计算机组成原理

第八章 CPU 的结构和功能

大　　纲

五、中央处理器(CPU)

(一)CPU 的功能和基本结构

(二)指令执行过程

(三)数据通路的功能和基本结构

(四)控制器的功能和工作原理

(五)异常和中断机制

1.异常和中断的基本概念

2.异常和中断的分类

3.异常和中断的检测与响应

(六) 指令流水线

1.指令流水线的基本概念

2. 指令流水线的基本实现

3.结构冒险、数据冒险和控制冒险的处理

4.超标量和动态流水线的基本概念

(七) 多处理器基本概念

1.SISD、SIMD、MIMD、向量处理器的基本概念

2.硬件多线程的基本概念

3.多核处理器（multi-core）的基本概念

4.共享内存多处理器（SMP）的基本

Contents

8.1 CPU 的结构

一、 CPU 的功能

1. 控制器的功能

取指令 — 指令控制

分析指令

执行指令，发出各种操作命令 — 操作控制

控制程序输入及结果的输出 — 时间控制

总线管理 — 处理中断

处理异常情况和特殊请求 — 数据加工

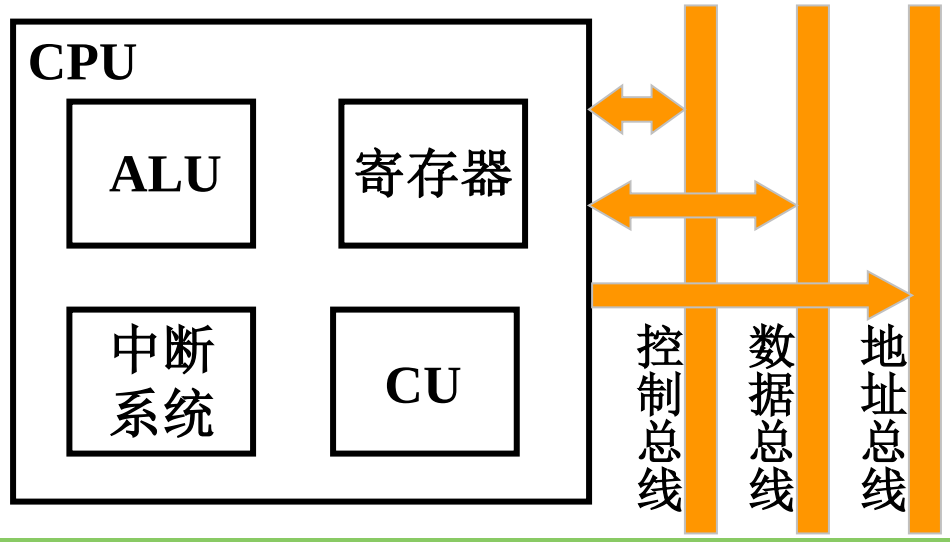
2. 运算器的功能

实现算术运算和逻辑运算

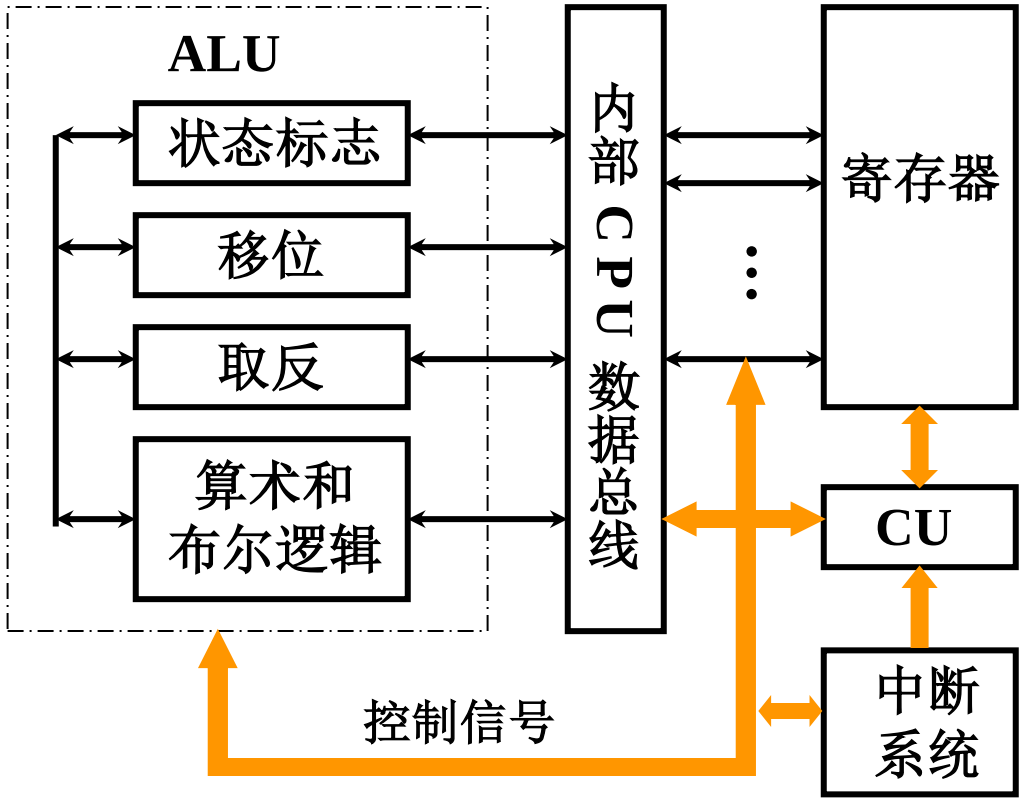
二、CPU 结构框图

1. CPU 与系统总线

指令控制		PC　IR
操作控制	}	CU　时序电路
时间控制	}	CU　时序电路
数据加工		ALU　寄存器
处理中断		中断系统



2. CPU 的内部结构



三、 CPU 的寄存器

8.1

1. 用户可见寄存器

(1) 通用寄存器　　存放操作数

可作 某种寻址方式所需的 专用寄存器

(2) 数据寄存器　　存放操作数（满足各种数据类型）

两个寄存器拼接存放双倍字长数据

(3) 地址寄存器　　存放地址，其位数应满足最大的地址范围

用于特殊的寻址方式　段基值　栈指针

(4) 条件码寄存器　存放条件码，可作程序分支的依据

如 正、负、零、溢出、进位等

2. 控制和状态寄存器

(1) 控制寄存器

PC → MAR → M → MDR → IR

控制 CPU 操作

其中 MAR、MDR、IR　　用户不可见

PC　　用户可见

(2) 状态寄存器

状态寄存器　　存放条件码

PSW 寄存器　　存放程序状态字

3. 举例　　Z8000　　8086　　MC 68000

四、 控制单元 CU 和中断系统 8.1

1. CU 产生全部指令的微操作命令序列

组合逻辑设计 硬连线逻辑

微程序设计 存储逻辑

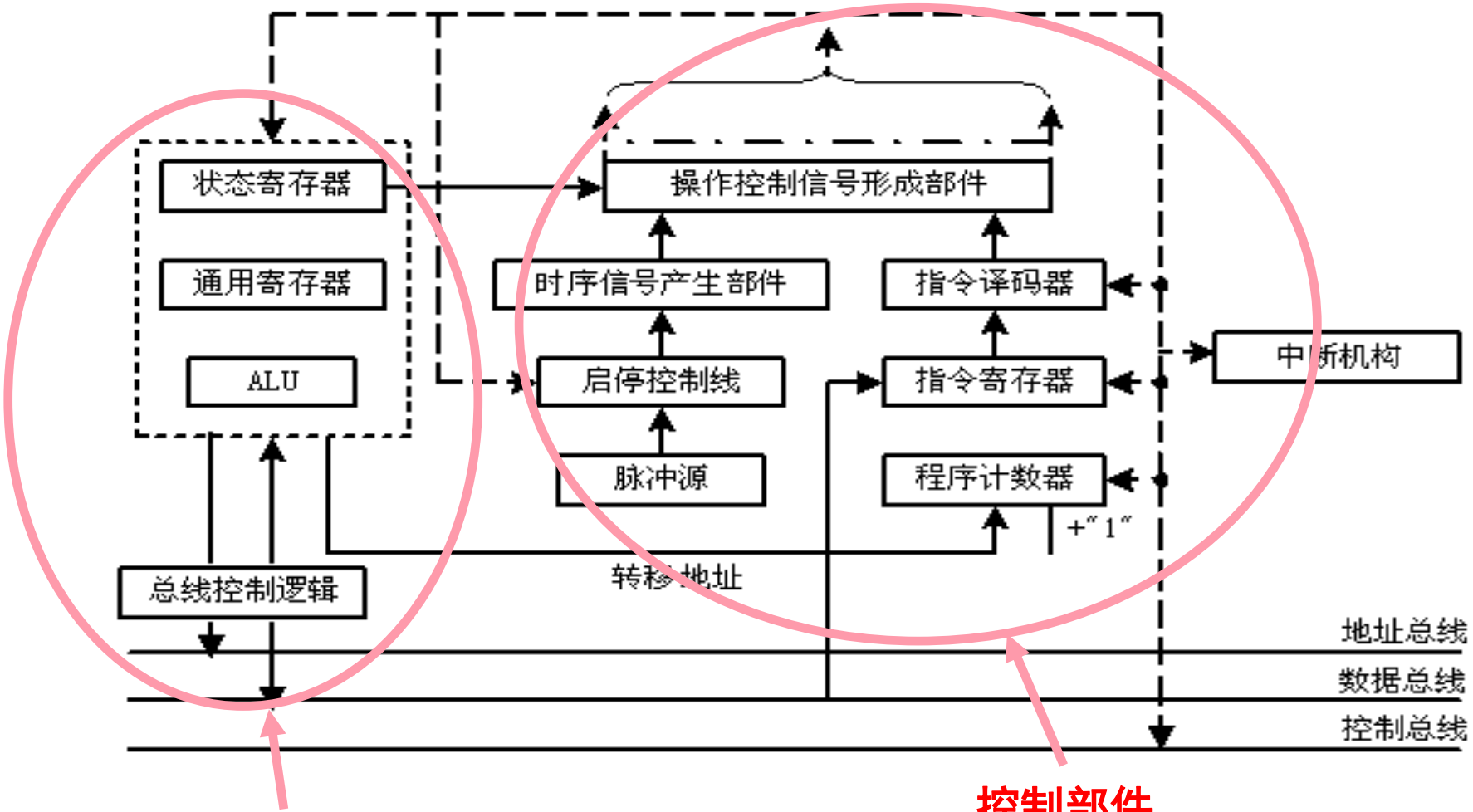
参见 第 4 篇

2. 中断系统 参见 8.4 节

五、ALU 参见 第 6 章

CPU基本组成原理图

指令寄存器----IR
程序计数器---PC



执行部件

控制部件

CPU 由 执行部件 和 控制部件组成

CPU 包含 数据通路 和 控制器

控制器 由 指令译码器 和 控制信号形成部件 组成

- PC、IR
- 指令译码器ID：对操作进行译码，向控制器提供操作的特定信号。
- 时序发生器：产生各种时序信号
  - 脉冲源：通常由石英晶体振荡器构成。产生一定频率的脉冲信号作为整个机器的时钟脉冲，是机器周期和工作脉冲的基准信号。在机器刚加电时，产生总清信号(reset)
  - 启停线路：保证可靠地送出或封锁完整的时钟脉冲，控制时序信号的发生或停止，从而启动机器工作或停机。

- 时序控制信号形成部件：当机器启动后，在CLK时钟作用下，根据当前正在执行的指令的需要，产生相应的时序控制信号，并根据被控功能部件的反馈信号调整时序控制信号。
- 状态寄存器（PSR）：存放PSW,PSW表明了系统基本状态，是控制程序执行的重要依据。

设置

【2010统考真题】下列寄存器中，汇编语言程序员可见的是(　)。

A 存储器地址寄存器(MAR)

B 程序计数器(PC)

C 存储器数据寄存器(MDR)

D 指令寄存器(IR)

提交

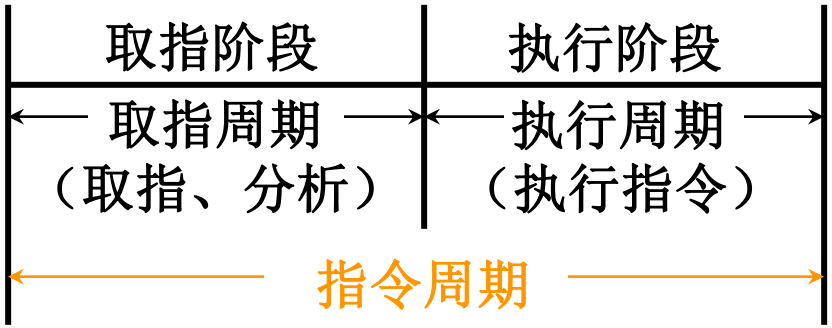
8.2 指 令 周 期

一、 指令周期的基本概念

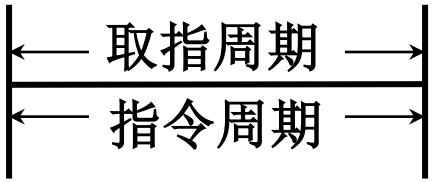
1．指令周期

取出并执行一条指令所需的全部时间。

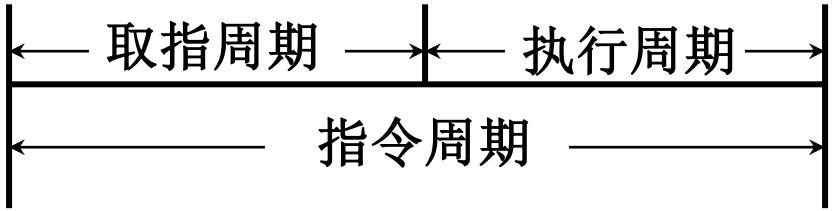
完成一条指令 { 取指、分析 —— 取指周期
执行 —— 执行周期



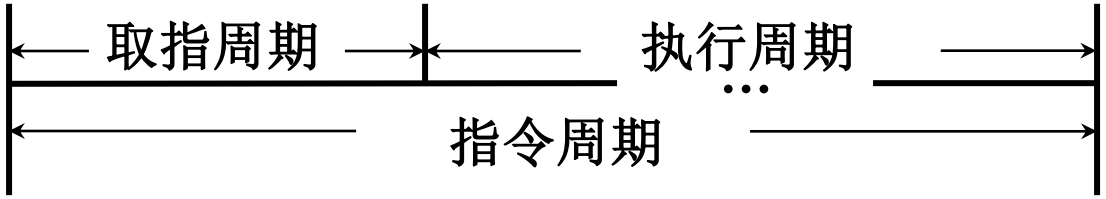
2. 每条指令的指令周期不同



NOP



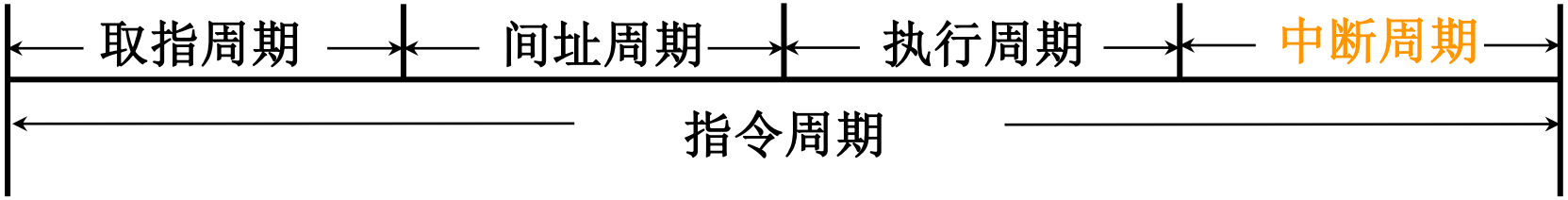
ADD mem



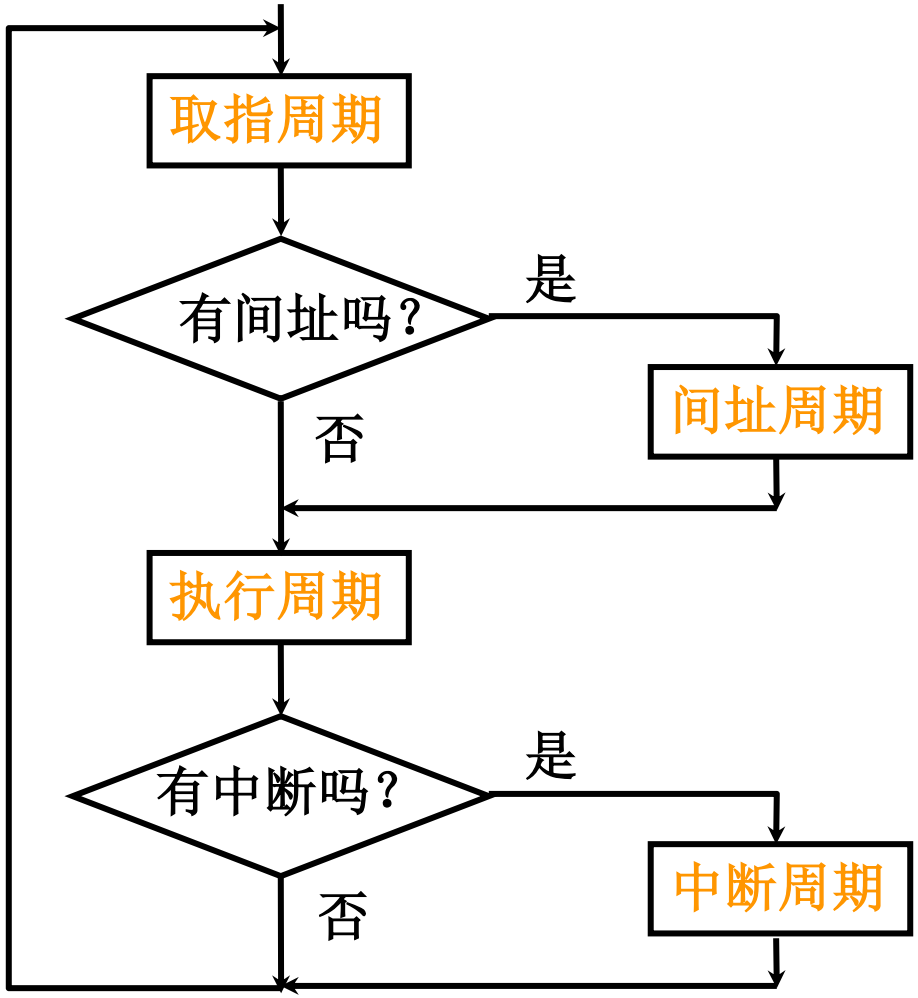
MUL mem

3. 具有间接寻址的指令周期

4. 带有中断周期的指令周期



5. 指令周期流程

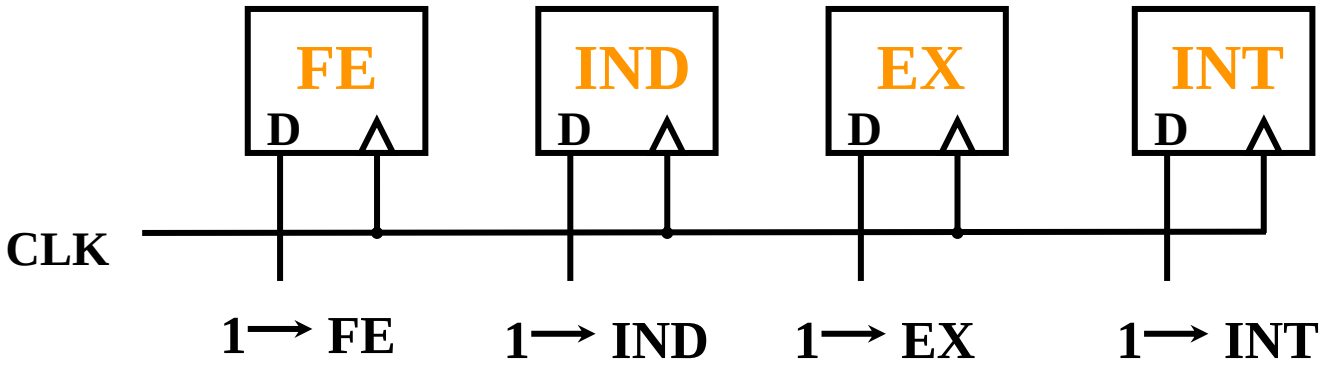


6. CPU 工作周期的标志

CPU 访存有四种性质

取 指令	取指周期
取 地址	间址周期
存/取 操作数或结果	执行周期
存 程序断点	中断周期

CPU 的
4个工作周期



设置

【2009统考真题】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

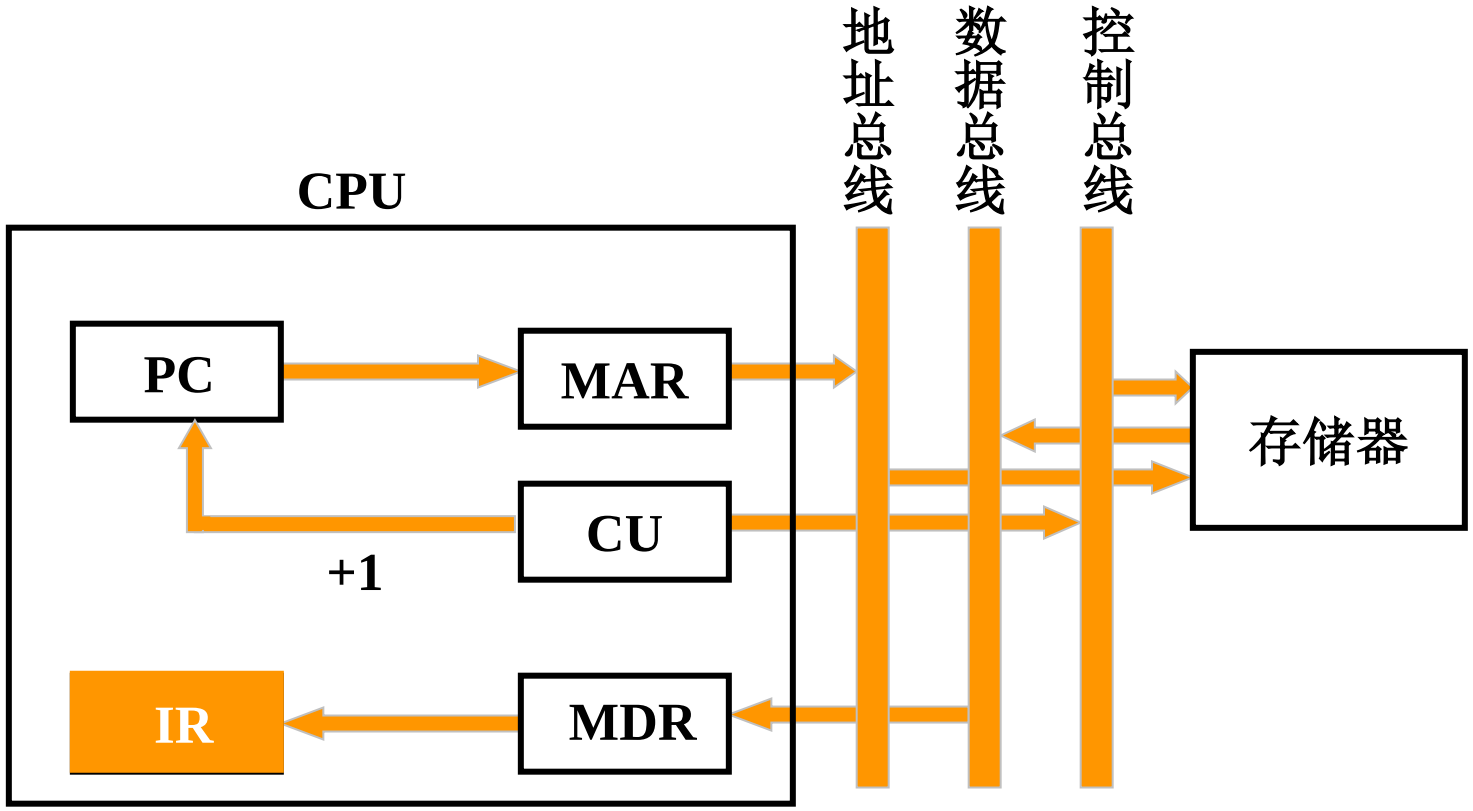
D 指令和数据所在的存储单元

提交

二、 指令周期的数据流

8.2

1. 取指周期数据流

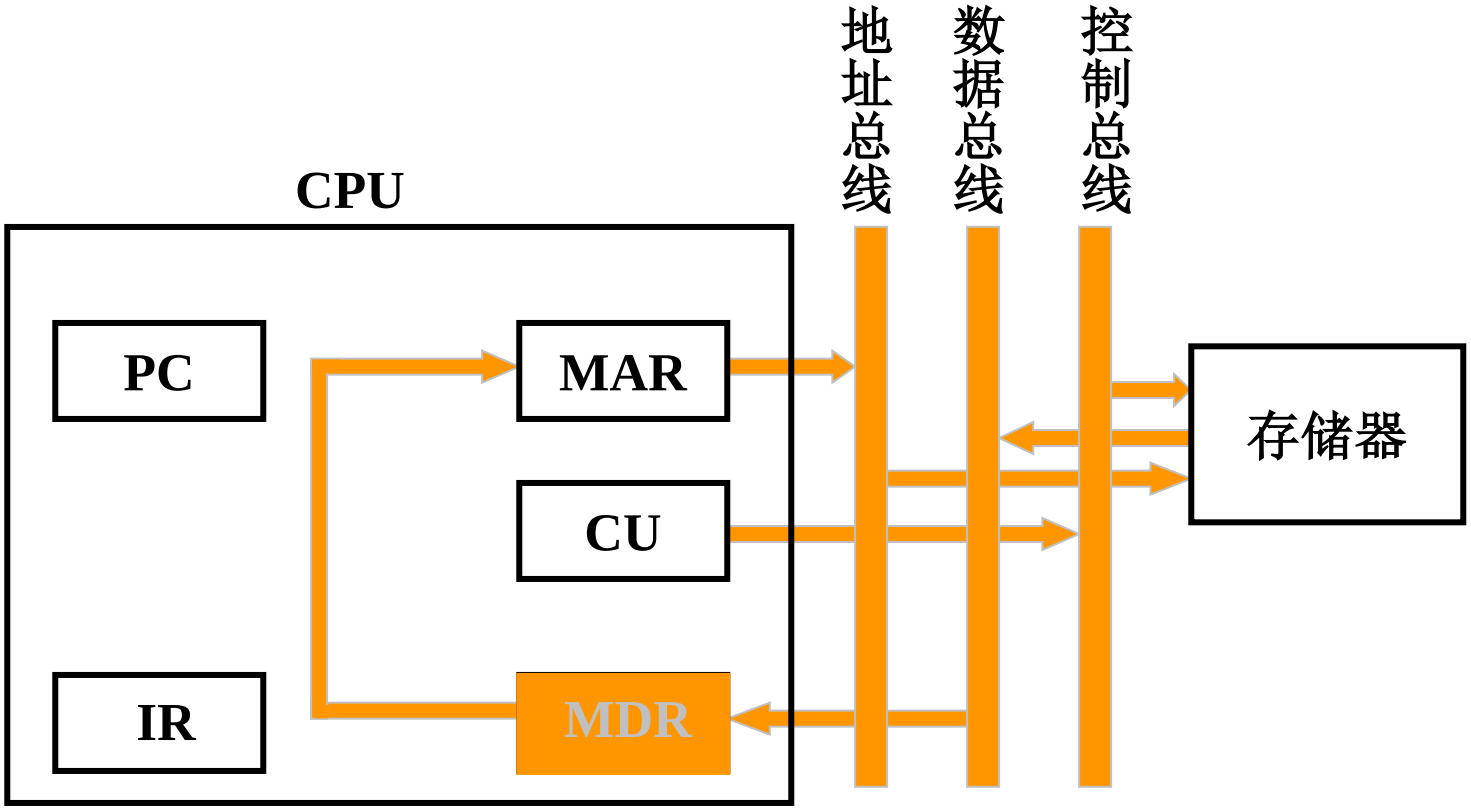


PC→MAR →M→MDR →IR

二、 指令周期的数据流

8.2

2. 间址周期数据流



Ad(IR)/ **Ad(MDR) →MAR→M→MDR**

二、 指令周期的数据流

3. 执行周期数据流

不同指令的执行周期数据流不同

中断周期完成的三个操作

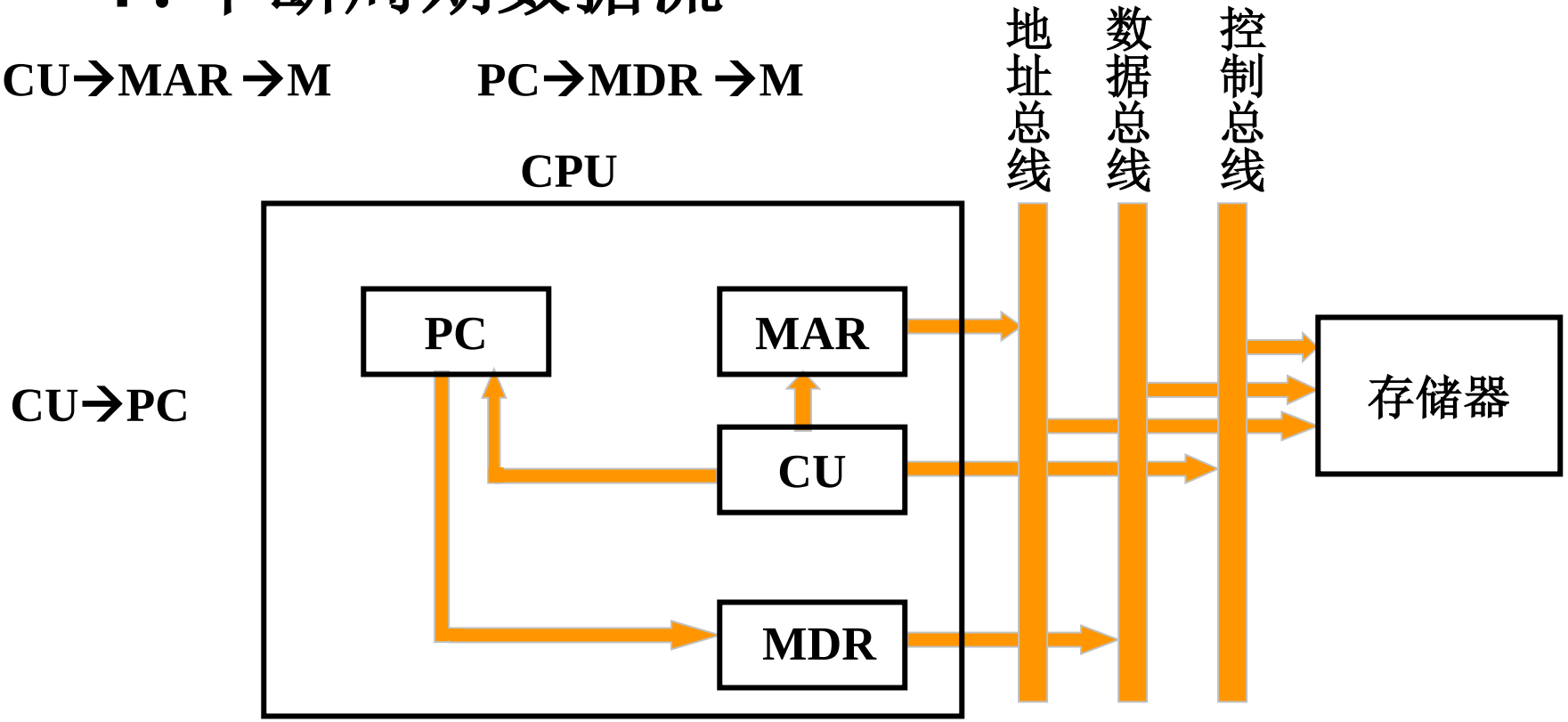
① 保存程序断点；

② 寻找到中断服务程序入口地址；

③ 关中断。

4 . 中断周期数据流

CU→MAR →M　　PC→MDR →M

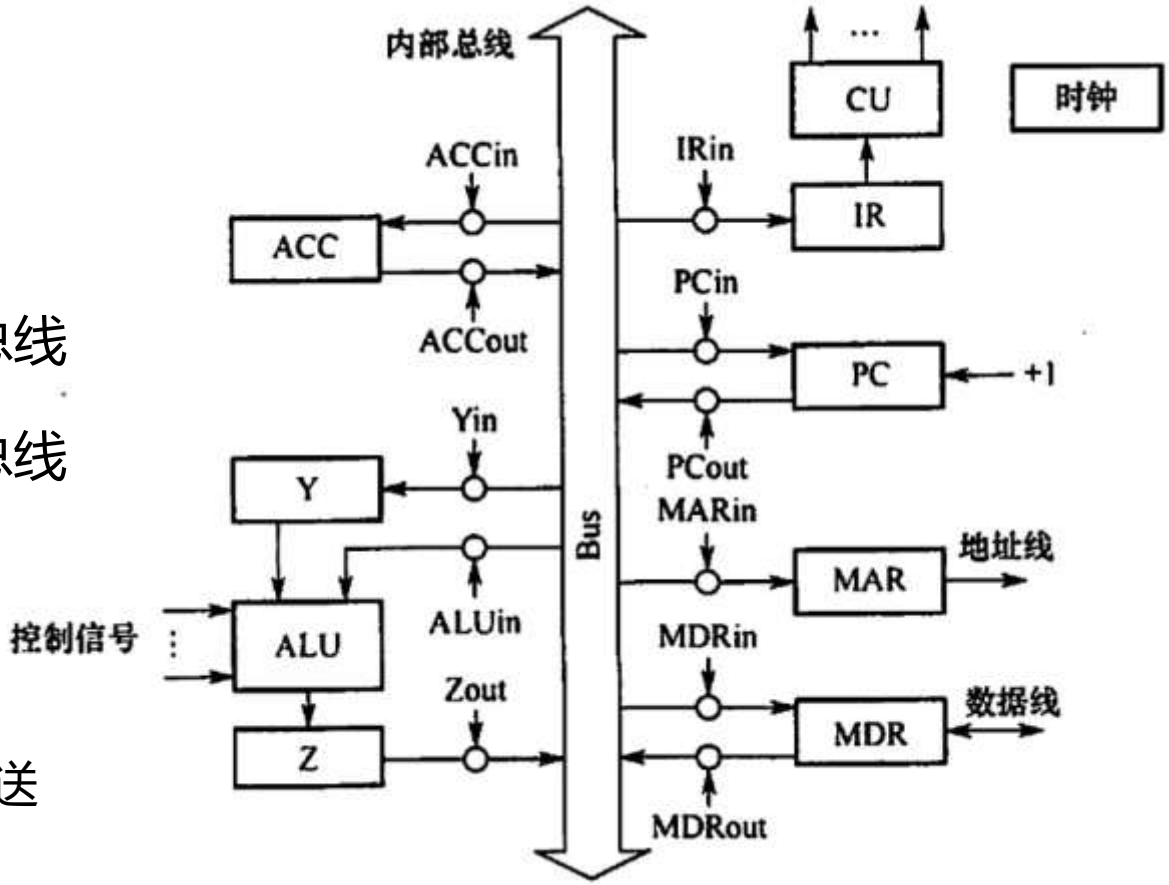


三、数据通路（datapath）

- **什么是数据通路（DataPath）？**
  - 数据在功能部件之间传送的路径称为数据通路包括数据通路上流经的部件，如ALU、通用寄存器、状态寄存器、异常和中断处理逻辑等。
- 数据通路描述了信息从什么地方开始，中间经过哪个寄存器或多路开关，最后传送到哪个寄存器，这些都需要加以控制。
- 由控制信号建立数据通路。

数据通路的基本结构

- CPU内部的单总线方式
- CPU内部的多总线方式
- 专用的数据通路方式
- 内部总线：同一部件内部总线
- 系统总线：各部件之间的总线
- 数据通路建立的方式：
  - 寄存器之间的数据传送
  - 主存和CPU之间的数据传送
  - 执行算术或者逻辑运算



- 数据通路结构直接影响CPU内各种信息的传送路径，数据通路不同，指令执行过程的微操作序列的安排也不同，它关系着微操作信号形成部件的设计。

【2021统考真题】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

提交

8.3 指令流水

一、如何提高机器速度

1. 提高访存速度

   高速芯片　　Cache　　多体并行

2. 提高 I/O 和主机之间的传送速度

   中断　　DMA　　通道　　I/O 处理机　　多总线

3. 提高运算器速度

   高速芯片　　改进算法　　快速进位链

- 提高整机处理能力

  高速器件　　改进系统结构，开发系统的并行性

二、系统的并行性

1. 并行的概念

并行 {
- 并发　两个或两个以上事件在 同一时间段 发生
- 同时　两个或两个以上事件在 同一时刻 发生

时间上互相重叠

2. 并行性的等级

过程级（程序、进程）	粗粒度	软件实现
指令级（指令之间） （指令内部）	细粒度	硬件实现

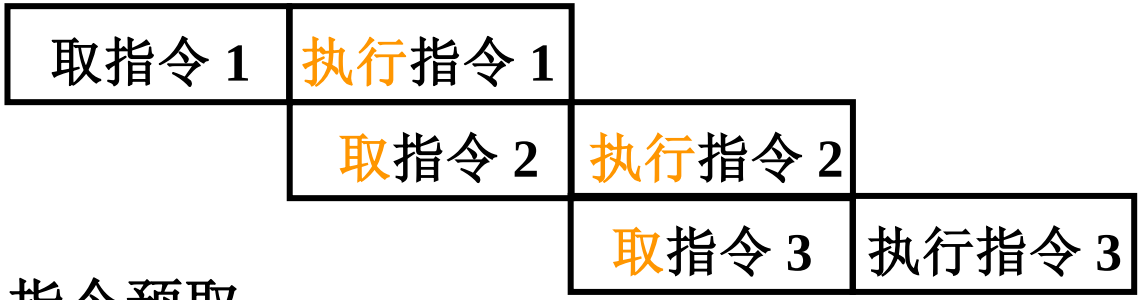
三、指令流水原理

1. 指令的串行执行

取指令　　取指令部件　　完成　　总有一个部件 空闲

执行指令　　执行指令部件　　完成

2. 指令的二级流水



指令预取

若 取指 和 执行 阶段时间上 完全重叠

指令周期 减半　速度提高 1 倍

3. 影响指令流水效率加倍的因素

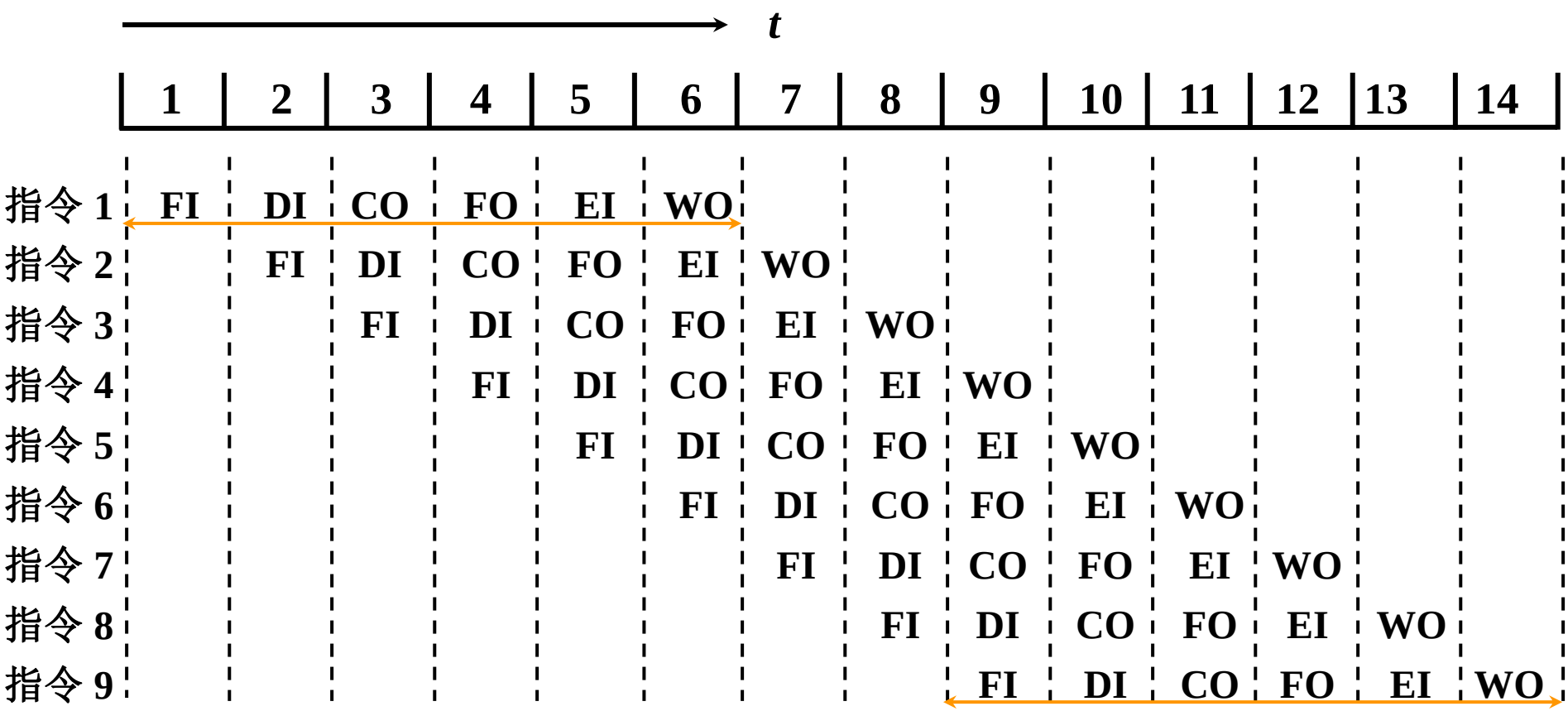
(1) 执行时间 > 取指时间

(2) 条件转移指令 对指令流水的影响

必须等 上条 指令执行结束，才能确定 下条 指令的地址，

造成时间损失　　猜测法

4. 指令的六级流水



完成一条指令 6 个时间单位

串行执行 6 × 9 = 54 个时间单位

六级流水 14 个时间单位

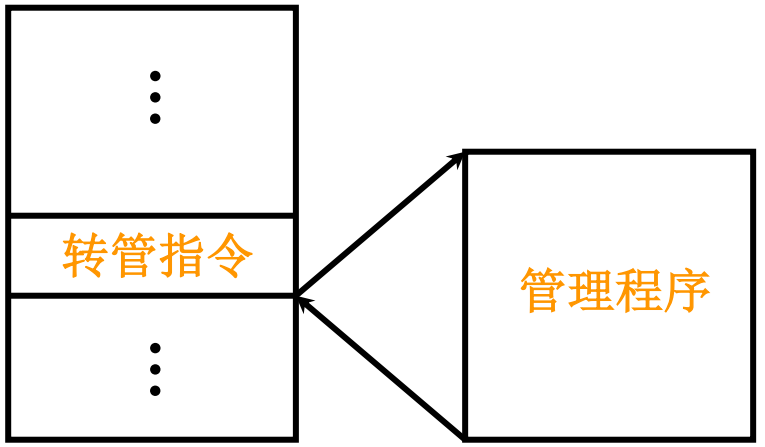
8.4 中断系统

一、概述

1. 引起中断的各种因素

(1) 人为设置的中断

如　转管指令



(2) 程序性事故　溢出、操作码不能识别、除法非法

(3)　硬件故障

(4)　I/O 设备

(5)　外部事件　　用 键盘中断 现行程序

中断的分类

1. 按中断来源分：

（1）内中断：发生在主机内部的中断称为内中断

（2）外中断：由主机外部事件引起的中断称为外中断

2. 按中断服务程序入口地址的获取方式分

（1）向量中断：外部设备在提出中断请求的同时，通过硬件自动形成中断服务程序入口地址。CPU响应中断后，将根据向量地址直接转入相应中断服务程序。这种具有产生向量地址的中断称为向量中断。

（2）非向量中断：非向量中断在产生中断的同时不能直接提供中断服务程序入口地址，而只产生一个中断查询程序的入口地址。需要通过中断查询程序确定中断源和中断服务程序的入口地址。

3. 按中断源位置分：
   - 硬件中断
   - 软件中断
4. 按是否可屏蔽分：CPU根据需要可以禁止响应某些中断源的中断请求。
   - 可屏蔽中断：CPU可以禁止响应的中断
   - 不可屏蔽中断：CPU必须响应的中断。

2. 中断系统需解决的问题

(1) 各中断源 如何 向 CPU 提出请求？

(2) 各中断源 同时 提出 请求 怎么办？

(3) CPU 什么 条件、什么 时间、以什么 方式 响应中断？

(4) 如何 保护现场？

(5) 如何 寻找入口地址？

(6) 如何 恢复现场，如何 返回？

(7) 处理中断的过程中又 出现新的中断 怎么办？

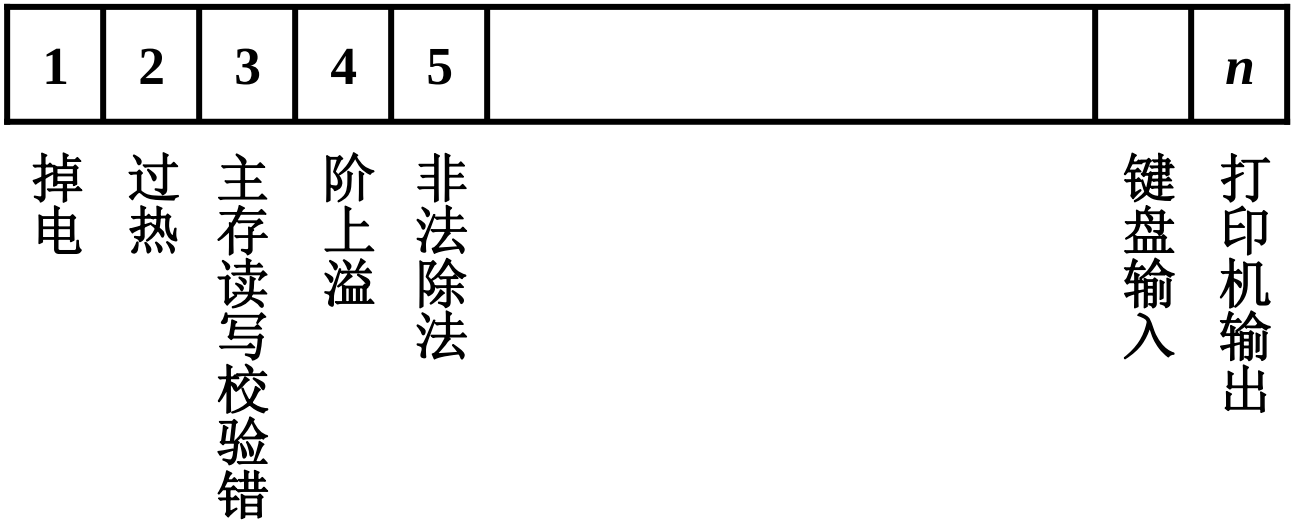
硬件 + 软件

二、中断请求标记和中断判优逻辑 8.4

1. 中断请求标记 INTR

一个请求源 一个 INTR 中断请求触发器

多个INTR 组成 中断请求标记寄存器



INTR 分散 在各个中断源的 接口电路中

INTR 集中 在 CPU 的中断系统 内

2. 中断判优逻辑

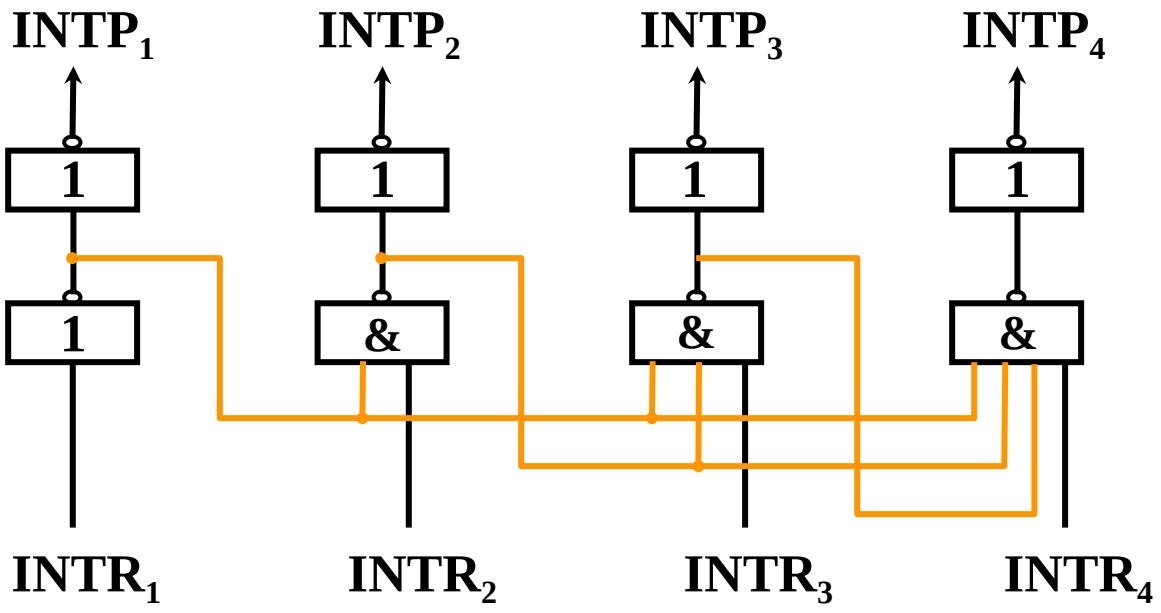
8.4

(1) 硬件实现（排队器）

① 分散 在各个中断源的 接口电路中　链式排队器

参见 第五章

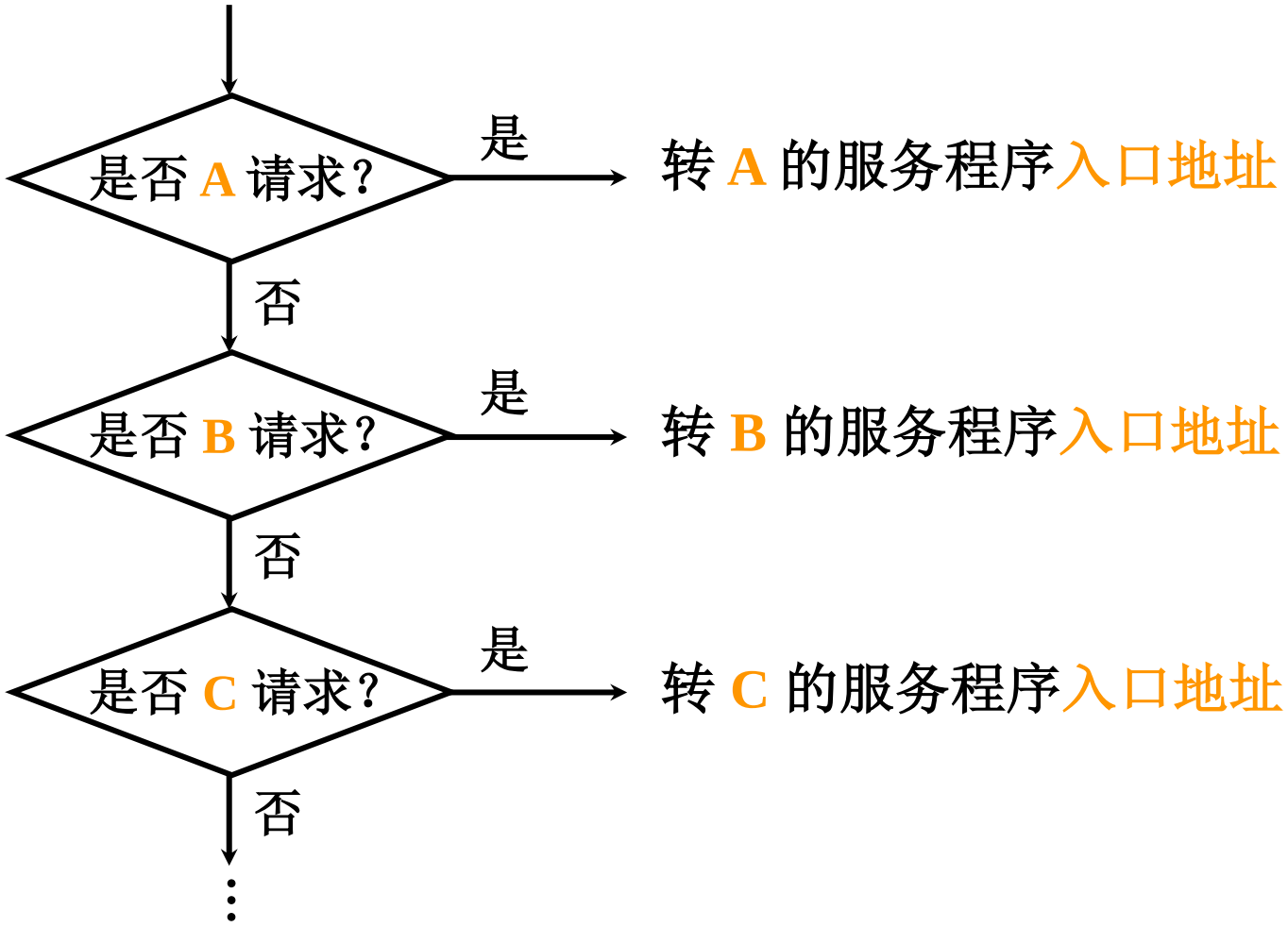
② 集中 在 CPU 内



$INTR_1$、$INTR_2$、$INTR_3$、$INTR_4$ 优先级 按 降序 排列

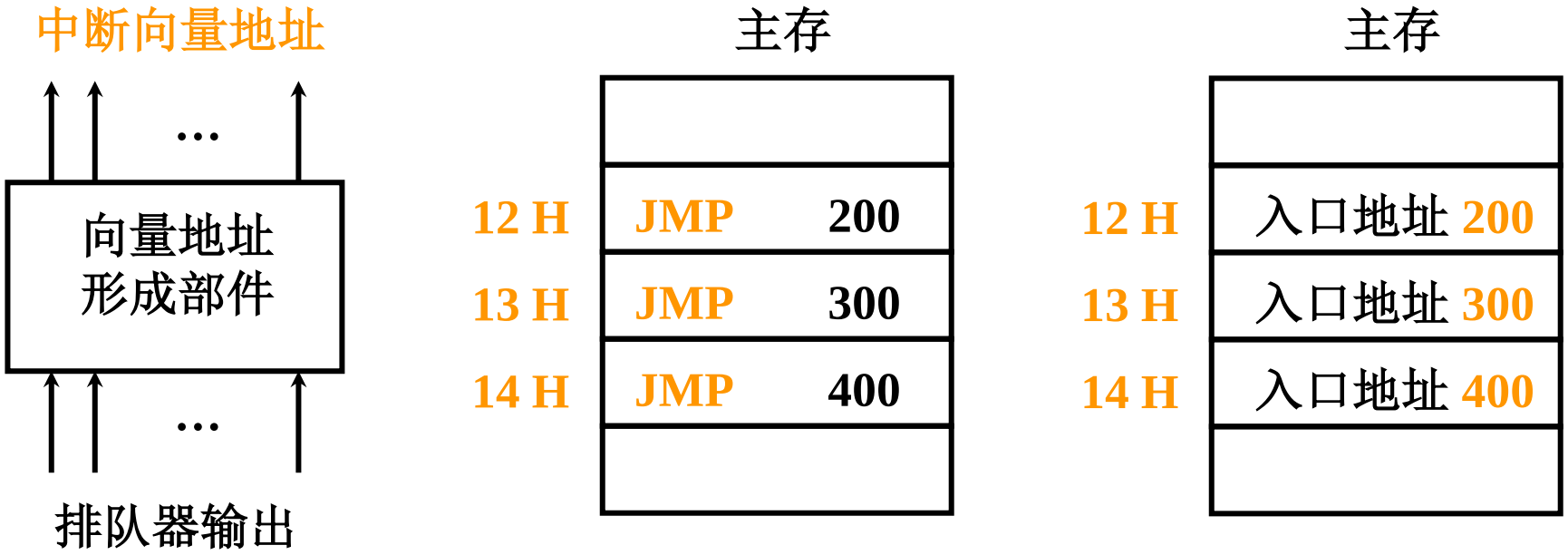
(2) 软件实现（程序查询）

A、B、C 优先级按 降序 排列



三、中断服务程序入口地址的寻找 8.4

1. 硬件向量法

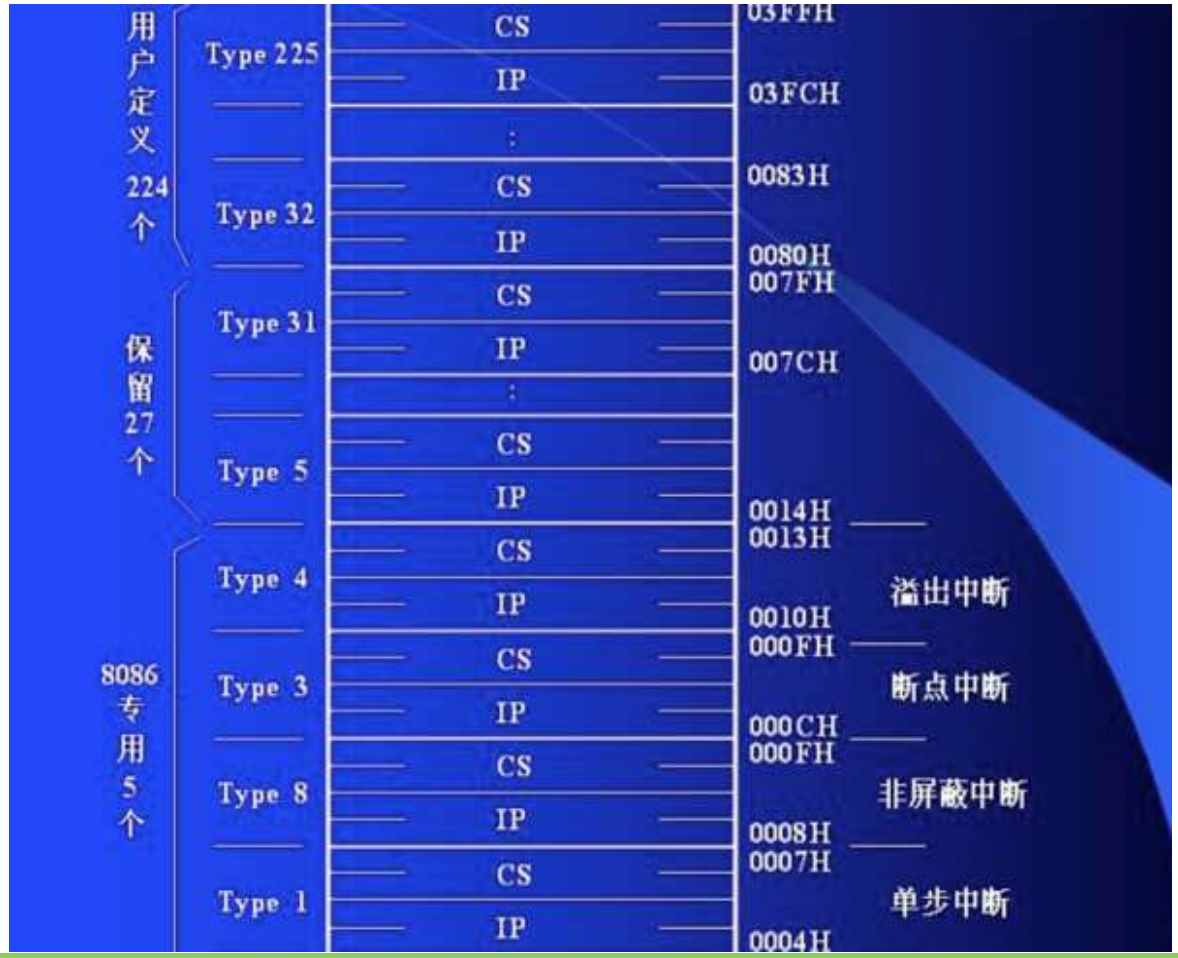


向量地址　12H、13H、14H

入口地址（中断向量）　200、 300、 400

8086的中断向量表

- 专用中断：0～4，占中断向量表0000～0013H。
- 系统备用中断：5～31，占中断向量表0014～007FH
- 用户使用中断： 32～255，占中断向量表0080～03FFH



40

2. 软件查询法

八个中断源 1，2，…　8 按 降序 排列

中断识别程序（入口地址 M）

地 址	指 令	说 明
M	SKP　DZ $1^{\#}$	$1^{\#}$　D = 0 跳（D为完成触发器）
	JMP　$1^{\#}$ SR	$1^{\#}$　D = 1 转$1^{\#}$ 服务程序
	SKP　DZ $2^{\#}$	$2^{\#}$　D = 0 跳
	JMP　$2^{\#}$ SR	$2^{\#}$　D = 1 转$2^{\#}$ 服务程序
	⋮	
	SKP　DZ $8^{\#}$	$8^{\#}$　D = 0 跳
	JMP　$8^{\#}$ SR	$8^{\#}$　D = 1 转$8^{\#}$ 服务程序

四、中断响应

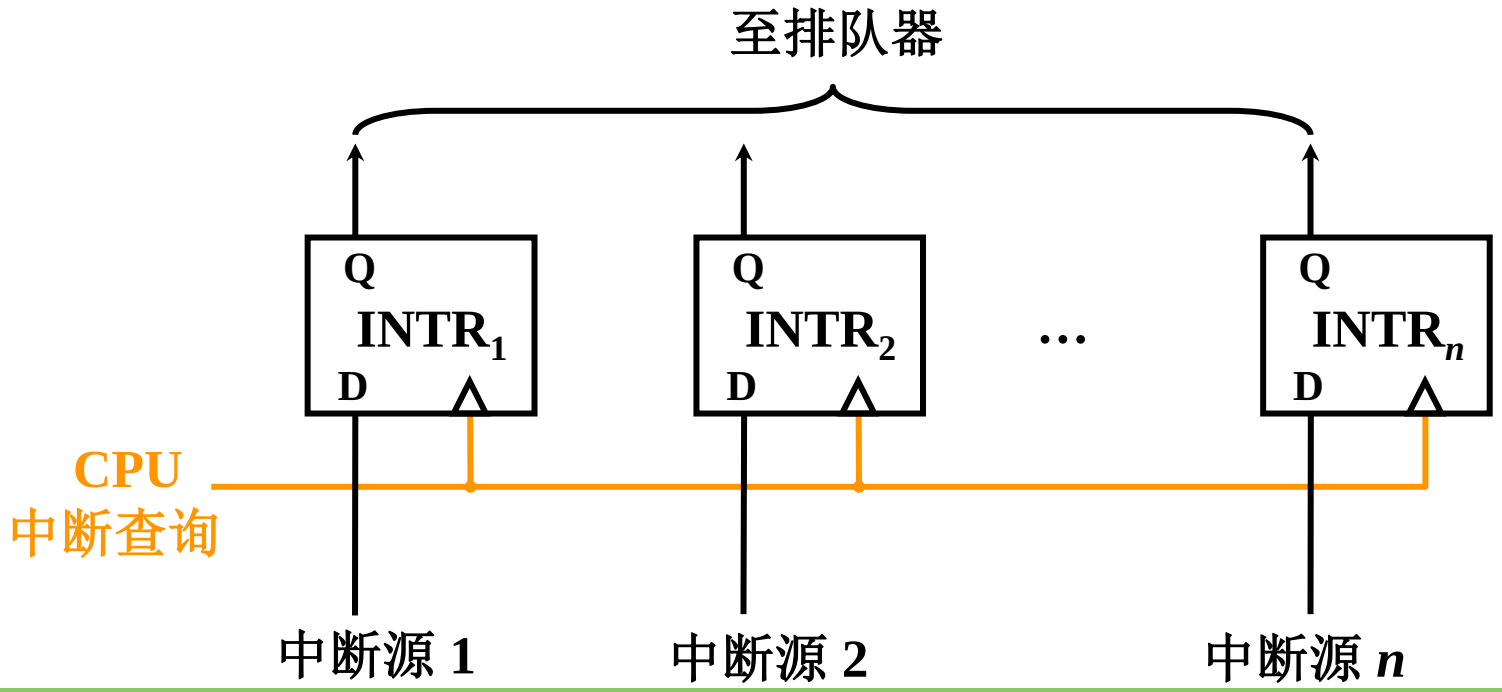
8.4

1. 响应中断的 条件

允许中断触发器　EINT = 1（可被开中断指令置1也可被关中断指令置0）

2. 响应中断的 时间

指令执行周期结束时刻由CPU 发查询信号



3. 中断隐指令（中断周期内）

8.4

(1) 保护程序断点

断点存于 特定地址（ 0 号地址）内　　断点 进栈

(2) 寻找中断服务程序入口地址

向量地址 ⟶ PC　（硬件向量法）

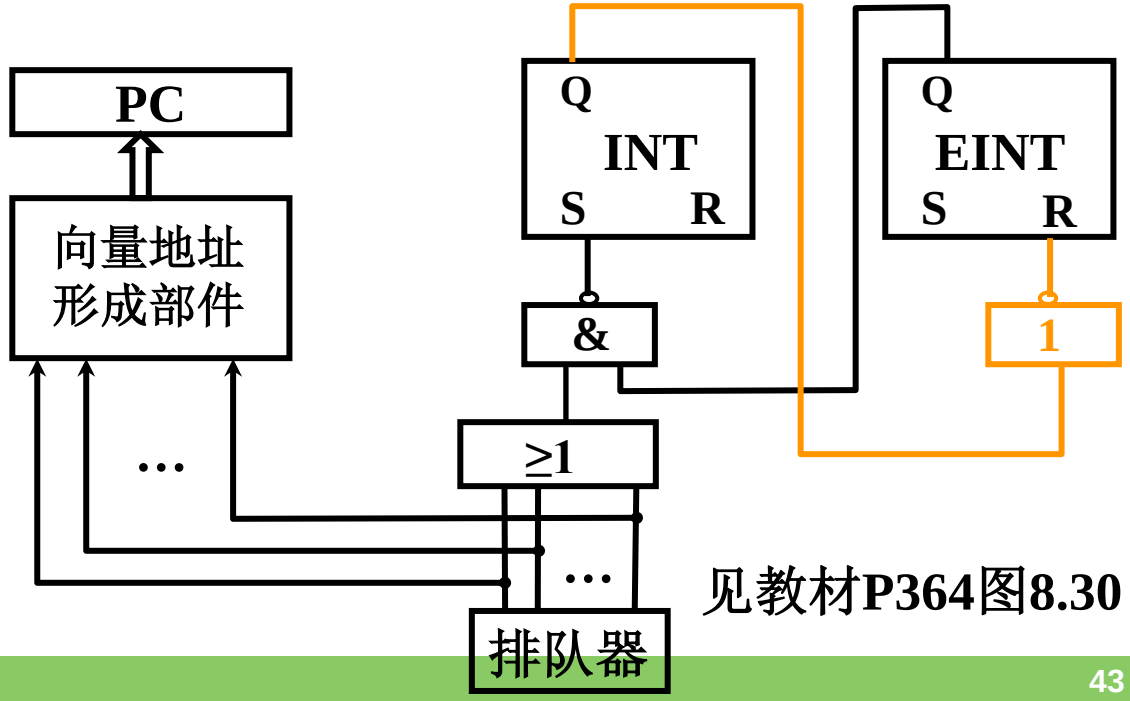
中断识别程序 入口地址　M ⟶ PC　（软件查询法）

(3) 硬件 关中断

INT 中断标记

EINT 允许中断

R – S 触发器



见教材P364图8.30

五、保护现场和恢复现场 8.4

1. 保护现场
   - 断点 —— 中断隐指令 完成
   - 寄存器 内容 —— 中断服务程序 完成
2. 恢复现场 —— 中断服务程序 完成

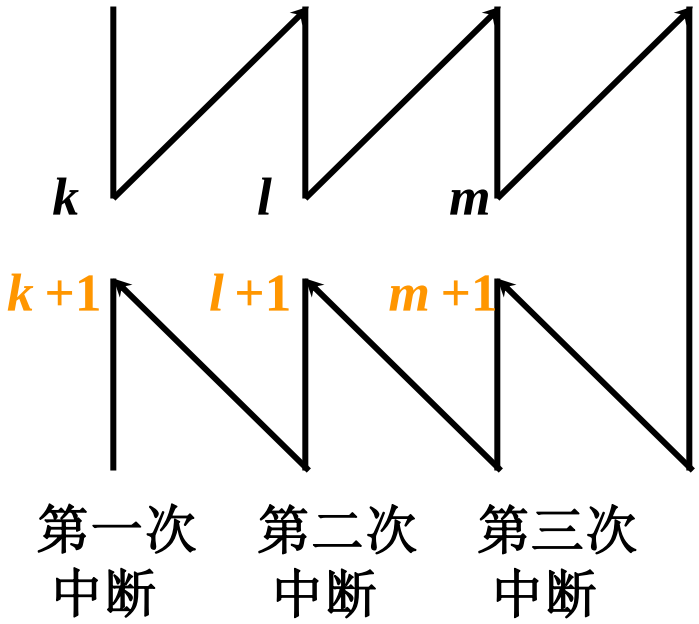
中断服务程序

保护现场	PUSH
其它服务程序	视不同请求源而定
恢复现场	POP
中断返回	IRET

六、中断屏蔽技术

8.4

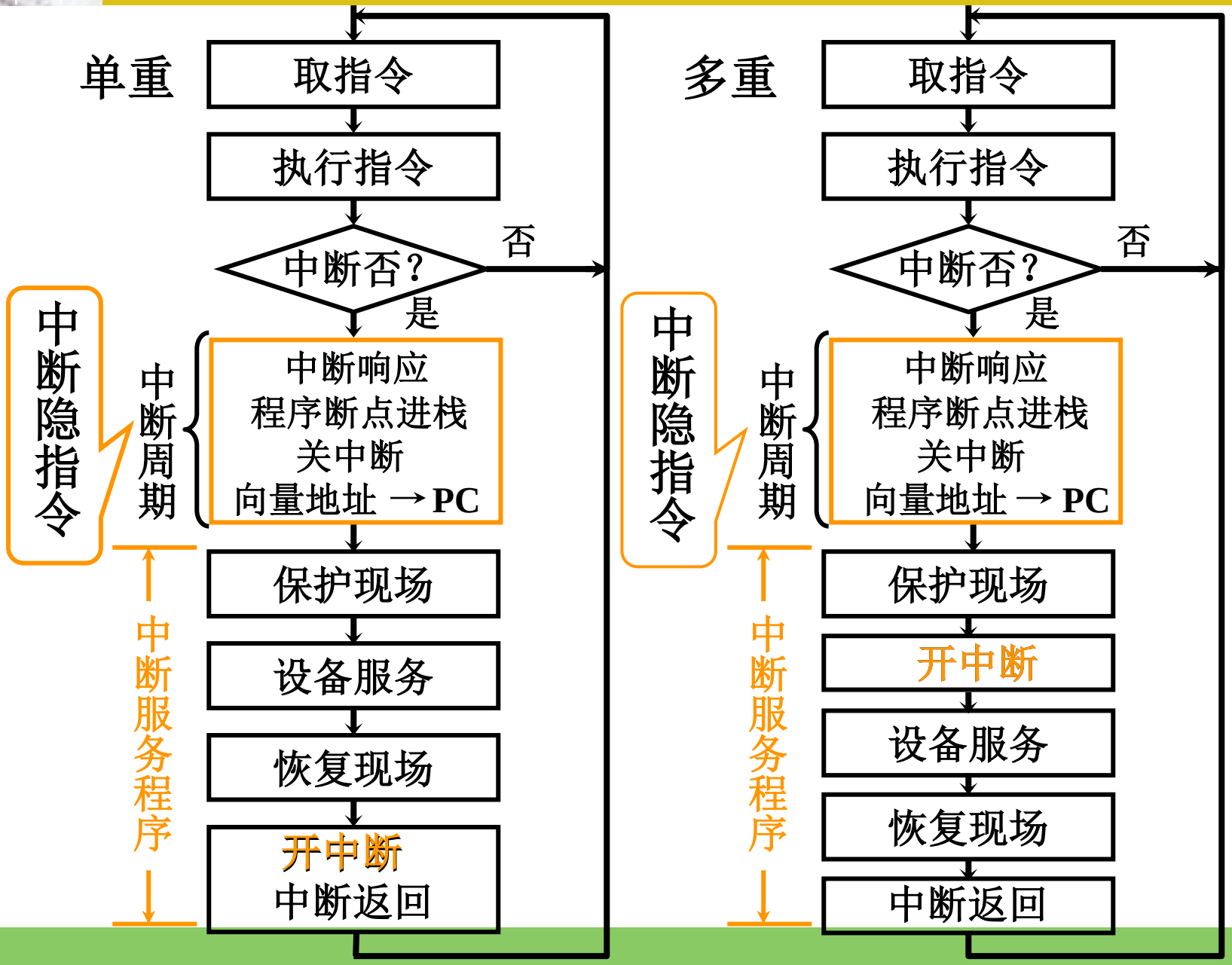
1. 多重中断的概念



程序断点 $k+1$, $l+1$, $m+1$

单重中断和多重中断的服务程序流程

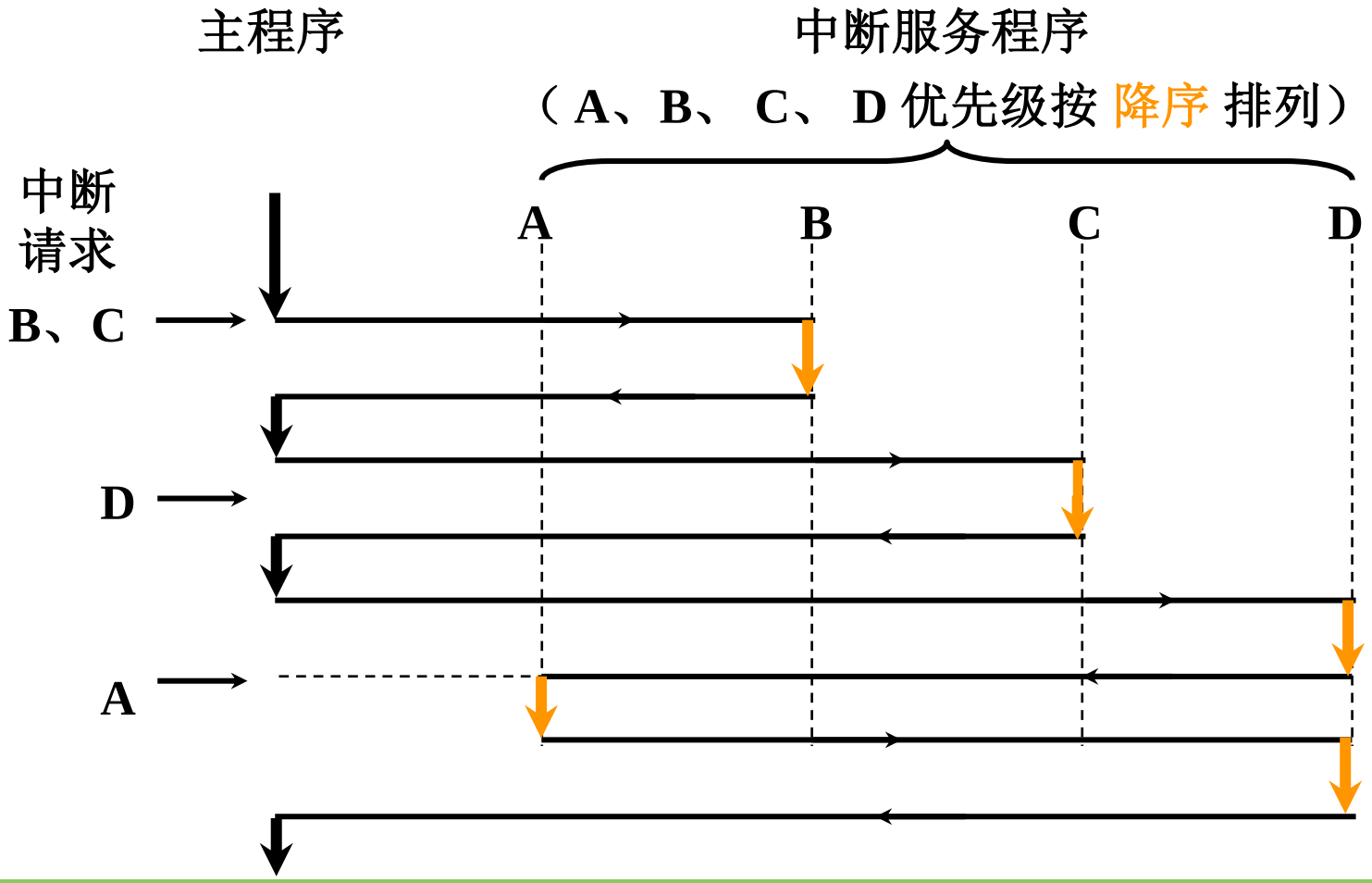
5.5



2. 实现多重中断的条件

(1) 提前 设置 开中断 指令

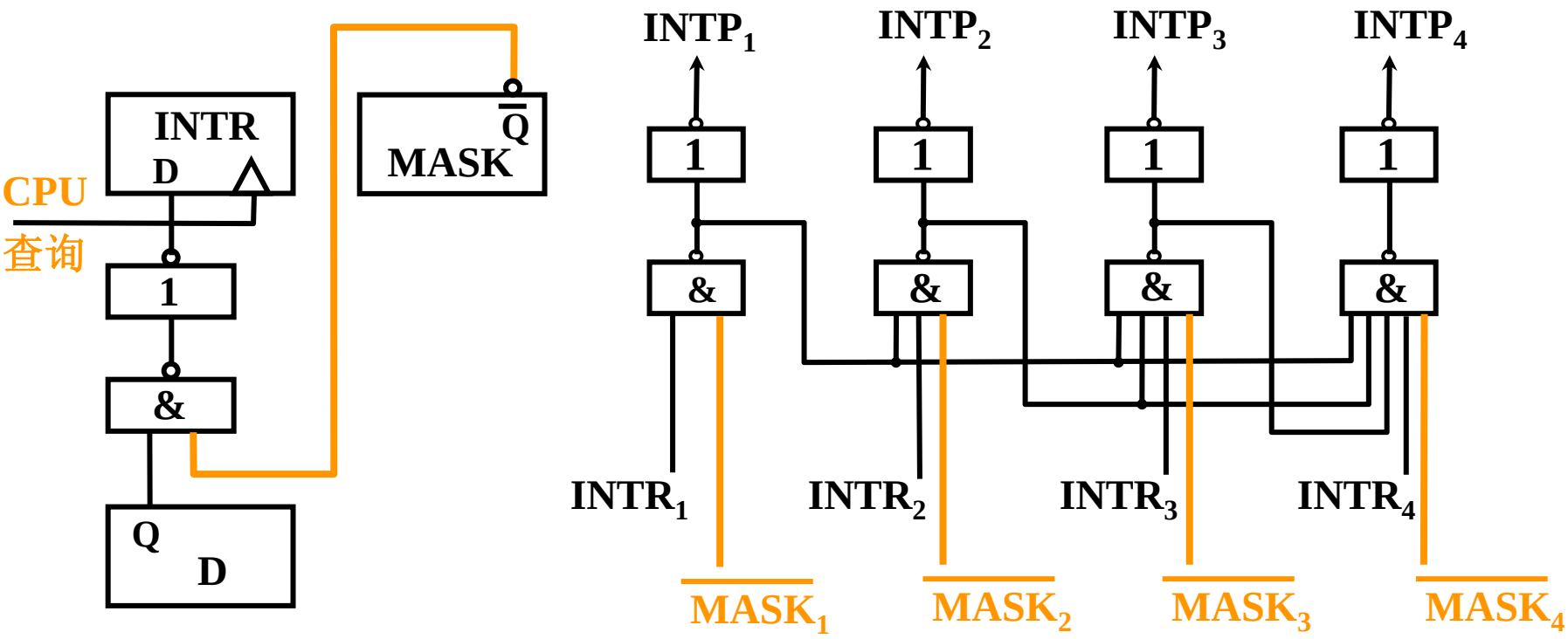
(2) 优先级别高 的中断源 有权中断优先级别低 的中断源



3. 屏蔽技术

8.4

(1) 屏蔽触发器的作用



MASK = 0（未屏蔽）　　**$MASK_i$ = 1**（屏蔽）

INTR 能被置 “1”　　**$INTP_i$ = 0**（不能被排队选中）

(2) 屏蔽字

16个中断源 1，2，3，… 16 按 降序 排列

1 屏蔽
0 未屏蔽

优先级	屏 蔽 字
1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2	0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
3	0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1
4	0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1
5	0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
6	0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1
⋮	⋮
15	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
16	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

(3) 屏蔽技术可改变处理优先等级

响应优先级　　不可改变

处理优先级　　可改变（通过重新设置屏蔽字）

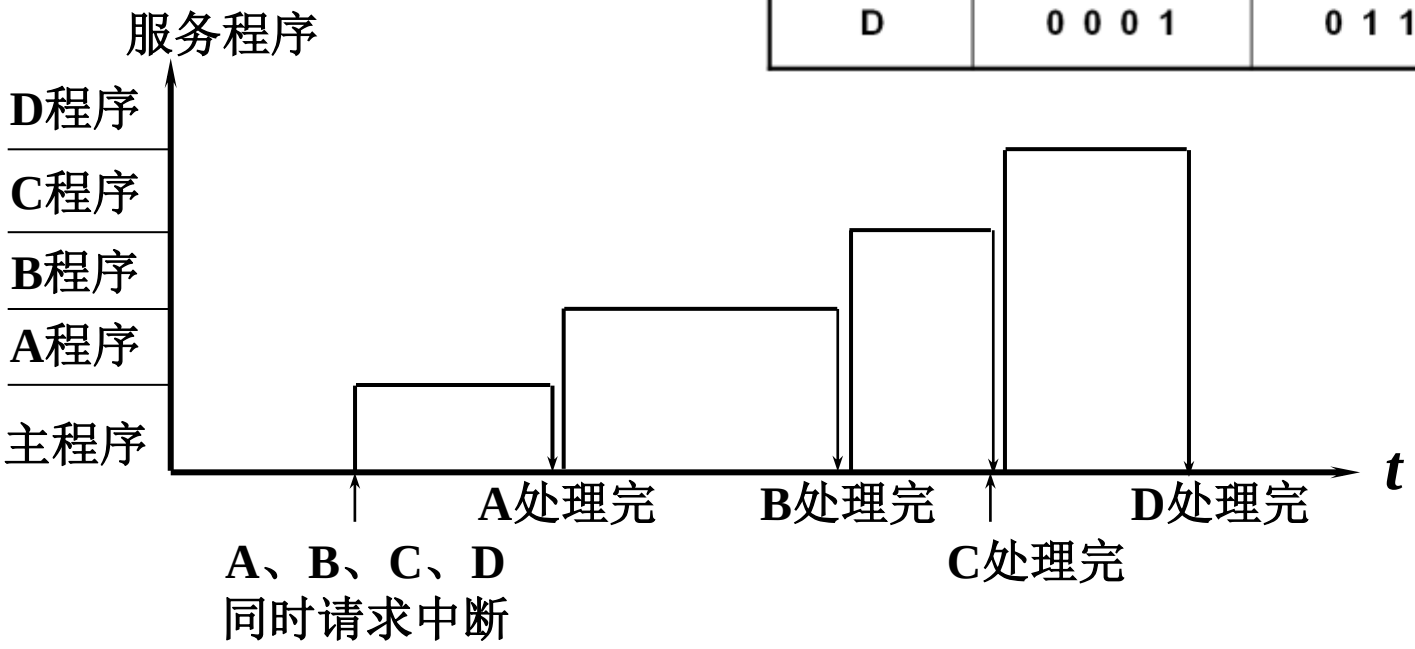
中断源	原屏蔽字	新屏蔽字
A	1 1 1 1	1 1 1 1
B	0 1 1 1	0 1 0 0
C	0 0 1 1	0 1 1 0
D	0 0 0 1	0 1 1 1

响应优先级　A→B→C→D 降序排列

处理优先级　A→D→C→B 降序排列

(3) 屏蔽技术可改变处理优先等级

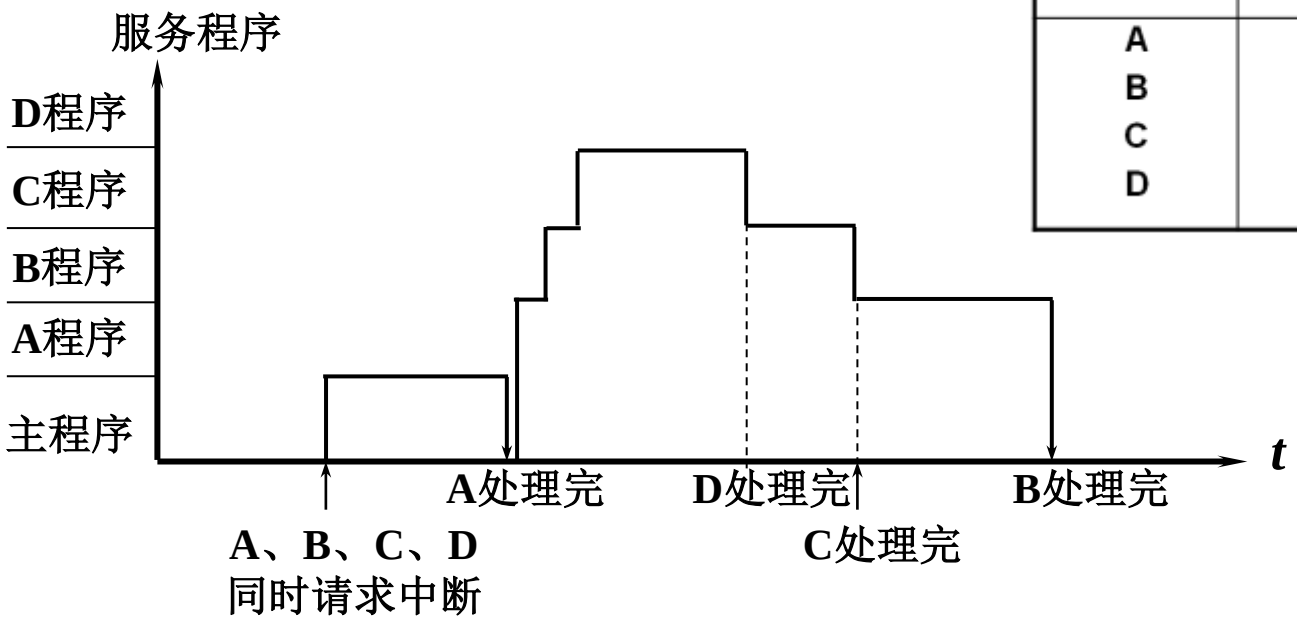
中断源	原屏蔽字	新屏蔽字
A	1 1 1 1	1 1 1 1
B	0 1 1 1	0 1 0 0
C	0 0 1 1	0 1 1 0
D	0 0 0 1	0 1 1 1



CPU 执行程序轨迹（原屏蔽字）

(3) 屏蔽技术可改变处理优先等级

中断源	原屏蔽字	新屏蔽字
A	1 1 1 1	1 1 1 1
B	0 1 1 1	0 1 0 0
C	0 0 1 1	0 1 1 0
D	0 0 0 1	0 1 1 1



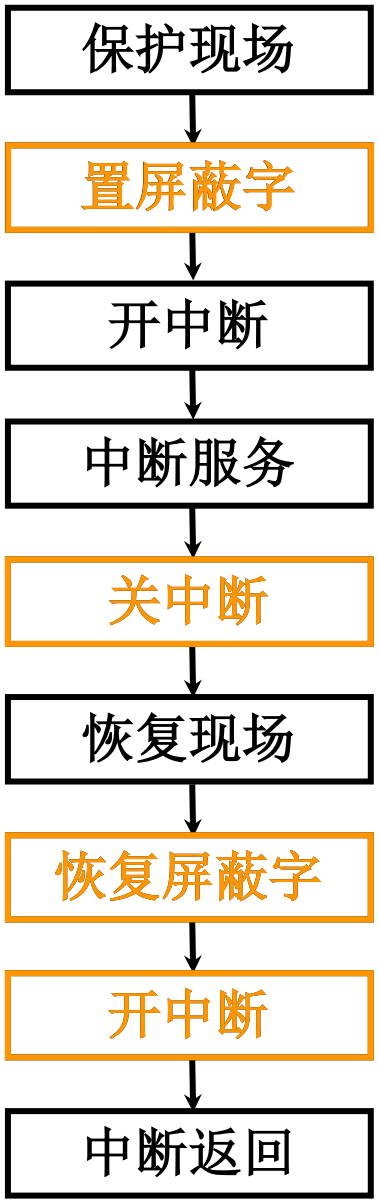
CPU 执行程序轨迹（新屏蔽字）

(4) 屏蔽技术的其他作用

可以 人为地屏蔽 某个中断源的请求

便于程序控制

(5) 新屏蔽字的设置



4. 多重中断的断点保护

(1) 断点进栈　　中断隐指令 完成

(2) 断点存入“ 0 ” 地址　　中断隐指令 完成

中断周期　　0 ⟶ MAR

命令存储器写　1⟶ W

PC ⟶ MDR　　断点 ⟶ MDR

(MDR) ⟶ 存入存储器

三次中断，三个断点都存入 “ 0 ” 地址

？ 如何保证断点不丢失？

课后习题

1. 某机有五个中断源，按中断响应的优先顺序由高到低为**L0,L1,L2,L3,L4**，现要求优先顺序改为**L4,L2,L3,L0,L1**，写出各中断源的屏蔽字。

中断源	屏蔽字 0 1 2 3 4
L0	
L1	
L2	
L3	
L4	

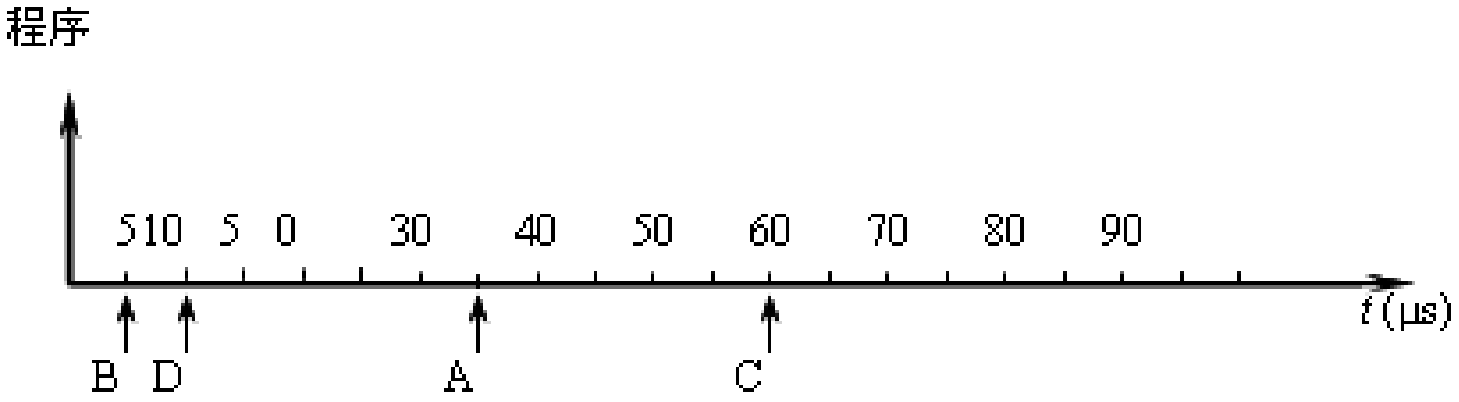
中断源	屏蔽字 0 1 2 3 4
L0	1 1 0 0 0
L1	0 1 0 0 0
L2	1 1 1 1 0
L3	1 1 0 1 0
L4	1 1 1 1 1

课后习题

2. 设某机有四个中断源A、B、C、D，其硬件排队优先次序为A > B > C > D，现要求将中断处理次序改为D > A > C > B。

（1）写出每个中断源对应的屏蔽字。

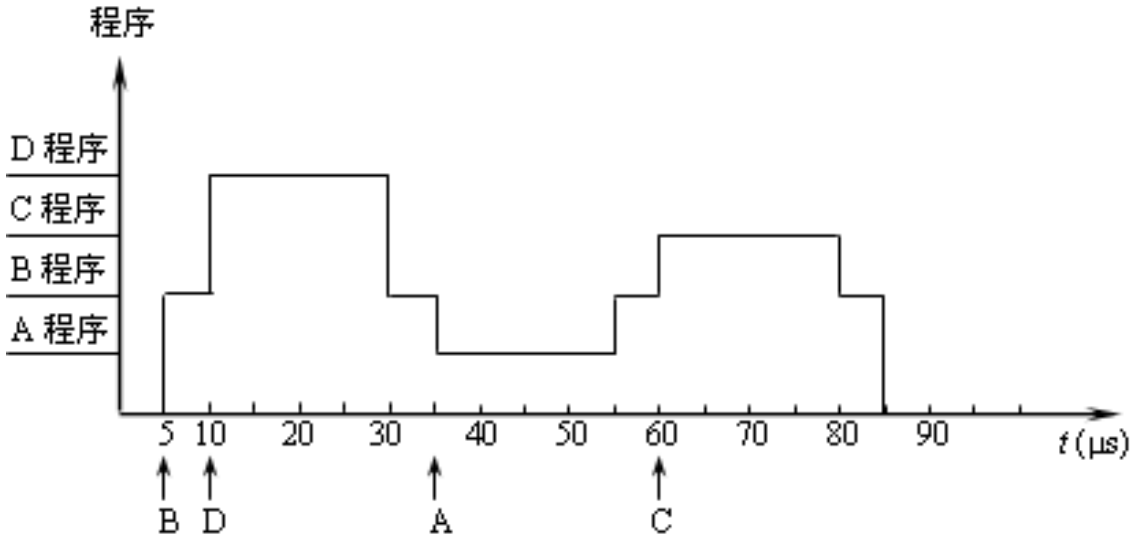
（2）按下图时间轴给出的四个中断源的请求时刻，画出CPU执行程序的轨迹。设每个中断源的中断服务程序时间均为20μs。



答：（1）在中断处理次序改为D > A > C > B后，每个中断源新的屏蔽字如表所示。

中断源	屏蔽字			
	A	B	C	D
A	1	1	1	0
B	0	1	0	0
C	0	1	1	0
D	1	1	1	1

（2）根据新的处理次序，CPU执行程序的轨迹如图所示



Thank You !