汇编语言程序设计

Review of Assembly Language Programming

Vincent Tam

目录

Contents

- 汇编语言编程基础
- PC的指令系统
- 汇编程序设计

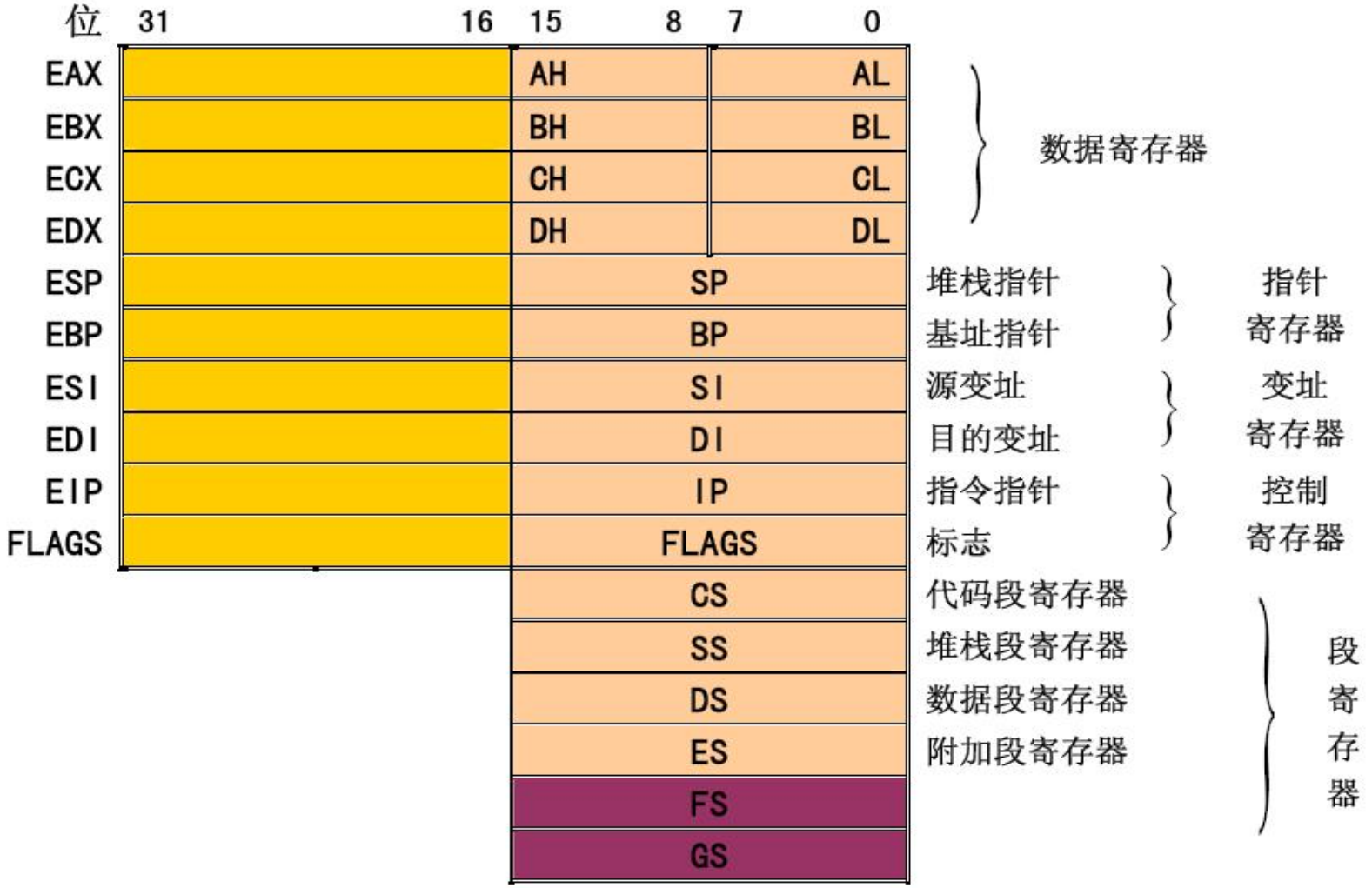
汇编语言编程基础

第一部分

CPU的三种运行模式

- 实模式
  - 没有多任务，所有程序都在最高特权
  - 程序独占整个CPU
- 保护模式
  - Windows、Linux的运行模式
  - 分页分段机制、特权级保护、多任务
- 虚拟8086模式
  - 以任务形式在保护模式下运行
  - 为了支持旧的8086程序

程序可见寄存器组


位
31
16
15
8
7
0
EAX
EBX
ECX
EDX
ESP
EBP
ESI
EDI
EIP
FLAGS
AH
AL
BH
BL
CH
CL
DH
DL
SP
BP
SI
DI
IP
FLAGS
CS
SS
DS
ES
FS
GS
数据寄存器
堆栈指针
基址指针
指针
寄存器
源变址
目的变址
变址
寄存器
指令指针
标志
控制
寄存器
代码段寄存器
堆栈段寄存器
数据段寄存器
附加段寄存器
段
寄
存
器

程序可见寄存器组

通用寄存器 / 数据寄存器

- 数据寄存器
  - 8位： AL AH BL BH CL CH DL DH
  - 16位：AX BX CX DX
  - 32位：EAX EBX ECX EDX [386]

程序可见寄存器组

通用寄存器 / 指针寄存器

- 堆栈指针寄存器SP、ESP（386以上）
  - 存放当前堆栈段栈顶偏移量，总是与SS堆栈段寄存器配合存取堆栈中的数据。
  - 实模式使用SP，保护模式使用ESP。
- 基址指针寄存器BP、EBP（386以上）
  - 存放地址的偏移量部分或数据。若存放偏移量时，缺省情况与SS配合。
  - 实模式使用BP，保护模式使用EBP。

程序可见寄存器组

通用寄存器 / 变址寄存器

- 变址寄存器SI、DI、ESI、EDI
  - 存放地址的偏移量部分或数据。若存放偏移量时，缺省情况与DS配合。
  - 实模式使用SI、DI，保护模式使用ESI、EDI。

程序可见寄存器组

段寄存器

- 段寄存器存放段基址。在实模式下存放段基地址（高16位），在保护模式下存放段选择符。

- 段选择符：用以选择描述符表中的一个描述符。
  - 描述符描述段的基地址、长度和访问权限。

程序可见寄存器组

段寄存器

- 代码段寄存器CS
  - 指定当前代码段，代码段中存放当前运行的程序段。
- 数据段寄存器DS
  - 指定当前运行程序所使用的数据段。
- 堆栈段寄存器SS
  - 指定当前堆栈段。
- 附加段寄存器ES
  - 指定当前运行程序所使用的附加数据段。
- FS、GS（386）
  - 指定当前运行程序的另外两个存放数据的存储段

程序可见寄存器组

控制寄存器

- 指令指针寄存器IP、EIP
  - 与CS段寄存器配合指出下一条要执行指令的地址，其中存放偏移量部分。
- 标志寄存器FLAGS
  - 也被称为状态寄存器，由运算结果特征标志和控制标志组成。

程序可见寄存器组

控制寄存器 / 标志寄存器

- 运算结果特征标志
  - 记录程序中运行结果的特征。
  - CF、PF、AF、ZF、SF、OF
- 控制标志
  - 控制处理器的操作，要通过专门指令才能使其变化。
  - IF、DF、TF

程序可见寄存器组

控制寄存器 / 标志寄存器 / 运算结果特征标志

- CF(Carry Flag) - 进位标志
  - 运算结果最高位是否向前产生进位或借位
- PF(Parity Flag) - 奇偶标志
  - 计算结果最低8位含1个数是否为偶数
- AF(Auxiliary carry Flag) - 辅助进位标志
  - 计算结果最低4位是否向前产生进位或借位
- ZF(Zero Flag) - 零标志 (记录是否为0)
- SF(Sign Flag) - 符号标志 (记录是否为负)
- AF(Overflow Flag) - 溢出标志 (记录是否溢出)

程序可见寄存器组

控制寄存器 / 标志寄存器 / 控制标志

- IF(Interrupt Flag) - 中断允许标志
  - =1，允许CPU响应外部可屏蔽中断请求INTR
- DF(Direction Flag) - 方向标志
  - 专门服务于字符串操作指令
  - =1，表示串操作数地址为自动减量(高地址到低)
  - =0，相反
- TF(Trap Flag) - 陷阱标志
  - 用于程序调试
  - =1，CPU处于单步方式
  - =0，处于连续方式

存储顺序

小端方式和大端方式

- 以小端方式为例
  - 低字节在前，低位保存在内存的低地址中
- 小端/大端方式一定是按字节存储的，但是可以以字节、字、双字等各种形式读出。

- 将9025H按小端方式存储到内存的1000H单元，则结果为
  - 1000H：25H
  - 1001H：90H

实模式寻址

分段管理

- 物理地址的计算方法
  - 10H * 段基址 + 偏移量

I/O地址空间

概念

- 外设与主机的信息交换是通过外设接口进行的。
- 不同的外设接口中含有的寄存器数量不同。
- 系统给每个接口中的寄存器赋予一个端口地址。

- 这些端口地址组成的地址空间为I/O地址空间。

PC的指令系统

第二部分

寻址方式

- 与数据有关的寻址方式
  - 使用MOV指令
- 与转移地址有关的寻址方式
  - 使用JMP指令

寻址方式

与数据有关的寻址方式 / 立即寻址方式

- MOV EAX, 立即数
- 用于给寄存器或者内存单元赋初值。

- 例子
  - MOV EAX,1234H

寻址方式

与数据有关的寻址方式 / 寄存器寻址方式

- 操作数直接包含在寄存器中。
  - 由指令指定寄存器号。

- 例子
  - MOV BX,AX

寻址方式

与数据有关的寻址方式 / 直接寻址方式

- 操作数的有效地址EA直接包含在指令中。

- MOV AL,[78H]
  - [78H]是一个普通变量的有效地址
- MOV EBX,ES:MEM
  - 段超越前缀ES：使用ES所指向的附加数据段
- MOV AL,VAR
  - VAR是内存变量名，代表一个内存单元的符号地址

寻址方式

与数据有关的寻址方式 / 直接寻址方式

- 有效地址存放在代码段的指令操作码之后，但操作数本身在存储器中，所以必须先求出操作数的物理地址。
- 普通变量缺省情况是存放在DS所指向的数据段，但允许使用段超越前缀指定为其它段。
- 物理地址=段基址*10H + 有效地址EA

寻址方式

与数据有关的寻址方式 / 直接寻址方式

操作类型	约定段寄存器	允许指定的段寄存器	偏移量
1. 指令	CS	无	IP
2. 堆栈操作	SS	无	SP
3. 普通变量	DS	ES、SS、CS	EA
4. 字符串指令的源串地址	DS	ES、SS、CS	SI
5. 字符串指令的目标串地址	ES	无	DI
6. BP用作基址寄存器	SS	DS、ES、CS	EA

寻址方式

与数据有关的寻址方式 / 寄存器间接寻址方式

- 操作数有效地址在基址寄存器BX、BP或变址寄存器SI、DI中，而操作数在存储器中的寻址方式。
- 若指令中使用的是BX、SI、DI、EAX、EBX、ECX、EDX、ESI、EDI，则缺省情况操作数在数据段，即它们默认与DS段寄存器配合。
- 若使用的是BP、EBP、ESP，则缺省情况默认与SS段寄存器配合。
- 均允许使用段超越前缀。

寻址方式

与数据有关的寻址方式 / 寄存器间接寻址方式

- MOV AL,[BX]
  - DS:[BX] -> AL
- MOV AX,[BP]
  - SS:[BP] -> AX

寻址方式

与数据有关的寻址方式 / 寄存器相对寻址方式

- 操作数的有效地址是一个基址(BX、BP)或变址寄存器(SI、DI)的内容和指令中给定的一个位移量(disp)之和。
- 386以上允许使用任何32位通用寄存器。位移量可以是一个字节、一个字、一个双字（386以上）的带符号数。
- 有效地址
  - EA = （基址 < 或变址 > 寄存器）+ disp
  - EA = （32位通用寄存器）+ disp

寻址方式

与数据有关的寻址方式 / 寄存器相对寻址方式

- MOV AL,TABLE[BX]
  - MOV AL,[BX+TABLE]
  - (DS:[BX+TABLE]) -> AL
  - 访问一维数组
    - TABLE是数组起始地址的偏移量
    - 寄存器中是数组元素的下标乘以元素的长度(占用字节数)

- MOV AL,8[BX]
  - MOV AL,[BX+8]

寻址方式

与数据有关的寻址方式 / 基址变址寻址方式

- EA = (基址寄存器) + (变址寄存器)
- 缺省使用段寄存器的情况由基址寄存器决定。若使用BP、ESP或EBP，缺省与SS配合；若使用BX或其它32位通用寄存器，则缺省与DS配合。
- 允许使用段超越前缀。

寻址方式

与数据有关的寻址方式 / 基址变址寻址方式

- MOV AL,[BX][SI]
  - MOV AL,[BX+SI]
  - (DS:[BX+SI]) -> AL
  - 访问一维数组
    - BX存放数组起始地址的偏移量
    - SI存放数组元素的下标乘以元素的长度

寻址方式

与数据有关的寻址方式 / 相对基址变址寻址方式

- EA = (基址寄存器) + (变址寄存器) + disp
- 缺省使用段寄存器的情况由基址寄存器决定。若使用BP、ESP或EBP，缺省与SS配合；若使用BX或其它32位通用寄存器，则缺省与DS配合。
- 允许使用段超越前缀。

寻址方式

与数据有关的寻址方式 / 相对基址变址寻址方式

- MOV AL,ARY[BX][SI]
  - MOV AL,[BX+SI+ARY]
  - (DS:[BX+SI+ARY]) -> AL
  - 访问二维数组
    - ARY为数组起始地址的偏移量
    - BX=行下标*一行占用的字节数(某行首与数组起始地址距离)
    - SI=列下标*一列占用的字节数(某列与所在行首的距离)

寻址方式

与转移地址有关的寻址方式 / 段内直接寻址方式

- 要转向的有效地址
  - EA = (IP) + { 8位 / 16位 } disp
  - EA = (EIP) + { 8位 / 32位 } disp
- 短转移
  - JMP SHORT L1
- 近转移
  - JMP L2 或 JMP NEAR PTR L2

寻址方式

与转移地址有关的寻址方式 / 段内间接寻址方式

格　式	举　例	注　释
JMP　通用寄存器	JMP　BX	16 位转向地址在 BX 中，其值送给 IP
	JMP　EAX	32 位转向地址在 EAX 中，其值送给 EIP
JMP　内存单元	JMP　WORD PTR VAR	16 位转向地址在 VAR 字型内存变量中
	JMP　WORD PTR [BX]	16 位转向地址在 BX 所指向的内存变量中
	JMP　DWORD PTR DVAR	32 位转向地址在 DVAR 双字型内存变量中
	JMP　DWORD PTR [EBX]	32 位转向地址在 EBX 所指向的内存变量中

寻址方式

与转移地址有关的寻址方式 / 段间直接寻址方式

- 执行时把偏移量送给IP，段基址送给CS，即可实现段间直接转移。

- MOV FAR PTR L1

寻址方式

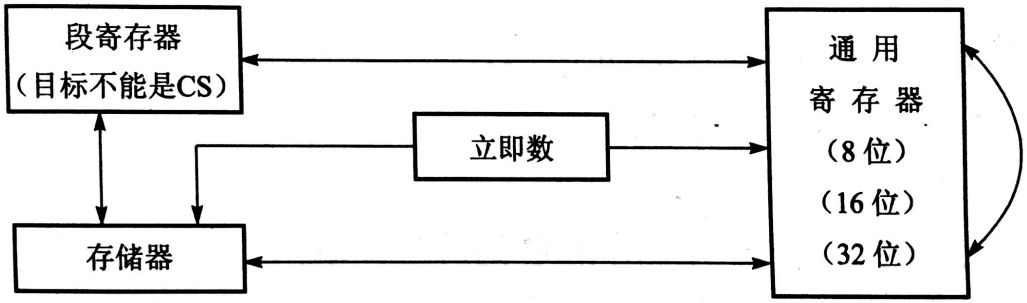
与转移地址有关的寻址方式 / 段间间接寻址方式

- 用一个双字内存变量中的低16位取代IP值，高16位取代CS值，从而实现段间转移。
- MOV DWORD PTR [BX]
  - DWORD表示[BX]指向一个双字变量

- 当操作数长度为32位时，则用一个三字内存变量(48位)中的低32位取代EIP值，高16位取代CS值。
- MOV FWORD PTR [EBX]
  - FWORD表示[EBX]指向一个三字变量

数据传送指令

通用数据传送指令 / 传送指令MOV



- 立即数不能作为目标操作数
- 立即数不能直接送段寄存器
- 目标寄存器不能是CS
- 段寄存器之间不能相互传送
- 两个存储单元之间不能相互传送

数据传送指令

通用数据传送指令 / 带符号扩展的数据传送指令MOVSX

- 只有386以上机器提供
- 用于对带符号数的扩展
- MOVSX DST,SRC
  - SRC -> DST，DST空出的位用SRC的符号位填充
  - DST必须为16或32位
  - SRC可以是8或16位，或存储器操作数，但不可为立即数

数据传送指令

通用数据传送指令 / 入栈指令PUSH

- PUSH SRC
  - 8086、8088
    - SRC是16位寄存器操作数或存储器操作数，不可为立即数
  - 80286以上
    - SRC可以是16或32位(386)……，也可以是立即数
- 功能
  - 先修改堆栈指针使其指向新的栈顶
    - SRC为16位，则SP-2 -> SP
    - SRC为32位，则SP-4 -> SP
  - 然后把SRC压入栈顶单元

数据传送指令

通用数据传送指令 / 出栈指令POP

- POP DST
  - DST可以是16或32位(386)的寄存器操作数和存储器操作数
  - 也可以是除CS寄存器外的任何段寄存器
- 功能
  - 先把堆栈指针所指向单元的内容弹出到DST
  - 然后修改堆栈指针以指向新的栈顶
    - DST为16位，则SP+2 -> SP
    - DST为32位，则SP+4 -> SP

数据传送指令

通用数据传送指令 / 交换指令XCHG

- XCHG OPR1,OPR2
  - OPR可以是8、16、32位(386)存储器操作数或寄存器操作数，不能是立即数
  - 其中之一必须是寄存器操作数

数据传送指令

输入输出指令 / 输入指令IN

- IN ACR,PORT
  - 把外设端口PORT的内容传给累加器ACR
- 传送8、16、32(386)的数据，相应的累加器选择AL、AX、EAX。
- 端口号在0-255之间，可以直接写在指令中；
- 端口号大于255，通过DX寄存器间接寻址。

数据传送指令

输入输出指令 / 输出指令OUT

- OUT PORT,ACR
- 把累加器中的内容传送给外设端口
- 传送8、16、32(386)的数据，相应的累加器选择AL、AX、EAX。
- 端口号在0-255之间，可以直接写在指令中；
- 端口号大于255，通过DX寄存器间接寻址。

数据传送指令

地址传送指令 / 传送有效地址指令LEA

- LEA REG,SRC
  - 源操作数SRC必须是存储器操作数
- 把操作数的有效地址传给指定寄存器

- LEA BX,ASC
  - 同MOV BX,OFFSET ASC
- LEA BX,ASC[SI]
  - 把DS:[SI+ASC]中的16位偏移量送BX

数据传送指令

标志传送指令

- 16位标志进栈指令PUSHF
  - SP-2 -> SP，压入FLAGS到栈顶单元
- 16位标志出栈指令POPF
- 32位标志进栈指令PUSHFD
  - ESP-4 -> ESP，压入EFLAGS到栈顶单元
- 32位标志出栈指令POPFD
- 标志送AH指令LAHF
  - FLAGS低8位送AH寄存器
- AH送标志寄存器指令SAHF
  - AH寄存器内容送标志寄存器FLAGS低8位

算术运算指令

类型转换指令

- 字节扩展成字CBW
  - AL中符号位扩展到AH中
- 字扩展成双字CWD
  - AX符号位扩展到DX中
- [386可用]字扩展成双字CWDE
  - AX符号位扩展到EAX中
- [386可用]双字扩展成四字CDQ
  - EAX符号位扩展到EDX中