IA-32指令系统概述

南京大学
计算机科学与技术系
袁春风
email: cfyuan@nju.edu.cn
2015.6

Intel处理器

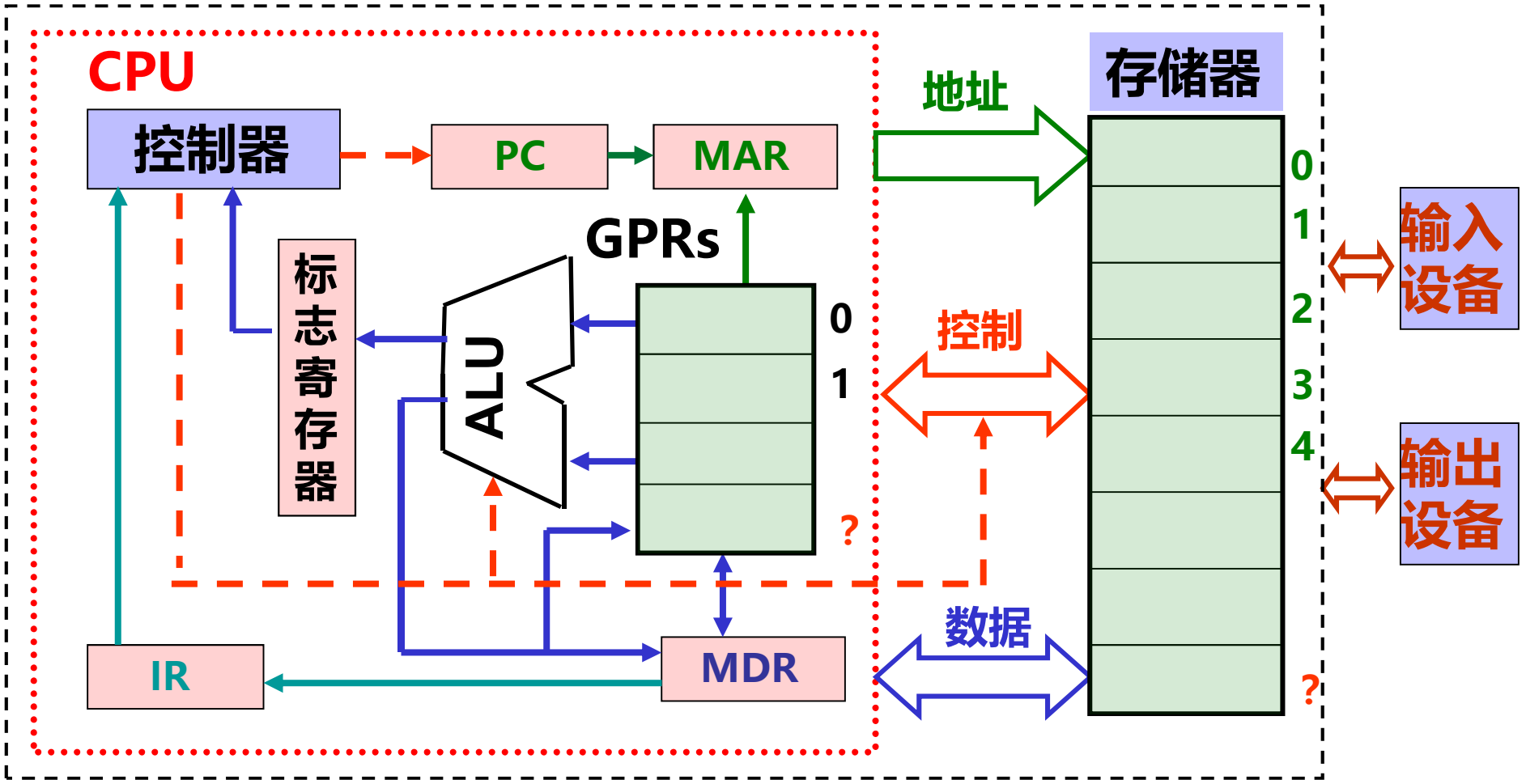
x86前产品	4004 • 4040 • 8008 • 8080 • iAPX 432 • 8085	
x87（外置浮点运算器）	8/16位总线: 8087 16位总线: 80187 • 80287 • 80387SX 32位总线: 80387DX • 80487	已停产
x86-16（16位）	8086 • 8088 • 80186 • 80188 • 80286	
x86-32/IA-32（32位）	80386 • 80486 • Pentium（OverDrive、Pro、II、III、4、M） • Celeron（M、D） • Core	
x86-64/Intel 64（64位）	Pentium（4（部份型号）、Pentium D、EE） • Celeron D（部份型号） • Core 2	
EPIC/IA-64（64位）	Itanium	
RISC	i860 • i960 • StrongARM • XScale	
微控制器	8048 • 8051 • MCS-96	
x86-32/IA-32	EP80579 • A100 • Atom（CE、SoC）	现有产品
x86-64/Intel 64	Xeon（E3、E5、E7、Phi） • Atom（部分型号） • Celeron • Pentium • Core（i3、i5、i7）	
EPIC/IA-64	Itanium 2	

IA-32/x64指令系统概述

- x86是Intel开发的一类处理器体系结构的泛称
  - 包括 Intel 8086、80286、i386和i486等，因此其架构被称为“x86”
  - 由于数字并不能作为注册商标，因此，后来使用了可注册的名称，如Pentium、PentiumPro、Core 2、Core i7等
  - 现在Intel把32位x86架构的名称x86-32改称为IA-32
  - IA是Intel Architecture的缩写
- 由AMD首先提出了一个兼容IA-32指令集的64位版本
  - 扩充了指令及寄存器长度和个数等，更新了参数传送方式
  - AMD称其为AMD64，Intel称其为Intl64（不同于IA-64）
  - 命名为“x86-64”，有时也简称为x64

IA-32的体系结构是怎样的呢？

寄存器个数及各自功能？寄存器宽度？存储空间大小？编址单位？
指令格式？指令条数？指令操作功能？寻址方式？数据类型？
小端/大端？标志寄存器各位含义？PC位数？I/O端口编址方式？……



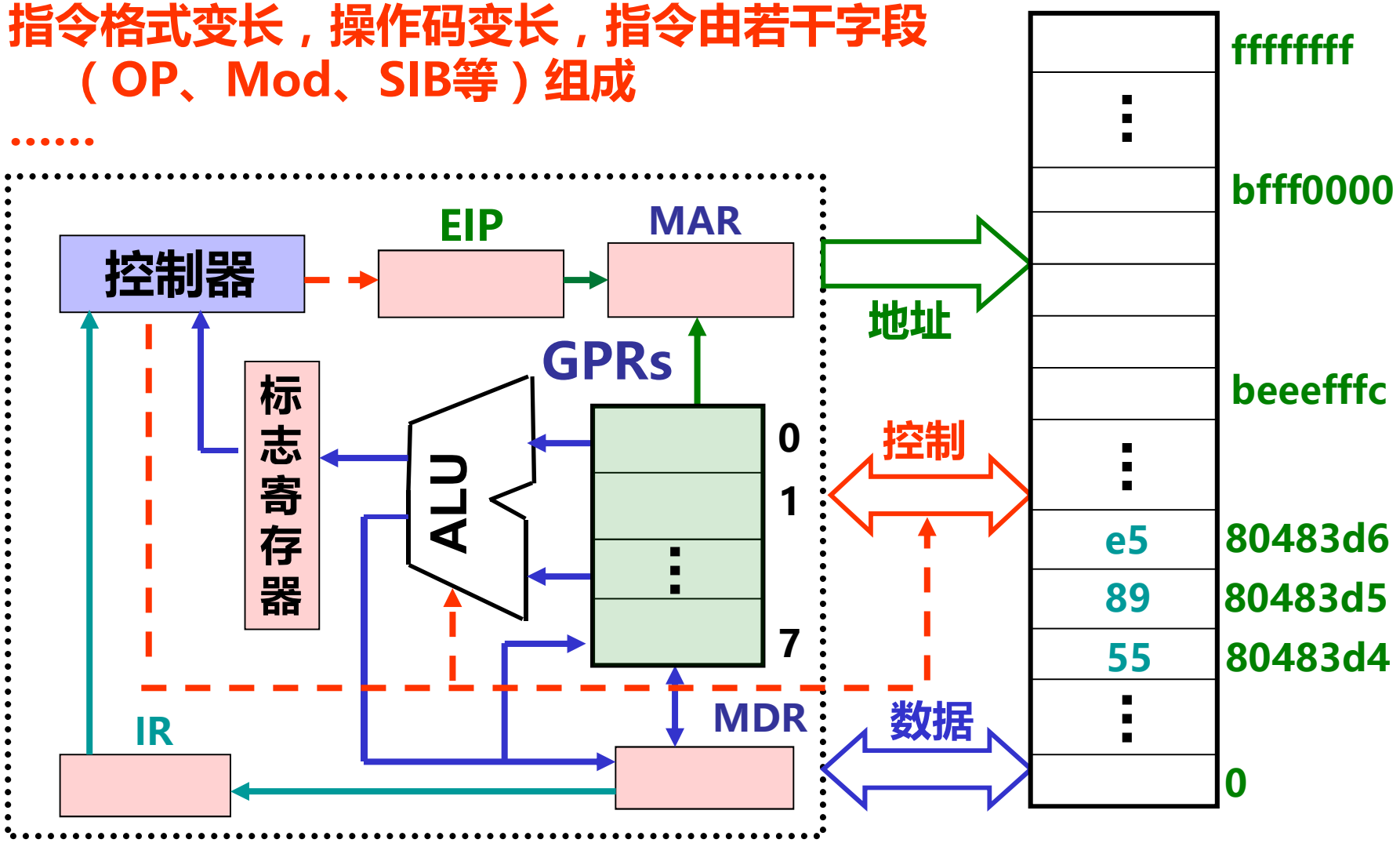
IA-32的体系结构是怎样的呢？

8个GPR（0~7），一个EFLAGs，PC为EIP

可寻址空间4GB（编号为0~0xFFFFFFFF）

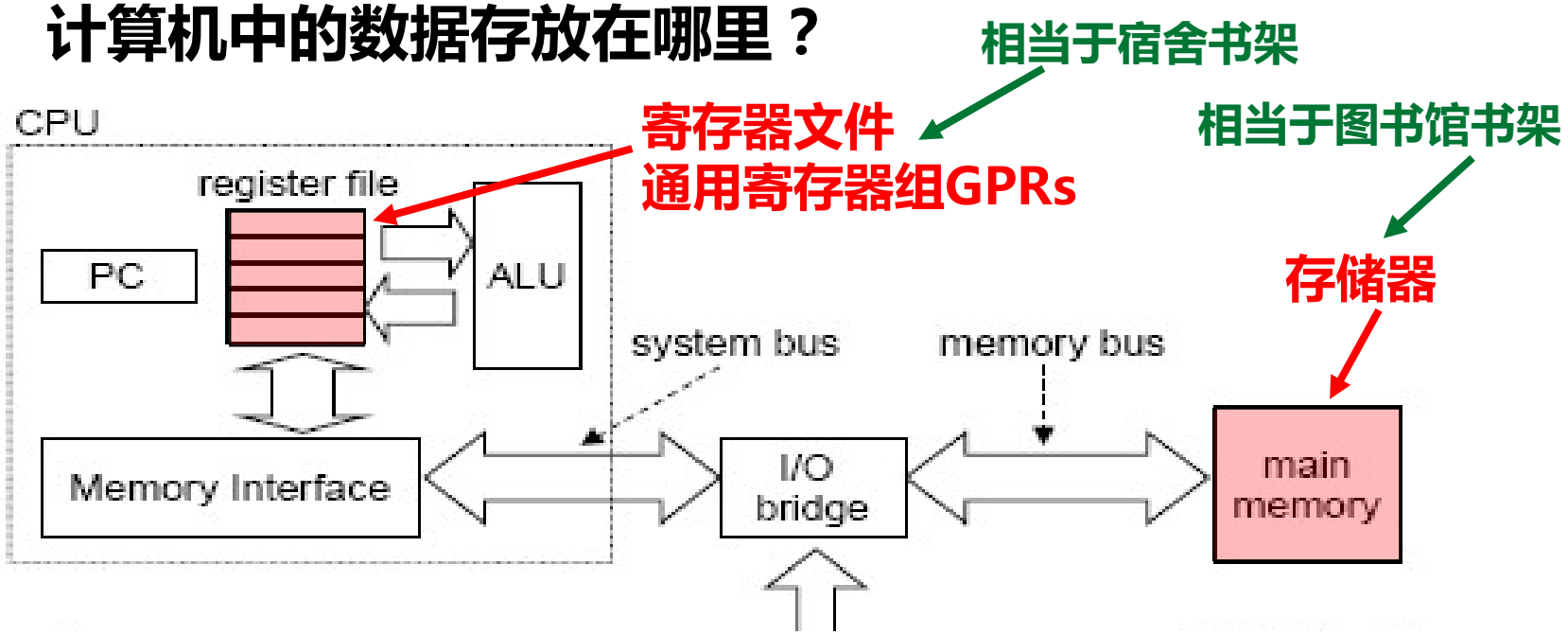
指令格式变长，操作码变长，指令由若干字段（OP、Mod、SIB等）组成

……



计算机中数据的存储

- **计算机中的数据存放在哪里？**



指令中需给出的信息：

操作性质（操作码）

源操作数1 或/和 源操作数2　（立即数、寄存器编号、存储地址）

目的操作数地址　（寄存器编号、存储地址）

存储地址的描述与操作数的数据结构有关！

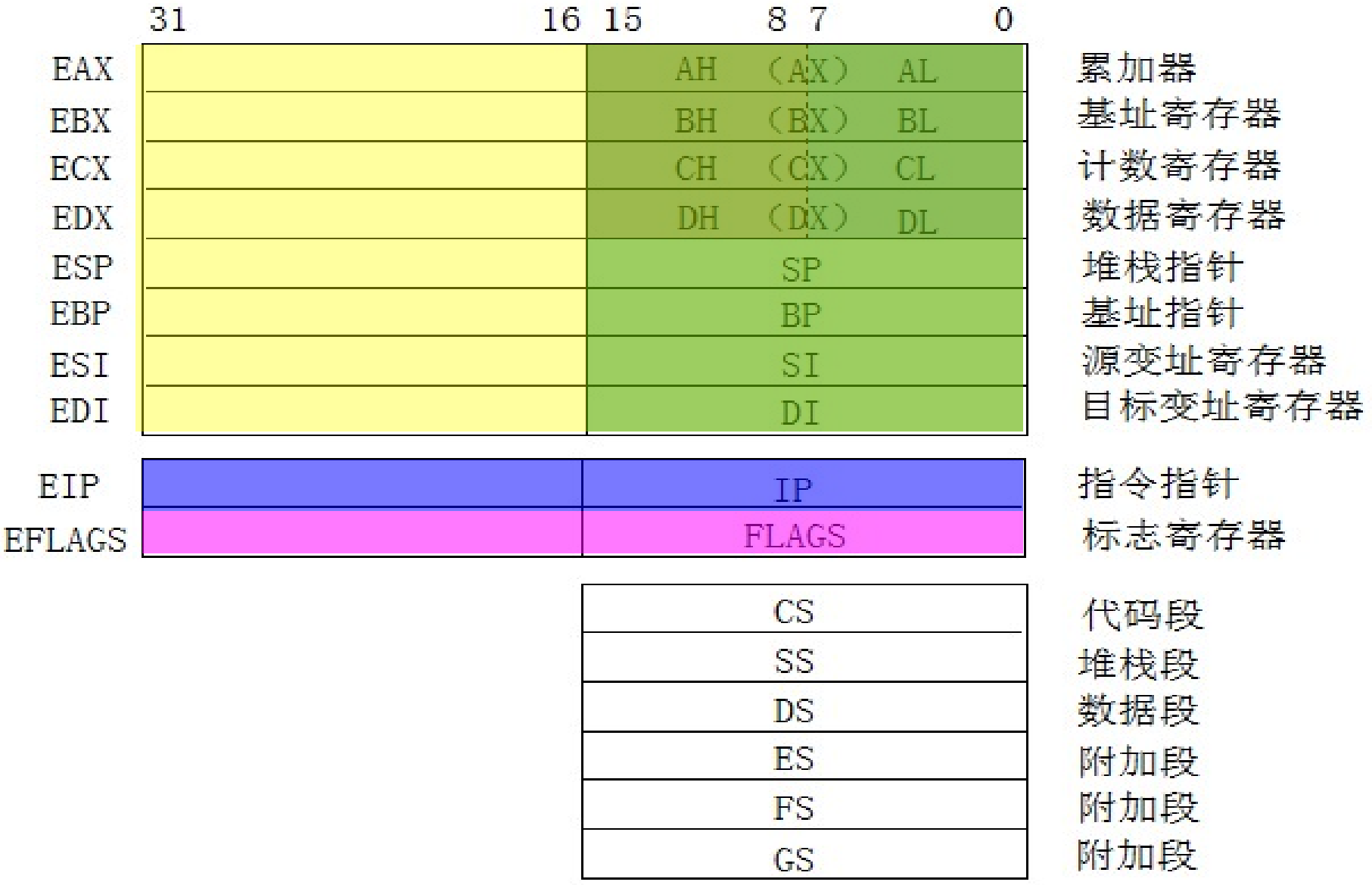
IA-32支持的数据类型及格式

C 语言声明	Intel 操作数类型	汇编指令长度后缀	存储长度（位）
(unsigned) char	整数 / 字节	b	8
(unsigned) short	整数 / 字	w	16
(unsigned) int	整数 / 双字	l	32
(unsigned) long int	整数 / 双字	l	32
unsigned) long long int	–	–	2×32
char *	整数 / 双字	l	32
float	单精度浮点数	s	32
double	双精度浮点数	l	64
long double	扩展精度浮点数	t	80 / 96

IA-32架构由16位架构发展而来，因此，虽然字长为32位或更大，但一个字为16位，长度后缀为 w；32位为双字，长度后缀为 l

long double实际长度为80位，但分配96位=12B（按4B对齐）

IA-32的寄存器组织


31
16
15
8
7
0
EAX
AH
（AX）
AL
累加器
EBX
BH
（BX）
BL
基址寄存器
ECX
CH
（CX）
CL
计数寄存器
EDX
DH
（DX）
DL
数据寄存器
ESP
SP
堆栈指针
EBP
BP
基址指针
ESI
SI
源变址寄存器
EDI
DI
目标变址寄存器
EIP
IP
指令指针
EFLAGS
FLAGS
标志寄存器
CS
代码段
SS
堆栈段
DS
数据段
ES
附加段
FS
附加段
GS
附加段

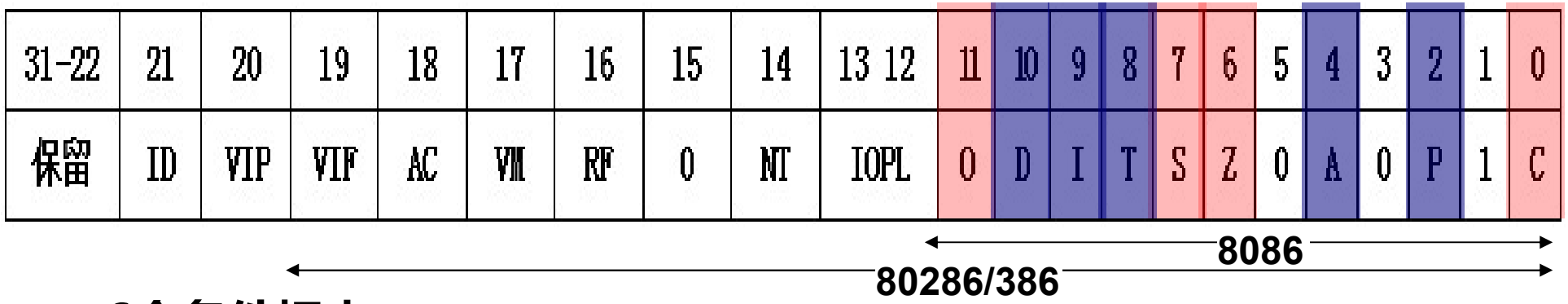
IA-32的寄存器组织

编号	8 位寄存器	16 位寄存器	32 位寄存器	64 位寄存器	128 位寄存器
000	AL	AX	EAX	MM0 / ST(0)	XMM0
001	CL	CX	ECX	MM1 / ST(1)	XMM1
010	DL	DX	EDX	MM2 / ST(2)	XMM2
011	BL	BX	EBX	MM3 / ST(3)	XMM3
100	AH	SP	ESP	MM4 / ST(4)	XMM4
101	CH	BP	EBP	MM5 / ST(5)	XMM5
110	DH	SI	ESI	MM6 / ST(6)	XMM6
111	BH	DI	EDI	MM7 / ST(7)	XMM7

反映了体系结构发展的轨迹，字长不断扩充，指令保持兼容

ST(0) ~ ST(7)是80位，MM0 ~MM7使用其低64位

IA-32的标志寄存器



- 6个条件标志
  - OF、SF、ZF、CF各是什么标志（条件码）？
  - AF：辅助进位标志（BCD码运算时才有意义）
  - PF：奇偶标志

SKIP

- 3个控制标志
  - DF（Direction Flag）：方向标志（自动变址方向是增还是减）
  - IF（Interrupt Flag）：中断允许标志（仅对外部可屏蔽中断有用）
  - TF（Trap Flag）：陷阱标志（是否是单步跟踪状态）
- ……

回顾：计算机中的算盘长啥样？

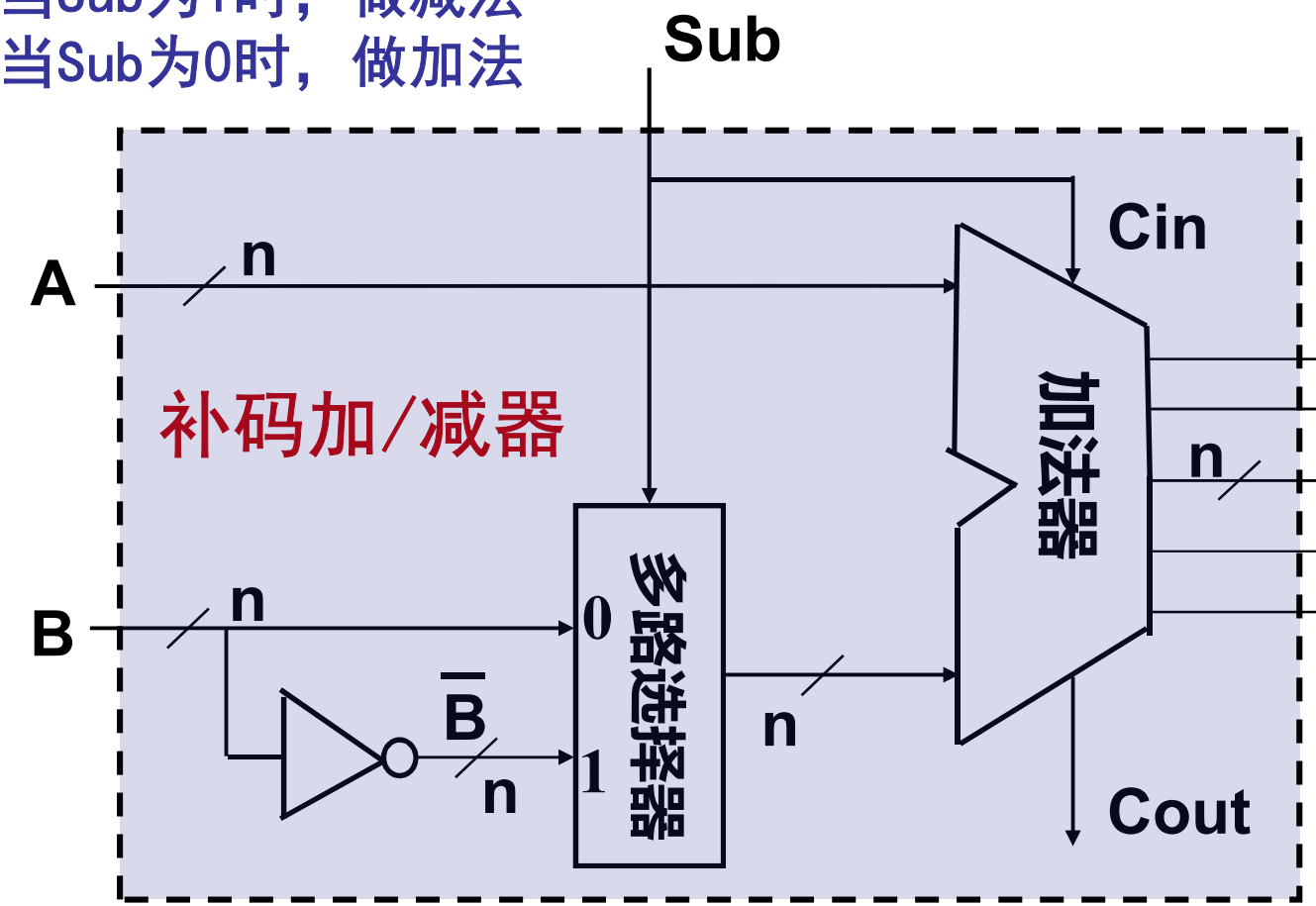
重要认识1：计算机中所有运算都基于加法器实现！

计算机中的算盘就是加法器！

重要认识2：加法器不知道所运算的是带符号数还是无符号数。

重要认识3：加法器不判定对错，总是取低n位作为结果，并生成标志信息。

当Sub为1时，做减法
当Sub为0时，做加法



各个标志如何生成呢？

零标志

符号标志

进/借位标志

BACK

IA-32的寻址方式

- 寻址方式
  - 如何根据指令给定信息得到操作数或操作数地址
- 操作数所在的位置
  - 指令中：立即寻址
  - 寄存器中：寄存器寻址
  - 存储单元中（属于存储器操作数，按字节编址）：其他寻址方式
- 存储器操作数的寻址方式与微处理器的工作模式有关
  - 两种工作模式：实地址模式和保护模式
- 实地址模式（基本用不到）
  - 为与8086/8088兼容而设，加电或复位时
  - 寻址空间为1MB，20位地址：(CS)<<4+(IP)
- 保护模式（需要掌握）
  - 加电后进入，采用虚拟存储管理，多任务情况下隔离、保护
  - 80286以上微处理器的工作模式
  - 寻址空间为2^{32}B，32位线性地址分段（段基址+段内偏移量）

保护模式下的寻址方式

寻址方式	说明
立即寻址	指令直接给出操作数
寄存器寻址	指定的寄存器R的内容为操作数
位移	LA=（SR）+A
基址寻址	LA=（SR）+（B）
基址加位移	LA=（SR）+（B）+A
比例变址加位移	LA=（SR）+(I)×S + A
基址加变址加位移	LA=（SR）+（B）+（I）+A
基址加比例变址加位移	LA=（SR）+（B）+(I)×S + A
相对寻址	LA=(PC)+A

存储器操作数

跳转目标指令地址

注：LA:线性地址　(X):X的内容　SR:段寄存器　PC:程序计数器　R:寄存器
A:指令中给定地址段的位移量　B:基址寄存器　I:变址寄存器　S:比例系数

- SR段寄存器（间接）确定操作数所在段的段基址
- 有效地址给出操作数在所在段的偏移地址
- 寻址过程涉及到“分段虚拟管理方式”，将在第6章讨论

SKIP

IA-32的寄存器组织

返回寻址方式

存储器操作数的寻址方式

```
int x;
float a[100];
short b[4][4];
char c;
double d[10];
```

Linux系统：
double型变量
按4B边界对齐

windows系统：
double型变量
按8B边界对齐

a[i]的地址如何计算？

104+i×4

i=99时，104+99×4=500

b[i][j]的地址如何计算？

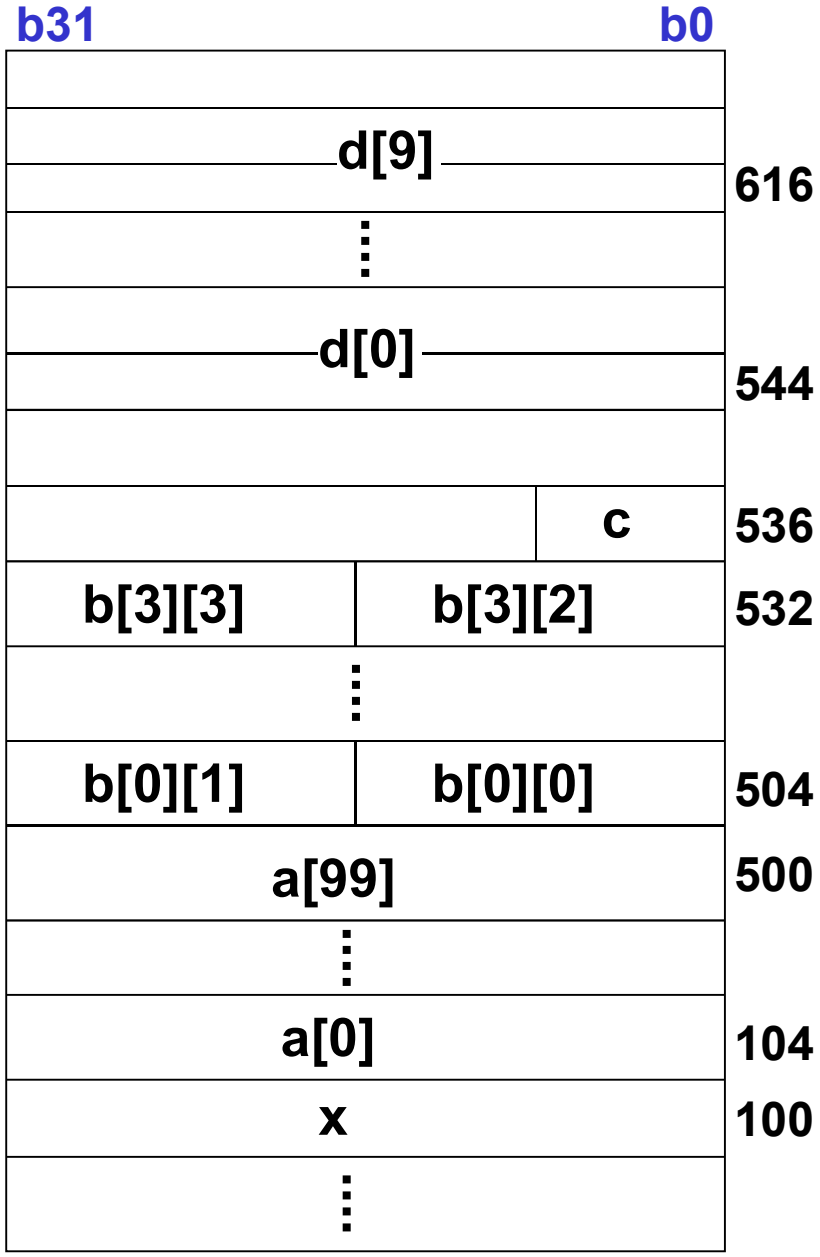
504+i×8+j×2

i=3、j=2时，504+24+4=532

d[i]的地址如何计算？

544+i×8

i=9时，544+9×8=616



存储器操作数的寻址方式

```
int x;
float a[100];
short b[4][4];
char c;
double d[10];
```

各变量应采用什么寻址方式？

x、c：位移 / 基址

a[i]：104+i×4，比例变址+位移

d[i]：544+i×8，比例变址+位移

b[i][j]： 504+i×8+j×2，

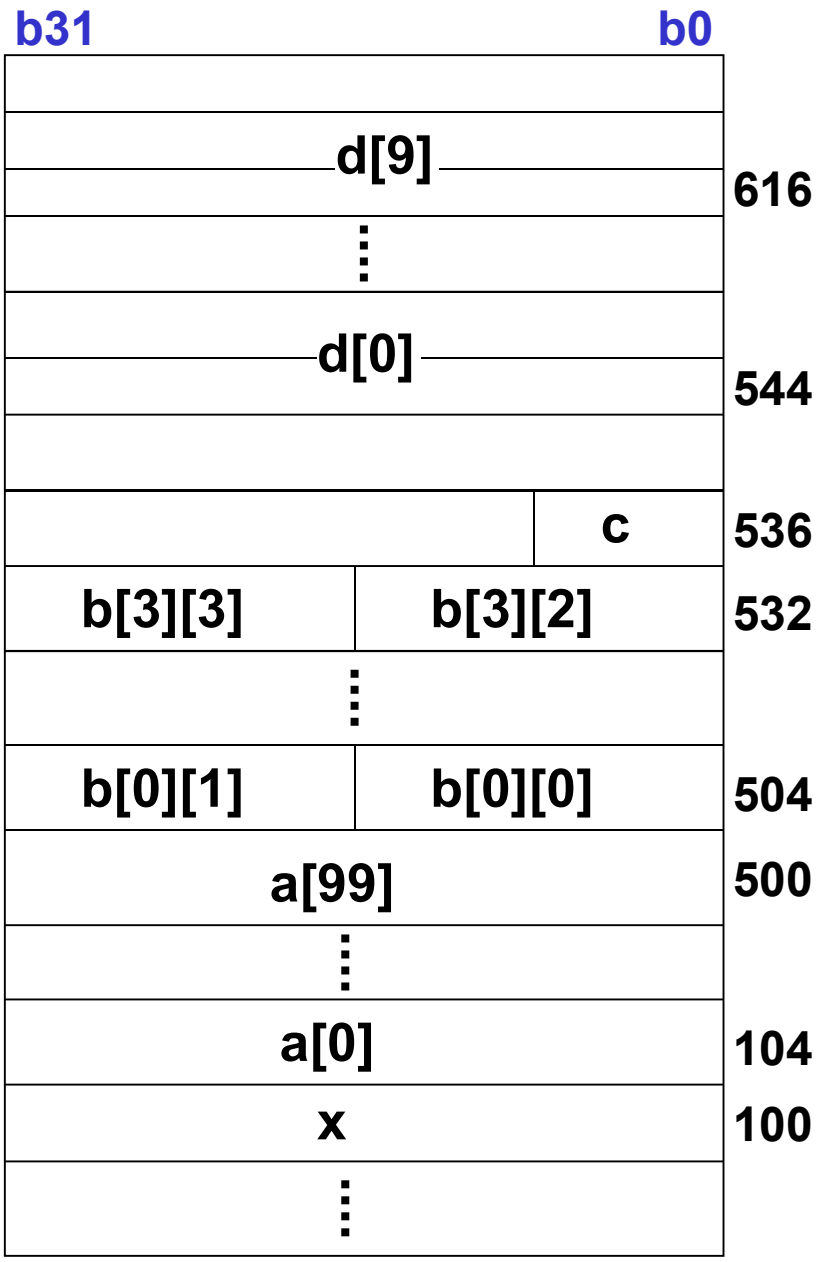
基址+比例变址+位移

将b[i][j]取到AX中的指令可以是：

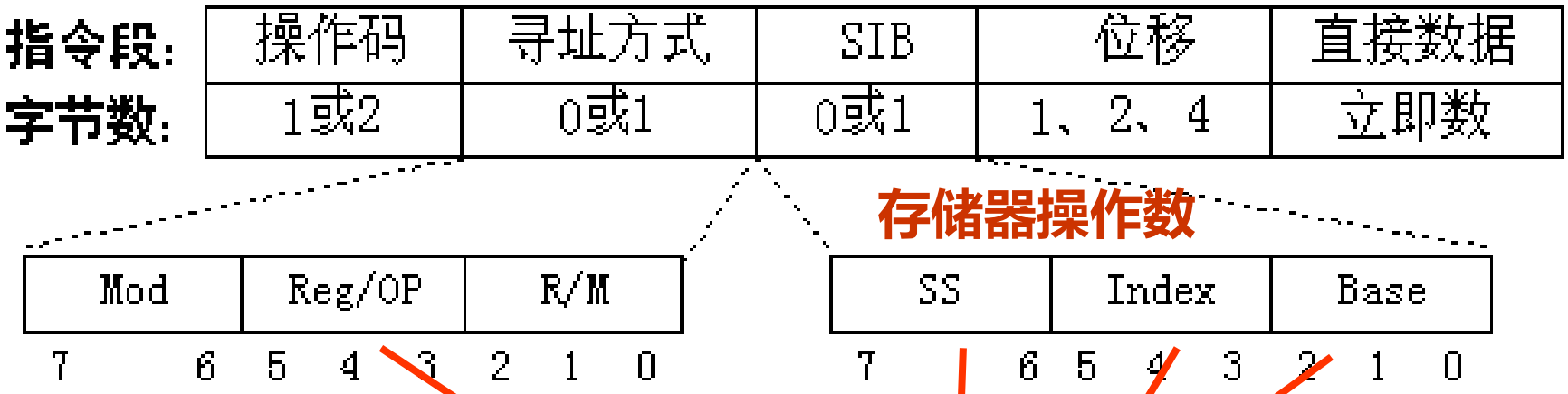
“movw 504(%ebp,%esi,2), %ax”

其中， i×8在EBP中，j在ESI中，

2为比例因子



IA-32机器指令格式



位移量和立即数都可以是：1B/2B/4B

SIB中基址B和变址I都可是8个GRS中任一个；SS给出比例因子

操作码：opcode; W：与机器模式（16 / 32位）一起确定寄存器位数（AL / AX / EAX）; D：操作方向（确定源和目标）

寻址方式（ModRM字节）：mod、r/m、reg/op三个字段与w字段和机器模式（16/32）一起确定操作数所在的寄存器编号或有效地址计算方式

```
8d 04 02    leal   (%edx,%eax,1), %eax
```

1000 1101 00 000 100 00 000 010

总结

- IA-32是典型的CISC（复杂指令集计算机）风格ISA
  - 8个通用寄存器（8位、16位、32位）
  - 2个专用寄存器：EIP（PC）、标志寄存器EFLAGS
  - 6个段寄存器（间接给出段基址）
  - 存储器地址空间为4GB，按字节编址，小端方式
  - 寻址方式
    - 立即、寄存器、存储器（SR:[B]+[I]*s+A）
    - 相对寻址
  - 变长指令字、变长操作码
- 汇编语言格式
  - Intel格式汇编
  - AT&T格式汇编（本课程使用）

段基址+有效地址（偏移量）

8(%edx,%eax,4)