

5.3 中断方式及接口

5.3.1 中断基本概念

1. 定义

CPU暂时中止现行政程序的执行，转去执行为某个随机事态服务的中断处理程序。处理完毕后自动恢复原程序的执行。

2. 实质与特点

(1) 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。

(2) 特点

随机性 { 随机发生的事态(按键、故障)
有意调用, 随机请求与处理的事态(调用打印机)
随机插入的事态(软中断指令插入程序任何位置)

注意**中断**与**转子**的区别。

3. 中断分类

(1) 硬件中断与软中断

由硬件请求信号引发中断

由软中断指令引发中断

(2) 内中断与外中断

中断源来自主机内部

中断源来自主机外部

(3) 可屏蔽中断与非屏蔽中断

可通过屏蔽字屏蔽该类请求; 关中断时不响应该类请求。

该类请求与屏蔽字无关; 请求的响应与开/关中断无关。

(4) 向量中断与非向量中断

由硬件提供服务程序入口地址

由软件提供服务程序入口地址

4. 中断典型应用

(1) 管理中、低速I/O操作

(2) 处理故障

(3) 实时处理

某事件出现的实际时间内及时处理，不是批量处理。

利用时钟中断定时采集参数，检测，调节。

(4) 人机对话

(5) 多机通信

5. 中断系统的组成

中断系统的硬、软界面

(1) 软件：服务程序、中断向量表

(2) 硬件 { 接口方面：请求、传递、判优逻辑
CPU方面：响应逻辑

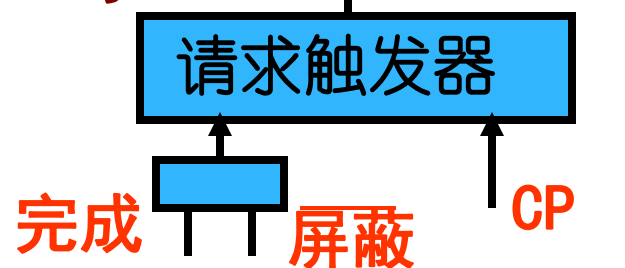
5.3.2 中断全过程（外中断）

1. 中断请求的提出与传递

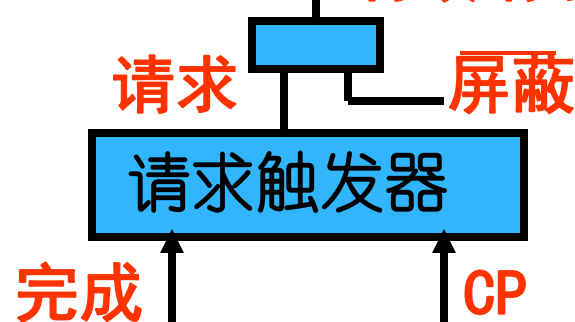
(1) 如何产生中断请求？

- 外设工作完成：“完成”标志为1
- CPU允许请求：“屏蔽”标志为0

先“屏蔽”，后请求

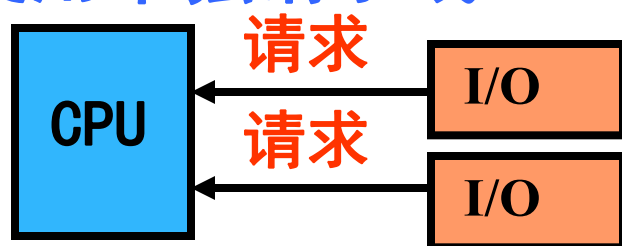


先请求，后“屏蔽”
有效请求

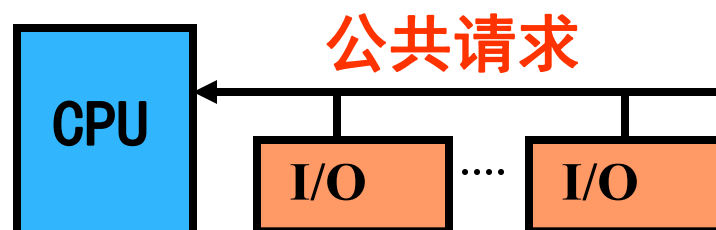


(2) 如何传送中断请求？

1) 使用单独请求线



2) 使用公共请求线



2. 中断判优

(1) 优先顺序 故障、DMA、外中断（输入、输出）

(2) CPU现执行程序与外设请求的判优

1) CPU设置允许中断标志 $\begin{cases} =1, & \text{开中断} \\ =0, & \text{关中断} \end{cases}$ (模型机采用)

2) CPU设置程序状态字的优先级字段

为现执行程序赋予优先级 $\begin{cases} < \text{外设请求优先级, 响应} \\ \geq \text{外设请求优先级, 不响应} \end{cases}$

(3) 各外设请求的判优

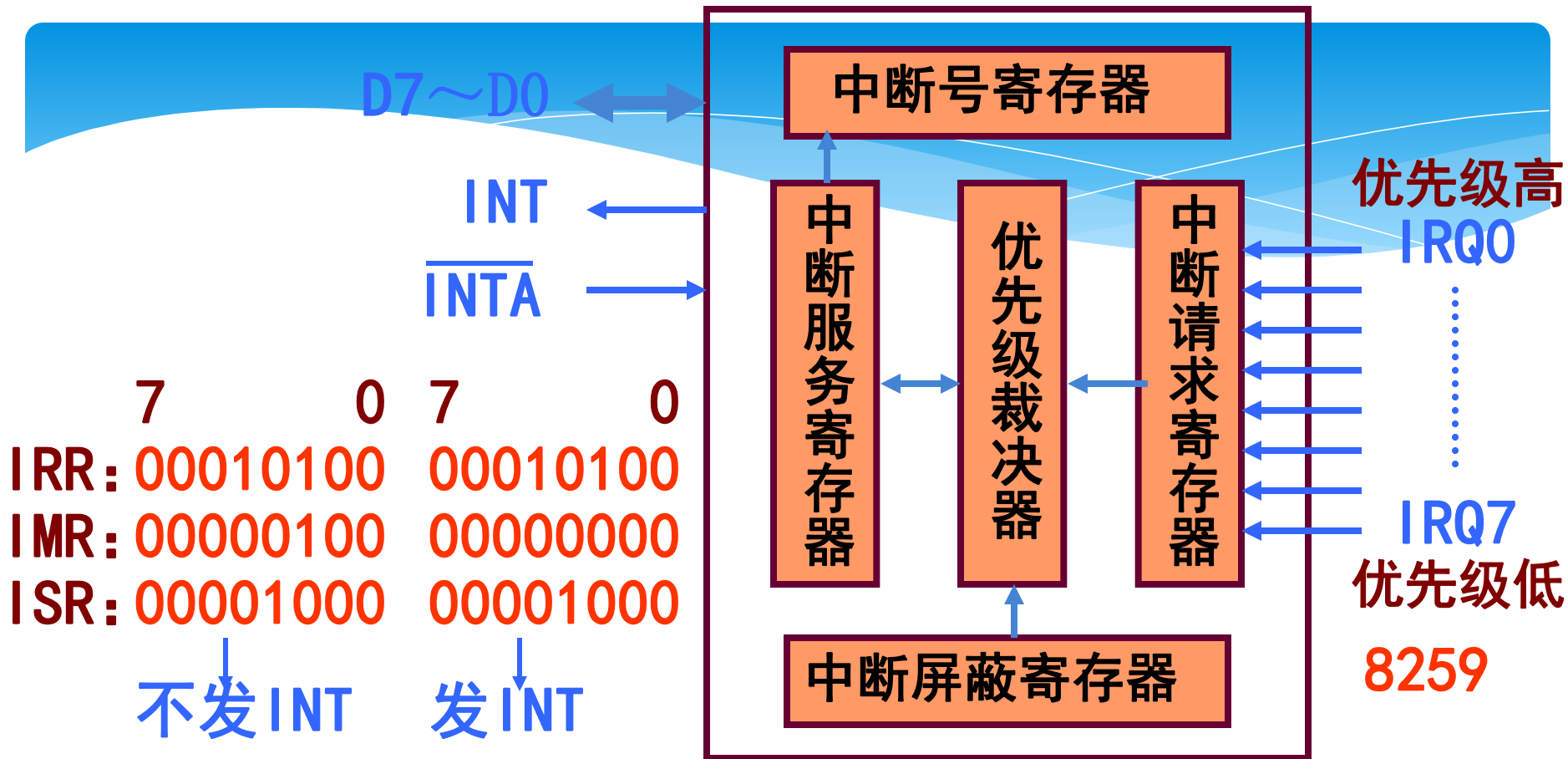
1) 软件判优

由程序查询顺序确定优先级。可灵活修改优先级。

2) 硬件判优

例. 中断控制器判优

中断控制器(如8259)集中解决请求信号的接收、屏蔽、判优、编码等问题。



中断请求 → 8259 (未屏蔽的请求判优, 生成相应中断号)
 → 公共请求 INT → CPU
 (CPU响应后, 取回中断号, 转入相应服务程序。)

3. 中断响应

(1) 响应条件

外设有请求，且未被屏蔽；CPU开中断；一条指令（非停机指令）结束；无故障、DMA等优先级更高的请求。

（2）如何获取中断服务程序的入口地址

1) 非向量中断

将服务程序入口组织在查询程序中；CPU响应时执行查询程序，查询中断源，转入相应服务程序。

2) 向量中断

将服务程序入口（**中断向量**）组织在**中断向量表**中；CPU响应时由硬件直接产生相应**向量地址**，按地址查表，取得服务程序入口，转入相应服务程序。

- **中断向量**：服务程序入口地址、服务程序状态字
- **中断向量表**：存放中断向量的表（一段存储区）
- **向量地址**：访问向量表的地址（指向中断向量的首址）

例1. 模型机向量表（从主存2#单元开始安排）

M按字编址。一个入口地址16位，占一个编址单元。

向量表

向量地址 = 中断号 + 2
(单元地址)

2#

3#

入口地址0
入口地址1
⋮

0号中断源

1号中断源

例2. IBM PC向量表

(从主存0#单元开始安排)

M按字节编址。一个入口地址32位，占4个编址单元。

向量表

向量地址 = 中断号 × 4

0#

4#

入口偏移0
入口基址0
入口偏移1
入口基址1
⋮

0号中断源

1号中断源

(3) 响应过程

向量中断方式:

CPU执行中
断隐指令
(硬件完成)

发响应信号INTA, 进入中断周期

关中断, 保存断点

获得中断号, 转换为向量地址,
查向量表

取中断向量, 转中断服务程序

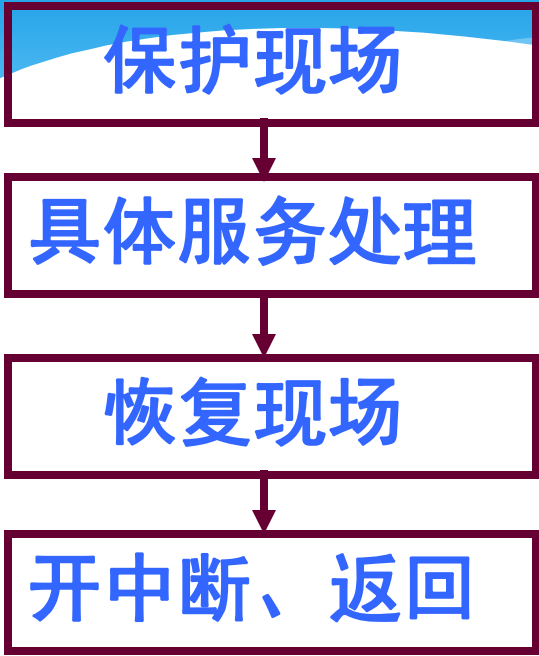
4. 中断处理

CPU执行中断服务程序。

(1) 单级中断: CPU响应后只处理一个中断源的请求, 处理完以后才能响应新的请求。

(2) 多重中断: 在某次中断服务过程中, 允许响应处理更高级别的中断请求。

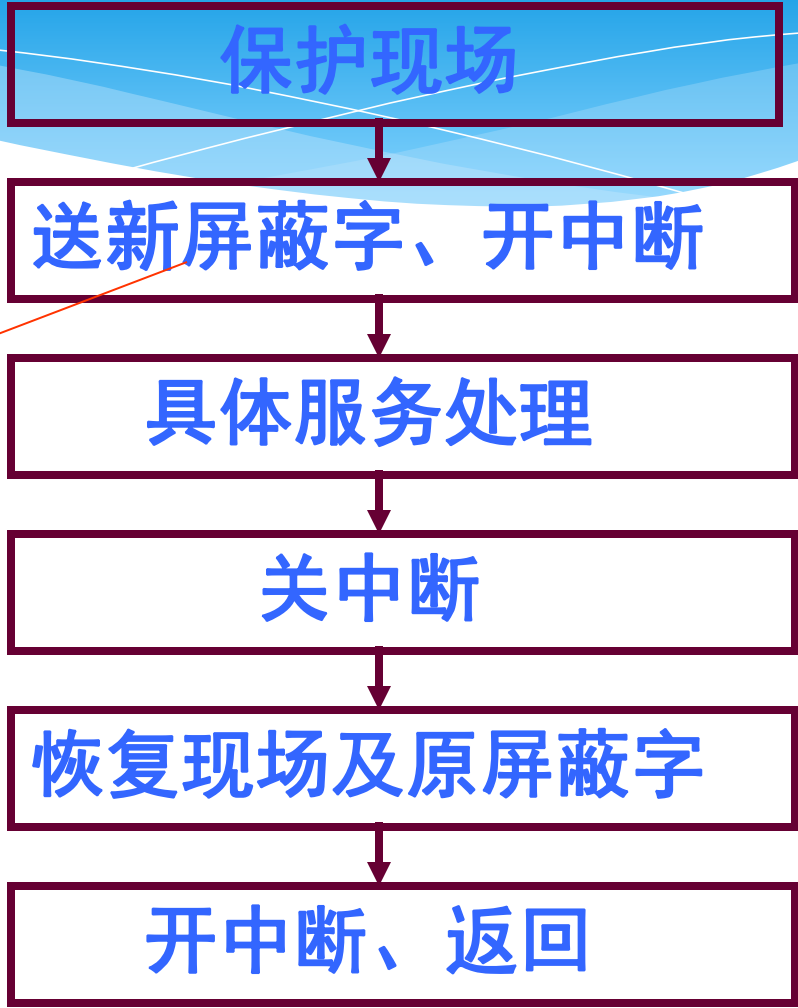
单级中断流程：



外中断：
数据传送

禁止同
级或更
低级别
的请求，
开放更
高级别
的请求

多重中断流程：



5.3.3 中断接口

1. 组成（寄存器级）

（1）寄存器选择

对接口寄存器寻址。

（2）命令字寄存器

接收CPU发向外设的命令字，转换为相应操作命令送外设。

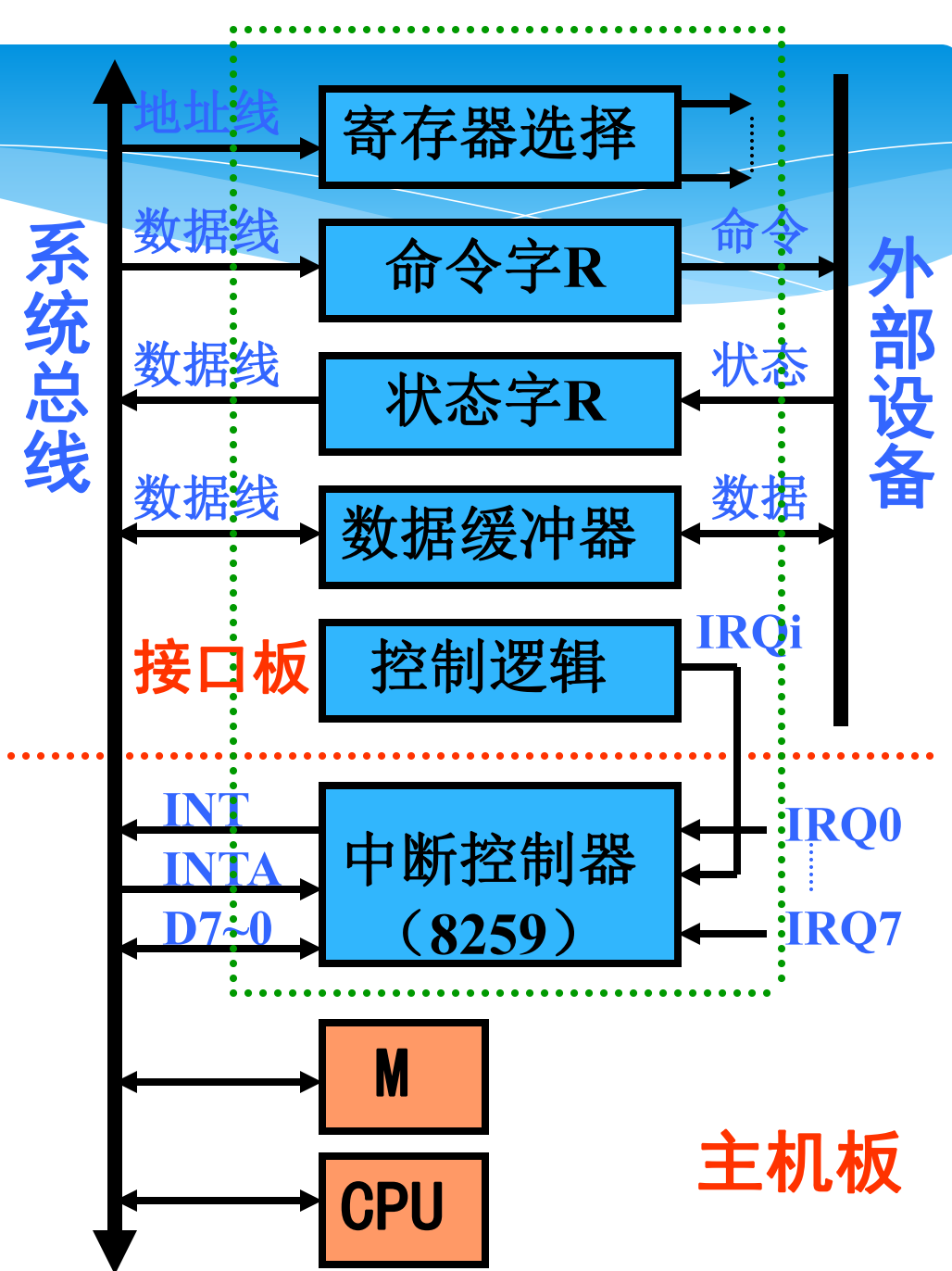
命令字格式的拟定：

用代码表示各种命令

代码位数
代码含义

（3）状态字寄存器

反映设备和接口的运行状态。

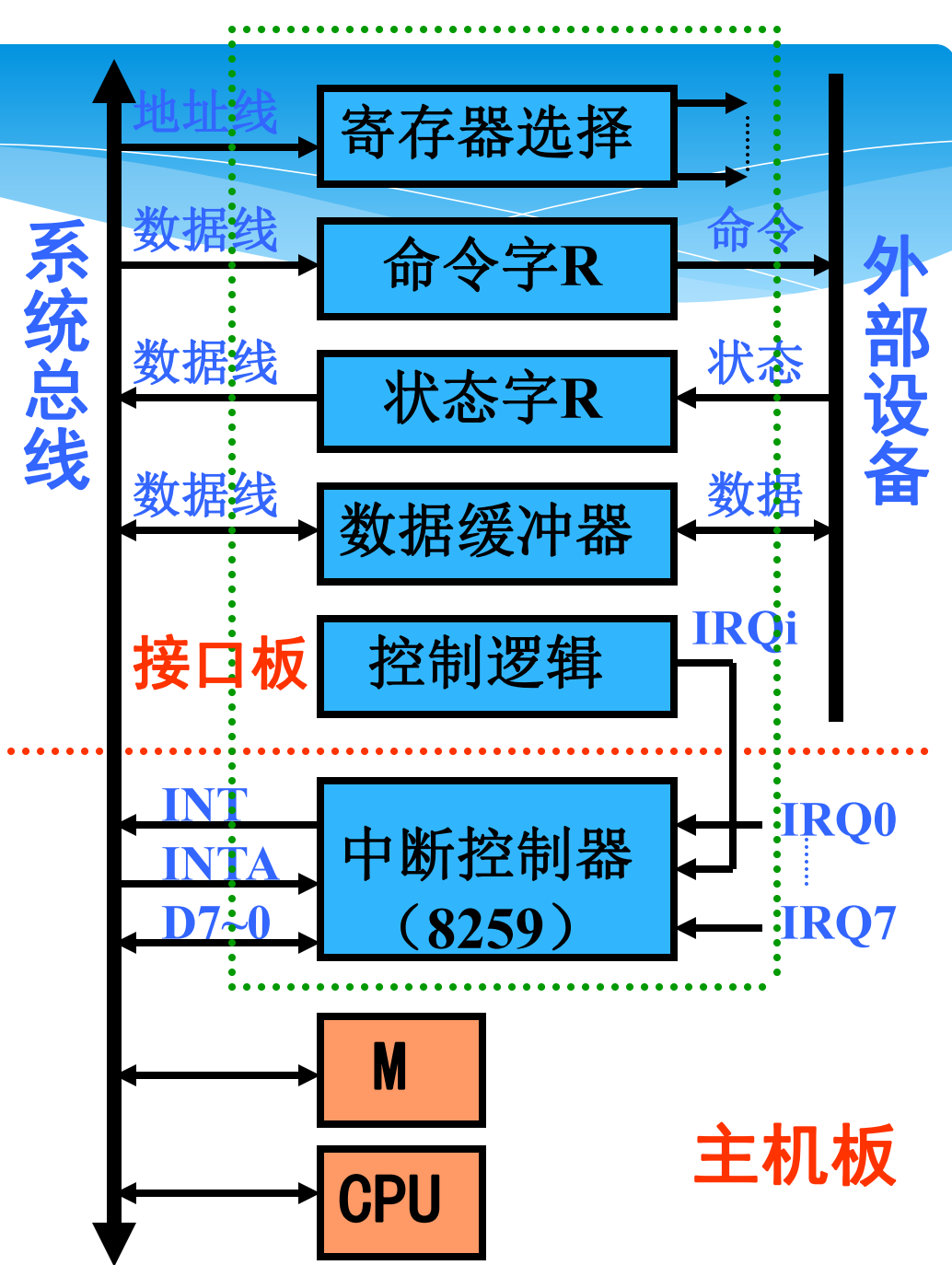


状态字格式的拟定：
用代码表示各种状态。

(4) 数据缓冲器
传送数据，实现缓冲。

(5) 控制逻辑
请求信号产生逻辑
电平转换逻辑
串-并转换逻辑(串口)
针对设备特性的逻辑

(6) 公用中断控制器
接收外设请求，判优，
送出公共请求；
接收中断批准，送出中
断号（中断类型码）。



2. 工作过程（外中断）

(1) 初始化：设置工作方式，送屏蔽字，送中断号(确定高位)。

(2) 发启动命令(送命令字)，启动设备。

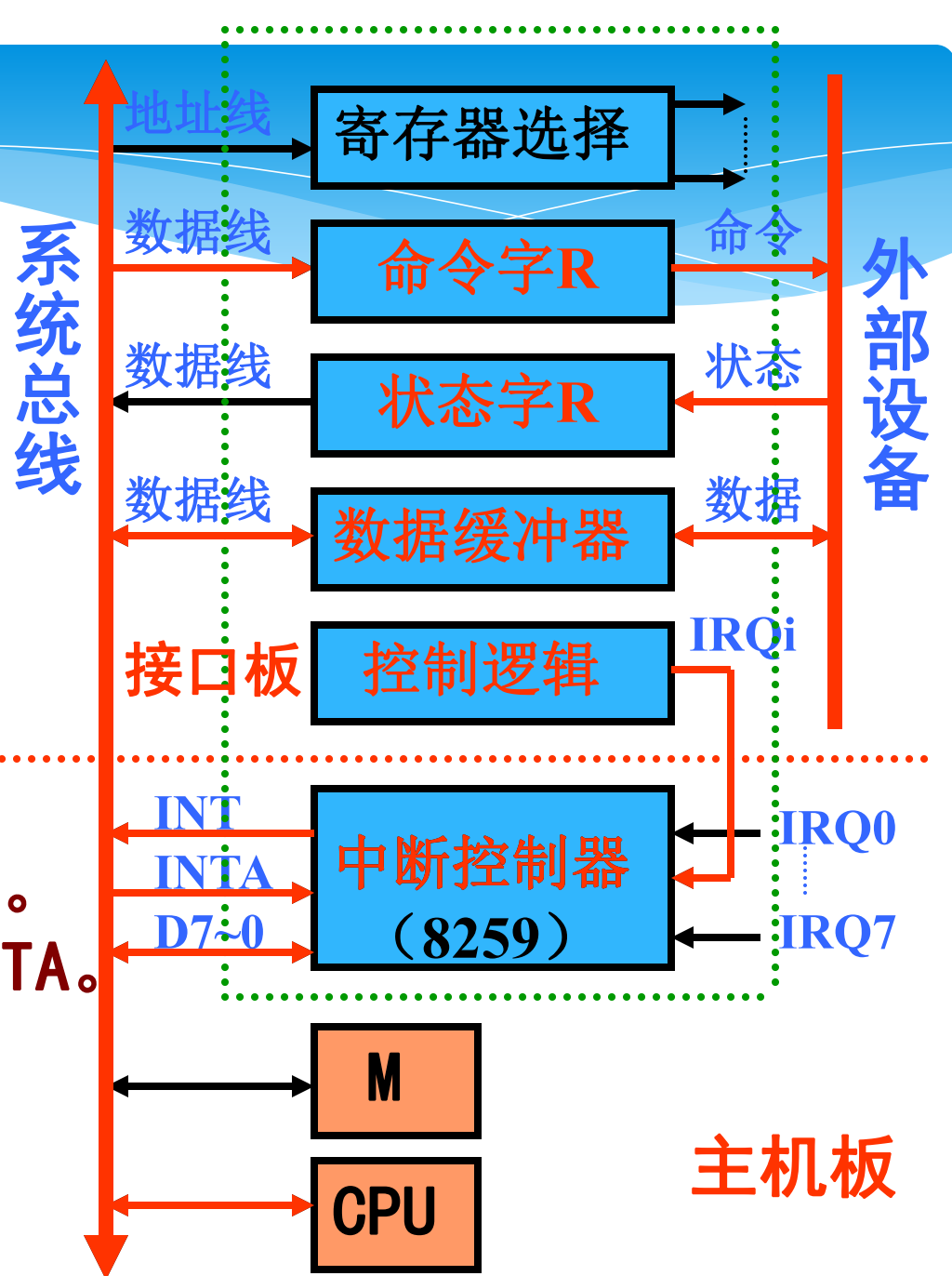
(3) 设备完成工作，申请中断。

(4) 中断控制器汇集各请求，经屏蔽、判优，形成中断号，并向CPU送INT。

(5) CPU响应，发批准INTA。

(6) 中断控制器送出中断号。

(7) CPU执行中断隐指令操作，进入服务程序。



3. 接口设计

涉及命令字、状态字格式的拟定，中断源的扩展。

例. 模型机需扩展两个外中断源，共用一个中断号。

主机发向外设的命令包括：启动、停止、数据选通；

外设的状态包括：忙、完成、出错。

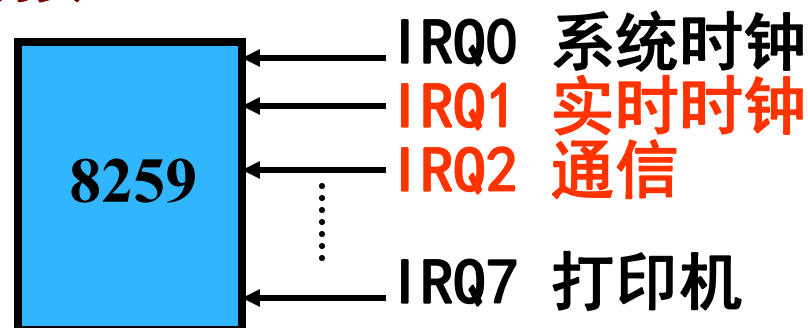
为两个扩展中断源设计中断接口。

模型机的外中断源安排：

通过IRQ2进行扩展。

(1) 接口组成

两个扩展中断源共用一个接口。



5	4	3	2	1	0
启动1	停止1	选通1	启动2	停止2	选通2

忙1	完成1	出错1	忙2	完成2	出错2
----	-----	-----	----	-----	-----

(2) 判断中断源

向量中断与非向量中断相结合 (软件扩展)

请求1与请求2在控制逻辑中形成公共请求IRQ2，送入8259参加判优；CPU响应后执行IRQ2服务程序 (向量中断过程)；

CPU在IRQ2服务程序中查询各设备状态，判中断源，转入相应设备服务程序 (非向量中断过程)。

