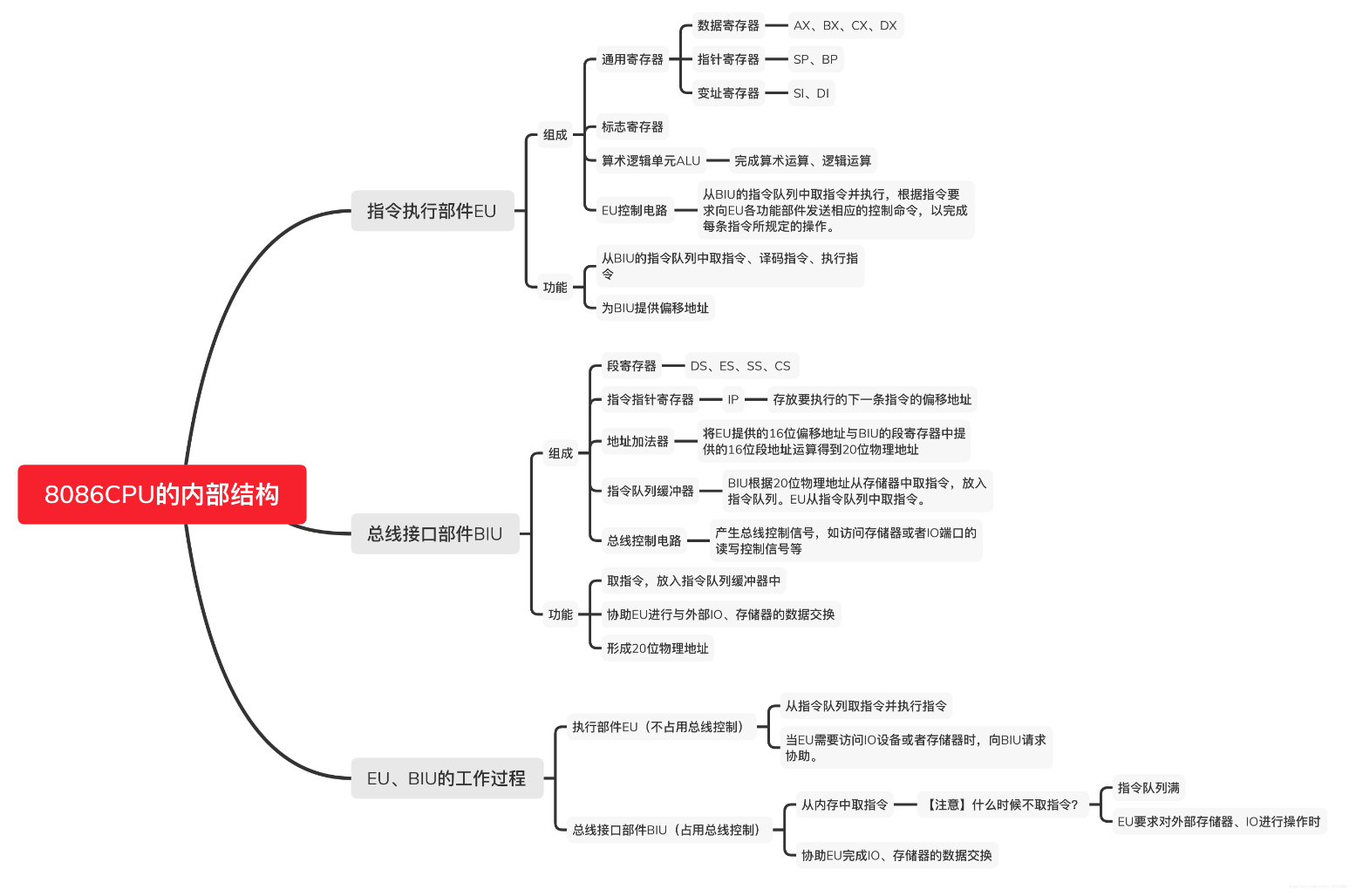
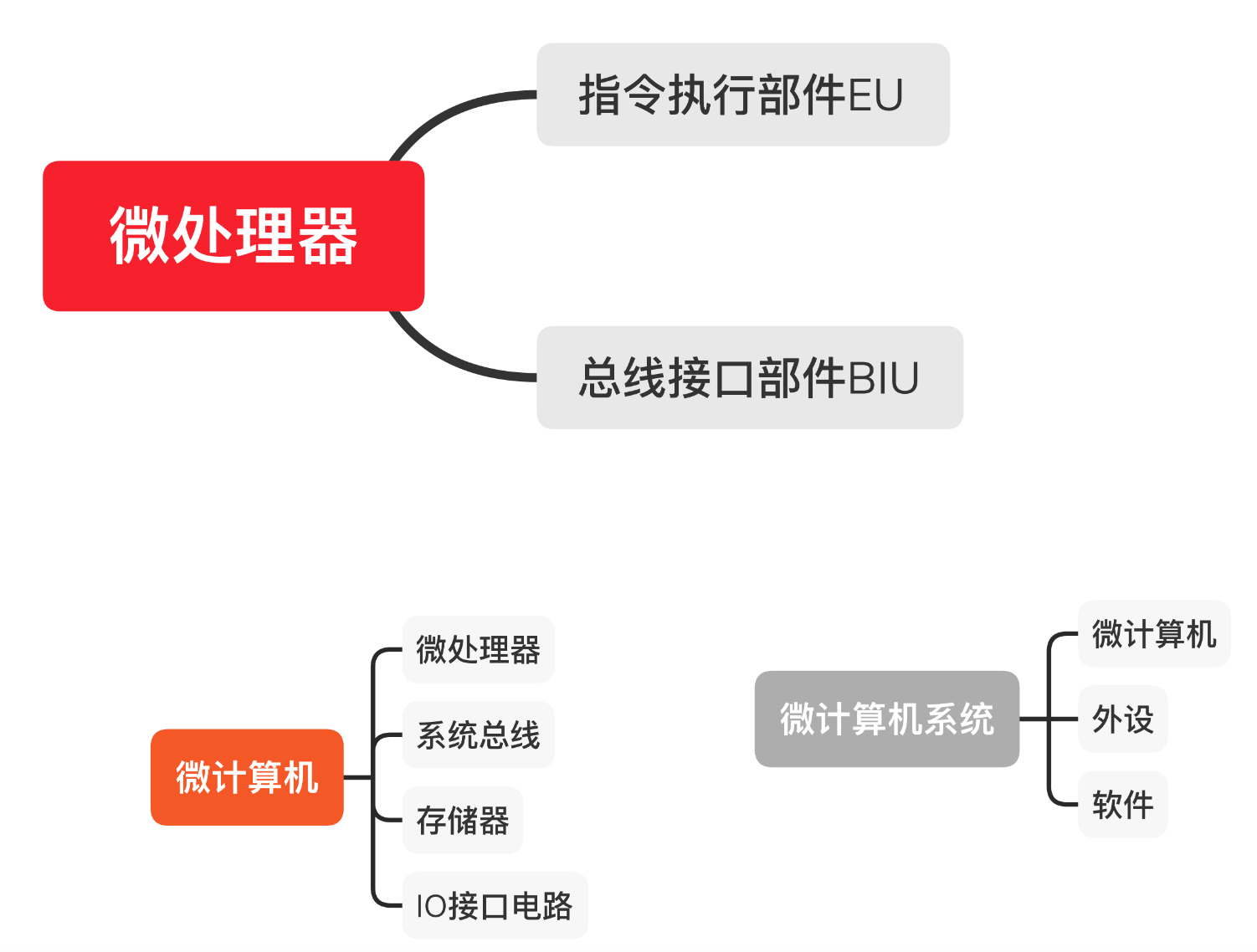
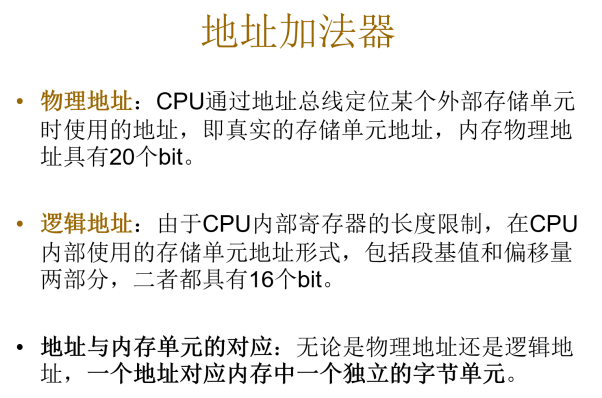
**1. 微处理器**

* EU（Execution Unit）:指令执行部件
* BIU（Bus Interface Unit）：总线接口部件





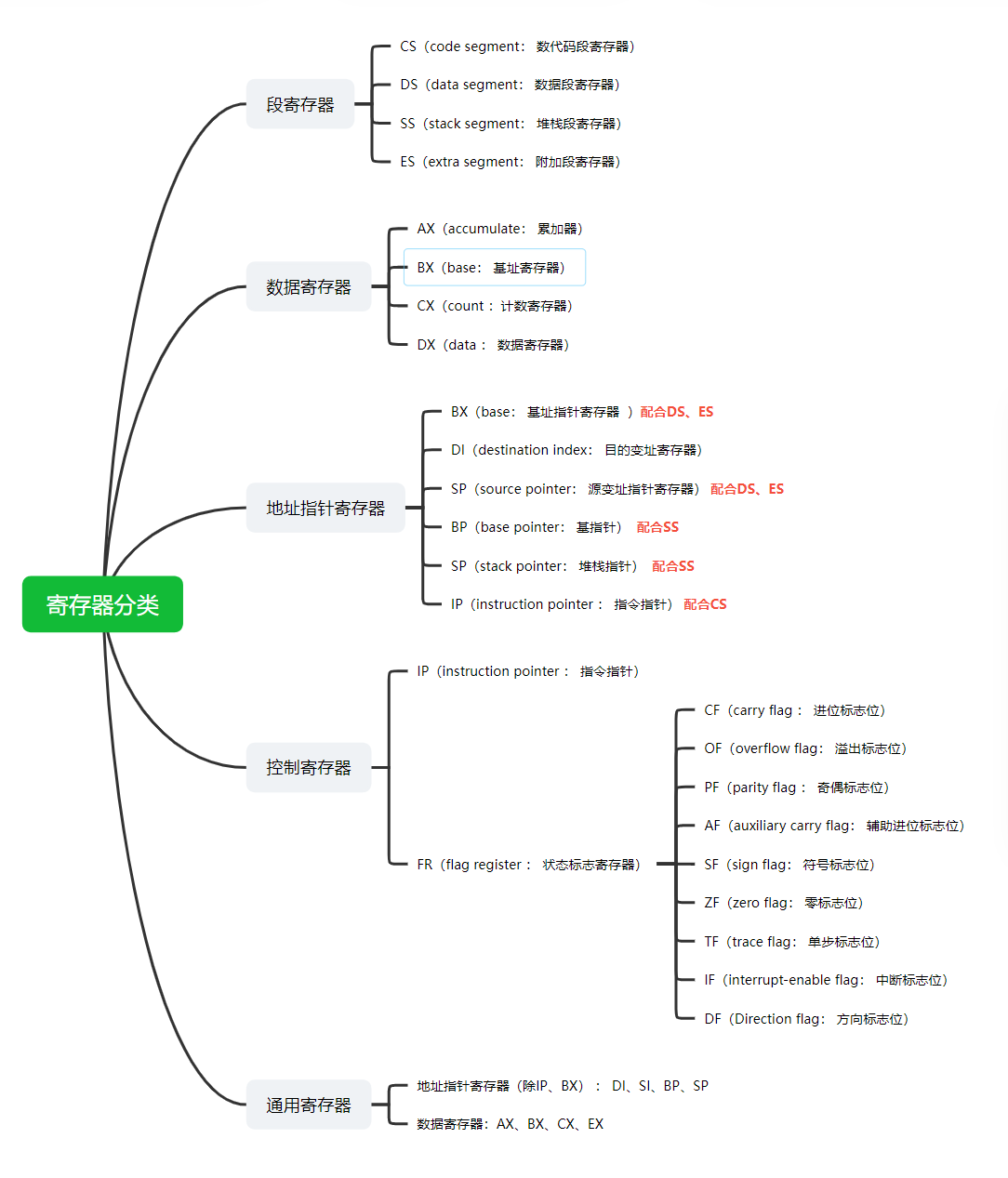
**2.地址加法器**



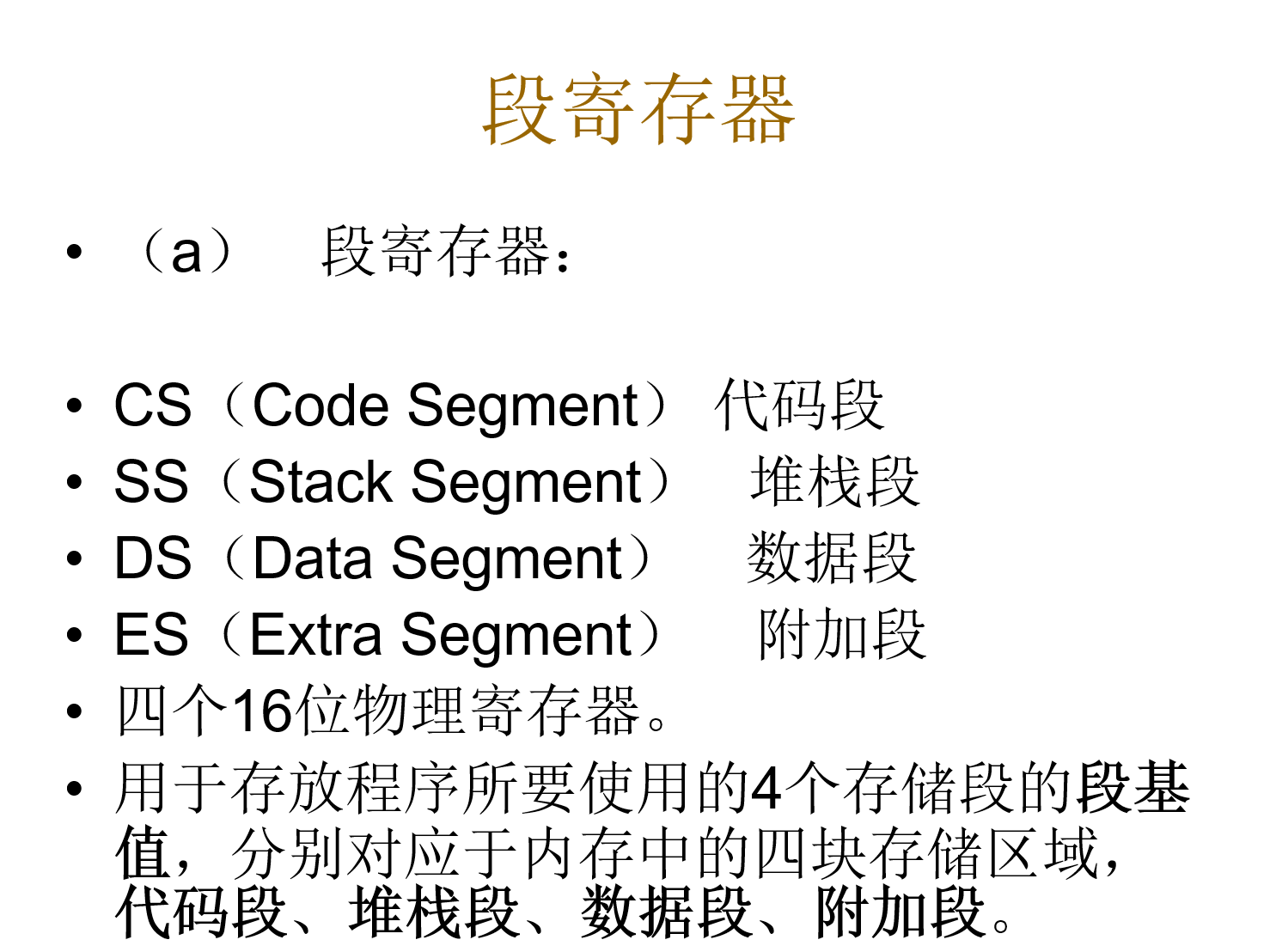
端基值 \* 16（左移4位） + 偏移量 = 物理地址（20位）

**任何一个逻辑地址可以转换为唯一的物理地址（反之不行）**

8086/8088 中的寄存器：总共有14个物理寄存器，逻辑上的寄存器有22个。



**3.段寄存器（CS、SS、DS、ES） ---16位**



* **代码段（必须的）**：用于存放程序的机器指令序列。
* **堆栈段（后进先出）**：用于存放程序使用堆栈指令所保存的数据；保存子程序调用指令提供的返回地址；保存中断断点。
* **数据段：**存放程序需要使用的数据，包括变量、数组、字符串等。
* **附加段：**根据实际需要确定。常用于存放串操作指令中的目的串。

**3.地址指针寄存器（BX、SI、DI、BP、SP、IP）--16位**



**BX（基址寄存器）**：

可以用于存放偏移量或偏移量分量。

和DS、ES这两个段寄存器配合使用

用于定位数据段或附加段中的内存单元；

**SI（源变址寄存器）：**

1. 用于存放偏移量或偏移量分量。
2. 和DS、ES这两个段寄存器配合使用
3. 用于定位数据段或附加段中的内存单元。

串操作指令中，SI指明 源串偏移量，所以被称为源变址寄存器。

**DI（目的变址寄存器）：**

用于存放偏移量或偏移量分量。

串操作指令中，DI指明 目的串偏移量，所以被称为目的变址寄存器。

**BP（基址指针寄存器）：**

1. 用于存放偏移量
2. 通常和SS段寄存器配合使用
3. 用于定位堆栈段中的内存单元。

**SP（堆栈指针）：**

1. 用于存放偏移量
2. 只能和SS段寄存器配合使用
3. 始终指向堆栈的栈顶
4. 在堆栈指令中隐含的使用它来定位栈顶数据。

**IP（指令指针）：**

1. 用于存放偏移量
2. 只能和CS段寄存器配合使用
3. 始终指向代码段中下一条将要读取到CPU指令队列的那条指令；
4. 修改内容：CPU在每读取一条指令到指令队列后自动进行的，使它指向要读取的下一条指令。
5. **转移指令可以隐含的修改IP寄存器中的内容**

**4.数据寄存器（AX、BX、CX、DX） --16位**

从逻辑上，一个数据寄存器可以看成三个寄存器。Eg.AX

1. 看作一个16位的数据寄存器AX
2. 看作一个低8位的数据寄存器AL
3. 看作一个高8位的数据寄存器AH
4. **标志寄存器（FR） --16位**

对FR的使用在指令里一般是隐含的

FR是按位操作的，每一个二进制位都有各自特定的含义

FR是实现程序分支、循环结构里必须的硬件基础

1. **状态标志位（CF、OF、PF、AF、SF、ZF）**

* CF(carry flag)： 进位标志位
* OF（overflow flag）： 溢出标志位
* PF（parity flag）： 奇偶标志位
* AF（auxiliary carry flag）：辅助进位标志位
* SF（sign flag）： 符号标志位
* ZF（zero flag）： 零标志位

**功能上：实现分支、循环程序结构的基础**

**器件上：依赖于运算器（ALU）的改变**

**CF（进位标志位）**

1. 含义：处理无符号数，标志着上次算术运算的最高位（字的第15位，字节的第7位）是否产生进位（加法指令）或借位（减法指令）

2. 值域：产生进位：CF = 1； 未产生进位 CF = 0

3. 改变条件：

* 操作指令会影响CF标志
* 用户把操作数看作**无符号数**

**提醒：不能用补码加法来实现减法运算！！！**

**在执行移位指令时，CF标志用于存放移出位的值**

**AF（辅助进位标志位）**

1. 含义：处理无符号数时，低字节的低4位是否向高4位产生进位或借位。

2. 值域： 产生： AF = 1； 不产生：AF = 0

3. 改变条件：

* 操作指令影响AF标志
* 用户把操作数看作无符号数

**SF（符号标志位） --FR的第七位**

1. 含义：判断计算结果是否为负数。

2. 值域：与结果数据的最高位保持一致 （补码的最高位就是符号位）

3. 改变条件：

* 操作指令影响SF标志
* 用户把操作数看作**有符号数**（完整编码，或包括编码最高位）

**OF（溢出标志位）**

1. 含义：处理有符号数时，用于指示运算结果是否超出了有符号数的表示范围。

2. 值域： 溢出 ： OF = 1； 未溢出 ： OF = 0

3. 改变条件：

* 操作指令影响OF标志
* 用户把操作数看作**有符号数**

**提醒：影响OF的操作数，要用补码进行运算！！！**

**PF（奇偶校验位） --位于FR的第2位**

1. 含义：判断得到的数据结果低八位中所含的1的个数的奇偶。

2. 值域： 偶数个“1”**（&全是0）：** PF = 1；奇数个“1”：PF = 0

3. 改变条件：

* 操作指令影响PF标志
* 只有低8位的数据能够影响PF

ASCII码占用一个字节，但是只有低7位是真正的码值，最高位（第7位）是校验位

如果ASCII码中有奇数个“1”，第7位取值为0；

如果ASCII码中有偶数个“1”，第7位取值为1；

**ZF（零标志位） ---位于FR的第6位**

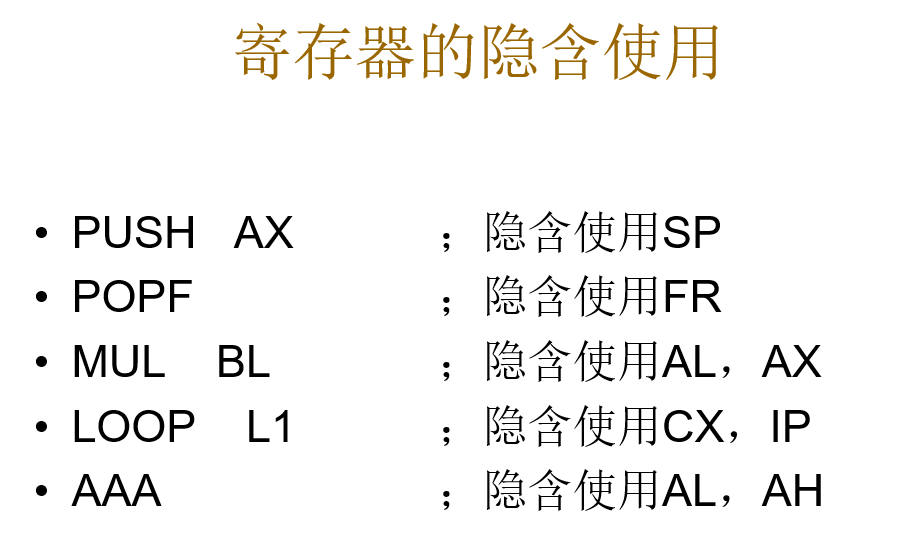
1. 含义：判断得到的数据结果是否为0

2. 值域： 是0 ： ZF = 1； 不是0 ： ZF = 0

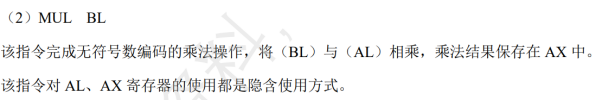
3. 改变条件：

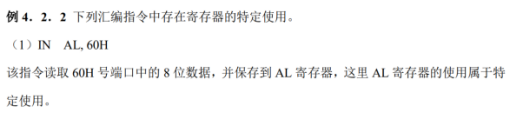
* 指令得到的结果所有位都是0

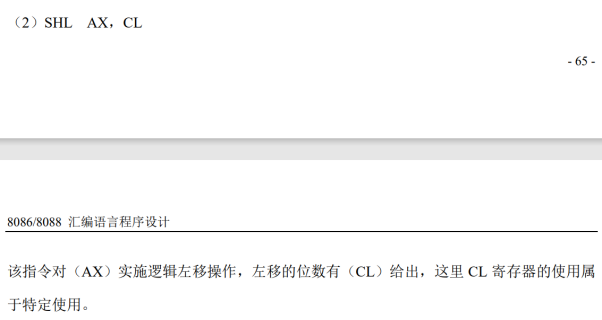
1. **寄存器的隐含使用**

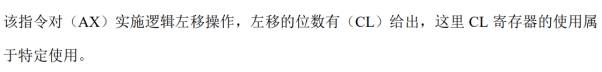












1. **寄存器的特定使用**

