第7章 中断技术

华中科技大学
电子信息与通信学院
School of Electronic Information and Communications

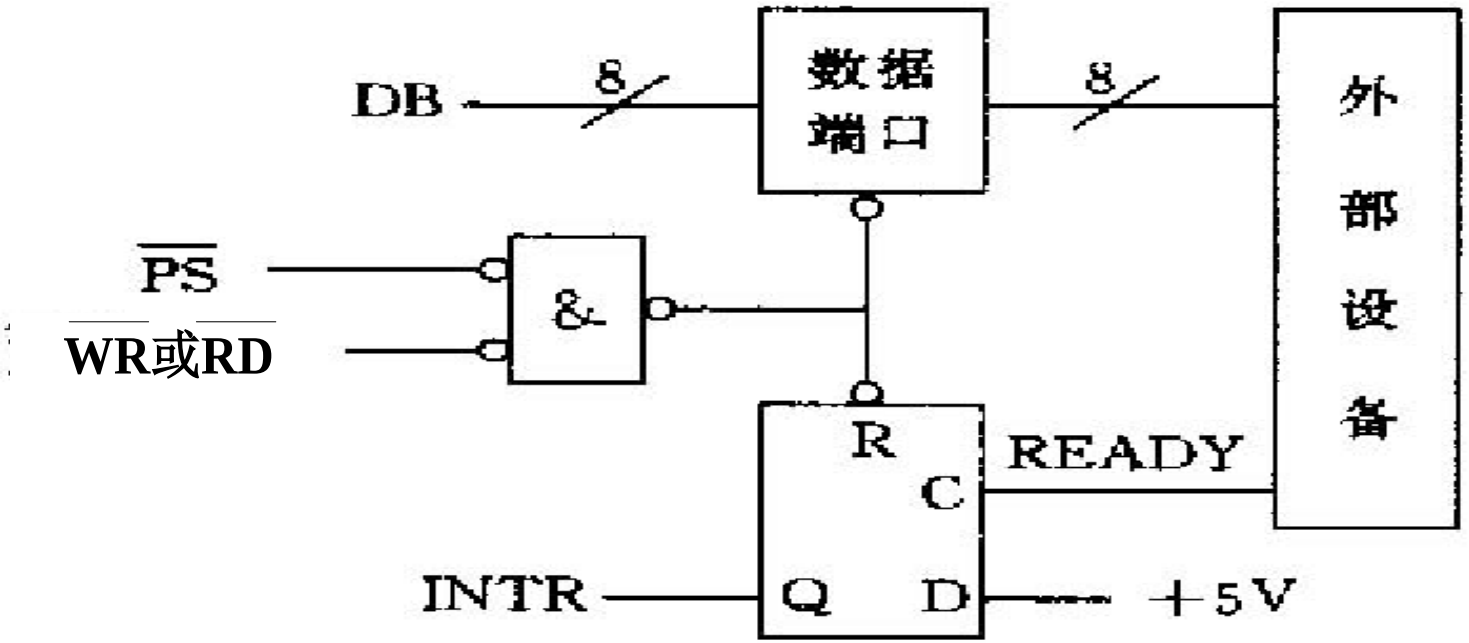
学习目标

- 了解中断控制器的基本构成
- 掌握AXI INTC的结构以及编程控制

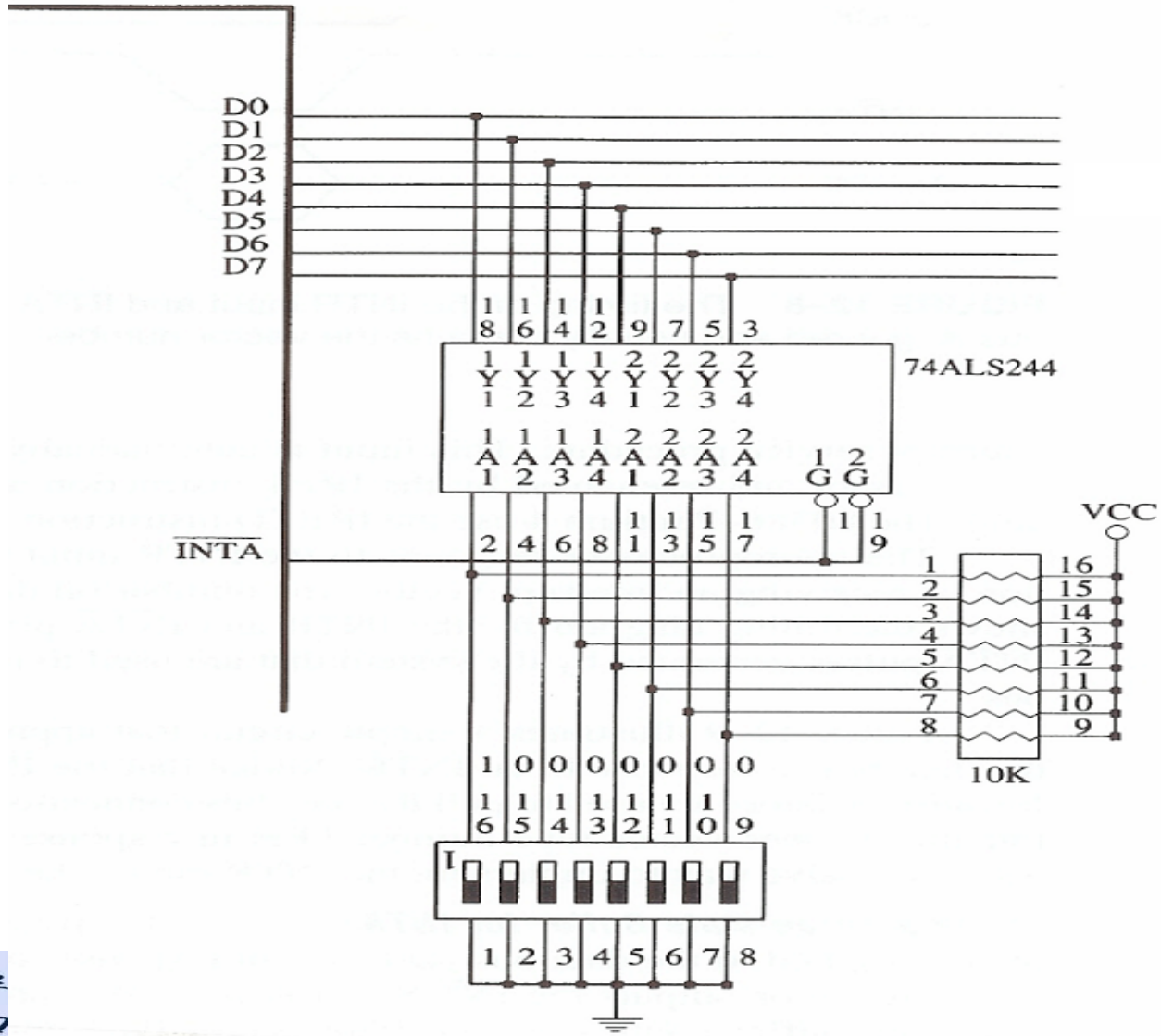
中断控制器构成

- 中断系统一般应具有以下功能：
  - 中断请求信号保持与清除，
  - 中断源识别，
  - 中断允许控制，
  - 中断优先级设置。

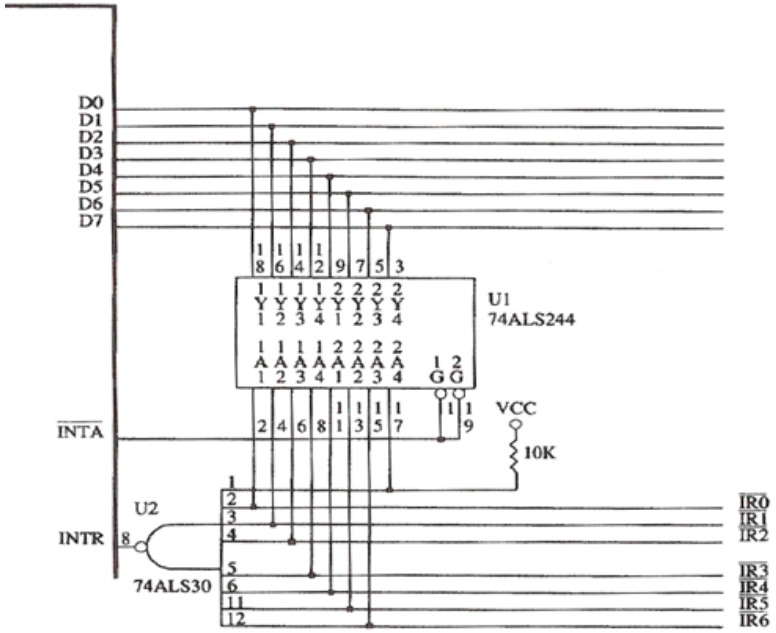
中断请求信号保持与清除



中断源识别——读取中断类型码



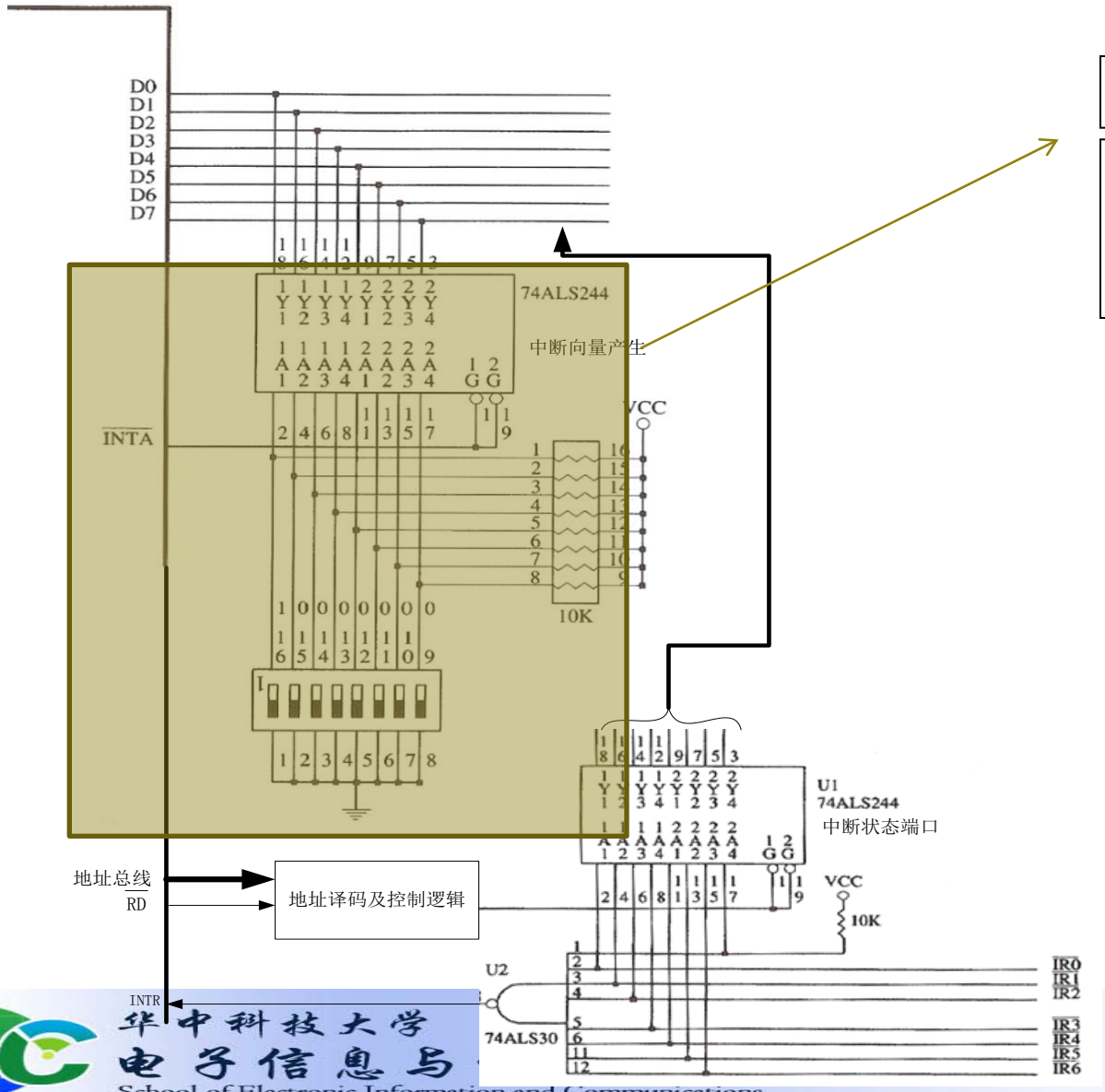
多中断源



同时产生多个中断，中断类型码种类太多（256）

	中断类型码							
中断源	D7	D6	D5	D4	D3	D2	D1	D0
IR0	1	1	1	1	1	1	1	0
IR1	1	1	1	1	1	1	0	1
IR2	1	1	1	1	1	0	1	1
IR3	1	1	1	1	0	1	1	1
IR4	1	1	1	0	1	1	1	1
IR5	1	1	0	1	1	1	1	1
IR6	1	0	1	1	1	1	1	1

软件查询中断源-单一向量

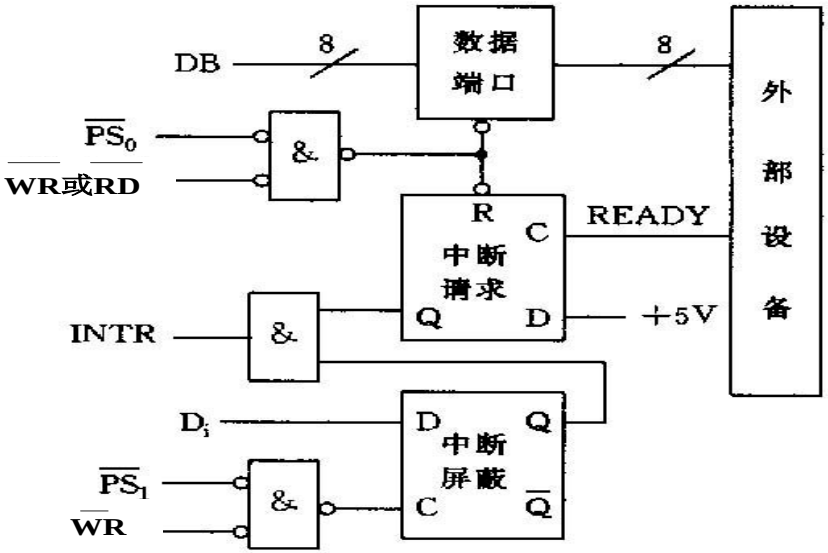


嵌入式处理器

固化，所有外设采用一个固定的中断类型码

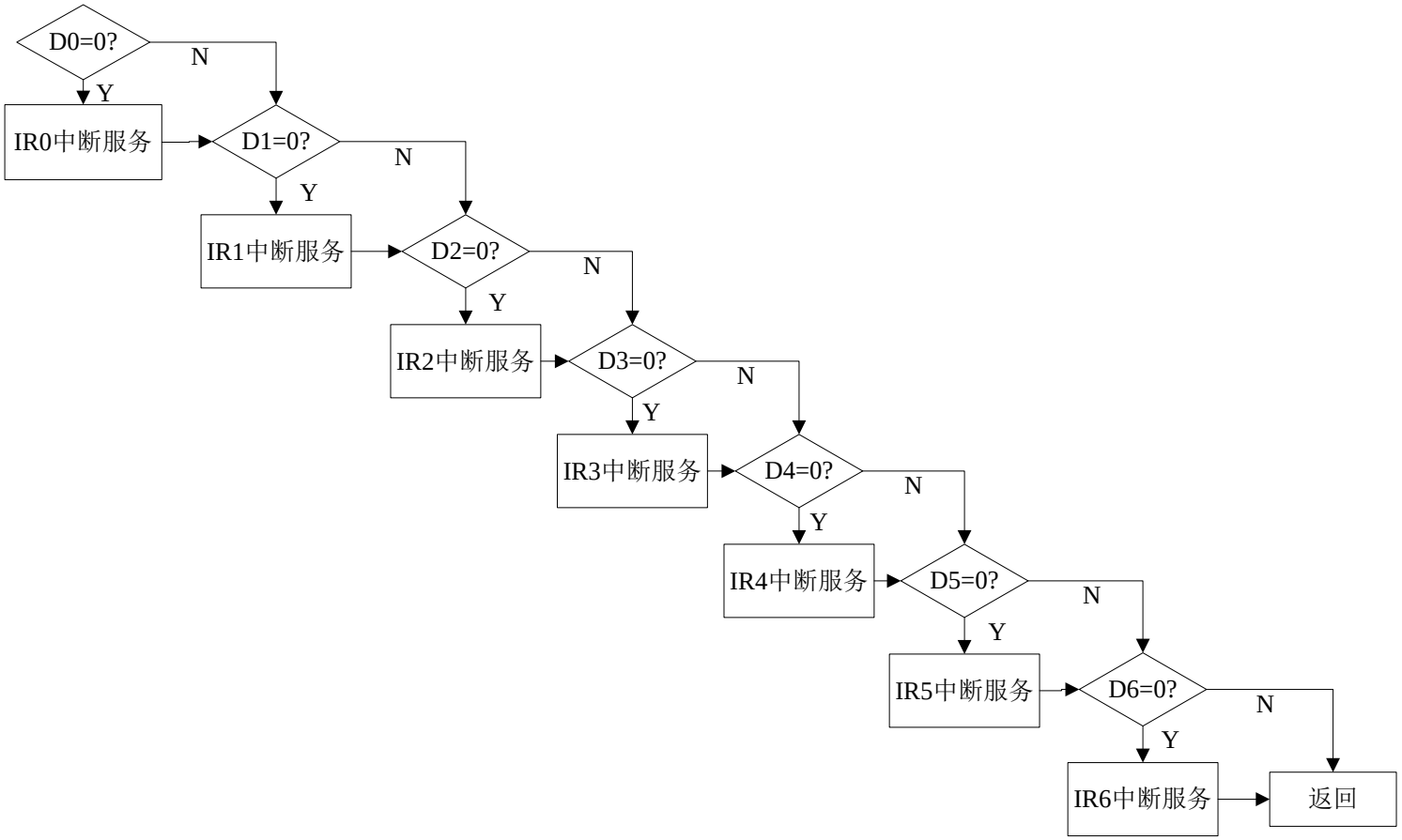
中断控制

- 微处理器用输出指令来控制中断屏蔽触发器的状态，从而控制是否接受某个特殊外设的中断请求
- 微处理器内部也有一个中断允许触发器，只有当其为“1”（即开中断），CPU才能响应外部中断

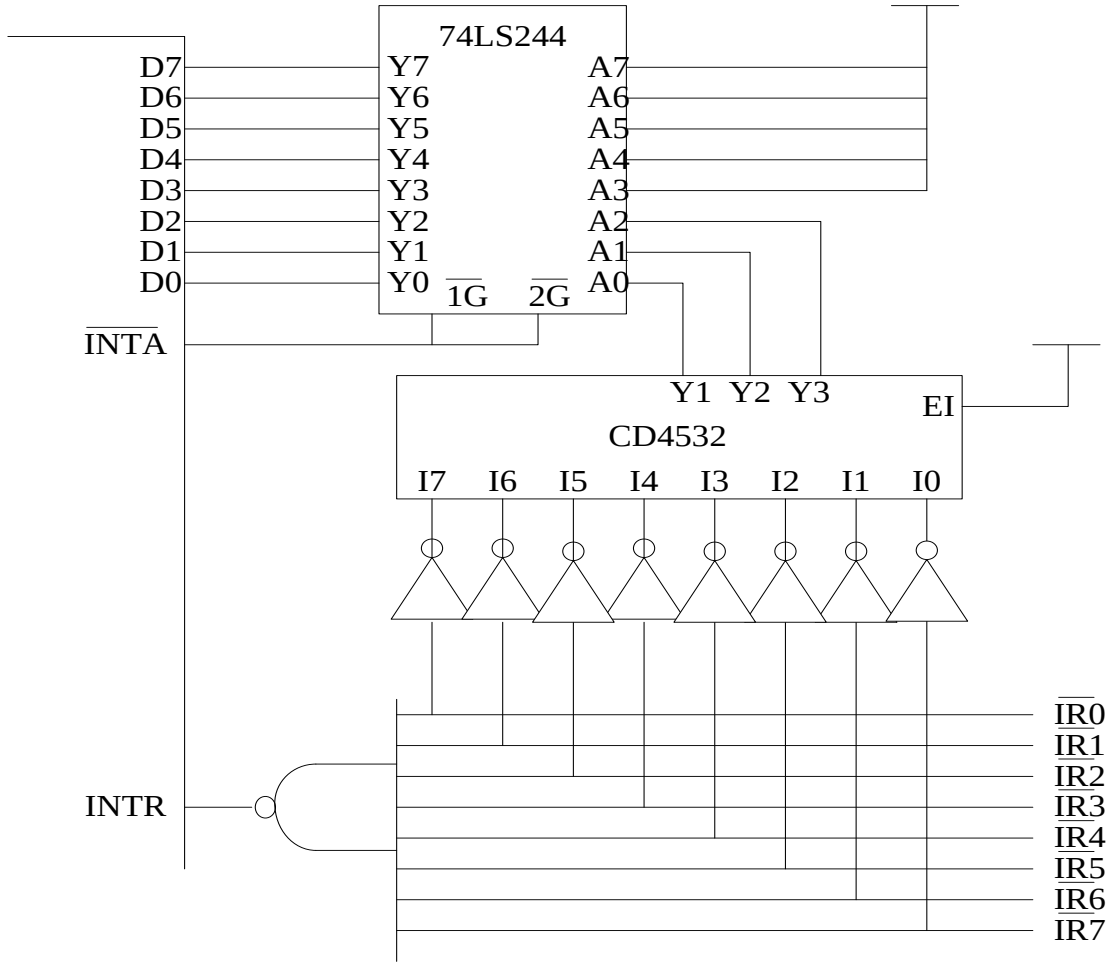


中断优先级

- 软件查询顺序决定

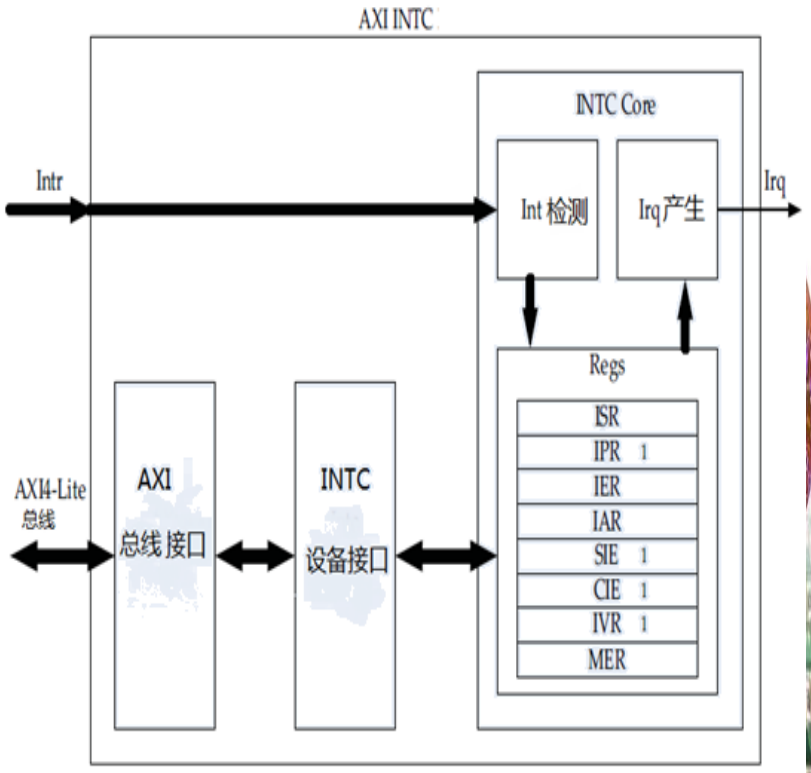


优先编码器


74LS244
D7
D6
D5
D4
D3
D2
D1
D0
Y7
Y6
Y5
Y4
Y3
Y2
Y1
Y0
1G
2G
A7
A6
A5
A4
A3
A2
A1
A0
INTA
Y1 Y2 Y3
EI
CD4532
I7 I6 I5 I4 I3 I2 I1 I0
IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7
INTR

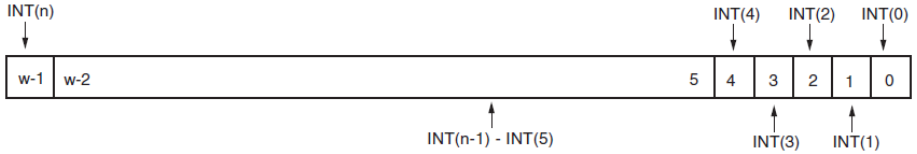
中断控制器举例-AXI INTC

- 支持32个中断源输入，每个中断源都可以配置为4种中断触发方式中的任意一种
- 一个中断请求信号输出，可配置为4种中断触发方式中的任意一种
- 可以级联
- 中断请求输入端的优先级根据所处位置决定，bit0具有最高优先级，bit31优先级最低
- 每个中断源可以单独屏蔽或开放，也可以同时屏蔽所有中断源



寄存器偏移地址

寄存器名称	偏移地址	允许操作	初始值	含义
ISR	0x0	Read / Write	0x0	中断请求状态寄存器
IPR（可选）	0x4	Read	0x0	中断悬挂寄存器
IER	0x8	Read / Write	0x0	中断屏蔽寄存器
IAR	0xC	Write	0x0	中断响应寄存器
SIE（可选）	0x10	Write	0x0	中断允许设置寄存器
CIE（可选）	0x14	Write	0x0	中断允许清除寄存器
IVR（可选）	0x18	Read	0x0	中断类型码寄存器
MER	0x1C	Read / Write	0x0	主中断屏蔽寄存器



ISR,IPR,IER,IAR,SIE,CIE

MER

中断处理过程

(1) 在中断请求输入端Intr上接受中断请求。

(2) 中断请求锁存在ISR中，并与IER相“与”，若使用优先级判断电路，那么将未屏蔽的中断送给优先级判定电路。

(3) 控制逻辑接受中断请求，输出Irq信号。

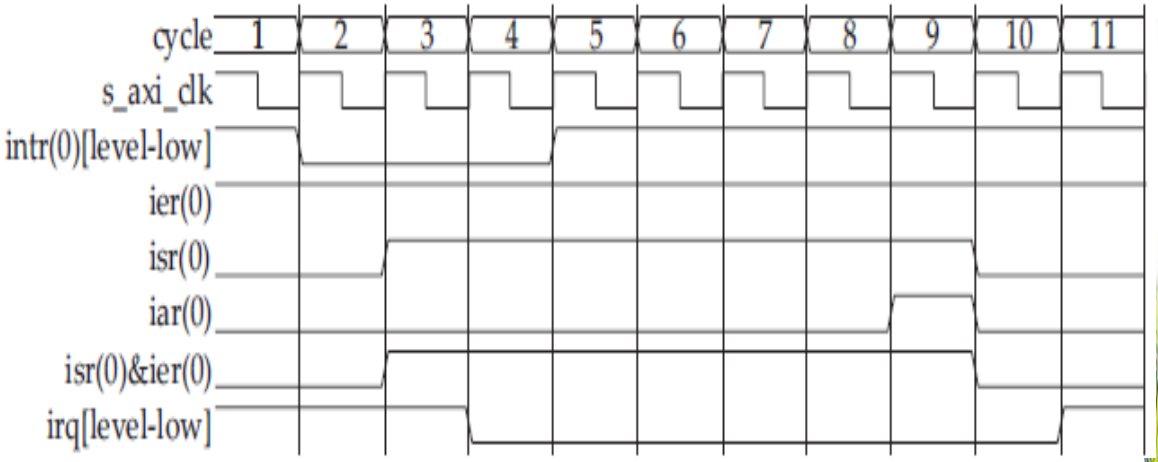
(4) 若使用优先级判断电路，优先级判定电路检出优先级最高的中断请求位，并将IVR设置为相应的值。

(5) 进入微处理器中断响应过程。若使用优先级判断电路，微处理器读取IVR识别当前优先级最高的中断请求源。若没有使用优先级判断电路，微处理器就需要读取ISR，识别产生中断的请求源。

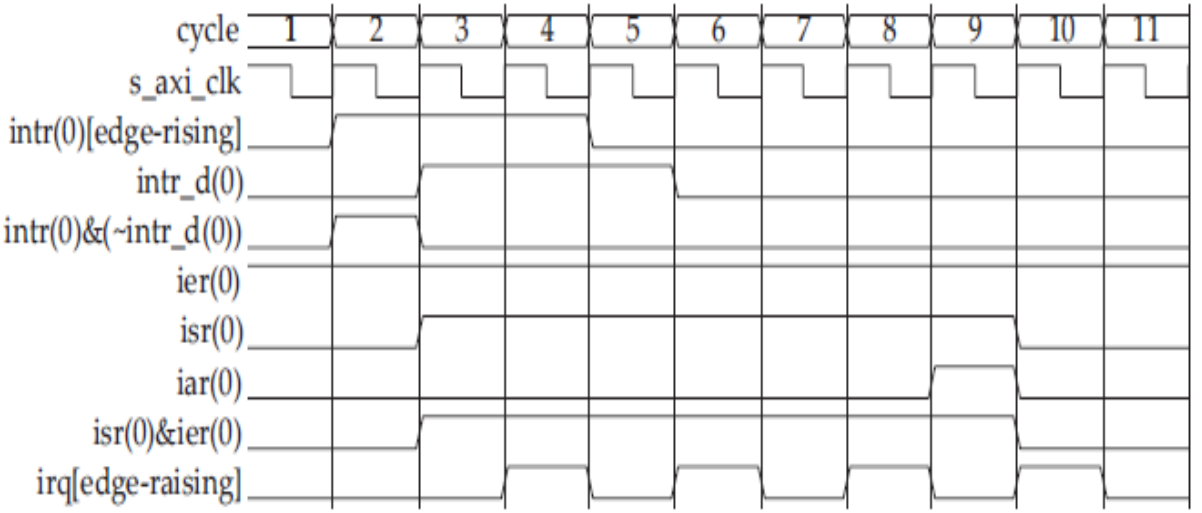
(6)微处理器向中断响应寄存器（IAR）对应的位写入1，使ISR相应位复位从而结束中断。

中断信号时序(4*4=16种）

Intr和Irq都为低电平中断触发方式



Intr和Irq都为上升沿中断触发方式



编程控制

- 初始化，通常包括两个方面：
  - 修改MER，使得HIE和ME都为1；
  - 修改IER，使得连接了中断请求源的相应位为1，允许中断请求
- 中断结束
  - 写IAR, 中断请求状态复位

- 某小字节序计算机系统利用AXI INTC作为中断控制器，该中断系统可以接收5个中断源的中断申请，分别将它们连接到中断请求端Intr0~4，请编写对AXI INTC进行初始化的程序段。假定AXI INTC的基地址为0x80000000。

IER的地址为0x80000008，MER的地址为0x8000001c。
允许Intr0~4的中断请求，在小字节序的计算机系统中，IER的值为0x0000001f。MER的值为0x00000003。采用Xilinx C语言控制的程序段为：

```
Xil_Out32(0x80000008, 0x0000001f);
Xil_Out32(0x8000001c, 0x00000003);
```

作业

- 1，2