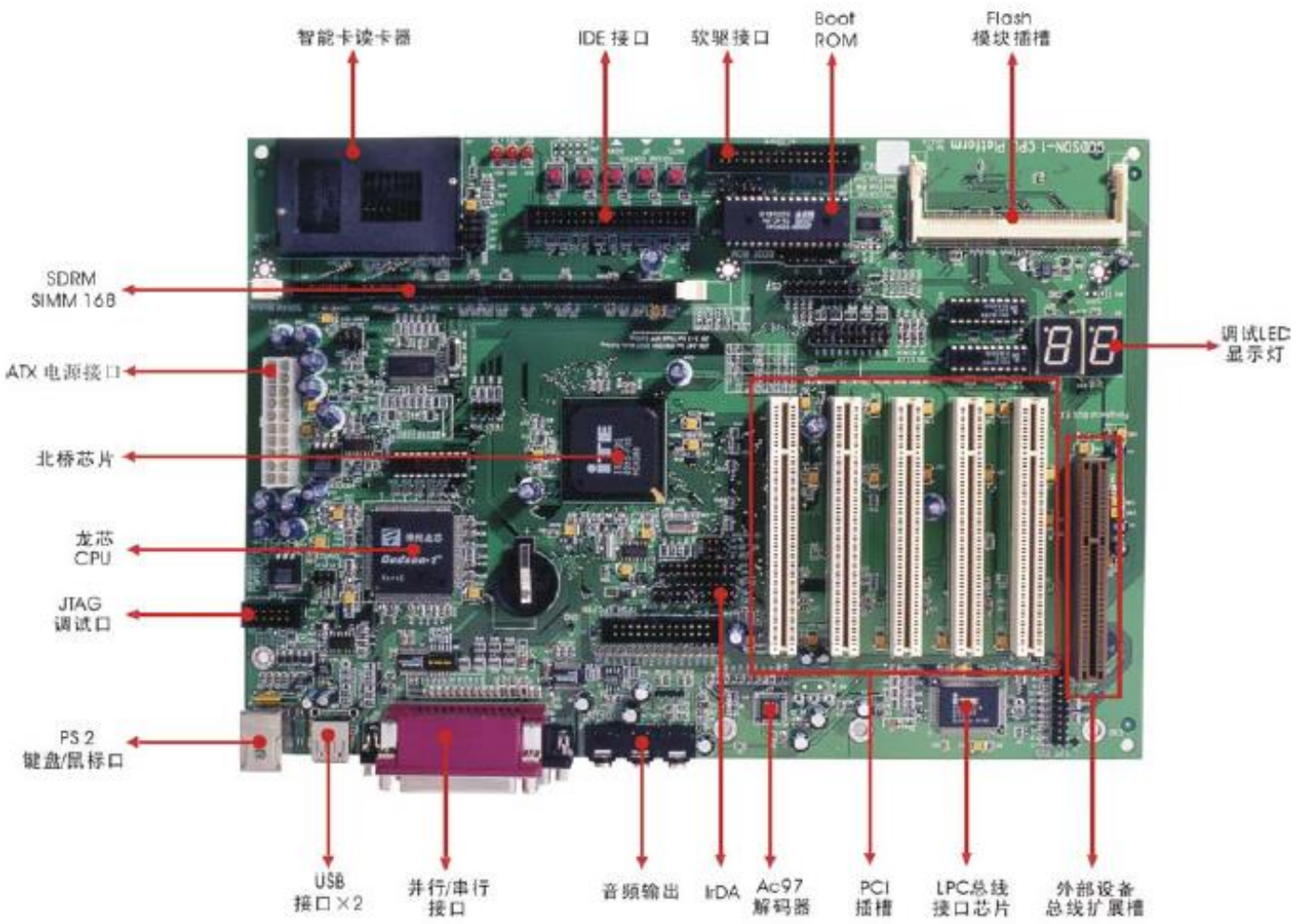
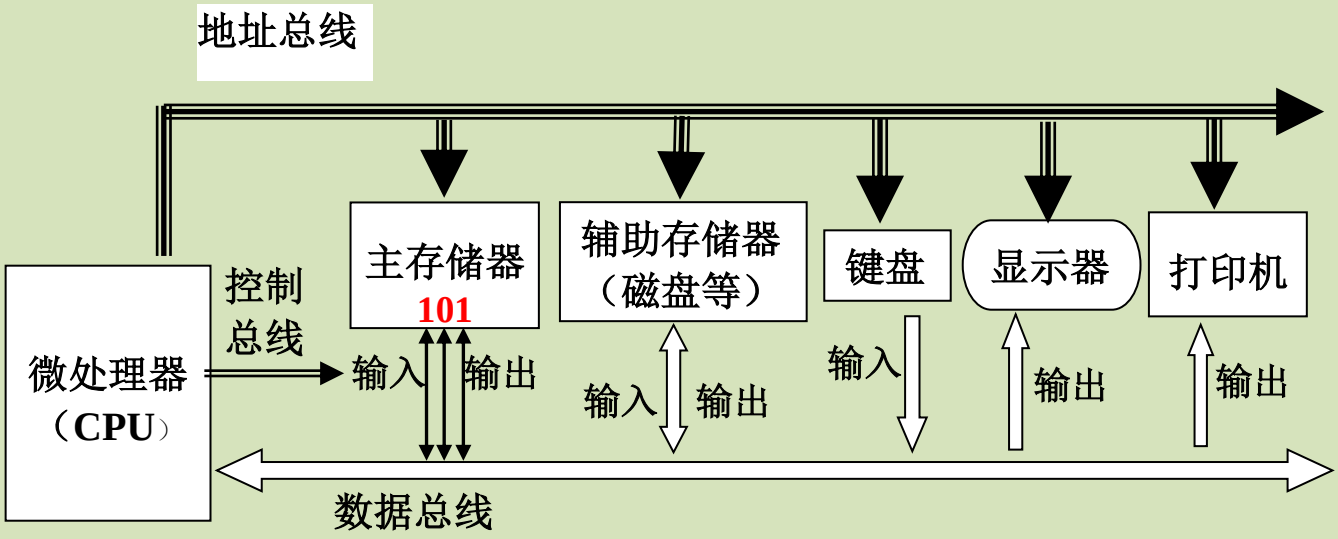
第三讲　计算机的内部结构

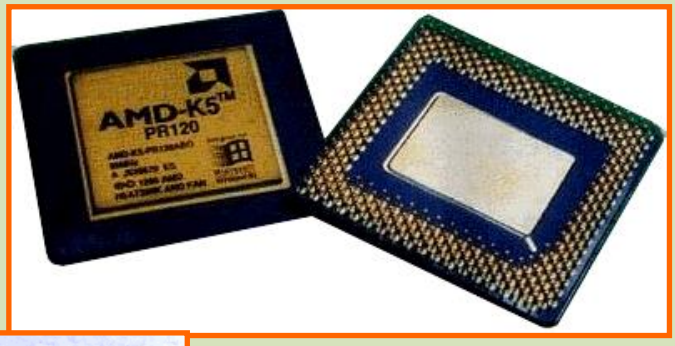


龙芯开发板布局图



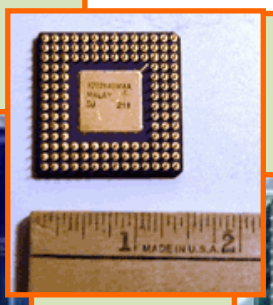
主要部件关系/总线/程序执行地点/进入主存结构/CPU 结构


C80286-6
S40172
I4370040
© INTEL '84


AMD-K5™
PR120


AMD
K6
3DNow!


AMD
K6
III
3DNOW!



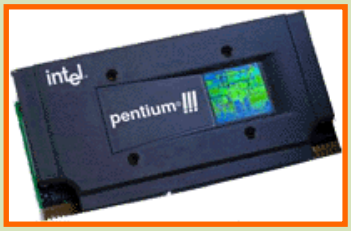

Cyrix
6x86


intel
i486™SX™

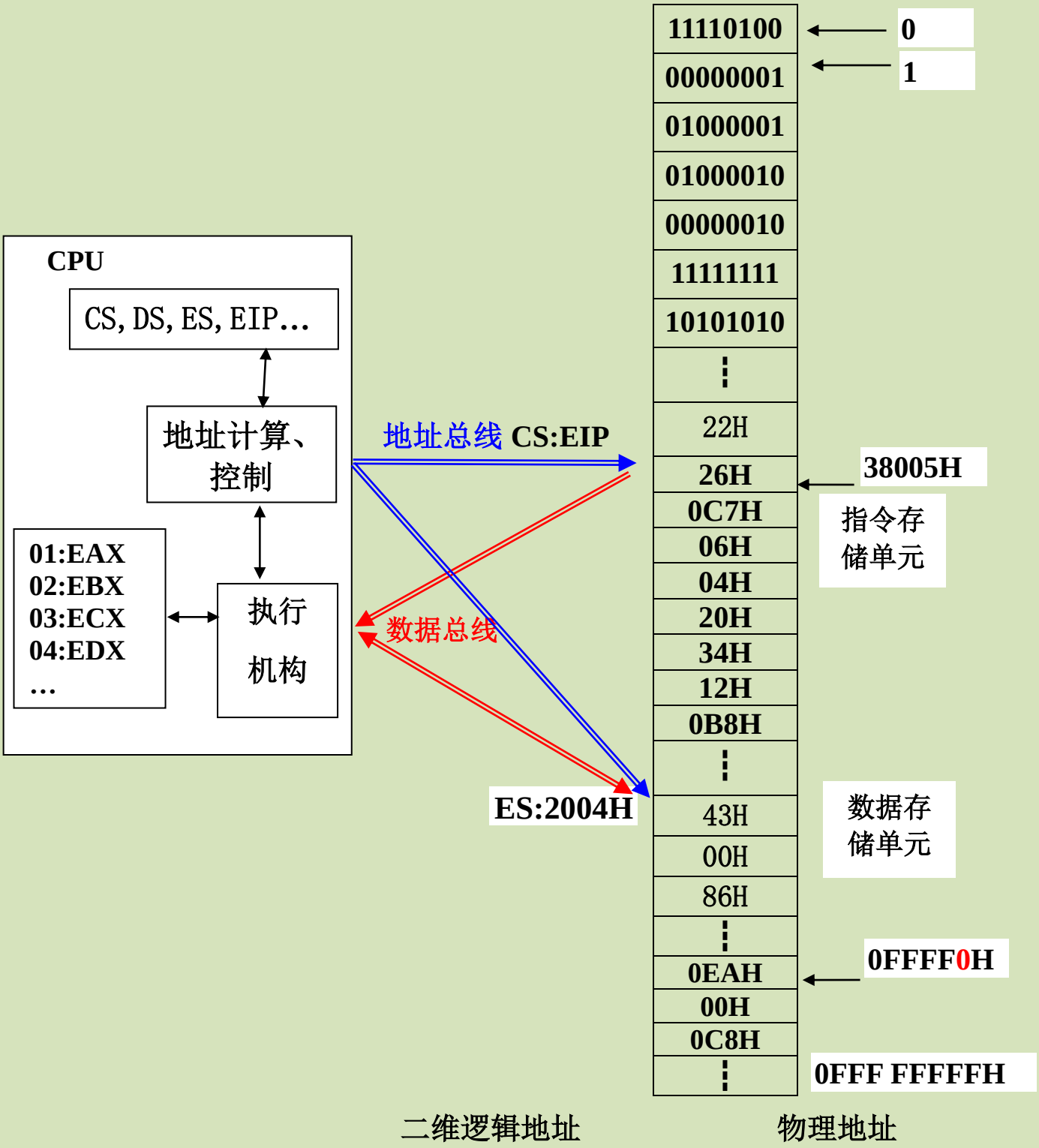

intel
Pentium
i200


AMD
Athlon
AMD


intel


intel
pentium III


pentium II
intel



作为一个机器，计算机的工作过程是不断地重复下列操作：

取命令 - 解释命令 - 执行命令 - 结果暂存 - 结果输出

CPU 内的寄存器（p6:1.2.2）

1. 通用寄存器组 (数据寄存器组、指示器和变址寄存器组)

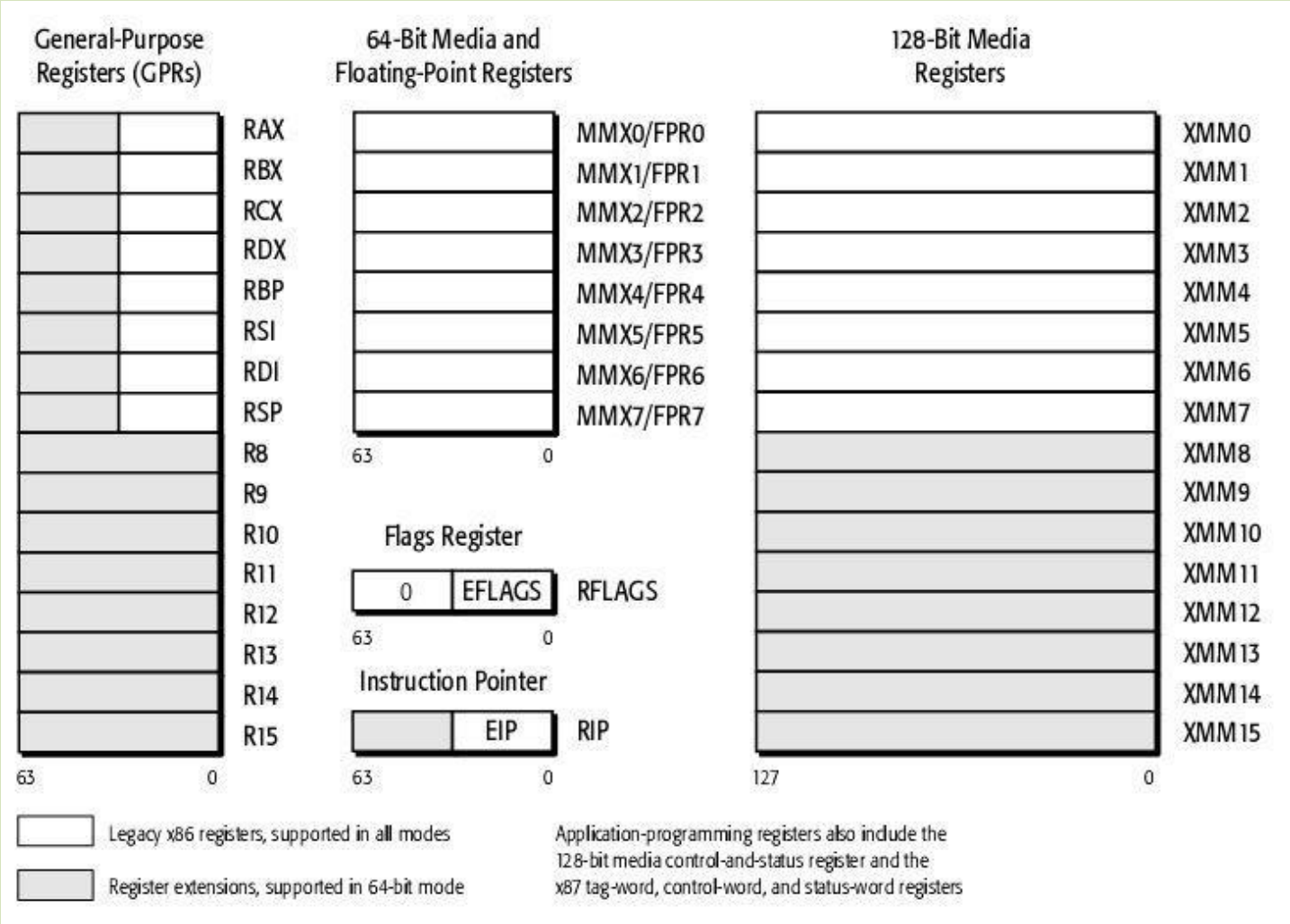
EAX		AH	AL
EBX		BH	BL
ECX		CH	CL
EDX		DH	DL
ESI		SI	
EDI		DI	
EBP		BP	
ESP		SP	
	31 16	15 8	7 0

累加器(AX) Adder/Accumulator
基址寄存器(BX) Base
计数寄存器(CX) Counter
数据寄存器(DX) Data
源变址寄存器 Source Index
目的变址寄存器 Destination Index
堆栈基址寄存器 Base pointer
堆栈指示器 Stack Pointer

（8086/80386 选用的寄存器组）

X64 多了 8 个通用寄存器：R8、R9、R10、R11、R12、R13、R14、R15，当然，它们都是 64 位的。另外还增加了 8 个 128 位 XMM 寄存器，不过通常用不着。

X32 中原有的寄存器在 X64 中均为扩展为 64 位，且名称的第一个字母从 E 改为 R。不过我们还是可以在 64 位程序中调用 32 位的寄存器，如 RAX（64 位）、EAX（低 32）、AX（低 16 位）、AL（低 8 位）、AH（8 到 15 位），相应的有 R8、R8D、R8W 和 R8B。



2. 段寄存器（总是 16 位）：指向主存储器首地址

CS 寄存器指示当前代码段，它指出了当前可访问代码段所在的存贮首地址；

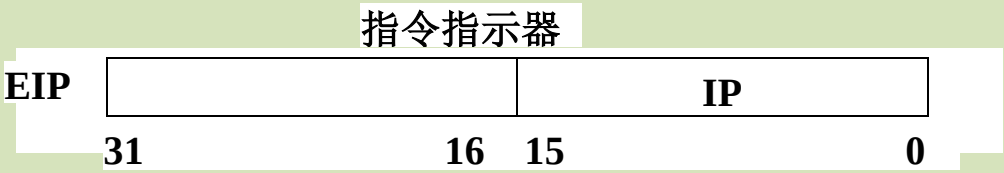
SS 寄存器指示当前堆栈段，它指出了当前可访问堆栈段所在的存贮首地址；

SS：SP　　SS：ESP

DS 寄存器指示当前数据段，它指出了当前可访问数据段所在的存贮首地址；

ES、FS、GS 寄存器指示当前附加数据段，它指出了当前可访问附加数据段所在的存贮首地址。

3. 指令指示器 IP（Instruction Pointer）/EIP：当程序正常执行时，IP/EIP 包含正在执行指令的下一条指令到代码段首址的偏移地址。



CS：IP　　16 位偏移

CS：EIP　　32 位偏移

4. 特殊/专用寄存器

控制 CPU 工作状态的 32 位寄存器 CR（多个）（CR0 最低位工作方式，=0 实方式；最高位是否分页，0=不分页）；

设置断点地址来调试程序的 32 位寄存器 DR（多个）；

系统地址寄存器：GDTR、LDTR、IDTR、TR（任务状态段寄存器）

5.标志寄存器(p23:1.5)：用来保存 CPU 执行完一指令后所处的状态信息。

FLAGS / EFLAGS

MOV AX，FLAGS

划分为条件标志、控制标志。

31	…	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							OF	DF	IF	TF	SF	ZF		AF		PF		CF
							溢出	方向	中断允许	跟踪	符号	零		辅助进位		奇偶		进位

条件标志：反映算术、逻辑运算指令执行后所得结果的特征。包括：

SF（操作结果的最高位=1 时为负数，将 SF 置为 1）；

ZF（操作结果为 0，将 ZF 置为 1）；

OF（操作结果有溢出错误，将 OF 置为 1）；

CF（操作中有进位或借位，将 CF 置为 1）；

AF、PF。

注意理解：（1） “是 则 1”，如：结果是零则 ZF =1。

（2）只观察当前执行指令中操作对象的信息，如最高位的相对性。

ADD AX，8

ADD AL，8

（3）注意溢出和进位的关系。

溢出：结果出错（无进位或有进位）。实际是有符号数的符号位反转。

如：01000000 B （64）

+ 01000000 B （64）

10000000 B （-128） （无进位，但有溢出）

10000000 B （-128）

+ 10000000 B （-128）

100000000 B （0） （有进位、有溢出）

进位：结果无错

如：01000000B （64）

+ 11000001B （-63）

1 00000001B （1） （有进位、无溢出）

原因：CPU 字长有限

控制标志：控制 CPU 工作状态。DF、IF、TF、。。。。。

标志寄存器有关操作

PUSHF 低 16 位 FLAGS 进栈

PUSHFD 32 位 EFLAGS 进栈

POPF

POPFD

LAHF （EFLAGS）$_{7\text{-}0}$ → AH

例（EFLAGS）= 00003487H, (AX)=0FFFFH

执行 LAHF 后,

（EFLAGS）= 00003487H, (AX)=0387FFH

SAHF （AH）→EFLAGS$_{7\text{-}0}$

STC

1 → CF

CLC

0 → CF

条件转移指令：(p103)

格式：JX 标号

功能：如果转移条件满足，则

标号的 EA（数值） → IP/EIP

否则，执行紧跟转移指令之后的那条指令。

例如：

```
    SUB   AX, BX
    JZ    NEXT
    INC   AX
NEXT:
```

NEXT：DEC　CX

┇

简单转移指令:根据单个标志位确定转移的条件。

标志位共有五个：CF=｛$^{0}_{1}$、OF=｛$^{0}_{1}$、SF=｛$^{0}_{1}$、ZF=｛$^{0}_{1}$、PF=｛$^{0}_{1}$，共十条。

- JC　（CF=1 转）有进位转
- JNC　（CF=0 转）无进位转

- JZ/JE　（ZF=1 转）结果等于 0 转
- JNZ/JNE（ZF=0 转）结果不等于 0 转

- JS　（SF=1 转）结果为负转
- JNS　（SF=0 转）结果为正转

- J0　（OF=1 转）有溢出转
- JN0　（OF=0 转）无溢出转

- JP/JPE　（PF=1 转）结果中 1 的个数为偶转移
- JNP/JPO　（PF=0 转）结果中 1 的个数为奇转移

特点：　只能是段内直接跳转，即：

（1）用立即数改变 IP、EIP 的值，不改变 CS。

（2）16 位段：最大转移距离 -32768～32767；

32 位段：最大转移距离-2G～2G。

}