 请求某种动作时，接口即发送中断请求信号给 CPU，申请中断。

3、多总线结构、PCI 总线辨析

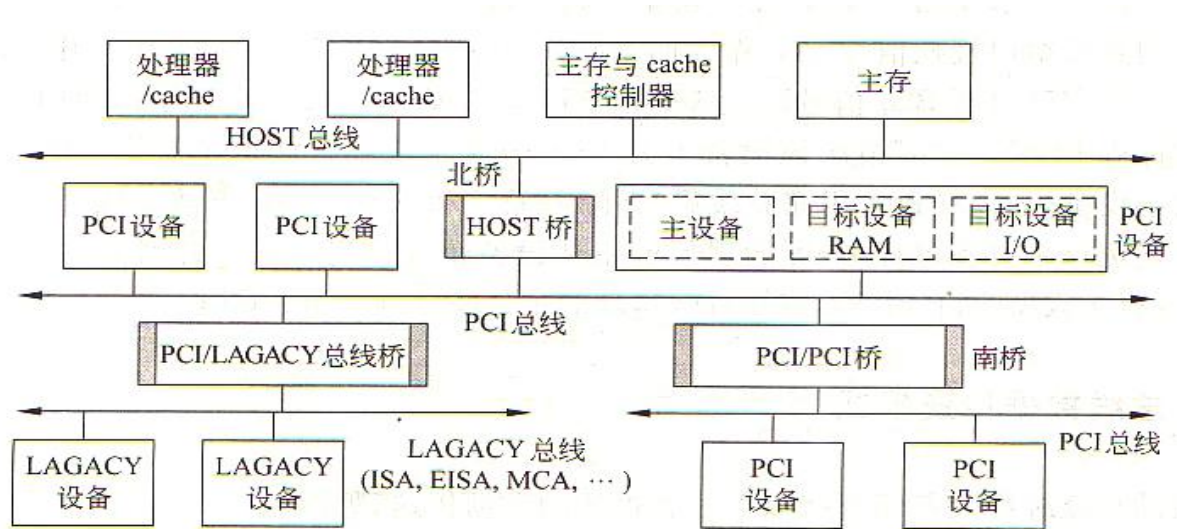


图 6.15 多总线结构框图

**HOST 总线：** 宿主总线，连接多 CPU、cache、主存、北桥。64 位数据线、32 位数据线、同步定时总线。

**PCI 总线：** 与处理器无关的高速外围总线，连接高速的 PCI 设备，32/64 位数据线、32 位地址线、同步定时、集中仲裁、猝发传送。

**LAGACY 总线：** 遗留总线，可以是 ISA、EISA、MCA 等传统总线，连接中、低速设备，保护用户以前的投资。

**桥的分类：** HOST 桥（北桥）、PCI/LAGACY 桥（南桥）、PCI/PCI 桥。

**桥的作用：**（1）连接两条总线，使彼此相互通信；

（2）总线转换部件，可以把一条总线上的地址空间映射到另一条总线的地址空间上，从而使系统中任意一个总线上的主设备都能看到同样的一份地址表；

（3）信号缓冲、电平转换、控制逻辑转换等。

# 第八章 输入输出系统

## 1、基本概念。

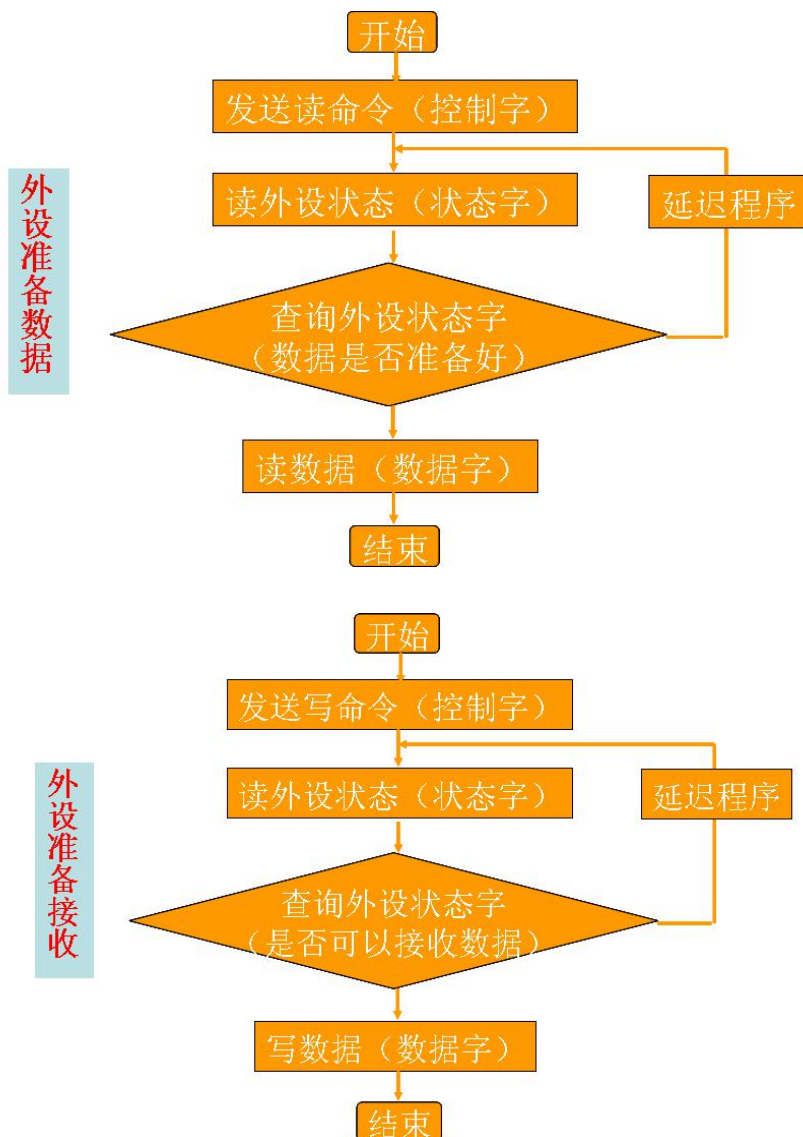
**直接内存访问 (DMA) 方式：** 是一种完全由硬件（称为 DMA 控制器）执行、在 I/O 设备和内存之间直接交换批量信息的工作方式，包括 DMA 请求、DMA 响应、DMA 传输、DMA 结束处理等步骤。在 DMA 传输过程中，DMA 控制器

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。

**外围处理机方式 PPU：**基本上是独立于主机工作，它有自己的指令系统，只是侧重于 I/O 管理。结构上接近一台计算机、或者就是一台通用计算机，也称为 I/O 管理前置机。

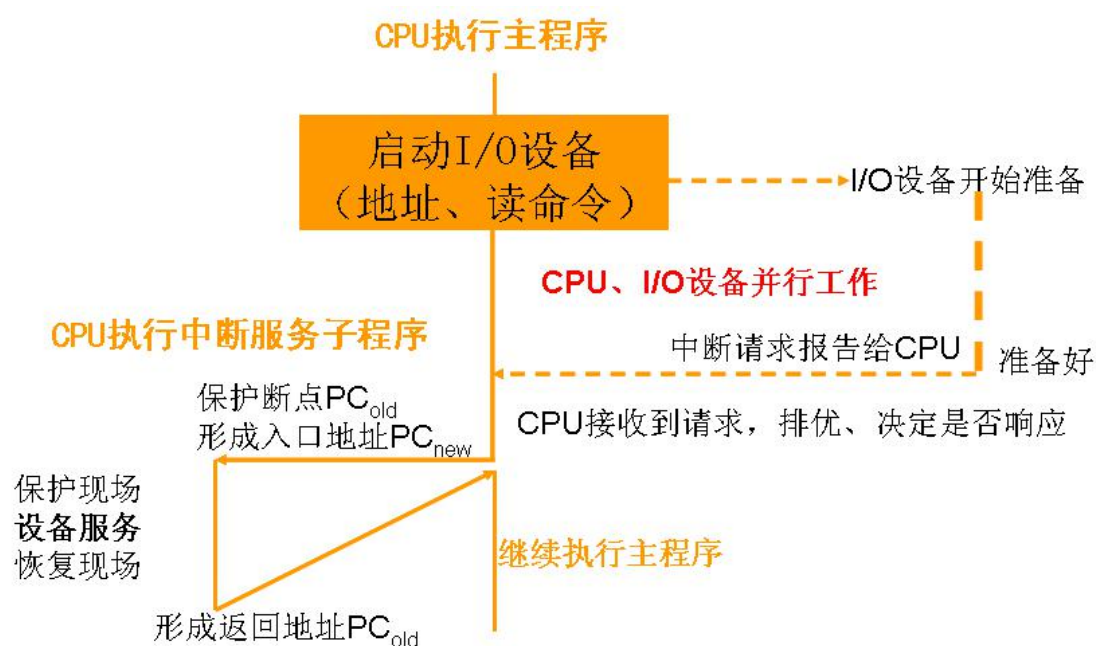
## 2、程序查询方式处理过程。





### 3、程序中断方式处理过程。

- (1) 一条指令完整执行完以后，查询是否有中断请求、是否有 DMA 请求等；
- (2) 如果有中断请求，则判断是否响应该中断（优先级等）；
- (3) 关中断（中断屏蔽置位）：单级中断系统中，一个中断被响应，必须该中断处理结束以后才能响应其它中断请求；多级中断系统中，允许中断嵌套，即允许高优先级中断低优先级的服务子程序；
- (4) 找出中断源：确定是哪个设备的中断请求；
- (5) 保存 PC：即保存返回地址，硬件实现自动推入堆栈保存；
- (6) 形成中断服务子程序的入口地址：常采用向量中断法；
- (7) 保存 CPU 现场：标志寄存器、通用寄存器内容送入堆栈保存；
- (8) 设备服务：CPU 与 I/O 设备之间交换一个机器字的数据；
- (9) 恢复 CPU 现场：从堆栈中把原来的状态寄存器、通用寄存器内容拿出来；
- (10) 开中断：本次中断请求处理完成，允许响应其它中断请求；
- (11) 中断返回：从堆栈中把返回地址拿出来，送 PC。





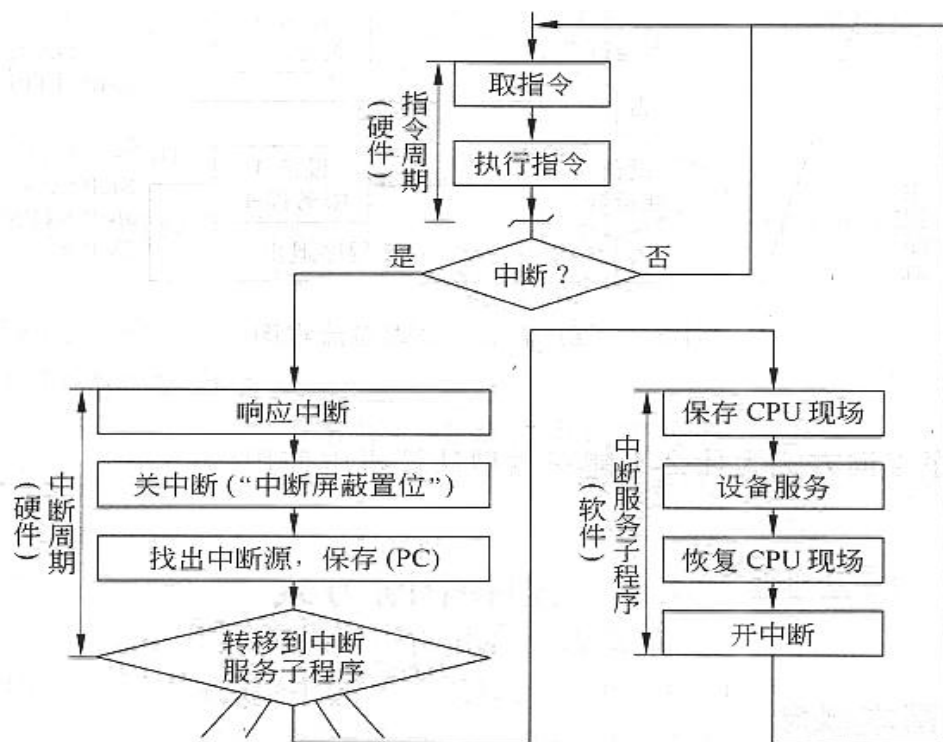
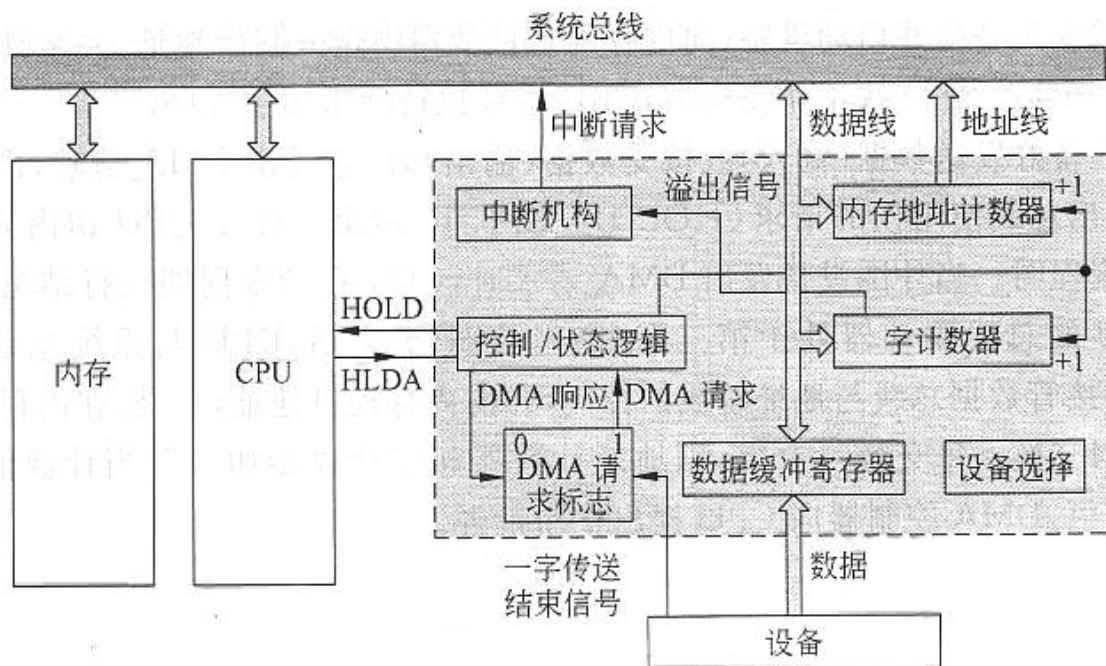


图 8.5 中断处理过程流程图

#### 4、基本 DMA 控制器的逻辑结构



##### 1、字计数器

CPU 初始化 DMA 控制器时，写入数据块长度初值。

初值通常以补码 ( $2^n - x$ ) 形式表示，如字计数器 8 位，数据块长度 125，则初值  $256 - 125 = 131$ 。

每传送一个字，字计数器加 1。当计数器溢出时，表示数据块传送结束，溢出信号通过中断机构，向 CPU 申请中断。

##### 2、内存地址计数器

CPU 初始化 DMA 控制器时，写入内存数据块起始地址。

每交换一个字数据后，地址计数器自动增量“加 1”。

# 第十一章 并行体系结构

## 1、基本概念

**并行性：**并行性是指计算机系统具有可以同时进行运算或者操作的特性，它包括同时性与并发性两种含义；同时性是指两个或两个以上的事件在同一时刻发生；并发性是指两个或两个以上的事件在同一时间间隔发生（如分时交替执行、重叠执行等）。

**VLIW 处理机：**由编译程序在编译时找出指令间潜在的并行性，进行适当调度安排，把多个能并行执行的操作组合在一起，成为一条具有多个操作段的超长指令。由这条超长指令去控制 VLIW 处理机中多个互相独立工作的功能部件，每个操作段控制一个功能部件，相当于同时执行多条指令。

**超线程技术：**同时调度多个线程执行，即多条指令流，按一定的策略往超标量流水线中交替/混合发射指令。流水线处理机可以同时处理来自不同线程的多条指令，可有效避免单指令流中的相关问题。

**向量处理机：**采用流水线技术实现向量处理，向量的分量源源不断地进入流水线，各个分量的处理时间重叠，整体上提高向量的处理速度。流水线满载以后，每个时钟周期向流水线发射一组分量，每个时钟周期输出一组分量的运算结果。

**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

## 2、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

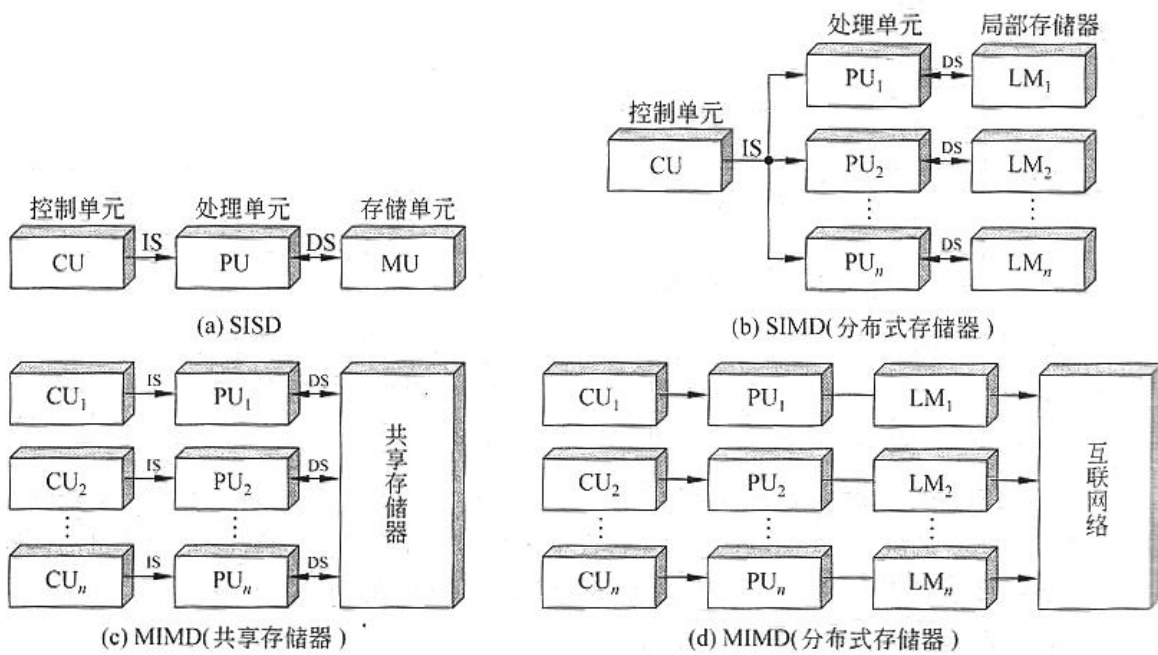


图 11.2 并行处理机的组成