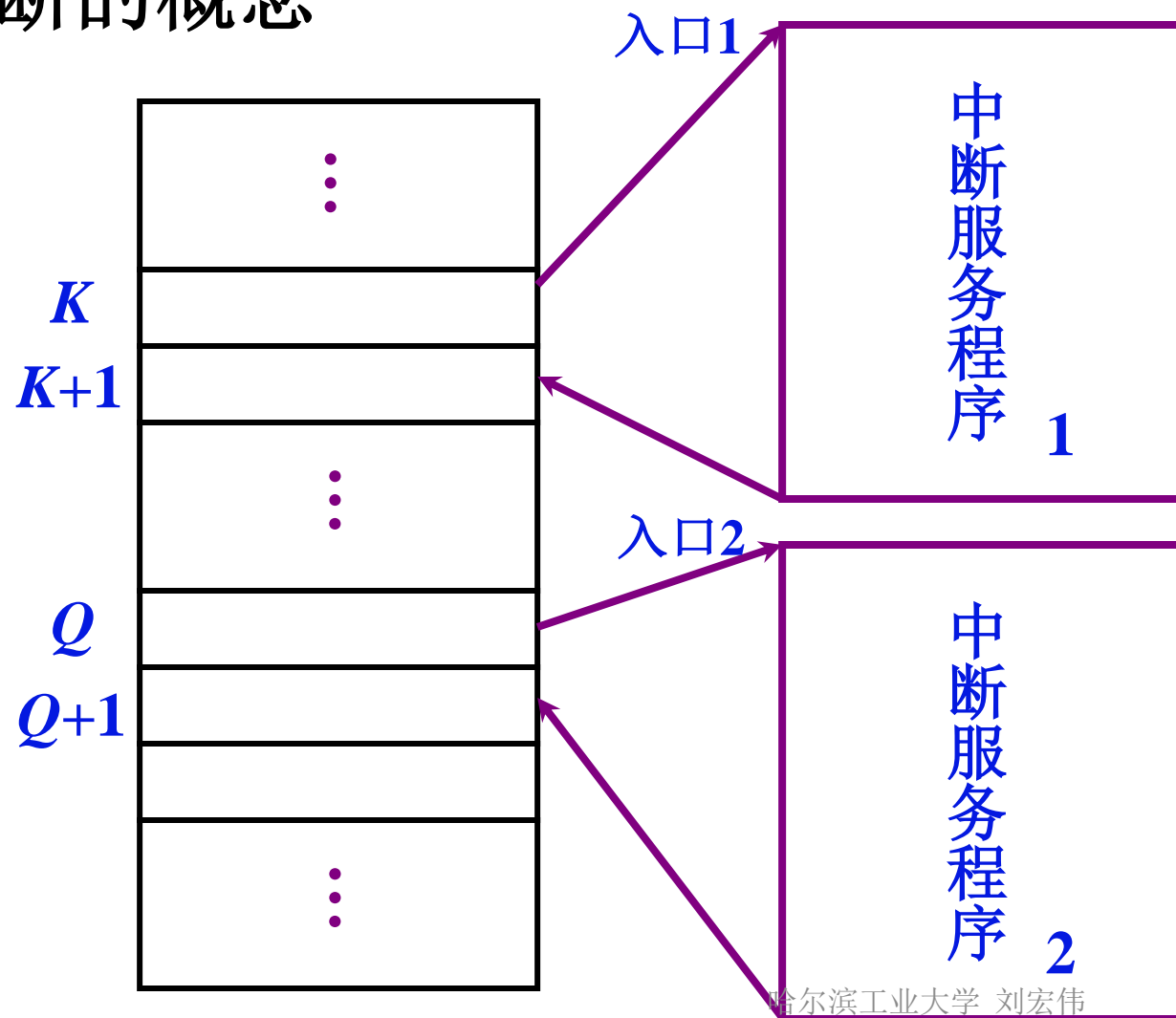


5.5 程序中中断方式

- 一、中断的概念
- 二、I/O中断的产生
- 三、程序中中断方式的接口电路
- 四、I/O 中断处理过程
- 五、中断服务程序流程

5.5 程序中中断方式

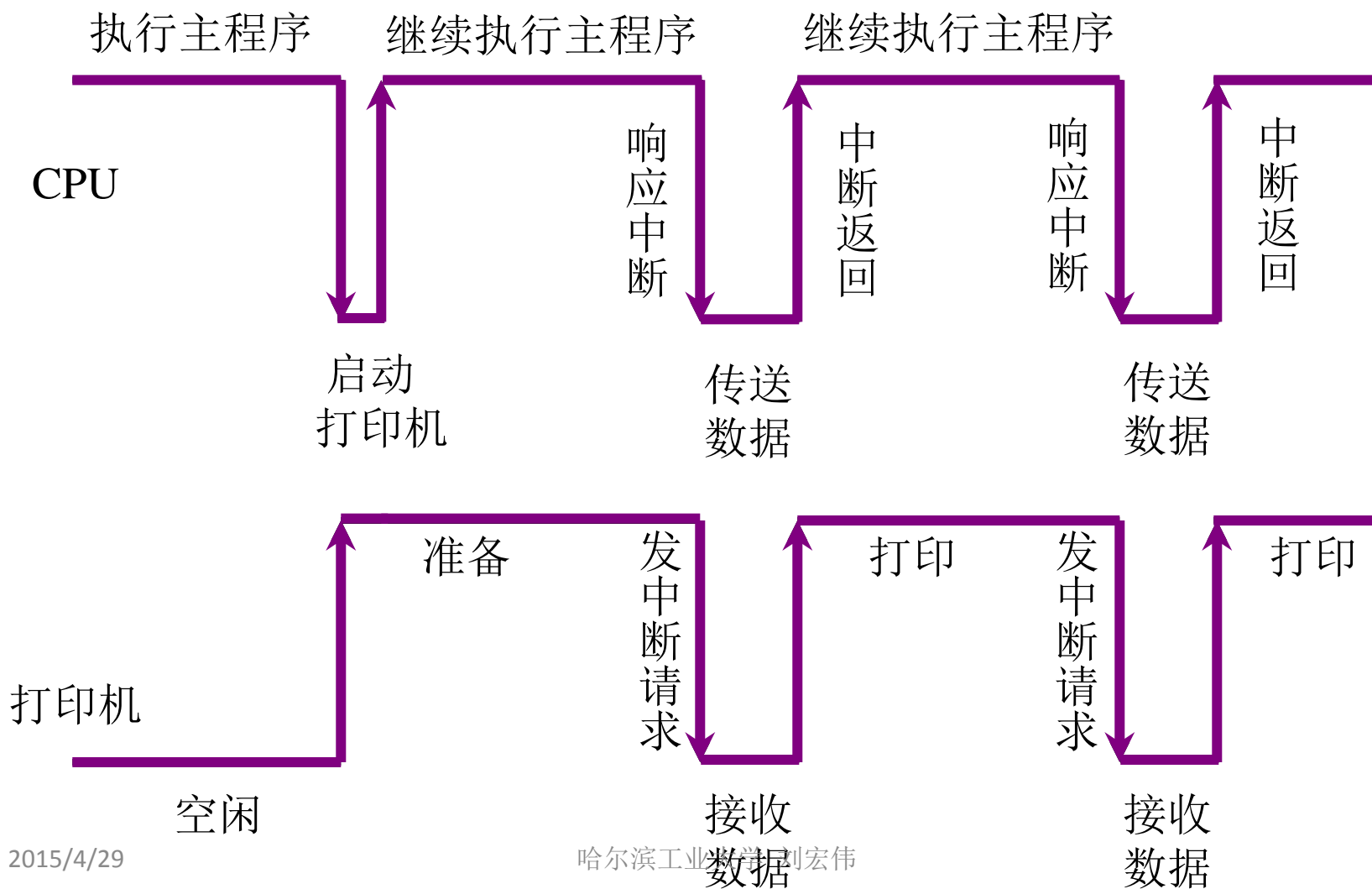
一、中断的概念



二、I/O 中断的产生

5.5

以打印机为例 CPU 与打印机部分并行工作



三、程序中断方式的接口电路

5.5

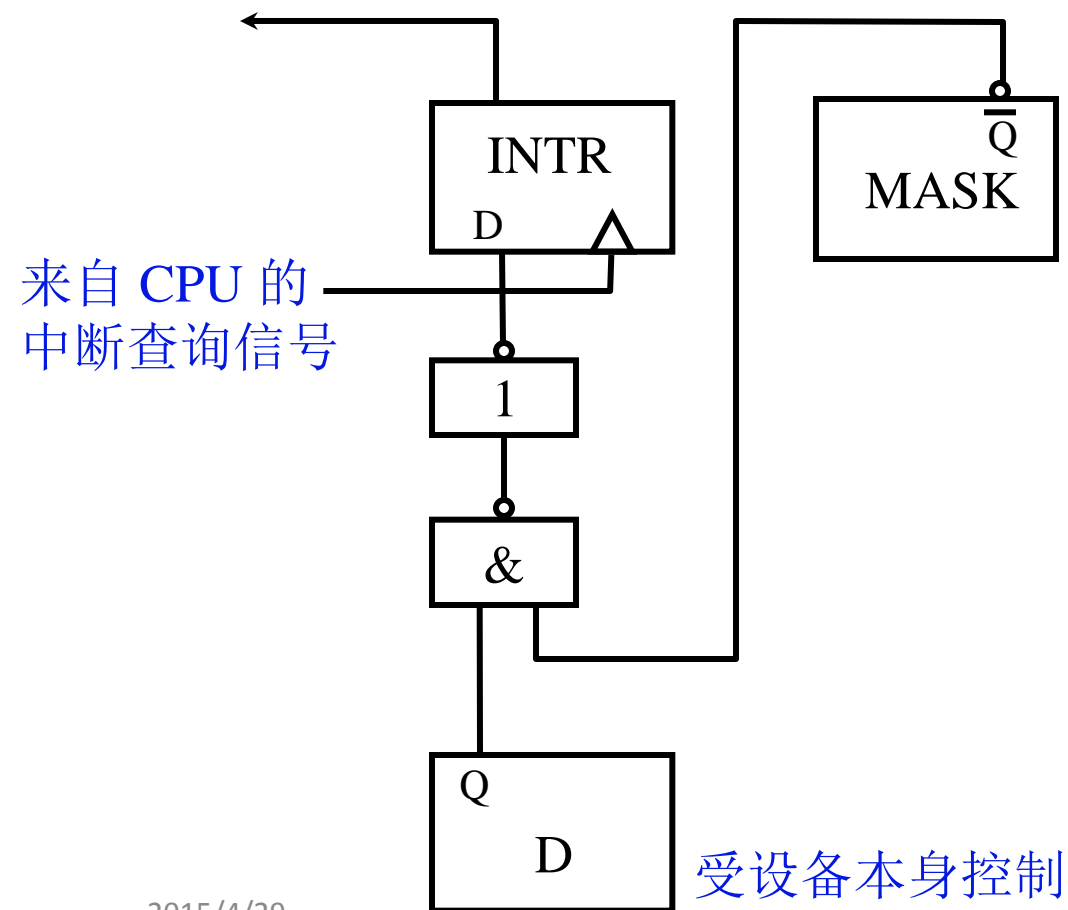
1. 配置中断请求触发器和中断屏蔽触发器
2. 排队器
3. 中断向量地址形成部件
4. 程序中断方式接口电路的基本组成

三、程序中中断方式的接口电路

5.5

1. 配置中断请求触发器和中断屏蔽触发器

中断请求



INTR

中断请求触发器

$INTR = 1$ 有请求

MASK

中断屏蔽触发器

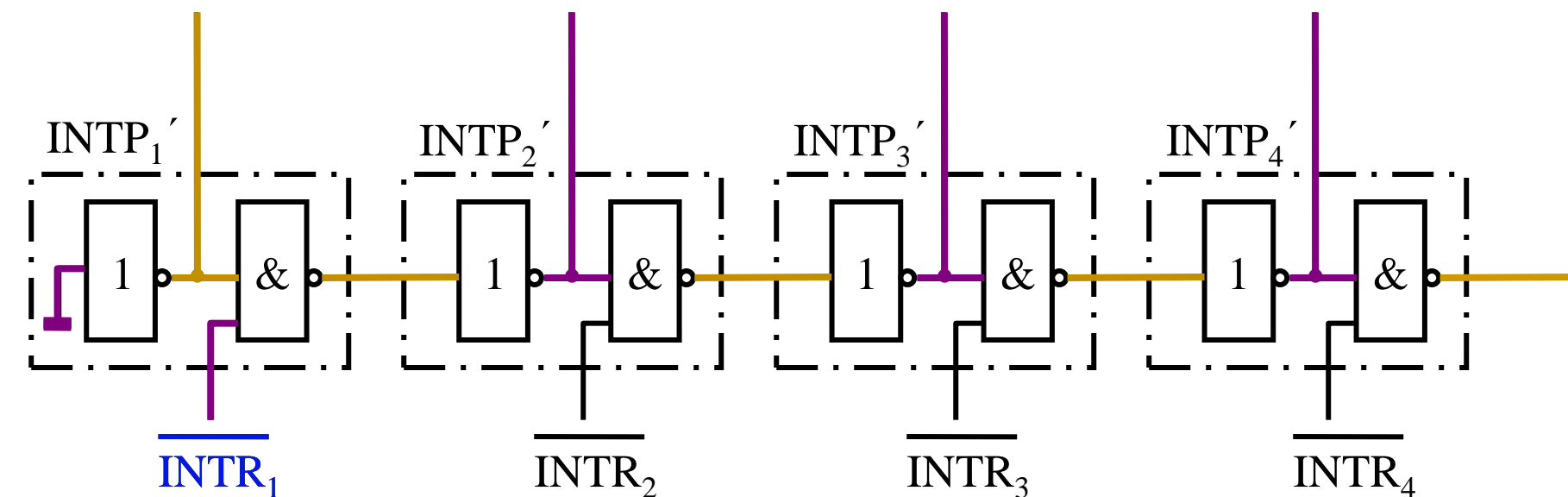
$MASK = 1$ 被屏蔽

D 完成触发器

2. 排队器

5.5

排队 { 硬件 在 CPU 内或在接口电路中（链式排队器）
软件 详见第八章



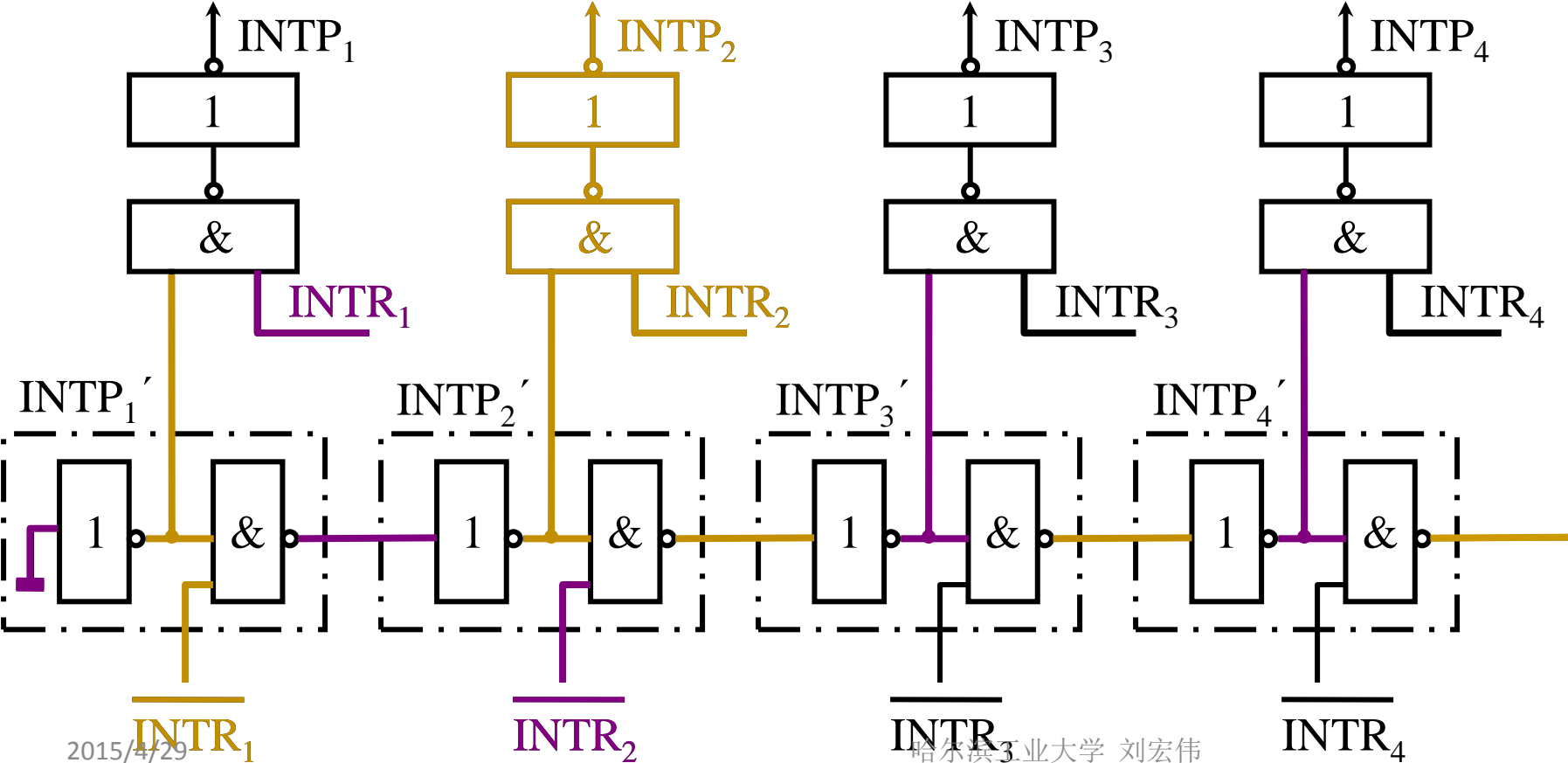
设备 1[#]、2[#]、3[#]、4[#] 优先级按 降序排列

$\text{INTR}_i = 1$ 有请求 即 $\overline{\text{INTR}}_i = 0$

2. 排队器

5.5

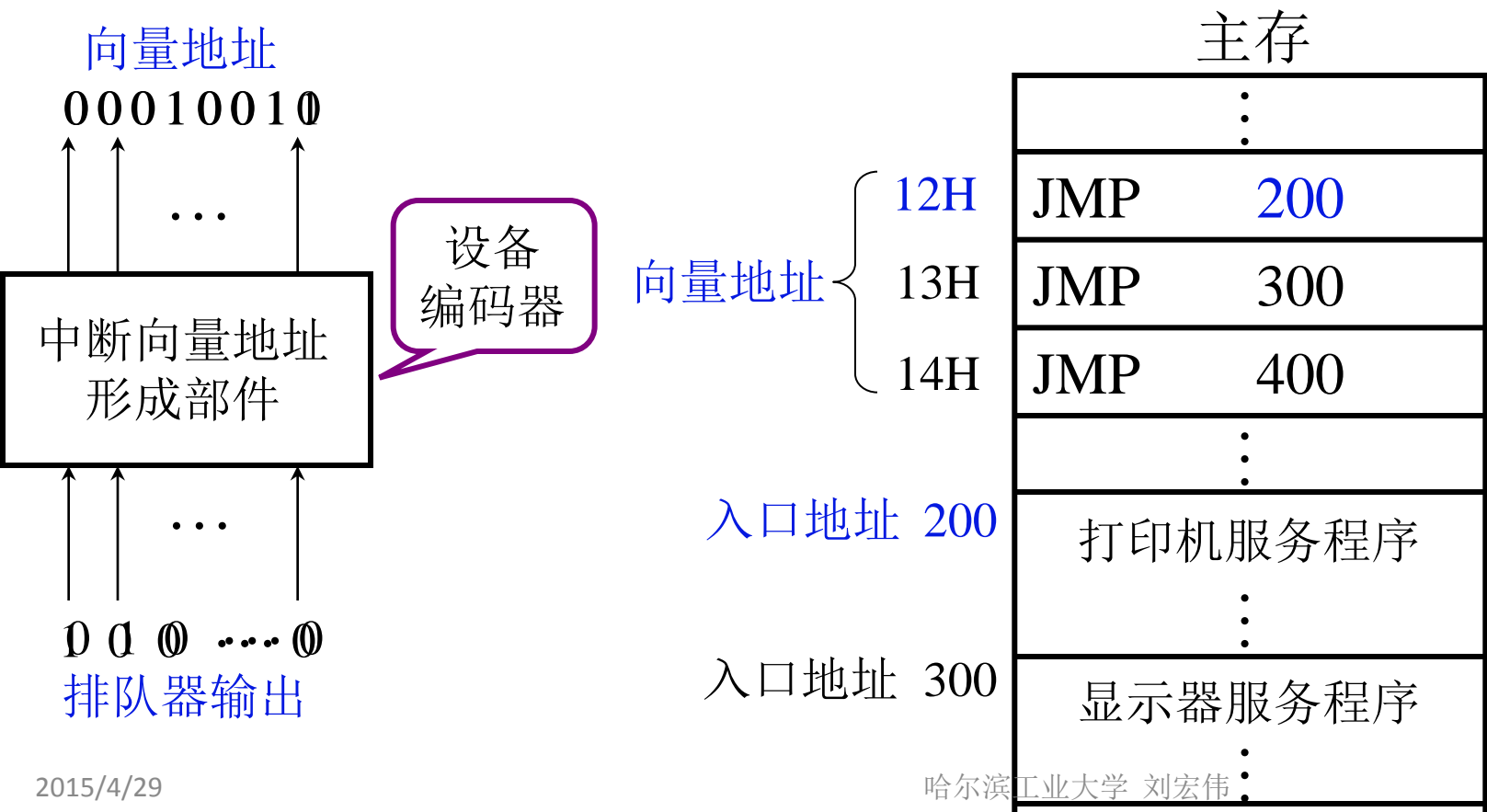
排队 { 硬件 在 CPU 内或在接口电路中（链式排队器）
软件 详见第八章



3. 中断向量地址形成部件

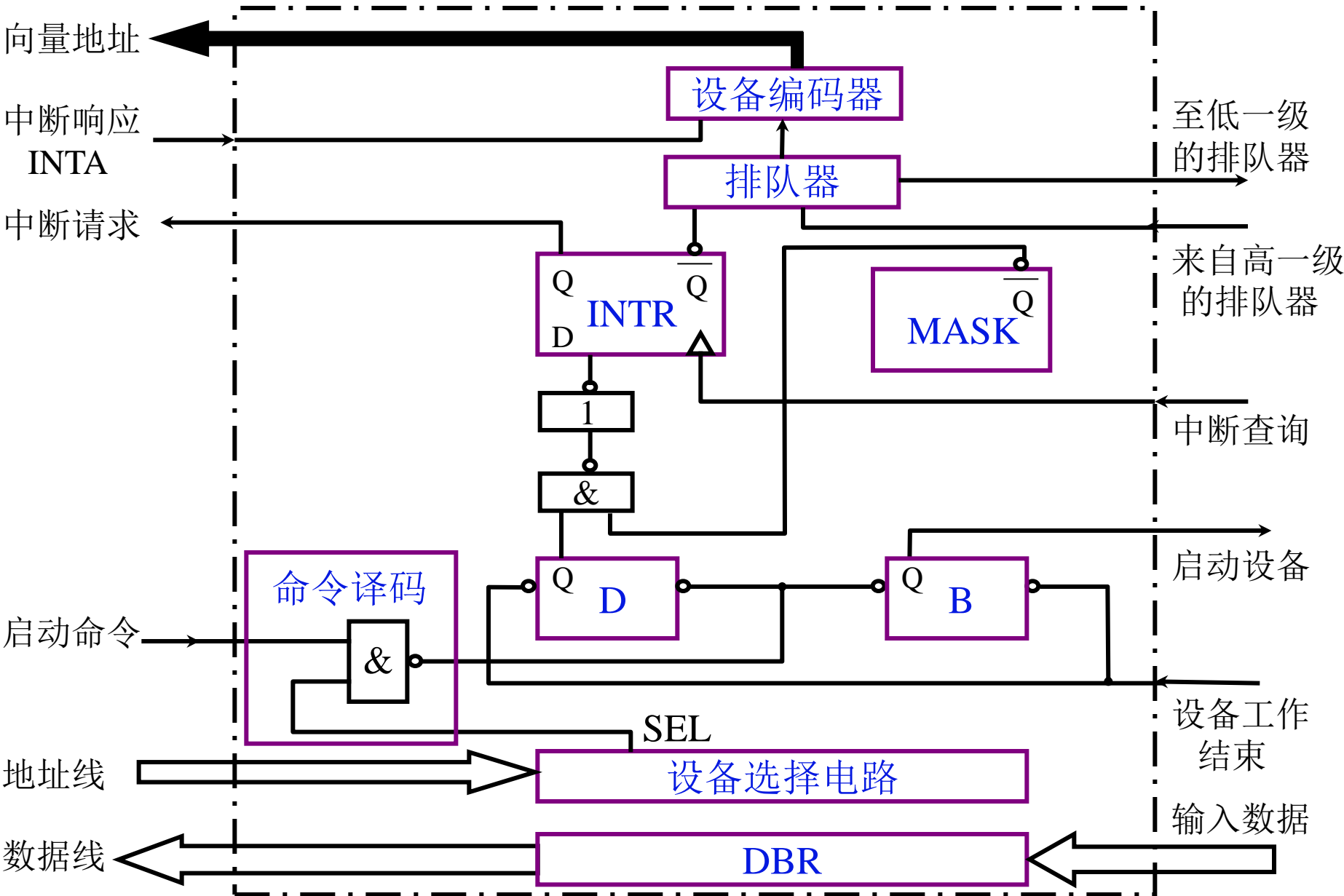
5.5

入口地址 { 由软件产生 详见第八章
 由硬件向量法 由 硬件 产生 向量地址
 再由 向量地址 找到 入口地址



4. 程序中断方式接口电路的基本组成

5.5



四、I/O 中断处理过程

5.5

1. CPU 响应中断的条件和时间

(1) 条件

允许中断触发器 **EINT = 1**

用 **开中断** 指令将 **EINT** 置 **“1”**

用 **关中断** 指令将 **EINT** 置 **“0”** 或硬件 **自动复位**

(2) 时间

当 **D = 1**（随机）且 **MASK = 0** 时

在每条指令执行阶段的结束前

CPU 发 **中断查询信号**（将 **INTR** 置 **“1”**）

5.5 程序中中断方式

- 一、中断的概念
- 二、I/O中断的产生
- 三、程序中中断方式的接口电路
- 四、I/O 中断处理过程
- 五、中断服务程序流程

五、中断服务程序流程

5.5

1. 中断服务程序的流程

(1) 保护现场

{ 程序断点的保护 中断隐指令完成
 寄存器内容的保护 进栈指令

(2) 中断服务

对不同的 I/O 设备具有不同内容的设备服务

(3) 恢复现场 出栈指令

(4) 中断返回 中断返回指令

2. 单重中断和多重中断

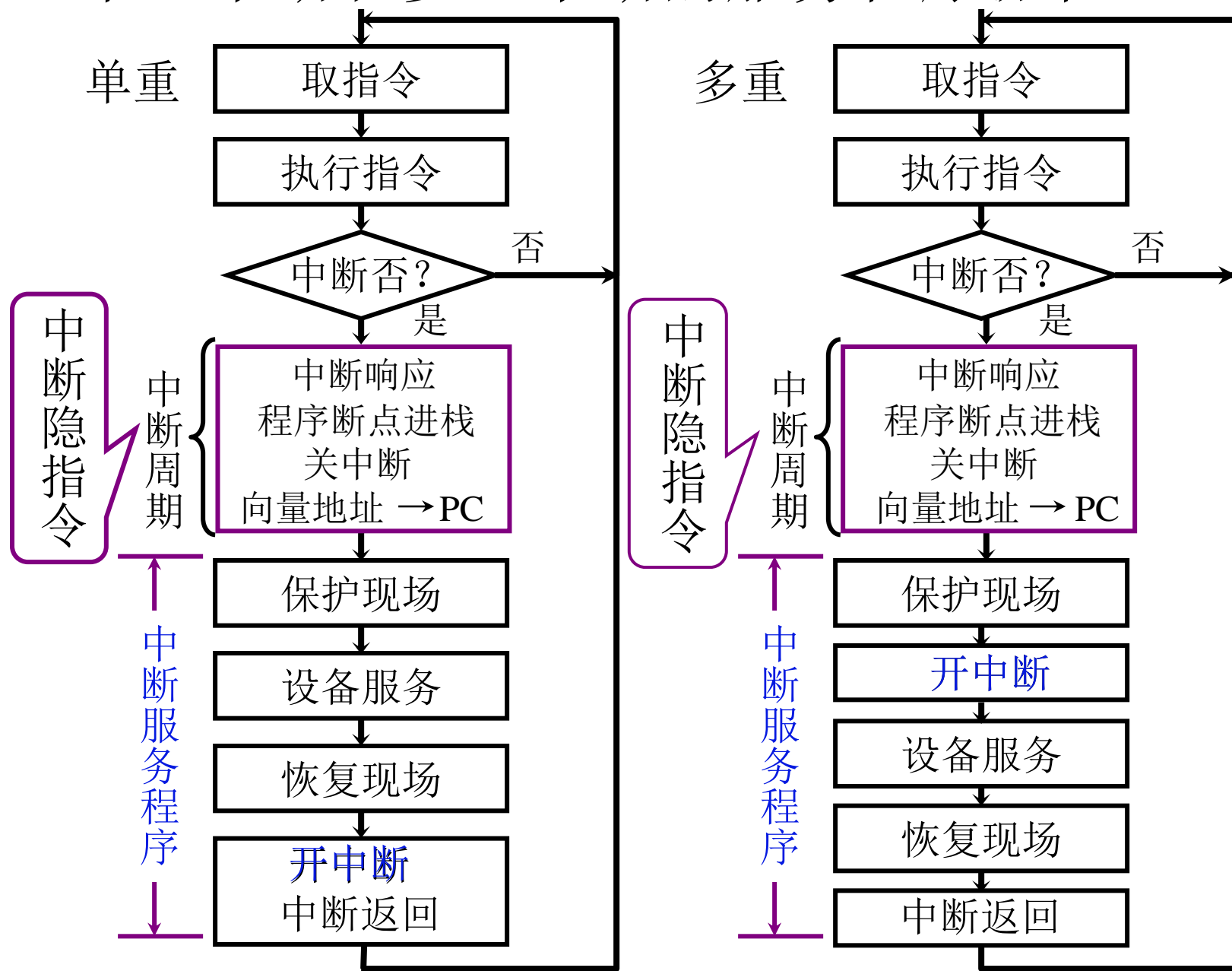
单重 中断 不允许中断 现行的 中断服务程序

多重 中断 允许级别更高 的中断源

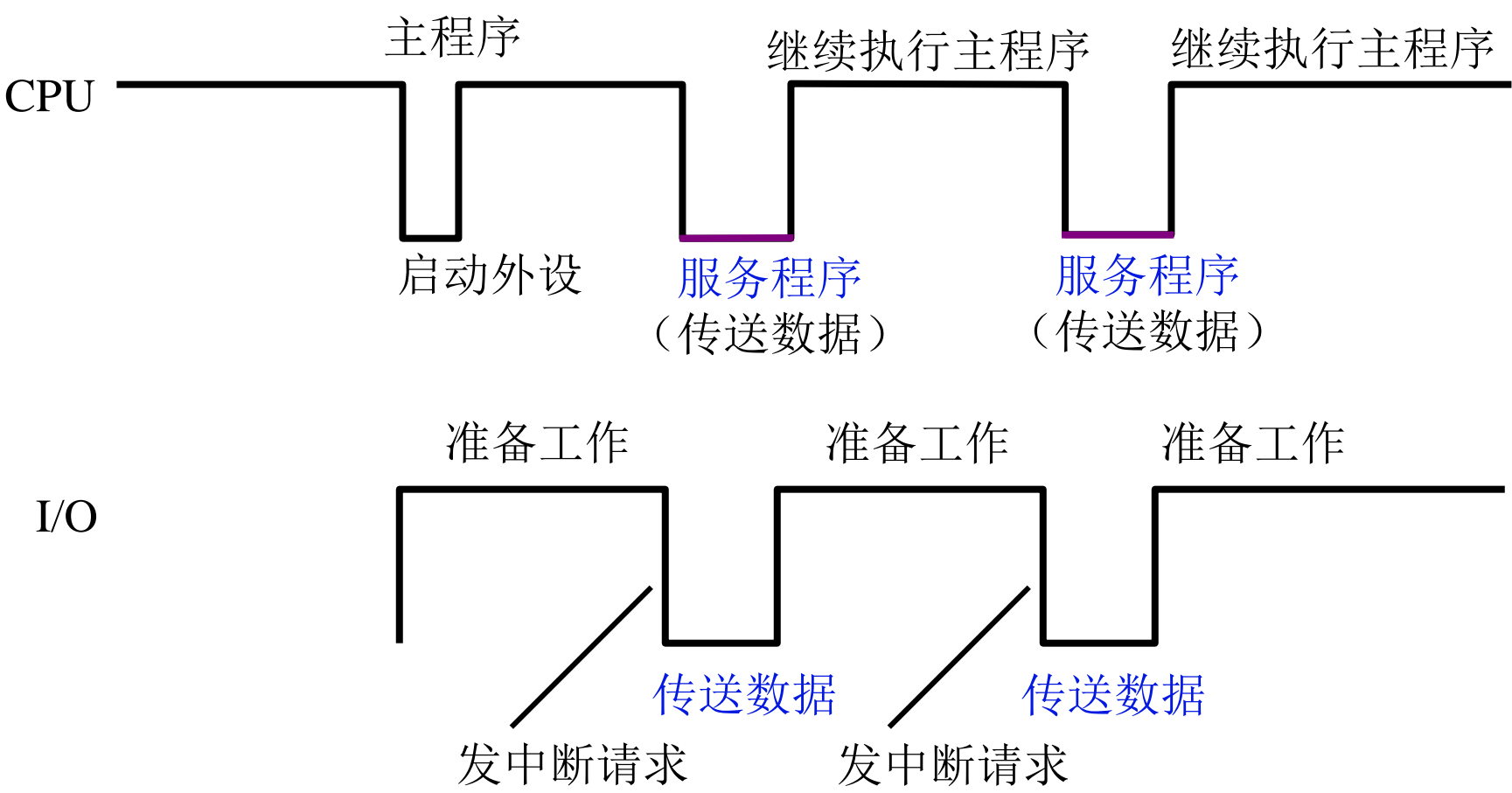
中断 现行的 中断服务程序

3. 单重中断和多重中断的服务程序流程

5.5



4. 主程序和服务程序抢占 CPU 示意图 5.5



宏观 上 CPU 和 I/O 并行 工作
微观 上 CPU 中断现行程序 为 I/O 服务