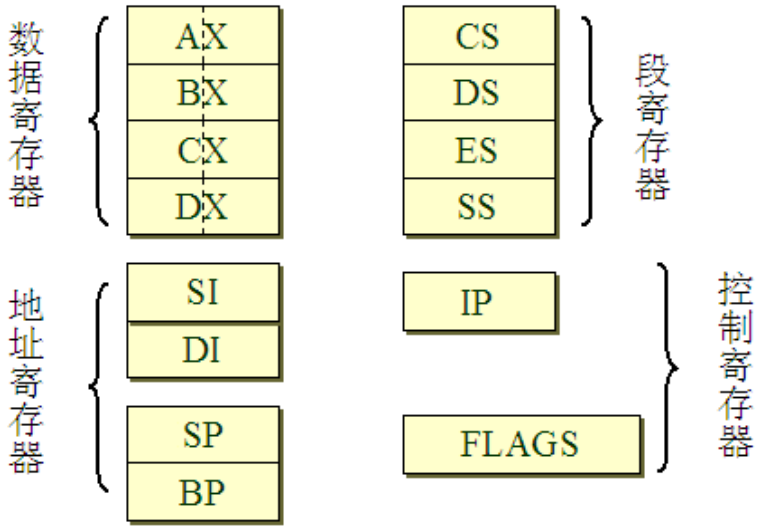
复习

信息学院 别丽华

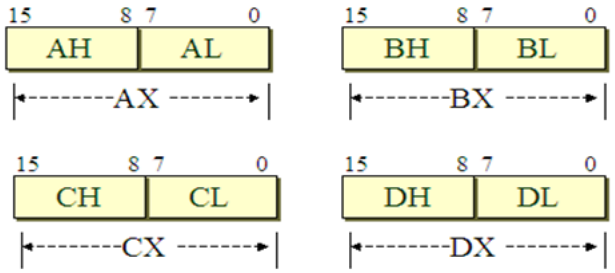
寄存器

- 8086CPU寄存器（16位 ）

- 14个寄存器：
  AX、BX、CX、DX、SI、DI、SP、BP、
  IP、CS、SS、DS、ES、PSW（FLAGS）



- 通用寄存器
  - AX、BX、CX、DX
    - 检测点2.1 （2）



数据长度、大小、表示范围

- 段寄存器
  - CS、DS、SS、ES

- 存储器的分段管理

8086内部为16位地址， 20位地址总线，寻址1M空间。

每段最大64K字节，最少16个字节。

物理地址=段地址×16+偏移地址

- 检测点2.2

- 8086汇编语言中把逻辑段分为四种类型：代码段、数据段、附加段和堆栈段。

段名	段寄存器	偏移地址
代码段	CS	IP
数据段	DS	BX、SI、DI等地址寄存器
附加段	ES	BX、SI、DI等地址寄存器
堆栈段	SS	SP或BP

- 检测点2.3
- 问题3.8-10
- 检测点3.2 P71

Debug命令查看寄存器内容：实验1

内存访问

- 存储器中数据的组织
- 大端方式
- 小端方式 (8086 CPU)

1、DS和[address]

```
mov bx,1000H
mov ds,bx
mov al,[0]  ；省略时，默认为Ds
```

检测点3.1 （2）（p55）

- mov指令的形式：

mov 寄存器，数据

mov 寄存器，寄存器

mov 寄存器，内存单元

mov 内存单元，寄存器

mov 段寄存器，寄存器

练习：

在70000H处写入字型数据1122H，可以用以下的代码完成：

```
mov ax,7000H
mov ds,ax
mov ,ax,1122H
mov [0],ax
```

补全下面的代码，完成同样的功能：

在70000H处写入字型数据1122H。

```
mov ax,7000H
mov ss,ax
mov sp,2H.
mov ax,1122H
push ax
```

```
mov ax,7000H
mov ss,ax
mov sp,2
```

2.批量数据访问

- [bx]与loop
  - mov ax,[bx]
    mov al,[bx]

```
    mov cx,循环次数
s:
    循环执行的程序段
    loop s
```

联用：问题5.4，程序5.8，,5.9
实验4 （1）（2）P121

- 段前缀

- 检测点6.1 P129
- 实验5 （5）（6）

- P140第7章(!)
  - [bx+idata]:数组的处理，大小写转换
  - [bx+si]
  - [bx+di]
  - [bx+si+idata]
  - [bx+di+idata]

问题7.6-7.9

P164 表8.2

问题8.1

- 在8086CPU 中，只有4个寄存器（**bx、bp、si、di**）

  可以用在“[...]”中来进行内存单元的寻址。

  可以单个出现，或以四种组合出现：

  bx和si、bx和di、

  bp和si、bp和di

转移——修改CS/IP

- Jmp
  - jmp short 标号 --- 段内短转移
    - (IP)=(IP)+8位位移
  - jmp near ptr 标号 --- 段内近转移
    - (IP)=(IP)+16位位移
  - jmp far ptr 标号 ----段间转移
    - (CS)=标号所在段的段地址；
    - (IP)=标号所在段中的偏移地址。
      - ---------- 根据位移转移

- jmp 16位寄存器
  - IP =（16位寄存器）
- 转移地址在内存中：
  - jmp word ptr 内存单元
  - jmp dword ptr 内存单元地址元地址
    - (CS)=(内存单元地址+2)
    - (IP)=(内存单元地址)

(IP) (CS)

检测点9.1 P183
检测点9.3

实验8 P187

Call和Ret---子程序

- 检测点10.1 10.2 10.4 10.5
- 实验10 1.3
- 实验11 P234

ret 和 retf

- Ret 指令用栈中的数据修改IP，实现近转移；

  （1）(IP)=((ss)*16+(sp))

  （2）(sp)=(sp)+2

- Retf 指令用栈中的数据，修改CS和IP，实现远转移；

  （1）(IP)=((ss)*16+(sp))

  （2）(sp)=(sp)+2

  （3）(CS)=((ss)*16+(sp))

  （4）(sp)=(sp)+2

- 检测点10.1 P191

call 指令

（1）将当前的 IP 或 CS和IP 压入栈中；
（2）转移。

- call　标号

(1) (sp) = (sp) – 2
((ss)*16+(sp)) = (IP)
(2) (IP) = (IP) + 16位位移

相当于

push IP
jmp near ptr 标号

--- 根据位移转移

检测点10.2　P192

Ax=?

内存地址	机器码	汇编指令
1000:0	b8 00 00	mov ax，0
1000:3	e8 01 00	call s
1000:6	40	inc　ax
1000:7	58	s:pop　ax

call far ptr 标号 ----- 段间转移

(1) (sp) = (sp) – 2

((ss) ×16+(sp)) = (CS)

(sp) = (sp) – 2

((ss) ×16+(sp)) = (IP)

(2) (CS) = 标号所在的段地址

(IP) = 标号所在的偏移地址

转移地址在指令中

- 检测点10.3 P192

call 16位寄存器

- (sp) = (sp) – 2
- ((ss)*16+(sp)) = (IP)
- (IP) = (16位寄存器)

转移地址在寄存器中

- 检测点10.4 P194

call word ptr 内存单元地址

```
push IP
jmp word ptr 内存单元地址
```

call dword ptr 内存单元地址

转移地址在内存

```
push CS
push IP
jmp dword ptr 内存单元地址
```

子程序

- 子程序的框架：

  标号:

  指令

  ret

书10.9-10.12 P200

PPT上扩充内容了解
进制转换举例

```
assume cs:code
code segment
main:   :
        :
        call sub1          ;调用子程序sub1
        :
        :
        mov ax,4c00h
        int 21h

sub1:   :                  ;子程序sub1开始
        :
        call sub2          ;调用子程序sub2
        :
        :
        ret                ;子程序返回

sub2:   :                  ;子程序sub2开始
        :
        :
        ret                ;子程序返回
code ends
end main
```

Mul和Div指令

例：用div 计算data段中第一个数据除以第二个数据后的结果，商存放在第3个数据的存储单元中

```
data segment
        dd 100001
        dw 100
        dw 0
data ends
mov ax,data
mov ds,ax
```

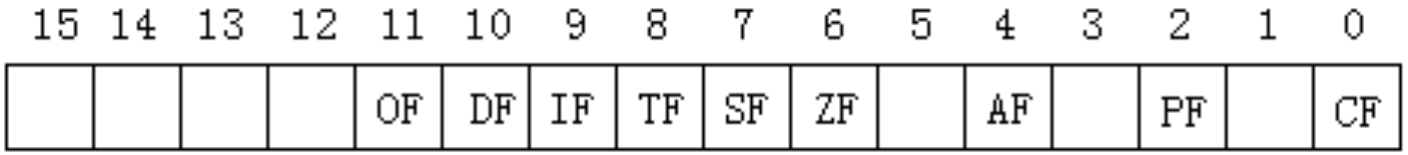
```
mov ax,ds:[0]        ;ds:0字单元中的低16位存储在ax中
mov dx,ds:[2]        ;ds:2字单元中的高16位存储在dx中
div word ptr ds:[4]
                ;用dx:ax中的32位数据除以ds:4字单元中的数据
mov ds:[6],ax        ;将商存储在ds:6字单元中
```

PPT上例题

参数传递的问题！

标志寄存器和条件转移

8086CPU的flag寄存器



CF：无符号数有意义
OF：有符号数有意义

Adc、sbb编程举例

检测点11.1-11.2

DF：方向标志，与串处理相关

Rep movsb

PPT例子

CMP指令和条件转移

检测点11.3

中断

- 8086CPU的中断过程：
  - （1）取得中断类型码；
  - （2）标志寄存器的值入栈
  - （3）设置标志寄存器的第8位TF 和第9位IF的值为0
  - （4）CS的内容入栈；
  - （5）IP的内容入栈；
  - （6）从内存地址为中断类型码*4 和中断类型码 *4+2 的两个字单元中读取中断处理程序的入口地址设置IP和CS。

中断程序的编写

```
assume cs:code
code segment
start: ISR安装程序
       设置中断向量表
       mov ax,4c00h
       int 21h
ISR:   XXXXX
       mov ax,4c00h
       int 21h
code ends
end start
```

rep movsb指令将ISR的代码送入0:0200处。

（1）传送的原始位置ds：si
（2）传送的目的位置es：di
（3）传送的长度--cx
（4）传送的方向cld/std

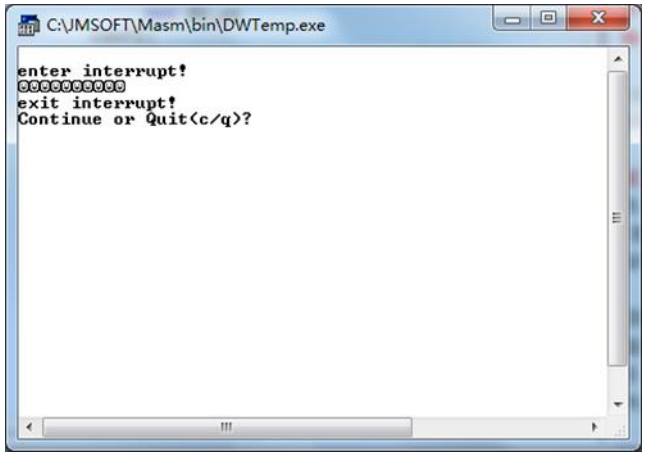
```
mov ax,0
mov es,ax
mov word ptr es:[0*4],200h
mov word ptr es:[0*4+2],0
```

若涉及到数据存储，则可以放在ISR内部。

```
Do0：
        jmp short do0start
          db   “overflow!”
Do0start:    mov ax,cs
             mov  ds,as
             mov si,202h
```

中断

- 练习：
- 利用中断类型号7cH，设计一个笑脸中断INT 7cH，实现下图所示效果，中断例程安装在0:200处。



系统中断调用、端口和宏

In/Out 指令的用法　　检测点14.2　P269

CMOS RAM芯片的访问和显示　　实验14

定义两数相加宏指令SUMM

```
SUMM  MACRO  X,Y,RESULT
        MOV  AX，X
        ADD  AX，Y
        MOV  RESULT，AX
ENDM
```