指令系统

机器指令处于硬件和软件的交界面

微指令是微程序级命令，属于硬件范畴

汇编指令是机器指令的汇编表示形式，即符号表示

指令包含的信息：操作码 + 源操作数 + 结果值 + 下一条指令地址（隐含在 PC 中）

指令集体系结构规定（考过一次）：

数据类型及格式，指令格式，寻址方式和地址空间大小

通用寄存器和控制寄存器定义

I/O 空间编址方式，中断结构，输入输出结构和数据传输方式，存储保护方式

指令格式

指令的基本格式：操作码 + 地址码

指令系统中指令长度是否相同

定长指令字结构：速度快，控制简单

变长指令字结构：字长整数倍

单字长指令，双字长指令，半字长指令（与机器字长比）

操作数地址码的数目不同：零地址指令（运算类的尽在堆栈类计算机中使用）、一地址指令、多地址指令

指令格式——指令操作码的扩展技术

指令操作码的长度决定了指令系统中完成不同操作的指令数

若某机器的操作码长度固定为K位，则它最多只能有2^K条不同指令

固定长度操作码：

优点：对于简化硬件设计，减少指令译码时间非常有利

缺点：指令少，浪费地址

可变长度操作码（分散地放在字的不同字段，前缀编码）：

优点：指令多，缩短指令平均长度，减少程序总位数，增加指令字所能表示的操作信息

缺点：译码复杂，控制器的设计难度增大

拓展操作码的一个重要原则：使用频度高的指令应分配短的操作码，使用频度低的指令相应地分配较长的操作码

[2017年] 某计算机按字节编址，指令字长固定且只有两种指令格式，其中三地址指令29条、二地址指令107条，每个地址字段为6位，则指令字长至少应该是（　）。

A. 24 位

B. 26 位

C. 28 位

D. 32 位

[2022年] 设计某指令系统时，假设采用 16 位定长指令字格式，操作码使用扩展编码方式，地址码为 6 位，包含零地址、一地址和二地址 3 种格式的指令。若二地址指令有 12 条，一地址指令有 254 条，则零地址指令的条数最多为（　　）。

A. 0

B. 2

C. 64

D. 128

二地址指令	OP(4b)	地址1(6b)	地址2(6b)
一地址指令	OP(4b + 6b)		地址(6b)
零地址指令	OP(4b + 6b + 6b)		

寻址方式

寻址方式的确定：在操作码中给定寻址方式、有专门的寻址方式位

指令寻址：

顺序寻址：通过程序计数器(PC) + n → PC（n为指令长度），自动生成下一条指令的地址

跳跃寻址：最终修改PC的值，通过转移类指令实现，是否跳跃受到状态寄存器和操作数的控制

绝对地址：由标记符直接得到

相对地址：相对于当前指令地址的偏移量

数据寻址：指令格式：操作码 | 寻址特征 | 形式地址A，有效地址EA

- 隐含寻址：不显示的给出操作数地址，在指令中隐含操作数地址（规定ACC为第二操作数地址）
- 立即寻址：不访存，A就是补码操作数
- 直接寻址：一次访存，EA = A
- 间接寻址：多次访存，以一次为例，EA = (A)
- 寄存器寻址：不访存，A为操作数所在的寄存器编号
- 寄存器间接寻址：访存多次，以一次为例，，为操作数地址所在的寄存器编号，EA = (Ri)
- 偏移寻址：EA = 偏移量 + 基础地址
  - 相对寻址：一次访存，A为偏移量，基础地址为自增后的(PC)，EA = A + (PC)，广泛用于转移指令，方便代码在程序内部浮动
  - 基址寻址：一次访存，A为偏移量，基础地址保存在基址寄存器BR中，EA = (BR) + A，面向操作系统，解决程序逻辑空间与存储物理空间的无关性，有利于多道程序的设计
  - 变址寻址：一次访存，A为基础地址，偏移量保存在变址寄存器IX中，EA = (IX) + A，面向用户，可用于数组处理，适合编写循环代码，A是数组首地址，IX不断加一。
- 堆栈寻址：操作数保存在堆栈中，有硬堆栈（寄存器构成）和软堆栈（主存划分的区域）

[2009年] 某机器字长为 16 位，主存按字节编址，转移指令采用相对寻址，由 2 字节组成，第一字节为操作码字段，第二字节为相对位移量字段。假定取指令时，每取一字节 PC 自动加 1 。若某转移指令所在主存地址为 2000H，相对位移量字段的内容为 06H，则该转移指令成功转移后的目标地址是（　）。

A. 2006H

B. 2007H

C. 2008H

D. 2009H

[2011年] 某机器有一个标志寄存器，其中有进位/借位标志 CF、零标志 ZF、符号标志 SF 和溢出标志 OF，条件转移指令 bgt（无符号整数比较大于时转移）的转移条件是（　）。

A. $CF + OF = 1$　　B. $\overline{SF} + ZF = 1$　　C. $\overline{CF + ZF} = 1$　　D. $\overline{CF + SF} = 1$

[2019年] 某计算机采用大端方式，按字节编址。某指令中操作数的机器数为 1234 FF00H，该操作数采用基址寻址方式，形式地址（用补码表示）为 FF12H，基址寄存器的内容为 F0000000H，则该操作数的 LSB（最低有效字节）所在的地址是（　）。

A. F000 FF12H

B. F000 FF15H

C. EFFF FF12H

D. EFFF FF15H

[2023年] 某运算类指令中有一个地址码为通用寄存器编号，对应通用寄存器中存放的是操作数或操作数的地址，CPU 区分两者的依据是（　）。

A. 操作数的寻址方式

B. 操作数的编码方式

C. 通用寄存器的编号

D. 通用寄存器的内容

汇编指令格式

	AT&T格式	Intel格式
大小写敏感性	仅支持小写（如movl）	不敏感（MOV/mov均可）
操作数顺序	源操作数在前，目的在后（mov $1，%eax）	目的在前，源在后（mov eax，1）
前缀格式	寄存器加%（%eax）立即数加$（$1）	无前缀（直接写eax或1）
内存寻址符号	使用圆括号（movl (%ebx)，%eax）	使用方括号（mov eax，[ebx]）

CISC 和 RISC

	CISC（复杂指令集）	RISC（精简指令集）
指令系统	复杂庞大（指令>200条）	简单精简（指令<100条）
指令字长	不固定	定长
可访存指令	不加限制	只有Load/Store指令
指令执行时间	相差较大	绝大多数一个周期（流水线下宏观一个周期）
指令使用频率	相差很大	都比较常用
通用寄存器数量	很少（微程序控制存储器占CPU>50%）	多（组合逻辑控制，硬布线仅占10%）
目标代码	难以用优化编译生成高效代码	优化编译生成高效代码
控制方式	绝大多数为微程序控制	绝大多数为组合逻辑控制
指令流水线	可通过一定方式实现	必须实现
兼容性	软件兼容低档机	无法兼容

[2009年] 下列关于 RISC 的说法中，错误的是（　　）

A. RISC 普遍采用微程序控制器

B. RISC 大多数指令在一个时钟周期内完成

C. RISC 的内部通用寄存器数量相对 CISC 多

D. RISC 的指令数、寻址方式和指令格式种类相对 CISC 少

CPU 的功能和基本结构

CPU功能：周而复始的执行指令、执行指令过程中，发现处理异常、每个指令结束，查询并响应中断

指令周期：CPU从主存中取出并执行一条指令的全部时间

机器周期（CPU周期）：内存中读取一个指令字的最短时间（通常由存取周期决定，指令字长为存储字长n倍，则n次取数据，机器周期为存取周期n倍）

执行各条指令的机器周期数可变，各机器周期的长度可变

[2016年] 某计算机的主存储器空间为 4 GB，字长为 32 位，按字节编址，采用 32 位字长指令字格式。若指令按字边界对齐存放，则程序计数器（PC）和指令寄存器（IR）的位数至少分别是（　）。

A. 30，30

B. 30，32

C. 32，30

D. 32，32

CPU 执行指令的过程

指令周期：指令周期 → 取值周期 → 间址周期（取操作数的有效地址）→ 执行周期 → 中断周期

指令执行过程：

- 取指令 → 缺页，越界，访问权限保护错
- PC+1送PC
- 指令译码 → 非法指令
- 进行主存地址运算
- 取操作数 → 缺页异常，越界，访问权限保护错
- 算数逻辑运算，存结果 → 溢出，缺页
    - 以上每步都需检测“异常”，自动切换异常处理程序

指令执行方案：

- 单指令周期：每条指令在一个时钟周期内完成，时钟周期选取时间最长的访存指令，必须使用多总线
- 多指令周期：把一个指令的执行分成多个阶段，每个阶段在一个时钟周期内完成（时钟周期以最复杂阶段所花时间为准）规定每个阶段最多只能完成1次访存或寄存器堆读/写或ALU
- 流水线方案

[2009年] 冯·诺依曼计算机中指令和数据均以二进制形式存放在存储器中，CPU 区分它们的依据是（　）。

A. 指令操作码的译码结果
B. 指令和数据的寻址方式
C. 指令周期的不同阶段
D. 指令和数据所在的存储单元

数据通路

- 数据通路由两种元件组成：
    - 组合逻辑元件（操作元件）：加法器、多路选择器MUX、算术逻辑单元ALU、译码器等
    - 时序逻辑元件（存储元件）：具有存储功能的元件（寄存器和存储器）
- 单总线数据通路：
    - 所有寄存器的输入输出端连接在同一条公共通路上
    - 存在冲突，同一时钟周期仅传送一个数据，即仅一个寄存器输出，可多寄存器读数据
    - 时钟周期往往以一个寄存器输出开始，寄存器输入结束，选择最长的时间，故以Read/Write为准
- 三总线数据通路：多总线上传送不同数据，提高效率
- 专用数据通路方式

[2016年] 单周期处理器中所有指令的指令周期为一个时钟周期。下列关于单周期处理器的叙述中，错误的是（　）。

A. 可以采用单总线结构数据通路

B. 处理器时钟频率较低

C. 在指令执行过程中控制信号不变

D. 每条指令的 CPI 为 1

[2021年] 下列关于数据通路的叙述中，错误的是（　　）。

A. 数据通路包含 ALU 等组合逻辑（操作）元件

B. 数据通路包含寄存器等时序逻辑（状态）元件

C. 数据通路不包含用于异常事件检测及响应的电路

D. 数据通路中的数据流动路径由控制信号进行控制

微程序控制器

将每条机器指令编成一个微程序，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

微命令：是微操作的控制信号 （控制序列最小单位）

微操作：是微命令的执行过程（最基本、不可再分操作）

微指令：是若干微命令的集合

微周期：指读取一条微指令并执行相应微操作所需时间

微程序：微命令的有序集合（一条指令功能由一段微程序实现）

伪指令编码方式

- 直接编码法：微指令的微命令字段中每位都代表一个微命令，选不选用只要将对应位置置0或1
- 字段直接编码法：将微命令字段分成若干段，把互斥微命令组合装同一段，每个字段独立编码，每个编码代表一种微命令（000表示无微指令，一个字段段中n条指令需要 $\lceil log_2(n+1) \rceil$ 位）
- 字段间接编码法：一个字段的微指令由另一个字段中的微指令解释（两层译码，缩短指令字长，但削弱并行控制能力）
- 下条微地址的确定：
  - 增量（计数器）法：下条微指令地址隐含在微程序计数器uPC中
  - 断定（下址字段）法：在本条微指令中明显指定下条微指令地址

控制器的实现比较

	硬布线控制器	微程序控制器
执行速度	快	慢
规整性	繁琐、不规整	规整
应用场合	RISC	CISC
易扩充性	不易扩展	易扩展

[2012年] 某计算机的控制器采用微程序控制方式，微指令中的操作控制字段采用字段直接编码法，共有 33 个微命令，构成 5 个互斥类，分别包含 7、3、12、5 和 6 个微命令，则操作控制字段至少有（　　）。

A. 5 位

B. 6 位

C. 15 位

D. 33 位

[2014年] 某计算机采用微程序控制器，共有 32 条指令，公共的取指令微程序包含 2 条微指令，各指令对应的微程序平均由 4 条微指令组成，采用断定法（后继地址字段法）确定下条微指令地址，则微指令中后继地址字段的位数至少是（　）。

A. 5

B. 6

C. 8

D. 9

异常和中断

异常（内中断）：

陷阱 （返回下一条）：陷阱指令自愿陷入

故障 （返回当前）：当前指令引起的意外事件

访存缺页属于异常（故障）

除0异常是故障，处理手段是终止该进程

终止(不会返回)： 硬件检测到致命错误，区分故障终止看是否与当前指令有关

外中断（返回下一条指令）：区分异常还是外中断看来自外部内部

[2015年] 内部异常（内中断）可分为故障（fault）、陷阱（trap）和终止（abort）三类。下列有关内部异常的叙述中，错误的是（　）。

A. 内部异常的产生与当前执行指令相关

B. 内部异常的检测由 CPU 内部逻辑实现

C. 内部异常的响应发生在指令执行过程中

D. 内部异常处理后返回到发生异常的指令继续执行

[2020 年] 下列关于“自陷”(Trap，也称陷阱)的叙述中，错误的是（　）

A. 自陷是通过陷叶指令预先设定的一类外部中断事件

B. 自陷可用于实现程序调试时的断点设置和单步跟踪

C. 自陷发生后 CPU 将转去执行操作系统内核相应程序

D. 自陷处理完成后返回到陷阱指令的下一条指令执行

指令流水线——流水线冒险处理

数据冒险（流水线按序流动，只会出现写后读）

- 硬件阻塞（stall）：控制复杂，要加阻塞电路和转发检测电路
- 软件插入NOP指令：最差的做法，硬件开销小
- 合理实现寄存器堆的读/写操作，同一周期内先写后读，但是不能解决所有数据冒险
- 旁路/转发技术：把数据从流水段寄存器直接送到ALU的输入端
  - Load-use数据冒险无法利用转发旁路技术解决冲突，因为load指令最早在第四阶段MEM后产生结果，故不能解决load指令和随后第一条指令间的数据冒险，需阻塞一个时钟周期
- 编译优化：调整指令顺序（不能解决所有数据冒险）

控制冒险：

- 分支预测技术：静态分支预测、动态分支预测
- 延迟分支（软件）：将分支指令无关的指令调到分支指令后，用于填充延迟槽（延迟损失时间片）

结构冒险：同一部件同时被不同指令所使用

- 设置多个部件，避免冲突。如指令存储器IM和数据存储器DM分开
- 寄存器读口和写口独立开，时钟周期的前半个执行写操作，后半个读操作，避免冲突

指令流水线——高级流水线

- 超标量流水线技术（空分复用）：每个时钟周期内可并发多条独立的指令
- 超长指令字技术（时分复用）：由编译程序挖掘出指令间潜在的并行性，将多条能并行操作的指令合成一条
- 超流水技术（指令的潜在并行性）：在一个时钟周期内再分段，在一个时钟周期内一个功能部件使用多次

[2013年] 某 CPU 主频为 1.03GHz，采用 4 级指令流水线，每个流水段的执行需要 1 个时钟周期。假定 CPU 执行了 100 条指令，在其执行过程中，没有发生任何流水线阻塞，此时流水线的吞吐率为（　）。

A. 0.25×10^9 条指令/秒

B. 0.97×10^9 条指令/秒

C. 1.0×10^9 条指令/秒

D. 1.03×10^9 条指令/秒

[2019年] 在采用 “取指、译码/取数、执行、访存、写回” 5 段流水线的处理器中，执行如下指令序列，其中 s0、s1、s2、s3 和 22 表示寄存器编号。下列指令对中，不存在数据冒险的是(　)。

```
I1：  add s2，s1，s0          //r[s2] ← R[s1]+R[s0]
I2：  load s3，0(t2)          //R[s3] ← M[R[t2]+0]
I3：  add s2，s2，s3          //R[s2] ← R[s2]+R[s3]
I4：  store s2，0(t2)         //M[R[t2]+0] ← R[s2]
```

A. I1 和 I3

B. I2 和 I3

C. I2 和 I4

D. I3 和 I4

[2020年] 下列给出的处理器类型中，理想情况下，CPI 为 1 的是（　）。

I. 单周期 CPU

II. 多周期 CPU

III. 基本流水线 CPU

IV. 超标量流水线 CPU

A. 仅 I、II

B. 仅 I、III

C. 仅 II、IV

D. 仅 III、IV

总线

- 分时：同一时刻只允许有一个部件向总线发送信息
- 共享：总线上可以挂接多个部件，各个部件之间互相交换的信息可以通过这组线路分时共享
- 猝发传输：一个总线周期内传输存储地址连续的多个数据字的总线传输方式
- 芯片内总线：在芯片内部各元件之间提供连接
- 系统总线：系统主要功能部件间提供连接
  - 数据总线 （双向）：数据线宽度反映一次能传送数据的位数，数据（指令、操作数、中断信号）
  - 地址线 （单向）：地址线宽度反映最大的寻址空间
  - 控制线：传输控制信号如时钟、复位、总线请求/允许、中断请求/回答、存储器读/写、I/O读/写、传输确认等
- 通信总线：在主机和I/O设备之间或计算机系统之间提供连接
- 串行总线：一条传输线（长距离传输，价格低廉）
- 并行总线：成本高，布线空间干扰，随着技术提高，并行总线不一定比串行总线快

总线——系统总线的结构

- 单总线结构：将CPU、主存、I/O设备都挂载到一组总线上
- 双总线结构：一条是主存总线，连接CPU、主存、通道，另一条是I/O总线用于在多个外部设备和通道间传输
- 三总线结构：主存总线、I/O总线、直接内存访问 DMA 总线
- 四总线结构：CPU总线、系统总线、高速总线、扩充总线
- 总线的传输周期：一次总线操作所需要的时间，总线传输周期由若干总线时钟周期构成（申请阶段、寻址阶段、传输阶段、结束阶段）
- 总线带宽（最高传输速率）= 总线工作频率 × 总线宽度

总线定时方式

- 同步定时方式：控制线上用一个时钟信号进行定时，有确定的通信协议
- 异步定时方式：用握手信号定时（应答方式），异步总线传送由CPU和设备控制器通信双方协调控制，传送操作按需分配时间
  - 不互锁方式
  - 半互锁方式
  - 全互锁方式
- 半同步方式：解决异步方式对噪声敏感问题，就绪和应答等握手信号都在时钟的上升沿有效
- 请求-回答方式：CPU启动一次读或写事物，然后等待存储器回答
- 分离总线事物方式：CPU启动一次读/写事务后释放总线，存储器启动一次回答事务，请求使用总线

[2009年] 假设某系统总线在一个总线周期中并行传输 4 字节信息，一个总线周期占用 2 个时钟周期，总线时钟频率为 10MHz，则总线带宽是（ ）。

A. 10MB/s

B. 20MB/s

C. 40MB/s

D. 80MB/s

[2012年] 某同步总线的时钟频率为 100MHz，宽度为 32 位，地址/数据线复用，每传输一个地址或数据占用一个时钟周期。若该总线支持突发（猝发）传输方式，则一次“主存写”总线事务传输 128 位数据所需要的时间至少是（ ）。

A. 20 ns

B. 40 ns

C. 50 ns

D. 80 ns

[2014年] 某同步总线采用数据线和地址线复用方式，其中地址数据线有 32 根，总线时钟频率为66MHz，每个时钟周期传送两次数据（上升沿和下降沿各传送一次数据），该总线的最大数据传输速率（总线带宽）是（　）。

A. 132MB/s

B. 264MB/s

C. 528MB/s

D. 1056MB/s

I/O 接口

- I/O接口：I/O设备控制器及其插座（网卡、显卡、键盘适配器、磁盘控制器，可编程中断控制器、网络控制器），磁盘驱动器就是磁盘本身，不是I/O接口
- 功能：
  - 数据缓冲：匹配主机和外设工作速度的匹配
  - 错误或状态检测：提供状态寄存器，保存错误或状态信息供CPU查用
  - 控制和定时：提供控制和定时逻辑，接受系统总线来的控制定时信号
  - 数据格式转换：外部接口数据转换为内部接口格式（串并等）
  - 主机和设备通信

- I/O端口：指接口电路中可被CPU直接访问的寄存器
- 统一编址（存储器映射方式）
  - I/O端口当作存储器单元进行地址分配
  - 不需要专门输入输出指令，占用存储器地址，速度慢
  - RISC常用，以不同地址码区分
- 独立编址（I/O映射方式）
  - I/O端口地址与存储器地址相互独立
  - 需要专门的输入输出指令，I/O指令类型少，程序灵活性差
  - I/O指令是指令系统的一部分，是机器指令的一类，为了反映与I/O设备交互的特点，格式和其他通用指令相比有所不同

[2012年] 下列有关 I/O 接口的叙述中，错误的是（　）。

A. 状态端口和控制端口可以合用同一个寄存器

B. I/O 接口中 CPU 可访问的寄存器称为 I/O 端口

C. 采用独立编址方式时，I/O 端口地址和主存地址可能相同

D. 采用统一编址方式时，CPU 不能用访存指令访问 I/O 端口

I/O 传输方式——轮询方式

- 程序直接控制（无条件传送）：对简单外设定时（同步）进行数据传送
- 程序查询方式（条件传送）：I/O设备将状态放到一个状态寄存器，操作系统阶段性查询状态寄存器中特定状态

I/O 传输方式——程序中断方式

中断的处理过程

中断检测（硬件）

中断响应（中断隐指令，硬件）

条件：CPU处于开中断状态、在一条指令执行完后、至少要有一个未被屏蔽的中断请求

过程： 关中断 → 保护断点和程序状态（PC/PSW到堆栈）→ 识别中断源

中断处理（中断服务程序，软件）

保存现场和屏蔽字

执行中断服务程序

恢复现场和屏蔽字

开中断、中断返回

- 中断类型号：用于得到中断向量，可等价
- 中断向量：中断服务程序的入口地址
- 中断向量地址：中断服务程序的入口地址的地址
- 中断响应优先级：多个中断同时请求时选择哪个响应，硬件决定
- 中断处理优先级：设置中断屏蔽标志，改变中断服务程序执行完的顺序

硬件故障属于最高级

非屏蔽中断 > 可屏蔽中断

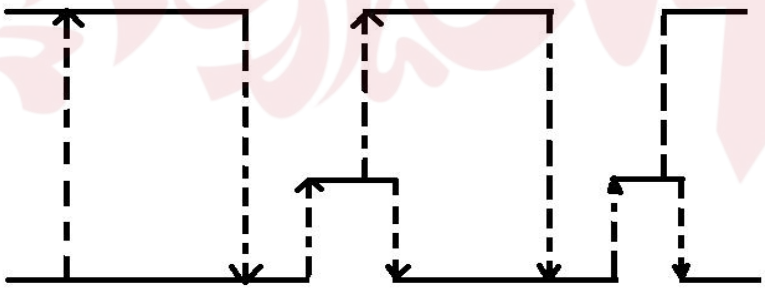
DMA请求 > I/O设备传送的中断请求

高速设备 > 低速设备

输入设备 > 输出设备

实时设备 > 普通设备

访管 > 程序性 > 重新启动



I/O 传输方式——DMA 方式

- I/O设备需要进行数据传送时，向DMA发送DMA请求，DMA控制器向CPU发送I/O请求，CPU响应后让出系统总线，由DMA控制器接管总线，磁盘等高速外设成批地直接和主存进行数据交换
- DMA传送方式：停止CPU访问主存、DMA和CPU交替访存、周期挪用（周期窃取）

过程：

- 预处理：由CPU完成一些必要的准备工作（寄存器置初值、设置传送方向、启动该设备
- 数据传送：DMA的数据传输可以以单字节（或字）为基本单位，也可以以数据块为基本单位，数据传送阶段完全由DMA控制）
- 后处理：DMA控制器向CPU发送中断请求，CPU执行中断服务程序做DMA结束处理

I/O 传输方式

	中断方式	DMA方式
数据传送控制	程序控制（CPU介入）	硬件控制（DMA控制器直接管理）
CPU介入程度	需要程序切换（保护和恢复现场）	仅需预处理与后处理
响应机制	执行完当前指令周期后响应中断	立即响应DMA请求
适用场景	CPU控制，适用于低速设备	DMA控制器控制，适用于高速设备
优先级	优先级低于DMA	优先级高于中断
异常处理	能处理异常事件	仅用于传送数据，不能处理异常

[2010年] 单级中断系统中，中断服务程序内的执行顺序是（　）。

I. 保护现场　　II. 开中断　　III. 关中断　　IV. 保存断点

V. 中断事件处理　　VI. 恢复现场　　VII. 中断返回

A. I → V →VI →II →VII

C. III → IV → V → VI → VII

B. III → I → V → VII

D. IV → I → V →VI→VII

[2011年] 某计算机处理器主频为 50MHz，采用定时查询方式控制设备 A 的 I/O，查询程序运行一次所用的时钟周期数至少为 500 。在设备 A 工作期间，为保证数据不丢失，每秒需对其查询至少 200 次，则 CPU 用于设备 A 的 I/O 的时间占整个 CPU 时间的百分比至少是（　）。

A. 0.02%

B. 0.05%

C. 0.20%

D. 0.50%

[2013 年] 下列关于中断I/O方式和DMA方式比较的叙述中，错误的是（　）。

A. 中断 I/O 方式请求的是 CPU 处理时间，DMA 方式请求的是总线使用权

B. 中断响应发生在一条指令执行结束后，DMA 响应发生在一个总线事务完成后

C. 中断 I/O 方式下数据传送通过软件完成，DMA 方式下数据传送由硬件完成

D. 中断 I/O 方式适用于所有外部设备，DMA 方式仅适用于快速外部设备

[2014 年] 若某设备中断请求的响应和处理时间为 100 ns，每 400ns 发出一次中断请求，中断响应所允许的最长延迟时间为 50 ns，则在该设备持续工作过程中，CPU 用于该设备的 I/O 时间占整个 CPU 时间的百分比至少是(　　)。

A. 12.5%

B. 25%

C. 37.5%

D. 50%

[2020 年] 下列事件中，属于外部中断事件的是（　）。

I. 访存时缺页

II. 定时器到时

III. 网络数据包到达

A. 仅 I、II

B. 仅 I、III

C. 仅 II、III

D. I、 II 和 III

[2021年] 下列是关于多重中断系统中 CPU 响应中断的叙述，错误的是（　）。

A. 仅在用户态（执行用户程序）下，CPU 才能检测和响应中断

B. CPU 只有在检测到中断请求信号后，才会进入中断响应周期

C. 进入中断响应周期时，CPU 一定处于中断允许（开中断）状态

D. 若 CPU 检测到中断请求信号，则一定存在未被屏蔽的中断源请求信号

[2023年] 下列关于 I/O 控制方式的叙述中，错误的是（　）。

A. 查询方式下，通过 CPU 执行查询程序进行 I/O 操作

B. 中断方式下，通过 CPU 执行中断服务程序进行 I/O 操作

C. DMA 方式下，通过 CPU 执行 DMA 传送程序进行 I/O 操作

D. 对于 SSD、网络适配器等高速设备，采用 DMA 方式输入/输出