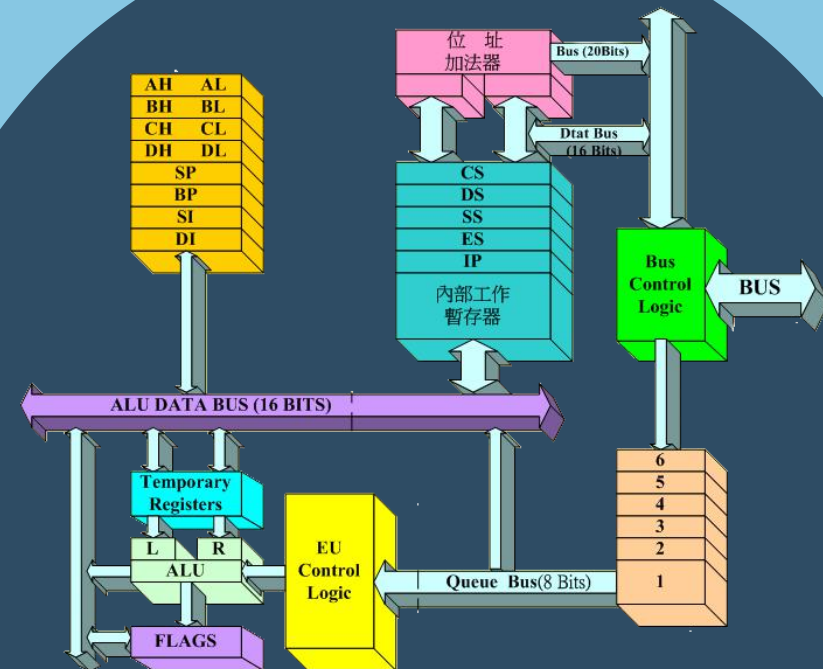


# 带进(借)位的加减法

贺利坚 主讲



汇编语言程序设计  
Assembly Language

# adc-带进位加法指令

💻 adc是带进位加法指令，它利用了CF位上记录的进位值。

- 📄 格式：adc 操作对象1,操作对象2
- 📄 功能：操作对象1=操作对象1+操作对象2+CF
- 📄 例：adc ax,bx 实现的功能是：(ax)=(ax)+(bx)+CF

## 💻实例

指令	mov al,98H add al,al adc al,3	mov ax,1 add ax,ax adc ax,3	mov ax,2 mov bx,1 sub bx,ax adc ax,1
结果	(ax)=34H	(ax)=5	(ax)=4
解释	adc执行时，相当于计算： (ax)+3+CF=30H+3+1=34H	adc执行时，相当于计算： (ax)+3+CF=2+3+0=5	adc执行时，相当于计算： (ax)+1+CF=2+1+1=4

# adc指令应用：大数相加

💻问题：8086指令提供add指令，完成8位或16位加法，有更大的数相加时，如何做？

👉 32位、64位、24位？

💻例：编程计算1EF000H+201000H，结果放在ax（高16位）和bx（低16位）中

💻解决思路：先将低16位相加，然后将高16位和进位值相加

💻程序：

```
mov ax,001EH
mov bx,0F000H
add bx,1000H
adc ax,0020H
```

```
      001E F000H
+)    0020 1000H
-----
                        ???
```

💻例：计算 1E F000 1000H+20 1000 1EF0H，结果放在ax（高16位），bx（次高16位），cx（低16位）中。

💻解决思路：.....

💻程序：

```
mov ax,001EH
mov bx,0F000H
mov cx,1000H
add cx,1EF0H
adc bx,1000H
adc ax,0020H
```

```
      001E F000 1000H
+)    0020 1000 1EF0H
-----
                                   ???
```

# 128位数据的相加

问题：编写一个子程序，对两个128位数据进行相加。

名称：add128

功能：两个逆序存放的128位数据进行相加

```
data segment
dw 0A452H,0A8F5H,78E6H,0A8EH,8B7AH,54F6H,0F04H,671EH
dw 0E71EH,0EF04H,54F6H,8B7AH,0A8EH,78E6H,58F5H,0452H
data ends
```

数据为128位，需要8个字单元，由低地址单元到高地址单元，依次存放由低到高的各个字。

分析

- ds:si指向存储第一个数的内存空间
- ds:di指向存储第二个数的内存空间
- 运算结果存储在第一个数的存储空间中。

```
671E 0F04 54F6 8B7A 0A8E 78E6 A8F5 A452H
+)0452 58F5 78E6 0A8E 8B7A 54F6 EF04 E71EH
_____
      ???
```

code segment start : mov ax,data mov ds,ax mov si,0 mov di,16 mov cx,8 call add128 mov ax,4c00h int 21h add128: ; 定义子程序 code ends end start	;寄存器压栈 sub ax,ax s: mov ax,[si] adc ax,[di] mov [si],ax inc si inc si inc di inc di loop s ; 寄存器出栈 ret	push ax push cx push si push di          pop di pop si pop cx pop ax
---	---	--


讨论

sub ax, ax可否替换为mov ax, 0

两个inc di是否可以替换为add di, 2

# sbb指令

 sbb：带借位减法指令

 格式：sbb 操作对象1,操作对象2

 功能：  
操作对象1=操作对象1-操作对象2-CF

 与sub区别：利用CF位上记录的借位值

 比如：sbb ax,bx

 实现功能： $(ax) = (ax) - (bx) - CF$

 应用：对任意大的数据进行减法运算

 例如：计算003E1000H-00202000H  
结果放在ax，bx中

 程序

```
mov bx,1000H
```

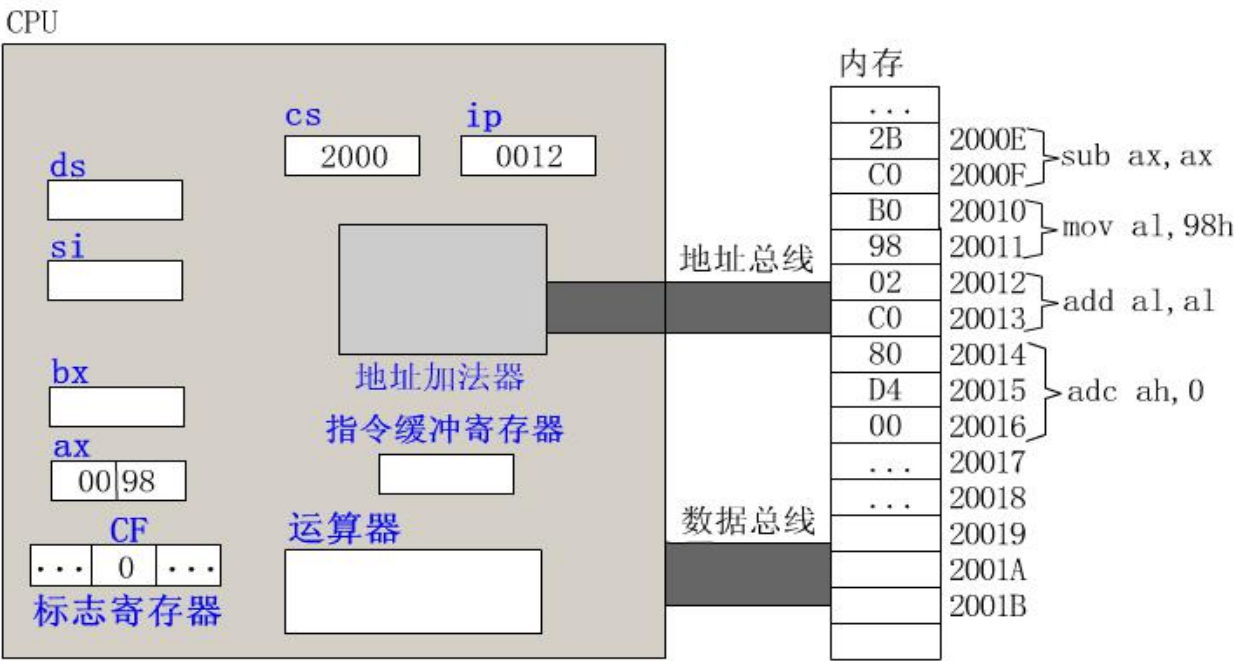
```
mov ax,003EH
```

```
sub bx,2000H
```

```
sbb ax,0020H
```

```
      003E 1000H
- )  0020 2000H
-----
                ???
```

# adc指令执行过程



adc指令的执行过程