

# 《计算机组成原理与结构》重要知识点

## 第一章 概论

### 一、基本概念

#### 1. 冯·诺依曼体制？存储程序方式？

冯·诺依曼体制包含三个要点：（1）采用二进制代码表示信息，以便计算机识别；（2）采用存储程序工作方式，才能使计算机自动地对信息进行处理；（3）由存储器、运算器、控制器、输入/输出设备等功能部件组成计算机硬件系统。

存储程序工作方式：事先编制程序，事先存储程序，自动、连续地执行程序。

#### 2. 控制流？数据流？

控制流：控制计算机工作的信息，即指令或命令。

数据流：计算机加工处理的对象，即数值和非数值数据。

传统的诺依曼机采用控制流（指令流）驱动方式：按指令序列依次读取指令，根据指令所包含的控制信息对数据进行处理，在程序执行过程中，始终由指令流驱动计算机工作。

数据流驱动方式是对传统诺依曼机工作方式的根本改变：只要数据准备好，有关指令就可并行执行，如数据流计算机。

#### 3. 模拟信号？数字信号？数字信号有哪两种？

模拟信号：在时间上连续变化的电信号，用信号的某些参数模拟信息。

数字信号：在时间上或空间上断续变化的电信号，依靠彼此离散的多位信号的组合表示信息。

数字信号有两种：脉冲信号和电平信号。

脉冲信号：在时间上离散的电信号，利用脉冲的有无表示不同的状态。电平信号：在空间上离散的电信号，利用信号电平的高低表示不同的状态。

#### 4. 总线及其组成？

**总线**是一组能为多个部件分时共享的公共的信息传送线路。总线由一组传送线与相应控制逻辑构成（包括 CPU 内设置控制逻辑、总线控制器）。按照传输的信息类型可分为地址总线、数据总线、控制总线。

#### 5. 接口的概念？

主机与外设之间的连接逻辑，控制外设的 I/O 操作。

#### 6. 基本字长？

一般指参加一次定点运算的操作数的位数。

#### 7. CPU 主频？时钟频率？

外部频率或基频，也叫系统时钟频率。CPU 主频=外频×倍频系数；CPU 主频是指 CPU 内核的工作频率，即 CPU 的时钟频率，计算机的操作在时钟信号的控制下分步执行，每个时钟信号周期完成一步操作，时钟频率的高低在很大程度上反映了 CPU 速度的快慢。

#### 8. 数据通路宽度？数据传输率？

数据通路宽度：是指数据总线一次能并行传送的数据位数。

数据传输率：是指数据总线每秒钟传送的数据量。

## 第二章 计算机中的信息表示

### 一、基本概念

#### 1. 一个数值型数据的完整表示需三个方面？

(1) 数的组合规则 - 进位计数制；(2) 小数点的位置确定 - 数的定点表示和浮点表示；(3) 符号的选择 - 带符号数的代码表示。

#### 2. 权、基数？

位权：在  $r$  进位制的数中，每个数位的数码所表示的数值等于该数码乘以一个与它所在数位相关的常数，这个常数称为该位的位权，简称权。

基数：在进位制中，各数位允许选用的数码个数，称为该进位制的基数，它等于该进位制各数位所允许的最大数码值加 1。

#### 3. 真值？机器数的定义？分类？

(1) 真值：在数的绝对值之前配上正 (+；通常可省略)、负 (-) 符号表示的数称为该数的真值。例如用十进制数表示的真值：159，-132。用二进制数表示的真值：1011、-1011 等。

(2) 机器数：在计算机内部使用的，连同数的符号一起数码化的数称为机器数。

(3) 原码、补码、反码、无符号数

#### 4. 数的定点表示与浮点表示？

定点数：在计算机中，小数点位置固定不变的数叫做定点数。

浮点数：小数点位置不固定，可随需要浮动的数称为浮点数。

规格化浮点数：是指浮点数的尾数部分用带符号定点小数表示，当  $R=2$  时，尾数用原码表示时其绝对值满足  $1/2 \leq |M| < 1$  (补码时是  $-1 \leq M < -1/2$  或  $1/2 \leq M$ )。

#### 5. 十进制数用原码、补码表示，以及无符号数、原码、补码的表示范围

注意+0，-0，-1 等特殊形式

#### 6. 补码加减运算

$$(X + Y)_{\text{补}} = X_{\text{补}} + Y_{\text{补}}$$

$$(X - Y)_{\text{补}} = X_{\text{补}} + (-Y)_{\text{补}}$$

#### 7. 双符号位运算，溢出判断

$$\text{溢出} = \overline{SA} \overline{SB} Sf + SA SB \overline{Sf}, \text{溢出} = Cf \oplus C, \text{溢出} = Sf1 \oplus Sf2$$

#### 8. 原码补码移位运算

(1) 正数补码/原码移位规则：

数符不变 (单：符号位不变；双：第1符号位不变)

空位补0 (右移时第2符号位移至尾数最高位)

(2) 负数补码移位

数符不变 (单：符号位不变；双：第1符号位不变)

左移空位补0

右移空位补1 (第二符号位移至尾数最高位)。

### 9. 1 位原码乘法运算规则（会运算）

- (a) 操作数、运算结果用原码表示；
- (b) 绝对值参与运算，符号单独处理；
- (c) 被乘数(B)、累加和(A)取双符号位；
- (d) 乘数末位( $C_n$ )为判断位 ( $C_n=0$ : 右移;  $C_n=1$ : +B, 右移)，其状态决定下一步操作；
- (e) 作  $n$  次循环（累加、右移）。

### 10. 原码不恢复余数除法（会运算）

## 二、计算题

1. IEEE754 短浮点数格式表示？
2. 原码一位乘法（要求运算过程）；
3. 原码不恢复余数除法（要求运算过程）

## 第三章 中央处理器

### 一、基本概念

#### 1. 指令？指令系统？

指令是指示计算机执行某类操作的命令，CPU 工作的主要依据。

指令集：计算机能执行的全部指令的集合，也称为指令系统

#### 2. 地址码？寻址方式？地址结构？

**地址码：**指令中用于指示操作数或者操作数地址的二进制代码。包括指令中给出操作数或操作数的地址、存放运算结果的地址、下一条指令的地址。

**寻址方式：**指令中以什么方式提供操作数或操作数地址，称为寻址方式。

(1) 显地址：在指令代码中明确给出了主存储器单元地址码或者寄存器号。

(2) 隐地址：如果地址是以隐含的方式约定，则在指令中并不给出地址码。

**地址结构：**指在指令中明确给出的地址。

(1) 简化地址结构的基本途径是什么？

【答】 在指令中减少显地址的数量，即使用隐地址方式给出地址，指令中的地址（段）个数就可减少。

(2) 减少指令中一个地址信息的位数的方法是什么？

【答】 采用寄存器寻址、寄存器间址等以寄存器为基础的寻址方式可以大大减少指令中一个地址的信息位数。

#### 3. 寻址方式

大致可将众多的寻址方式归纳为以下四大类，其它的寻址方式则是它们的变型或组合。

- ① 立即寻址。在读取指令时也就从指令之中获得了操作数，即操作数包含在指令中。
- ② 直接寻址类。直接给出主存地址或寄存器编号，从 CPU 内或主存单元内读取操作数。
- ③ 间接寻址类。先从某寄存器中或主存中读取地址，再按这个地址访问主存以读取操作数。
- ④ 变址类。指令给出的是形式地址（不是最终地址），经过某种变换（例如相加、相减、高低位地址拼接等），才获得有效地址，据此访问主存储器以读取操作数。

#### 4. 何谓隐式 I/O 指令？其主要特点是什么？

将外围设备接口的有关寄存器与主存统一编址，采用通用的数据传送指令实现 I/O 操作，这种 I/O 指令是隐含在传送指令中，所以又称为隐式 I/O 指令。

这类指令的特点是地址码本身可区分存储单元地址和 I/O 端口地址，不影响标志寄存器的内容。

#### 5. 主机调用外围设备，外设编制可采用那几种方式？

对 I/O 设备的编址方法实际上就是对 I/O 接口中有关寄存器及相应部件的编址方法，有两大类型。

① 外围设备单独编址。早期是为每台 I/O 设备分别分配一个设备码，每个设备下属  $n$  个接口寄存器，在 I/O 指令中给出设备码，并指明接口的哪个寄存器，从而实现 CPU 对外设的访问。现在普遍采用为各 I/O 接口的每个有关寄存器分别分配一种 I/O 端口地址，指令中给出端口地址，也就知道 CPU 访问哪一台设备及其接口寄存器。

② 外围设备与主存统一编址。将各 I/O 接口中的有关寄存器与主存的各编址单元统一编址，为它们分配统一的总线地址。在传送指令中给出这类总线地址，CPU 就可以访问相应的 I/O 设备及其接口寄存器。

#### 6. CPU 内部各寄存器的功能？

##### 可编程寄存器：

(1) 通用寄存器：可由 CPU 编程访问，能实现多种功能的寄存器。例如可提供操作数，存放运算结果，用作地址指针，作为变址寄存器、基址寄存器、计数器等。

(2) 程序计数器 PC：用来指示指令在存储器中存放位置的寄存器。PC 的内容是指令所在存储单元的地址，取指后，PC 内容增量计数，指向下一条指令的地址。

(3) 程序状态字 PSW：用来记录现行程序的运行状态和指示程序工作方式的寄存器。

(4) 堆栈指针 (SP)：仅 1 个，固定存放堆栈的栈顶单元的地址码。

##### 非编程寄存器：

(1) 暂存器：为避免破坏通用寄存器的内容，用来暂时存放某些中间结果的寄存器。

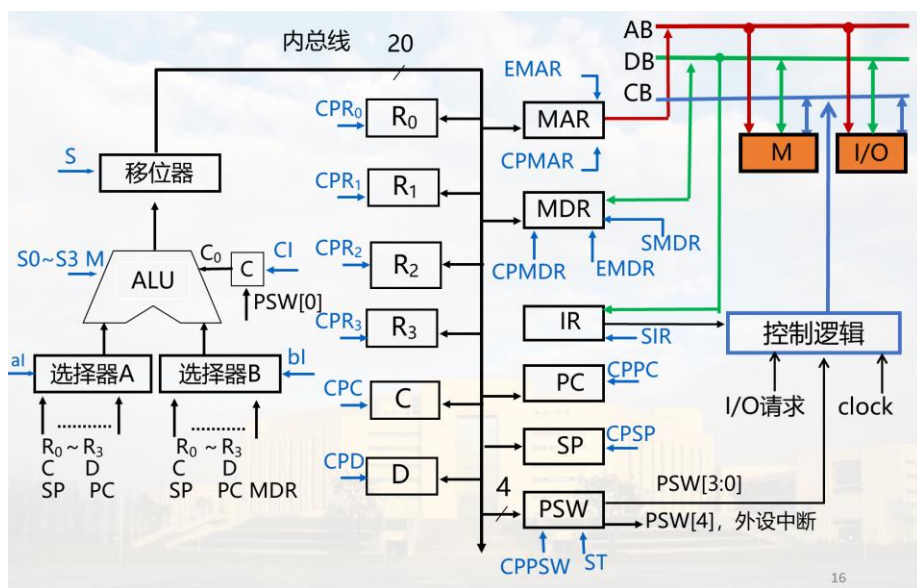
(2) 指令寄存器 IR：用来存放现行指令的寄存器。当需要执行某条指令时，先将该指令从存储器取出，并存入 IR 中，然后再对 IR 的内容进行译码。

(3) 地址寄存器 (MAR)：只有 1 个，读写存储器时，先要定位存储单元，因此设置 MAR 来存放目标单元的地址码。

(4) 数据缓冲寄存器 (MDR)：只有 1 个，过渡性地存放 CPU 与主存之间交换的数据。

#### 7. 数据通路结构？

为了使数据传送控制简单、集中，采用以 ALU 为中心的总线结构。内总线采用单向数据总线 (20 位)，主要由 ALU 部件、寄存器组；存储器；控制器构成。



## 8. 何谓同步控制方式，何谓异步控制？有何主要特征？应用场合？

**同步控制方式：**各项操作受统一时序控制。

(1) 特点：时钟周期时间固定，各步操作的衔接、各部件之间的数据传送受严格同步定时控制。

(2) 优缺点：时序关系简单，时序划分规整，控制不复杂；控制逻辑易于集中，便于管理；时间安排不合理。

(3) 应用：用于 CPU 内部、设备内部、系统总线操作(各挂接部件速度相近，传送时间确定，传送距离较近)。

**异步控制方式：**各项操作按不同需要安排时间，不受统一时序控制。

(1) 特点：无统一时钟周期划分，各操作间的衔接和各部件之间的信息交换采用异步应答方式。

(2) 优缺点：时间安排紧凑、合理；控制复杂

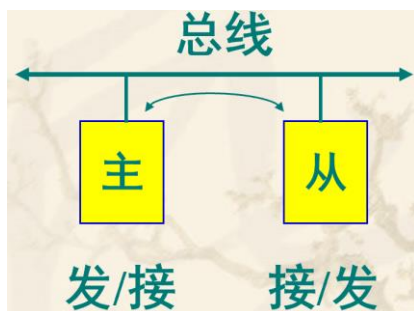
(3) 应用场合：用于异步总线操作(各挂接部件速度差异大，传送时间不确定，传送距离较远)。

## 9. 何谓主从设备，试举例说明。

**主设备：**申请并掌握总线权的设备。

**从设备：**响应主设备请求的设备。

例. 异步传送操作

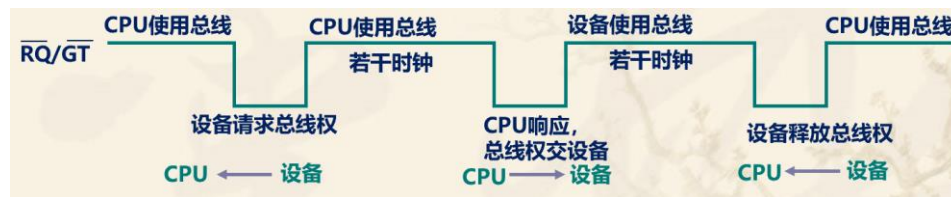


(1) 主设备获得总线控制权；(2) 主设备询问从设备；(3) 从设备准备好 - 接收否则再次询问。(4) 主设备发送/接收数据。(5) 主设备释放总线控制权

# 10. 同步控制中如何引入异步应答的，试举例说明。

以固定时钟周期作为时序基础，引入应答思想。

例. 8088 最大模式，用一根总线请求/应答线实现总线权的转移。



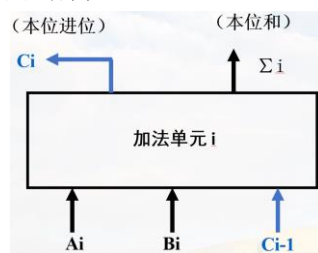
# 11. 时序信号？时序系统？

时序信号：周期、节拍、脉冲等频率型信号序列

时序系统：用来产生时序信号（如周期、节拍、脉冲等）的部件称为时序系统或时序发生器，它由一个振荡器和一组计数分频器组成。

# 12. 一位全加器的结构及关系表达式。

加法器结构：



逻辑关系：

$$\Sigma_i = (A_i \oplus B_i) \oplus C_{i-1}$$

$$C_i = A_i B_i + (A_i \oplus B_i) C_{i-1}$$

# 13. 判断下面叙述是否正确，说明理由：串行加法器中的进位链是串行进位链，并行加法器中的进位链只有并行进位链。

错误：并行加法器中的进位链可以并行串行结合。

并行加法器：多位全加器同时实现多位二进制数的加法。

串行加法器：只有一个全加器，数据逐位串行送入加法器中进行运算，一次只能运算一位，效率低。

# 14. 常用的加法器进位链结构有哪几种？

串行进位、并行进位

# 15. 组合逻辑控制器思想的基本要点是什么？

简单地讲，由硬连逻辑电路产生微命令的方式称为组合逻辑控制方式。它的基本思想如下：综合、化简产生微命令的条件，形成相应逻辑式，并用组合逻辑电路实现；执行指令时，由组合逻辑电路(微命令发生器)在相应时间发出所需微命令，控制有关操作。

# 16. 微程序控制思想的基本要点是什么？

简单地讲，由微指令译码产生微命令的方式称为微程序控制方式。它的基本思想如下：将若干微命令编制成一条微指令，控制实现一步操作；将若干微指令组成一段微程序，解释执行一条机器指令；将微程序事先存放在控制存储器中，执行机器指令时再取出。

### 17. 组合逻辑控制器、微程序控制器有何区别？

组合逻辑控制和微程序控制是用来控制如何产生微命令的两种方式。

**组合逻辑控制方式**是直接通过逻辑门电路产生微命令的，因而它的主要优点是产生微命令的速度很快。其主要缺点有两点：

①设计不规整，因而难于实现设计自动化；

②采用硬连逻辑，不易修改和扩展指令系统的功能。组合逻辑控制方式主要用于高速计算机或小规模计算机中。

**微程序控制方式**是通过执行微指令来产生微命令的，主要优点如下：

① 设计规整，设计效率高；

② 易于修改、扩展指令系统功能；

③ 结构规整、简洁，可靠性高；

④ 性价比高。其主要缺点是：产生微命令的速度慢，因为要多次访问控存读取微指令，访存速度限制了产生微命令的速度；

另外，机器的执行效率不高，因为微指令格式较简单，没有充分发挥数据通路本身所具有的并行能力。

微程序控制方式主要用于速度要求不高、功能较复杂的机器中，特别适合于系列机。

### 18. 组合逻辑控制器、微程序控制器的时序系统是如何划分的？

组合逻辑控制器的时序系统：

（1）工作周期：一个指令周期中，完成某一阶段操作所需的时间称为工作周期。

如取指周期、源周期、目的周期、执行周期等。

（2）时钟周期：CPU 执行一步操作所需的时间称为一个时钟周期。时钟周期作为时序基准，在一个计算机中其长度是固定不变的。

（3）时钟脉冲：对微操作进行定时控制。

微程序控制器的时序系统：

（1）微指令周期

（2）时钟脉冲

### 19. 微命令、微操作、微指令、微指令周期、微程序？

**微命令**：在计算机中用来控制微操作（如逻辑门的开或关、寄存器的打入或清除等操作）的控制命令称为微命令，也称为微操作控制信号。

**微指令**：将一步操作所需的微命令编写在一串代码中，这串代码称为微指令。它由微命令字段和微地址字段组成。

**微程序**：由若干条微指令组成一段微程序，用来解释执行一条机器指令。

**微指令周期**：读取并执行一条微指令所用的时间称为微指令周期。

## 二、计算题

1. 扩展操作码：根据条件计算指令条数？

2. 根据寻址方式寻找操作数？

3. 并行加法器中的串行进位链结构:  $C_n = G_n + P_n C_{n-1}$   
并行进位链结构:  $C_n = G_n + P_n G_{n-1} + \dots + P_n \dots P_1 C_0$

### 三、设计题:

CPU 的逻辑组成及工作机制

#### 1. CPU 的逻辑组成 (模型机框图)

- (1) CPU 的逻辑组成→模型机框图;
- (2) CPU 内每个寄存器的作用;
- (3) 总线的分类及定义;
- (4) 控制器的分类及区别;

#### 2. CPU 的指令流程

(1) 指令类型: MOV 指令、双操作数算术逻辑运算指令、单操作数算术逻辑运算指令、转移/返回指令、转子指令;

- (2) 核心是寻址方式: 立即寻址、R、(R)、-(R)、(R)+、@(R)+、X(R);

#### 3. 操作时间表的安排 (微命令的安排):

- (1) CPU 数据通路操作: 按照数据的流向分成四段

ALU 输入选择→AUL 功能选择→移位器功能选择→分配脉冲 (打入到寄存器中的脉冲);

- (2) 与访问主存有关的微命令。

## 第四章 存储子系统

### 一、基本概念

#### 1. 主存、缓存、外存?

缓存: 容量小、速度高; 主存: 容量较大、速度较高; 外存: 容量大、速度慢。

三级存储体系包括缓存 (cache)、内存和外存, 这种模式的优点是层次体系清晰、便于设计实现, 也利于系统调度管理, 能提高存储系统性能; 缺点是结构复杂, 管理和控制都比较复杂, 硬件成本高。

#### 2. 高速缓存 Cache 用来存放什么内容? 设置它的主要目的是什么?

(1) 用来存放当前活跃的数据与程序, 作为主存活跃区的副本; (2) 设置高速缓存是为了解决 cpu 与主存的速度匹配;

#### 3. 存储介质?

按存储介质, 分为半导体存储器, 磁表面存储器、光盘存储器。

#### 4. 何谓随机存取? 何谓顺序存取? 何谓直接存取? 请各试举一例。

随机存取是指按地址码访问存储器, 访存时间和存储单元的位置无关, 比如内存;

顺序存取是指按存储介质的存储顺序移动介质、定位存储区并读写数据, 比如磁带;

直接存取是指先寻道再定位扇区, 然后再读写数据, 比如硬盘。

#### 5. 静态存储器 SRAM、动态存储器 DRAM 存储原理?

静态存储器 SRAM (静态 MOS 型): 依靠双稳态电路内部交叉反馈的机制存储信息。功耗较大, 速度快, 作 Cache。

动态存储器 DRAM (动态 MOS 型): 依靠电容存储电荷的原理存储信息。功耗较小,



容量大,速度较快,作主存。

## 6. 动态刷新分为哪几种情况,各有什么特点?

最大刷新闻隔:大多数 DRAM 要求:保持 2ms-64ms 期间一定对所有单元刷新。

刷新方法:各动态芯片可同时刷新,片内按行刷新。

刷新周期(存取周期):刷新一行需要的时间。

刷新周期数:刷新一片芯片所需的周期数由芯片矩阵的行数决定。

对主存的访问:

CPU 访存:由 CPU 提供行列地址随机访问

芯片刷新:由刷新地址计数器提供地址定时刷新

刷新周期的安排方式:

集中刷新:2ms 内集中安排所有刷新周期。用于实时要求不高的场合

分散刷新:各刷新周期分散安排在存取周期中。造成主存利用率降低,用在低速系统中

异步刷新:刷新周期分散安排在 2ms 内。

## 7. 刷新、重写的区别?

动态 MOS 存储单元:

(1) 四管单元:保持原状态需要定期向电容补充电荷。读出是非破坏性读出,读出过程即实现刷新。

(2) 单管单元:单管单元是破坏性读出,读出后需重写。

刷新:动态存储器,需要定期补充电容以保持原来信息。

重写:破坏性读出后需要重写,以恢复原来的信息。

## 8. 全地址译码方式? 部分地址译码方式?

全译码方式:将除了与芯片连接的地址以外的所有高位地址用于译码产生片选信号。

部分译码方式:将除了与芯片连接的地址外的部分高位地址用于译码产生片选信号(易出现地址重叠)。

## 二、设计题:

半导体存储器逻辑设计:(地址分配、片选逻辑、逻辑框图),片选逻辑采用全/部分译码方式。

# 第五章 I/O 系统

## 一、基本概念

### 1. 系统总线上一般包含哪三组信息?

总线:一组为多个部件分时共享的信息传送线路。

特点:分时共享。特定时刻只允许一个部件送出数据到总线上。

总线周期:完成一次主存或 I/O 端口访问的时间

总线分类:

(1) 按传输信号的类型:数据总线(传输数据信息,决定总线宽度);地址总线(传输地址信息,决定寻址能力);控制总线(传输控制信息和状态信息)

(2) 数据传送格式:并行总线;串行总线

(3) 按时序控制方式:同步总线(由统一时序信号控制总线完成传送操作,在固定时钟周

期内完成数据传送, 由同步脉冲打入); 异步总线(无固定时钟周期, 以异步应答的形式控制传送); 扩展同步总线(以时钟周期为基础, 允许总线周期中时钟数可变)

(4) 按功能: 内总线(CPU 内部线); 局部总线(主板各芯片之间互连的总线); 系统总线(主板与插件板之间互连总线); 外总线(计算机系统与外部设备之间互连的总线)

## 2. 中断方式定义、实质、特点?

中断含义: CPU 暂时中止现行程序的执行, 转去执行为某个随机事件服务的中断处理程序。处理完毕后自动恢复原程序的执行。

实质: 程序切换

方法: 保存断点、保护现场、恢复现场、返回断点

时间: 一条指令结束时切换, 保证程序的完整性

特点: 随机性

随机发生的事态(按键、故障)

有意调用, 随即请求与处理的事态(调用打印机)

随机插入的事态(软中断指令可插入程序任何位置)

## 3. 中断向量、中断向量表、向量地址?

中断类型码: 每个中断源对应的编号

中断向量: 中断服务程序入口地址

中断向量表: 存放各中断服务程序的入口地址的单元

向量地址: 访问中断向量表的地址码, 可通过计算中断类型码得到

模型机中: 向量地址 = 中断号 + 2

IBM PC 向量表: 向量地址 = 中断号 \* 4

## 4. 何谓向量中断? 何谓非向量中断? 各有何优点和缺点?

是根据中断源获取服务程序入口

非向量中断: 由软件查询提供服务程序入口地址

向量中断: 由硬件直接提供服务程序入口地址

## 5. 单级中断、多重中断?

中断处理: 主要任务: CPU 执行中断服务程序

单级中断流程: 保护现场、中断服务处理、恢复现场、开中断、返回

多级中断流程:

保护现场:

送新屏蔽字、开中断(禁止同级别或更低级别的请求, 开发更高级别的请求)

中断服务处理

关中断

恢复现场及原屏蔽字

开中断

返回

## 6. 转子与中断调用的相似、区别?

相同点: 程序切换

区别:

(1) 子程序的执行由程序员事先安排, 而中断服务程序的执行则是由随机中断事件触发。

(2)子程序的执行受主程序或上层程序控制,而中断服务程序一般与被中断的现行程序无关。

(3)一般不存在同时调用多个子程序的情况,但可能发生多个外设同时向 CPU 发出中断服务请求的情况。

## 7. DMA 方式定义、实质、特点？

直接依靠硬件系统来控制主存和设备之间的数据传送，传送期间无需 CPU 干预，传送结束后通常通过中断方式通知 CPU。

特点：

响应随机请求

不影响 CPU 程序的执行，仅占用总线、无程序切换

大批量数据的简单传送

典型的应用：

主存和高速 I/O 设备之间的简单数据传送

大批量数据采集系统

动态存储器 DRAM 的自动刷新

DMA 传送过程：

程序准备：主程序实现初始化（对 DMA 控制器和接口  
传送请求

DMA 传送：存储器 - 直传 - I/O（硬件实现）

善后处理：执行中断处理程序

## 8. 程序直传方式，中断方式，DMA 方式之间的区别？

直接程序传送：依靠 CPU 直接执行相关的 I/O 程序来实现数据的输入和输出控制。

中断方式：设备提出中断请求，主机响应后与设备交换信息，接口中包含中断控制请求。

DMA 方式：直接依靠硬件系统来控制主存和设备之间的数据传送，传送期间无需 CPU 干预，传送结束后通常通过中断方式通知 CPU，支持高速外设与主存之间进行 DMA 方式交换数据。

## 9. 中断控制器 8259A 的工作原理？

当一个中断请求从 IRQ0 到 IRQ7 中的某根线到达 IMR 时，IMR 首先判断此 IR 是否被屏蔽，如果被屏蔽，则此中断请求被丢弃；否则，则将其放入 IRR 中。

8259 比较 ISR 和 IRR 中的优先级，若 IRR 有优先级高的外部中断请求，则通过发送一个 INT 信号给 CPU，通知 CPU 有一个中断到达。

CPU 收到这个信号后，会暂停执行下一条指令，然后发送一个 INTA 信号给 8259。

8259 收到这个信号后，将中断号送到数据总线上。

CPU 从数据总线读取中断号，形成中断服务程序的入口地址（中断周期）。转入中断服务程序。

中断请求 → Intel 8259 (未被屏蔽的请求判优, 生成相应中断号)

→ 公共请求INT → CPU →

→ CPU响应中断请求(如果CPU允许响应中断) →

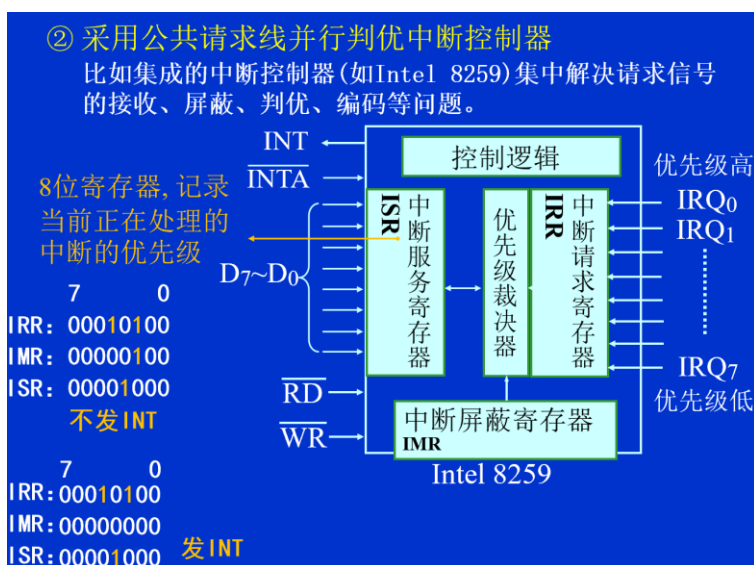
→ CPU发出响应信号 $\overline{INTA}$ 到Intel 8259 →

→ Intel 8259将中断号送往数据总线 →

→ CPU从数据总线读取中断号, 由中断号形成中断服务程序入口地址, 并转入中断服务程序)

## 二、计算题

1.在 8259A 中, 如何通过 IRR、IMR、ISR 的值, 判断中断接口是否向 CPU 发 INT 信号。



2.如何根据中断类型码(或中断号)生成向量地址。

向量地址: 访问中断向量表的地址码, 可通过计算中断类型码得到

模型机中: 向量地址=中断号+2

IBM PC 向量表: 向量地址 = 中断号 \* 4

## 《计算机系统结构》考试题型

- 一、 名词解释题 (9 分, 3 题 × 3 分)
- 二、 简答题 (20 分, 4 题 × 5 分)
- 三、 判断正误, 并说明原因 (20 分, 5 题 × 4 分)
- 四、 计算题 (24 分, 4 题 × 6 分)
- 五、 设计题: 存储器 (通过译码器片选, 全译码/部分译码) (12 分)
- 六、 综合题: CPU (模型机设计、指令流程、微命令设置) (15 分)