# 《计算机组成与系统结构》考试大纲

## 第1章 计算机系统概论

本章的学习目的：初步了解计算机系统的组成和计算机的工作过程，掌握常用的概念、名词术语，为以后各章的学习打下基础。

本章要掌握的主要内容：

1.计算机系统是由硬件和软件两大部分组成的，硬件是物质基础，软件是解题的灵魂。弄清硬件和软件的概念。

2.计算机硬件系统所包含的主要部分，各部分的功能及其组成框图。

3.计算机的工作过程，主要是周而复始地取出指令、解释指令和执行指令的过程。而指令周期是指取出指令和执行指令所需的时间。它包括取出指令、解释指令和执行指令两个阶段。

4.冯·诺依曼计算机的设计思想是采用二进制表示各种信息以及存储程序和程序控制。存储程序的概念是将解题程序（连同必须的原始数据）预先存入存储器；程序控制是指控制器依据所存储的程序控制全机自动、协调地完成解题任务。存储程序和程序控制统称为存储程序控制。它是电子数字计算机与其他计算工具的最大区别，是电子计算机之所以能高速进行大量计算工作的基础。

5.控制器和运算器合称为中央处理器CPU，当前CPU芯片还集成有存储管理部件、Cache等；CPU和内存储器合称为计算机主机。

6.指令字和数据均以二进制代码的形式存入存储器，计算机是如何区分出指令和数据的。

7.计算机系统的主要性能指标：字长、存储容量、运算速度等。

8.计算机的运算速度是指它每秒钟执行指令的条数。单位是MIPS（百万条指令每秒）

  
式中，n—指令的种类

fi —第i种指令在程序中出现的频度（%）

ti —第i种指令的指令周期

9.计算机系统按功能划分，通常为五级的层次结构：依次是微程序设计级、一般机器级、操作系统级、汇编语言级和高级语言级，每一级都可进行程序设计。

10.软件和硬件在逻辑功能的等效性及其例子。

11.本章主要的术语及概念：

运算器，控制器，中央处理器CPU，主机，存储器，I/O接口（适配器），I/O设备，总线，存储程序，程序控制，硬件，软件，运算速度，存储容量，单元地址，存储单元，程序，指令。

## 第2章 运算方法和运算器

本章的学习目的：弄清数据与文字在计算机中的表示法，定点加、减、乘、除运算的算法，浮点数的表示法及运算方法，逻辑运算的实现，定点、浮点运算器的组成及工作原理。

本章要掌握的主要内容：

1.进位计数制及不同计数制（十、二、十六）之间数的转换方法。

进位计数制有两个要素，一是基数R，二是位权Ri。R是指计数制中所用到的数码个数，如十进制为0~9共十个数字符号；Ri是指R进制数数位的固定倍数。

2.计算机广泛使用二进制计数制。

3.计算机中表示的二进制位数B和人们习惯的十进制数D之间的位数关系：

B = 3.32 D

可见，一位十进制数要用3.32位二进制数表示，这应与二进制编码的十进制数（BCD码）区分开来。

4.数值数据在计算机中有定点表示和浮点表示两种数据格式。

5.定点表示法的表数范围、精度及其特点。

6.浮点表示这一部分的内容是一个难点，主要掌握以下内容：

(1).浮点数的构成：N=RE×M

上式R是基数，通常R=2（也有R=8或R=16），对于同一台计算机，R是固定不变的，因此，计算机表示浮点数时只需表示指数（称为阶）E和尾数M。E包括阶符（指明指数的正负）和阶码（整数），用于指明小数点的实际位置。M为尾数，包括数符和尾数，M表示了数的精度和正负。它在机器中的表示如下：

|  |  |  |  |
| --- | --- | --- | --- |
| ES | E1 E2 … Em | MS | M1 M2 … Mn |

˙

|←阶符→| 阶码 |←数符→| 尾数 |

形式小数点

所表示的浮点数，其形式小数点的位置在Ms之后。由于整个数的小数点位置还应由阶来决定，即当E为正阶时，表明实际小数点的实际位置应右移；当E为负阶时，表明实际小数点的位置应左移。由于所表示的尾数部分，其最大的绝对值约等于1，因此，所能表示的最大数是由阶码的位数来确定，而表示数的精度应由尾数的位数n决定。

(2).规格化浮点数是尾数的最高位为非零数值的浮点数。

表示为 2-1≤|M|<1 （R=2）

规格化数使一个浮点数的表示是惟一的，而且能保留最多的有效数字，避免丢失运算精度。例：某运算结果：N=20001×0.0000000110001110，限定的尾数为8位，可得

N1=20001×0.00000001

或 N2=2-0111×0.11000111，这二个数的精度不同，N2有8位数的精度，而N1 只有1位数的精度。N1 是由N舍去尾数的低8位得到的，N2 则是由N 规格化后得到的。

(3).如何实现规格化？

当|M|≥1时，将尾数右移，每右移一位，阶码加1，称为向右规格化，简称右规；

当|M|<0.5时，将尾数左移，每左移一位，阶码减1，称为向左规格化，简称左规。

可见，规格化过程，就是自动调节比例因子的过程。应注意的是，尾数为零的浮点数不能规格化。

(4).规格化浮点数的表数范围：

设阶码为m位，尾数为n位（不包括阶符和尾符），则规格化浮点数的表数范围为：

×≤ ≤

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| - | …… | -× | × | …… |  |

上式中(2m-1)和-(2m-1)是m位阶码能表示的最大和最小的阶码，而和则是规格化尾数绝对值最小和最大的值。

（5）IEEE754标准—浮点表示标准化

① IEEE754的浮点数格式

符号位 指数 尾数

|  |  |  |
| --- | --- | --- |
| S | E | M |

阶用移码，尾数用原码，规格化尾数为1.M

\*个位上的1为隐含位

② 两种常用的IEEE754 浮点数据格式。

7.计算机中表示数的大小和正负的方法称为码制。机器数的表示有原码、补码、反码和移码四种形式，重点掌握原码和补码。

8.原码、补码的性质归纳：

(1) 补码的符号位作为数值的一部分看待，参加运算，而原码则不能。

(2) 原码的表数范围相对于0来说是对称的，

整数： -（2n-1）～ 0 ～ +（2n -1）

小数：-（1-2-n）～ 0 ～ +（1-2-n）

而补码则可多表示一个最小负数：

整数：-2n ～ 0 ～ +（2n-1）

小数：-1～ 0 ～ +（1-2-n）

(3) 零的原码有二种表示形式（例如定点小数）：

[+0]原=0.00…0，[-0] 原=1.00…0

而零的补码只有一种表示形式（例如定点整数）：

[+0]补=[-0]补=0 00…0

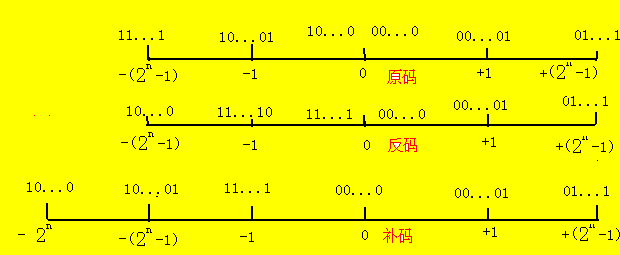
注意：(－1)补码的表示方法！

定点整数(－1)补=111…1 不是最小的一个数

定点小数(－1.0)补=100…0 是最小的一个数

以8位数为例：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10000000 | 10000001 | … | 11111111 | 00000000 | 00000001 | … | 01111110 | 01111111 |
| -128 | -127 | … | -1 | 0 | +1 | … | +126 | +127 |
| -1 | -0.1111111 | … | -0.0000001 | 0 | +0.0000001 | … | +0.1111110 | 0.1111111 |



（4）补码右移时，移空位（数的最高位）补上和符号相同的代码，补码左移时，移空位（数的最低位）补0；而原码左右移时，移空位均补上0。

（5）原码表示法便于输入输出，有利于实现乘除运算，不利于加减运算；补码表示法便于加减运算，乘除运算也有较好算法，故多被采用。

9.字符的ASCII码、字符串的表示方法。

10.汉字的表示方法包括汉字的输入编码，汉字内码和汉字字模码。

11.奇偶校验码校验位的生成，查错过程及查错功能。

12.补码加法的规则是任意两个数的补码之和等于该两数和之补码，即

[X]补 +[Y]补 = [X+Y]补  (mod 2)

对于定点小数来说，上式的先决条件是：-1≤x<1, -1≤y<1, -1≤x+y<1。

13.补码减法的运算公式：

[X-Y]补 =[X]补+[-Y]补  (mod 2)

在用补码表示的机器中，存储的是[x]补 和[y]补的机器数，而减法运算则是指令的要求，上式表明要做减法，必须从[y]补 求出[-y]补（称为对y求补），再把减法变为加法进行运算。

[-Y]补 = ¬[Y]补 +2 -n（各位变反，末位加1）

14.溢出的检测与处理

溢出是指当运算结果大于机器所能表示的最大正数（上溢）或小于机器所能表示的最小负数（下溢）。

采用双符号位补码（模4补码、变形补码）运算，便于判溢出。

15.由逻辑门电路组成的全加器的逻辑方程式：





16.计算机实现乘除运算的方法：

（1）用乘除运算子程序实现；

（2）在加法器和寄存器中增添控制线路实现；

（3）用阵列乘除法器实现。

17.原码一位乘法的算法（一般了解）：

（1）符号位单独处理，ZS＝XSYS

（2）从乘数的最低位开始，逐位与被乘数相乘，若该乘数位Yn-i+1 = 1，则部分积Pi-1 +|x|，若Yn-i+1 = 0，则Pi-1+0，相加后右移一位，得新的部分积Pi ，重复n次可得乘积的绝对值|P|。（可见，乘法过程变为+|x|或+0以及右移操作）

（3）给|P|置乘积的符号位Zs，可得[x×y]原

18.原码除法的运算规则（一般了解）

（1）商的符号位单独处理，即 qS＝XSYS

（2）商的尾数

商的原码[q]原=qs.q1q2…qn

（3）被除数X、除数Y、商q和余数rn之间的关系应满足：

X=q×Y+rn 0≤|rn|≤2 – n×Y

19. 计算机中的基本逻辑运算、逻辑运算的特点及其应用。

20.运算器的三种基本结构及其特点，运算器的实例。

21.浮点运算的算法：

浮点算术运算由阶和尾数两部分的运算组成，它们的运算可采用任何一种相应的定点运算的方法进行。

设两浮点数：，,则

（1）浮点加减法运算：

 Ex < Ey

或 = (Mx±My×)× Ex ≥ Ey

（2）浮点乘法运算：



（3）浮点除法运算：



22.浮点加减法运算的步骤：

（1）首先是对阶，就是使两个浮点数的阶码取得一致的过程。

通常用加法线路求阶差：



若 >0，即Ex >Ey，应将My 右移，每右移一位， -1，直至=0为止；

若<0，即Ex <Ey ，应将Mx 右移，每右移一位，+1，直至=0为止。

（2）取大阶Max (Ex ,Ey) 暂作结果的阶，将对阶后的尾数按指令要求相加或相减。

（3）将运算结果规格化（以双符号补码为例）

A.右规条件：运算结果两个尾符S0'S0状态不同，即:



右规的操作是尾数右移，阶码加1；

B.左规条件：结果非零且为正，尾数最高位M1 =0；或结果为负，尾数最高位M1=1，即：



左规的操作是尾数每左移一位，阶码减1。

（4）舍入处理

当尾数右移时，为减少误差，需进行舍入处理。常用的舍入法有“0舍1入法”和“恒置1法”。

（5）最后检测结果是否溢出。

浮点数的溢出是指运算结果的阶大于机器所能表示的最大正阶。若溢出，转中断处理或停机。

23.浮点运算流水线：

（1）线性流水线时钟周期的确定: τ= Max (τi) +τl =τm +τl

（2）K级线性流水线的加速比： Ck = =

28.本章主要的术语、概念。

进位计数制，码制，规格化浮点数，左规，右规，舍入，溢出，机器数，真值，原码，补码，求补，ASCII码，汉字内码，数据校验码，变形补码，浮点运算流水线，加速比。

## 第3章 存储系统

本章的学习目的：了解半导体存储元件的存储机理，由半导体存储器芯片组成主存的工作原理，高速缓冲存储器、多模块交叉并行存储系统和虚拟存储器的基本概念及工作原理，存储系统的层次结构，外存储器的工作原理及硬盘的主要技术指标。

本章要掌握的基本内容：

1.存储器的分类，主要掌握按存取方式分类和按在计算机系统中的作用分类。

2.存储系统的设计目标：在一定的成本下，获得尽可能大的存储容量，尽可能高的存取速度以及可靠性等。

3．存储系统的分级结构

（1）高速缓冲存储器

在计算机系统中用于存放最活跃的程序和数据的高速小容量存储器。

（2）主存储器

用于存放计算机运行期间的大量程序和数据的半导体存储器。

内存储器（简称内存）包括主存储器和高速缓冲存储器，是CPU 能直接访问的存储器。

（3）辅助存储器（外存储器）

存放当前暂不参与运行的程序和数据，需要时再与主存成批交换信息的存储器。例如磁表面存储器（磁盘、磁带）、光盘存储器。（在第7章介绍）

4.主存储器的技术指标

（1）存储容量

主存存储单元的总数，通常用字数或字节数表示。按字节编址的主存，存储容量的单位可用KB、MB、GB、TB等单位表示：

1KB=210 B， 1MB=220 B，1GB=230B， 1TB=240B

熟练掌握容量与地址码位数的关系

（2）存储周期Tmc

两次读/写操作之间所需的最短间隔时间。

Tmc 的单位是ns(纳秒)，1ns=10-9 s。

当前半导体存储器的Tmc 已小于10ns。

5.MOS静态存储元的组成及其存储二进制数的机理—用双稳态触发器的两个稳定状态表示1和0。

6.冯•诺依曼计算机的工作方式基本特点之一是按给定的地址访问存储器。地址译码通常用双译码的结构。（矩阵X×Y）

7.主存储器与CPU的连接，包括地址线、数据线和控制线的连接。

根据存储器容量的要求，可将若干存储器芯片按位、字、或字位进行扩展，如课本P136图3.25和图3.26所示。

所需某种规格存储器芯片数N的计算如下：



存储器扩展常用的译码器电路74LS138,74LS139芯片工作原理

8.四管、单管动态存储元的存储机理—用电容存储电荷的多少表示1和0。

9.动态存储器的刷新，三种主要的刷新方式。

10.半导体只读存储器的分类：

（1）掩膜式只读存储器（MROM）

是由制造厂家把信息“写入”，用户不能修改的存储器片。

（2）（一次性）可编程的只读存储器（PROM）

信息由用户编程写入，但不能“擦除”再写的存储器片。

（3）光可擦可编程的只读存储器（EPROM）

写入信息后可用紫外光擦除，再编程写入的只读存储器。

（4）闪速存储器(Flash Memory)的工作原理及其工作模式

闪速存储器是一种快速电擦除、可改写型的存储器。

11.解决主存与CPU速度不匹配的主要途径：

（1）在CPU内部设置多个通用寄存器或加长存储器的字长；

（2）采用并行操作的存储器；例如双端口、相联存储器和多模块交叉存储器；

（3）在CPU和主存之间插入高速缓冲存储器（Cache）。

12.双端口存储器是指同一个存储器具有两组相互独立的读写控制电路。双端口存储器是用硬件的冗余取得高带宽。在奔腾机中用作数据Cache。

13.多模块交叉存储器

多模块交叉存储器的基本原理是：把M=2n 个容量为L个存储单元的存储器模块进行交叉编址，使通常按地址自然递增访问存储器的操作依次发生在不同的存储模块中，由于每个存储模块都有自己的读/写电路和地址寄存器、数据缓冲寄存器，就能对不同存储模块同时访问，达到提高存储器工作速度的目的。

多模块交叉存储器的并行操作关键在于各存储模块的交叉编址。设有M个存储器模块，存储模块编号为J（J = 0，1，2，…，(M-1)），每个存储模块容量为L个存储单元，单个模块的单元顺 序 号 为i（i = 0，1，2，…，(L-1)）。则Mj 模块的编址模式为：



例如M=4，则用模4交叉编址

模块号 地址编址序列 最末二位地址状态

M0 0，4，8，…（4i+0）, …4(L-1)+0 00

M1 1，5，9，…（4i+1）, …4(L-1)+1 01

M2 2，6，10，…（4i+2）, …4(L-1)+2 10

M3 3，7，11，…（4i+3）, …4(L-1)+3 11

在理想的情况下，每 （Tmc —存储周期）可读/写一次。

多模块交叉主存系统是以硬件的冗余和交叉编址技术换取高带宽。

【主存带宽计算举例】设存储器容量为128K字，字长32位，模块数m＝4，分别用顺序方式和交叉方式进行组织。存储周期TMC＝80ns，数据总线宽度为32位，假定总线传送周期T＝15ns。问顺序存储器和交叉存储器的带宽各是多少?

【解】顺序方式和交叉方式读取m＝4个字的信息总量都为

q=32b×4＝128b

顺序存储器和交叉存储器连续读取4个字所需的时间分别为

T1=mTMC=4×80＝320ns=32×10-8(s)

T2=TMC+(m-1)τ=80+(4-1)×20＝140ns=14×10-8(s)

顺序存储器和交叉存储器的带宽各是

W1=128/(32×10-8)=4×108(b/s)

W2=128/(14×10-8)=9.14×108(b/s)

14.相联存储器

是按内容寻址的存储器，即用某项内容（关键字）作为地址来存取的存储器。

相联存储器主要用于存放Cache 的行标记，虚拟存储器的分段表、页表和快表。

15.高速缓冲存储器(Cache)

Cache是介于CPU与主存之间，用于存放当前最活跃的程序块和数据的高速小容量存储器。

Cache实现的工作原理是CPU运行程序的局部性原理，即指CPU执行的程序所使用的存储单元是相对集中或小批簇聚于相邻单元中。

Cache的命中率H是指CPU在Cache中访问到的次数n1与总的访问次数n之比。



不命中率（脱耙率）：（1- H）

在有Cache 的主存系统中，CPU访问存储器的平均周期：

T A =H×Tcc +(1-H)×T mc

上式中：Tcc —— Cache 的存储周期

Tmc ——主存的存储周期

Cache的访问效率：e=Tcc/TA

CPU与Cache、主存的存储层次。

**16.虚拟存储器**

**虚拟存储器的功能与特点**

**页式虚拟存储器**

**（1）页式虚存地址映射**

**（2）转换后援缓冲器（TLB ），快表**

**由于页表通常在主存中，因而即使逻辑页已经在主存中，也至少要访问两次物理存储器才能实现一次访存，这将使虚拟存储器的存取时间加倍。为了避免对主存访问次数的增多，可以对页表本身实行二级缓存，把页表中的最活跃的部分存放在高速存储器中，组成快表。这个专用于页表缓存的高速存储部件通常称为转换后援缓冲器(TLB)。保存在主存中的完整页表则称为慢表。**

**段式虚拟存储器**

**段页式虚拟存储器**

21.本章主要的术语、概念

存储元， ROM，RAM，Cache，主存，内存，外存（辅存），存储周期，静态存储器，动态存储器，刷新，写操作，读操作，多模块交叉存储器，双端口存储器，Cache 的命中率，相联存储器，快表**TLB**，慢表

## 第4章 指令系统

本章的学习目的：弄清计算机指令系统按功能划分的指令种类；两种指令系统计算机： CISC（复杂指令系统计算机）和RISC（精简指令系统计算机）指令的特点；指令和数据的寻址方式；堆栈及其应用。

本章要掌握的基本内容：

1.指令的基本格式及指令系统与硬件、软件之间的关系

|  |  |
| --- | --- |
| 操作码字段 OP | 地址码字段 A |

OP——指示指令的操作性质，用二进制代码表示，OP通过指令译码器进行解释。

A——通常用于指示操作数的地址或指令地址。

决定指令格式的主要因素有三个：一是操作的种类，二是地址的数目，三是寻址方式。

2. 操作码OP的结构

（1）操作码的位数n取决于操作的种类N

2n ≥N , 即n≥log2 N

3. 地址码结构

根据一条指令中所包含的地址个数，分为三地址、二地址、一地址和零地址指令。

4.指令的寻址方式有二种：

（1）顺序寻址方式，即指令在内存按序安排，指令地址由程序计数器PC提供。

（2）跳跃寻址方式，由程序控制类指令的执行形成下一条指令的地址。

5. 操作数寻址方式——形成操作数有效地址的方法。主要的寻址方式有：

（1）立即寻址方式（立时地址）

指令中的地址字段直接给出操作数本身。适用于指定固定的常数。

（2）直接寻址方式

地址字段直接给出操作数在内存的地址A，即有效地址E=A。

（3）间接寻址方式

指令中的地址字段指出操作数地址的地址。

间接寻址可根据间址的次数分为一次间址和多次间址，如E=(A)为一次间址；E=((A))为二次间址。

（4）寄存器寻址方式

A.寄存器直接寻址

地址字段给出寄存器的编号，该寄存器的内容就是操作数。

B.寄存器间接寻址

地址字段指定的寄存器，其内容是操作数的地址，有效地址E = (Rn)。

C.变址寄存器寻址

将变址寄存器的内容（变址值）与形式地址相加而得到有效地址。

E = (Rx) +D

D通常用补码表示，可以是正整数或负整数，变址范围：－2n-1 ~ (2n-1-1)，n为D的位数。

（5）段寻址方式

将段寄存器的基地址（左移4位）与偏移量相加形成内存地址的寻址方式（PC采用）。

6. 堆栈

按后进先出（LIFO）方式存取的存储单元的有序集合。计算机中堆栈的实现有二种结构，一种是寄存器堆栈（串联堆栈、下压堆栈），另一种是存储器堆栈。前者是在CPU中设置一组专门的具有对位串联的若干个寄存器组成，配合堆栈指令实现堆栈操作；后者则是在内存开辟专门用于堆栈的存储区，另加堆栈指针SP组成，配合堆栈指令实现其操作。由于存储器堆栈是使用容量较大的内存部分存储区，因此具有堆栈区的位置灵活和容量可变等特点，是常用的一种。

堆栈在计算机中的应用主要有：

a.为零地址指令提供操作数，例如堆栈处理器；

b.存放返回主程序得地址，实现子程序的嵌套；

c.存放多级中断的有关信息，实现多级中断的嵌套。

7. 精简指令系统的特点：

a.选用的是使用频率最高的一些简单指令；

b.指令长度固定，指令格式及寻址方式种类少；

c.只有取数和存数指令访问存储器，其余指令的操作均在寄存器之间进行。

8. 指令系统按功能划分的种类：传送类指令、算逻运算指令、程序控制类指令、I/O指令、其它指令等

【指令格式及寻址方式举例】

3）指令格式结构如下所示，试分析指令格式及寻址方式特点。

15 10 7 4 3 0

|  |  |  |  |
| --- | --- | --- | --- |
| OP |  | 目标寄存器 | 源寄存器 |

【解】指令格式及寻址方式特点如下：

① 单字长二地址指令。

② 操作码字段OP 6位可以指定26＝64条指令。

③ 源和目标都是通用寄存器(可分别指定32个寄存器)，所以是RR型指令，两个操作数均在寄存器中。

④ 这种指令结构常用于算术逻辑运算类指令。

9. 指令系统的发展演变

CISC：复杂指令集（Complex Instruction Set Computer）

具有大量的指令和寻址方式（更接近高级程序语言）

80/20原则：80%的程序只使用20%的指令

大多数程序只使用少量的指令就能够运行

RISC：精简指令集（Reduced Instruction Set Computer)

在数据通道中只包含最有用的指令

确保数据通道快速执行每一条指令

使CPU硬件结构设计变得更为简单

大部分RISC机具有以下特点：

（1） 指令系统设计时选择一些使用频率较高的简单指令，且选择一些很有用但不复杂的指令。

（2） 指令长度固定，指令格式种类少，寻址方式种类少。

（3） 只有取数/存数指令访问存储器，其余指令的操作都在寄存器之间进行。

（4） 采用流水线技术。超级标量及超级流水线技术，增加了指令执行的并行度，使得一条指令的平均指令执行时间小于一个机器周期。

（5） CPU中通用寄存器数量相当多，可以减少访存次数。

（6） 以硬布线控制逻辑为主，不用或少用微码控制。

（7） 采用优化的编译程序，力求有效地支持高级语言程序。

RISC三要素：

(1)一个有限的简单的指令集；指令长度固定，指令格式和寻址方式种类少，

(2)CPU配备大量的通用寄存器；只有取数/存数指令访问存储器，其余指令的操作均在寄存器之间进行。

(3)强调对指令流水线的优化。

10. 本章主要的术语、概念

指令，指令系统，操作码，地址码，形式地址，有效地址，寻址方式，顺序寻址，跳跃寻址，立即寻址，隐含寻址，直接寻址，间接寻址，寄存器寻址，寄存器间接寻址，段寻址，CISC， RISC，堆栈，压栈，出栈。

## 第5章 中央处理器

本章的学习目的：了解中央处理器的基本功能和组成，基本了解计算机内部的运行机制，组合逻辑控制器，微程序控制器及其设计技术，并行处理技术等。

本章要掌握的基本内容：

中央处理器CPU――计算机中用于解释和执行指令的部件。

1. CPU的功能：

（1）指令控制，即程序的顺序控制。

主要是由程序计数器PC（顺序寻址）和控制类指令的执行（跳跃寻址）实现的。

（2）操作控制

由执行指令的一系列微操作信号进行控制。

（3）时间控制

对各种操作实施时间上的控制。主要是由时序信号发生器等实现。

（4）数据加工

对数据实现算逻运算等的处理。

CPU的前三个功能主要是由控制器实现的，最后一个功能则是在运算器实现的。

2. CPU的组成

传统的CPU是由控制器和运算器两部分组成的，在巨大规模集成电路的CPU芯片中还包括存储管理部件及CPU内部的Cache。

控制器是指挥计算机各部件按指令要求进行操作的部件，是计算机的控制中心，其主要功能是：

(1) 控制取出指令，解释指令和执行指令；

(2) 中断的控制；

(3) 信息传送线的控制。

3. 控制器的基本组成

(1) 指令部件

即与指令有关的部件，它包括程序（指令）计数器PC、指令寄存器IR和指令译码器ID。

(2) 时序部件

时序部件主要包括时钟脉冲源及启停控制电路；时序信号的产生及控制部件。

(3) 操作控制器

用以产生微操作控制信号，控制各部件的操作。

(4) 地址寄存器AR及数据寄存器DR，状态条件寄存器PSR等。

(5) 中断机构及控制台。

4. 操作控制器的类型主要有二种：

(1) 组合逻辑控制器（硬布线逻辑）

(2) 微程序控制器（存储逻辑型）

5. 指令周期，CPU周期和时钟周期

(1) 指令周期

指取出并执行一条指令所需的时间。通常，其长短与指令的复杂程度有关。

(2) CPU周期（机器周期）

将指令周期划分为若干个相对独立的操作阶段，称为CPU周期。

(3) 时钟周期（节拍脉冲或T周期）

CPU周期包括若干个完成微操作的节拍脉冲。

6. 时序信号的作用和体制

时序信号的作用：为计算机各部分的协调工作提供时序标志。

时序信号的基本体制：电位-脉冲制。这是由于器件的特性所决定。

7. 时序信号发生器的组成

(1) 时钟脉冲源

由石英晶体振荡器和与非门组成的振荡电路组成，是主机信号的发源地。

(2) 环型脉冲发生器

循环移位寄存器和译码电路配合产生所需的节拍脉冲。

(3) 启停控制电路

启停控制电路是控制机器正常启动运行和停机操作的控制电路。对启停电路设计的要求主要是：启动时，一定要从第一个节拍电位的前沿开始工作，以保证机器运行时的第一个节拍脉冲有足够的宽度，使机器工作可靠；停机时，一定要在指令最末一个节拍脉冲结束后才关闭时序发生器。

【微处理器的数据传输速率】一个32位的微处理器，它有16位外部数据总线，总线的时钟频率是800MHz，假定一个总线事务的最短周期是2个时钟周期，问：（1）这个处理器的最大数据传输速率是多少MB/s？（2）如果将外部数据总线的宽度扩展为32位，那么处理器的最大数据传输速率提高到多少MB/s？

【解】（1）处理器的最大数据传输速率是

Dtr=(16/8)÷2T=2×(800÷2)=800 (MB/s) (T=1/F)

（2）如果将外部数据总线的宽度扩展为32位，那么处理器的最大数据传输速率提高到

Dtr=(32/8)÷T=4×(800÷2)=1600 (MB/s)

8. 组合逻辑控制器（硬布线控制器）

对不同指令中相同微操作的产生条件用逻辑代数的方法进行综合、化简，最后得到每个微操作产生的逻辑式并用组合逻辑电路实现之。组合逻辑控制器经常采用三级的时序信号：主状态周期（CPU周期）、节拍电位、节拍脉冲。

9. 微命令、微指令、微程序和微周期

微命令是控制部件向执行部件发出的各种最简单的控制命令，例如ALU BUS。

微指令是在一个CPU周期中，一组实现一定操作功能的微命令的组合。微指令通常包括操作控制字段和顺序控制字段。

微程序是由微指令组成的，用以实现指令功能的程序。

概括为：

微命令 组合 微指令 组合 微程序 组合 实现指令系统功能

微周期是执行一条微指令和取出下一条微指令时间之和。

10 微程序控制器组成原理及微程序控制的计算机的工作过程。

微程序控制的计算机的工作过程：

（1）执行取指微指令，依（PC）从内存读出指令，由指令操作码经地址转移逻辑形成执行该指令的微程序入口地址送到微地址寄存器μAR。

（2）根据（μAR）从控制存储器CM读出微指令送微指令寄存器MIR。

（3）操作控制字段（经译码）产生微命令，送入各功能部件执行。

（4）由地址转移逻辑形成下一微地址送μAR，再重复步骤（2）、（3）、（4），这就是微程序的执行过程。

11. 微命令编码的方法通常有三种

（1）直接表示法（直接控制法、不译法）

操作控制字段中的每一位二进制代码表示一个微命令，该位为1，表示有该微命令。

（2）编码表示法（字段编码法）

将微命令分段编码，经简单译码产生微命令。

分段编码的原则：

① 相互有一定关系又不能在同一微周期出现（相斥性）的微命令可编在同一字段内；可能在同一微周期出现（相容性）的微命令应编在不同的字段内。

② 分段应与数据通路的结构相适应，以便于微命令的设计、修改和检查。

③ 每个字段的位数不能太多（2～4位）。

（3）混合表示法

把直接表示法和编码表示法相混合使用。

12.微地址的形成

如何正确形成下一条微指令的地址，是关系到微程序能否正确执行的问题。取指令后，通常由指令的操作码经地址转移逻辑形成执行该指令的微程序入口；无转移时，意味着下一条微指令的地址在设计时已可定（已知）；判断转移时，由判别字段和微指令执行结果经地址转移逻辑获得下一微地址。

13. 组合逻辑控制器和微程序控制器的根本区别：

微操作信号的产生部件不同，产生方式不同。

14. 在并行处理技术中，并行性的两种含义：并发性和同时性。并行技术的三种形式：时间并行，空间并行和时空并行。

15. 本章主要的术语、概念

控制器　　　　　程序计数器PC　　　指令寄存器IR　　 　 程序状态字PSW

　 时序部件　　　　指令译码 微操作形成部件 　 微操作

指令周期　　　　 CPU周期　　　　　 节拍电位　　　　　 节拍脉冲

指令微流程　　　 组合逻辑控制　　　 微程序控制 　　　　 控制存储器CM

　　微命令　　　　　　微指令　　　　　微程序　　　　　　 微周期 　　　　　　 微地址 　　同时性　 　　　　并发性　　　　　　 并行性　　　　　　 时间重叠 　　资源重复　　　　顺序方式　　　　　 流水方式　　　　　 吞吐率

## 第6章 总线系统

本章的学习目的：掌握总线的工作原理，熟悉计算机各个模块之间的互联方法，掌握总线的类型、仲裁机制、定时方式、数据传送方式等。重点在总线结构、总线控制两个方面，要多记，在理解的基础上记忆。

本章要掌握的基本内容：

1. 总线的基本概念

总线是构成计算机系统的互连机构，是多个系统功能部件之间进行数据传送的公共通路。一个单处理器系统中的总线，大致分为三类：

(1)内部总线：CPU内部连接各寄存器及运算部件之间的总线。

(2)系统总线：CPU同计算机系统的其他高速功能部件，如存储器、通道等互相连接的总线。

(3)I/O总线：中、低速I/O设备之间互相连接的总线。

2.总线的特性

物理特性：指总线的物理连接方式，包括总线的根数，总线的插头、插座的形状，引脚线的排列方式等。

功能特性：描述总线中每一根线的功能。

电气特性：定义每一根线上信号的传递方向及有效电平范围。送入CPU的信号叫输入信号(IN)，从CPU发出的信号叫输出信号(OUT)。

时间特性:定义了每根线在什么时间有效。规定了总线上各信号有效的时序关系，CPU才能正确无误地使用。

1. 总线标准

ISA总线 (16位，带宽8MB/S)

EISA总线 (32位，带宽33.3MB/S)

VESA总线 (32位，带宽132MB/S)

PCI总线 (32位，带宽132MB/S，允许64位)

4. 总线带宽

指总线本身所能达到的最高传输速率。

单位：MB/s 兆字节每秒

1. 总线接口
2. 信息的传送方式

串行，并行，分时

7.接口的基本功能

(1)交换主机与外设的状态信息，如控制外设的启停，传送外设的忙、闲信息等；

(2) 匹配主机与外设的速度差异；

(3) 实现数据格式转换；

(4) 实现主机与外设之间的数据交换。

8.接口的分类

9.集中式仲裁

## 第7章 外存与I/O设备

本章的学习目的：

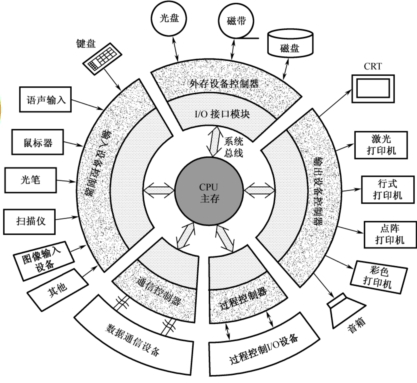
本章要掌握的基本内容：

1.外围设备的一般功能

在计算机和其他机器之间，以及计算机与用户之间提供联系。

2.外围设备的基本组成

3.外围设备的分类

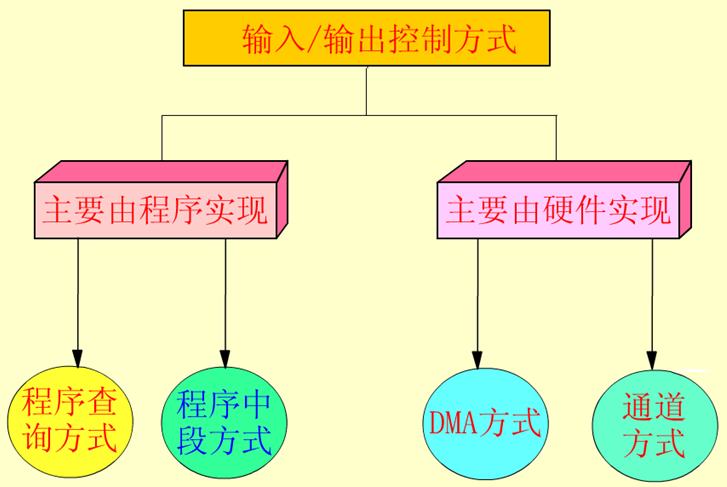


4.硬磁盘存储器

## 第8章 输入输出系统

本章要掌握的基本内容：

1. 外围设备的信息交换方式



1程序查询方式

　　程序查询方式是早期计算机中使用的一种方式。数据在CPU和外围设备之间的传送完全靠计算机程序控制，查询方式的优点是CPU的操作和外围设备的操作能够同步，而且硬件结构比较简单。但问题是，外围设备动作很慢，程序进入查询循环时将白白浪费掉CPU很多时间。这种情况同上述例子中第一种方法相仿，CPU此时只能等待，不能处理其他业务。即使CPU采用定期地由主程序转向查询设备状态的子程序进行扫描轮询的办法，CPU宝贵资源的浪费也是可观的。因此当前除单片机外，很少使用程序查询方式。

2程序中断方式

中断是外围设备用来“主动”通知CPU，准备送出输入数据或接收输出数据的一种方法。通常，当一个中断发生时，CPU暂停它的现行程序，而转向中断处理程序，从而可以输入或输出一个数据。当中断处理完毕后，CPU又返回到它原来的任务，并从它停止的地方开始执行程序。这种方式和我们前述例子的第二种方法相类似。可以看出，它节省了CPU宝贵的时间，是管理I/O操作的一个比较有效的方法。中断方式一般适用于随机出现的服务，并且一旦提出要求，应立即进行。同程序查询方式相比，硬件结构相对复杂一些，服务开销时间较大。

3直接内存访问(DMA)方式

用中断方式交换数据时，每处理一次I/O交换，约需几十微秒到几百微秒。对于一些高速的外围设备，以及成组交换数据的情况，仍然显得速度太慢。直接内存访问(DMA)方式是一种完全由硬件执行I/O交换的工作方式。这种方式既考虑到中断响应，同时又要节约中断开销。此时，DMA控制器从CPU完全接管对总线的控制，数据交换不经过CPU，而直接在内存和外围设备之间进行，以高速传送数据。这种方式和前述例子的第三种方法相仿，主要优点是数据传送速度很高，传送速率仅受到内存访问时间的限制。与中断方式相比，需要更多的硬件。DMA方式适用于内存和高速外围设备之间大批数据交换的场合。

4通道方式

DMA方式的出现已经减轻了CPU对I/O操作的控制，使得CPU的效率有显著的提高，而通道的出现则进一步提高了CPU的效率。这是因为，CPU将部分权力下放给通道。通道是一个具有特殊功能的处理器，某些应用中称为输入输出处理器(IOP)，它可以实现对外围设备的统一管理和外围设备与内存之间的数据传送。这种方式与前述例子的第四种方法相仿，大大提高了CPU的工作效率。然而这种提高CPU效率的办法是以花费更多硬件为代价的。



1. 程序中断方式的基本接口

程序中断由外设接口的状态和CPU两方面来控制.

在接口方面,有决定是否向CPU发出中断请求的机构,主要是接口中的“准备就绪”标志(RD)和“允许中断”标志(EI)两个触发器.

在CPU方面,有决定是否受理中断请求的机构,主要是“中断请求”标志(IRQ)和“中断屏蔽”标志(IM)两个触发器.

另外还有工作标志(BS)触发器.