

# 4.3 汇编语言程序设计的基本方

## ——子程序及过程定义

### 六. 子程序及过程定

子程序是完成确定功能的程序段，它可以被其它程序调用，在完成确定功能后，又可自动返回到调用程序处。在进行数值计算时，常常会用到  $SIN(X)$ ,  $COS(X)$ ,  $\sqrt{2}$  等或二进制数转换成BCD数及BCD数转换成ASC II 码等。对于这样一些常用的程序段，我们可以将其写成子程序，在需要时，只要按一定的格式调用，就可以实现相应的运算或转换，从而避免了程序中多次重复的书写这些程序，也节约了内存。

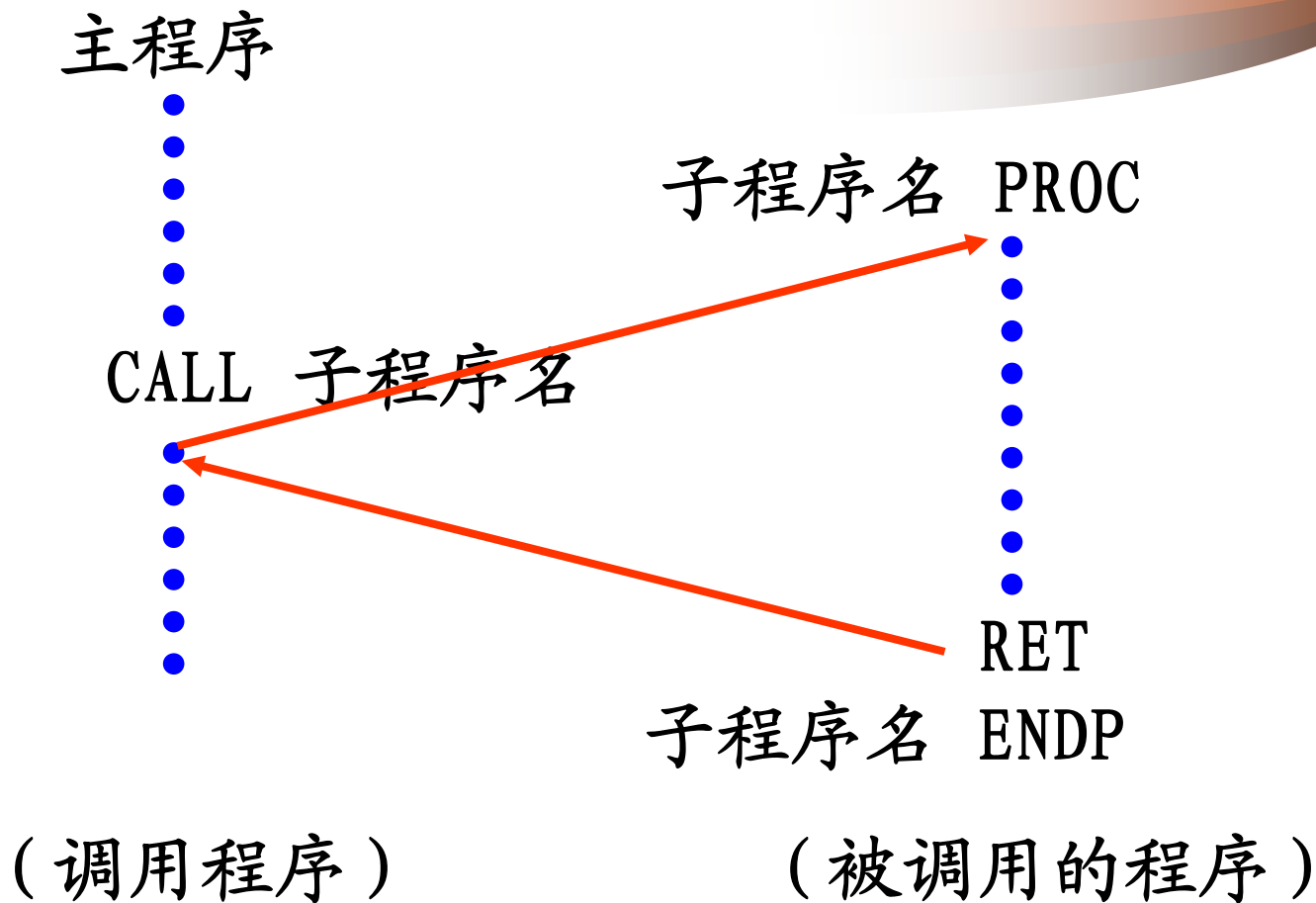
# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

同时，由于一些常用的子程序可以事先编好，一旦需要时可以随时调用，因而也方便了程序的编制和调用。但子程序的调用和返回需要占用时间，并且子程序通常具有一定的通用性和独立性，因此，在调用时，需要保存某些寄存器的内容，以防止子程序运行过程中破坏调用程序以产生的中间结果，所以，采用子程序在运行时间上将有所损失。

# 4.3 汇编语言程序设计的基本方法

——子程序及过程定义



# 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

## 1. 子程序的定义

定义格式:

子程序名 PROC [类型]; 子程序定义开始伪指令

⋮

} 子程序体

RET

子程序名 ENDP ; 子程序定义结束伪指令

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

一个子程序名一旦定义，就具有以下三个属性



# 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

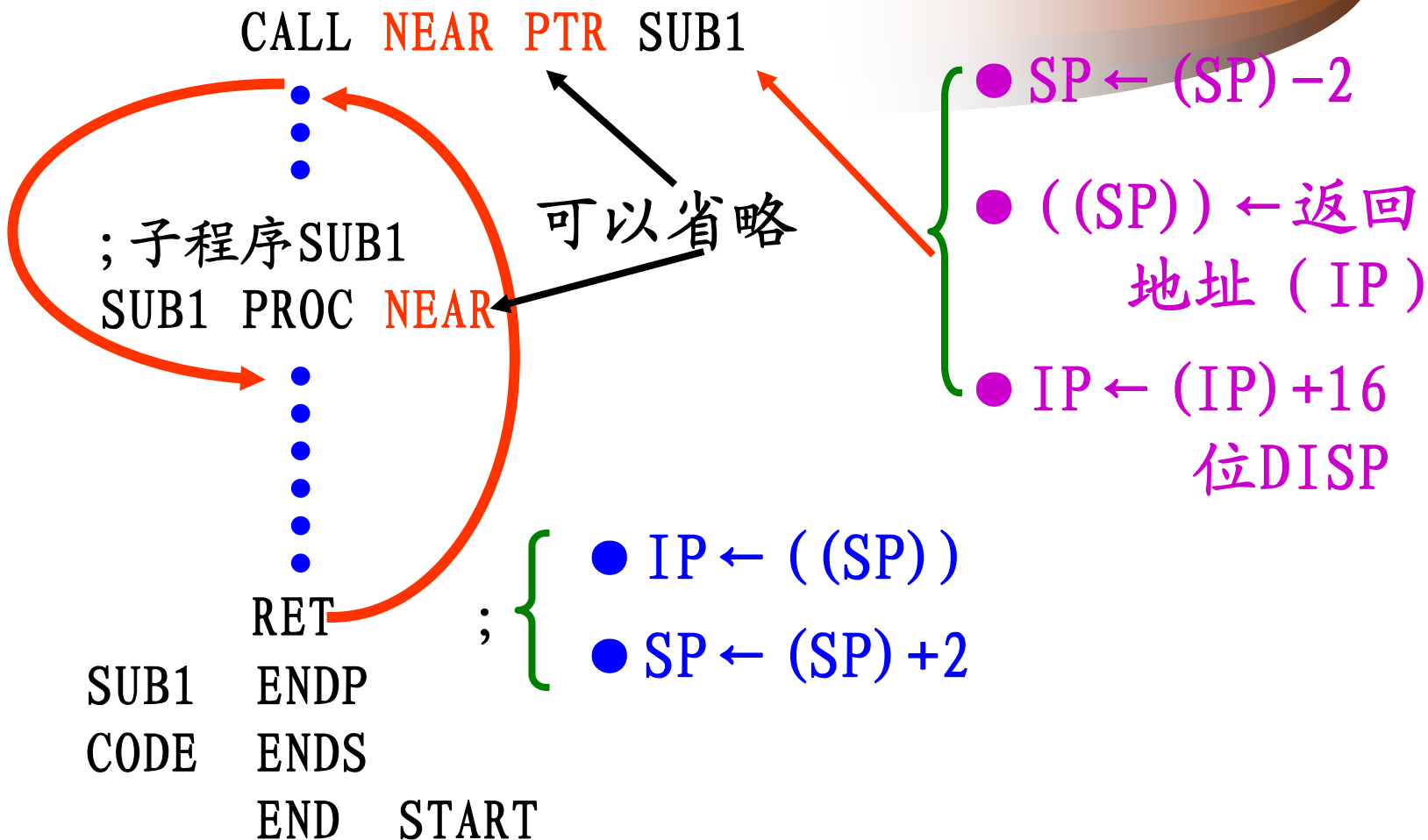
## 2. 子程序调用与返回指令

### ① 段内子程序调用与返回

```
                ; 主程序  
CODE SEGMENT  
                ASSUME CS: CODE  
START:         ●  
                ●  
                ●  
                ●  
                ●
```

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义



①  $SP \leftarrow (SP) - 2$   
 $((SP)) \leftarrow$  返回地址 (CS)  
 ②  $SP \leftarrow (SP) - 2$   
 $((SP)) \leftarrow$  返回地址 (IP)  
 ③  $IP \leftarrow$  OFFSET SUB2  
 $CS \leftarrow$  SEG SUB2

返回

①  $IP \leftarrow ((SP))$   
 $SP \leftarrow (SP) + 2$   
 ②  $CS \leftarrow ((SP))$   
 $SP \leftarrow (SP) + 2$

```

; 主程序
CODEM SEGMENT
ASSUME CS: CODEM
START:
    .
    .
    .
    CALL FAR PTR SUB2
    .
    .
    .
    MOV AH, 4CH
    INT 21H
CODEM ENDS
  
```

```

; 子程序
CODES SEGMENT
ASSUME CS: CODES
SUB2 PROC FAR
    .
    .
    .
    RETF ; 也可以写成RET
SUB2 ENDP
CODES ENDS
  
```

RETF ; 也可以写成RET

END START



# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

### 3. 子程序与主程序之间参数的传递

主程序在调用子程序时，应为子程序提供运算所需的初始数据，而子程序返回时，应将运算结果以一定的方式提交给主程序。这些问题，通常称为主程序与子程序之间的参数传递。

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

通常，我们将主程序给予程序提供的初始数据或获得初始数据的信息称为子程序的入口参数，而子程序返回给主程序的结果称为子程序的出口参数。主程序与子程序之间的参数传递可以有多种方式，通常有以下三种方法。

# 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

利用寄存器传递参数 ★

利用存储器单元传递参数

利用堆栈传递参数

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

### 4. 子程序说明文件

子程序编好以后是为了提供给其它程序调用的，因此，它必须以一定的格式说明自己的功能和使用方法，也就是要建立子程序的说明文件。说明文件的内容大致可以包括以下几个方面：

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

- (1) 子程序名;
- (2) 子程序完成的功能;
- (3) 子程序的入口参数及其传递方式;
- (4) 子程序的出口参数及其传递方式;
- (5) 子程序用到的寄存器;
- (6) 典型例子。

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

### 5. 子程序的嵌套

子程序的嵌套是指子程序本身再次调用子程序。如图4.3.18所示，主程序调用子程序SUB1，SUB1又调用子程序SUB2，SUB2又调用子程序SUB3。在返回时，也必须按层返回，SUB3子程序返回时将返回到CALL SUB3指令的下一条指令处；同样，SUB2子程序将返回到SUB1的调用指令处；最后SUB1子程序返回到主程序。嵌套调用过程中逐层调用及按层返回是由堆栈保证的。

# 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

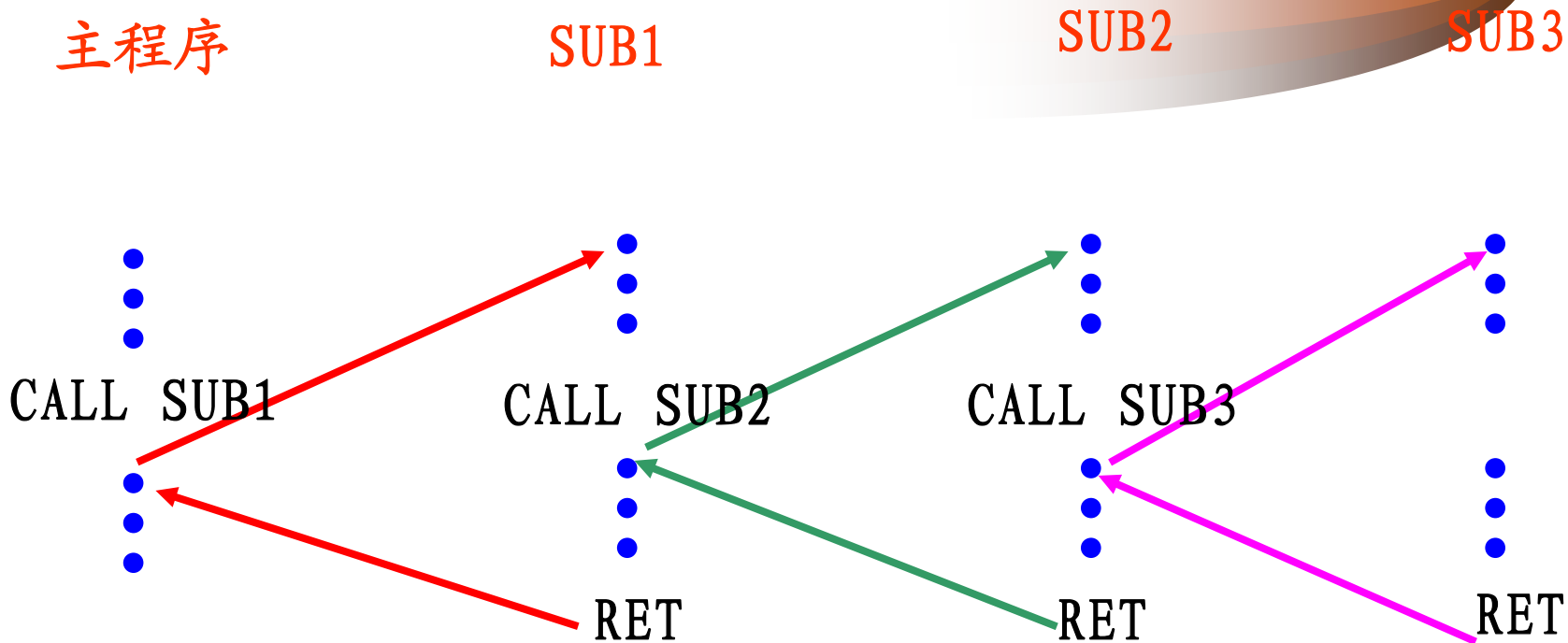


图4.3.18

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

### 6. 信息的保护与恢复

在子程序设计时，肯定要用到一些寄存器，而这些寄存器在主程序也用到了。我们期望这些寄存器的值在子程序调用与返回后，保持原来主程序中的值不变，这就需要对这些寄存器的值进行保护。而保护这些寄存器值的方法是利用堆栈操作。



# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

信息的保护与恢复方法有以下两种：

1. 在子程序中进行

2. 在主程序中进行

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

在子程序中对信息进行保护与恢复的方法

```
子程序名 PROC
```

```
    PUSH    AX
```

```
    PUSH    BX
```

```
    PUSH    CX    ; 信息的保护
```

```
    ●  
    ●  
    ●
```

```
    POP     CX
```

```
    POP     BX
```

```
    POP     AX    ; 信息的恢复
```

```
    RET
```

```
子程序名 ENDP
```

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

在主程序中对信息进行保护与恢复的方法

```
    ●  
    ●  
    ●  
    PUSH    AX  
    PUSH    BX  
    PUSH    CX    ; 信息的保护  
    CALL    子程序名  
    POP     CX  
    POP     BX  
    POP     AX    ; 信息的恢复  
    ●  
    ●  
    ●
```

# 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

## 7. 子程序设计举例

P125, 4. 3. 13

编写程序，完成将一组BCD数转换成16位二进制数。（BCD数转换成16位二进制数用子程序实现）

# 4.3 汇编语言程序设计的基本方法

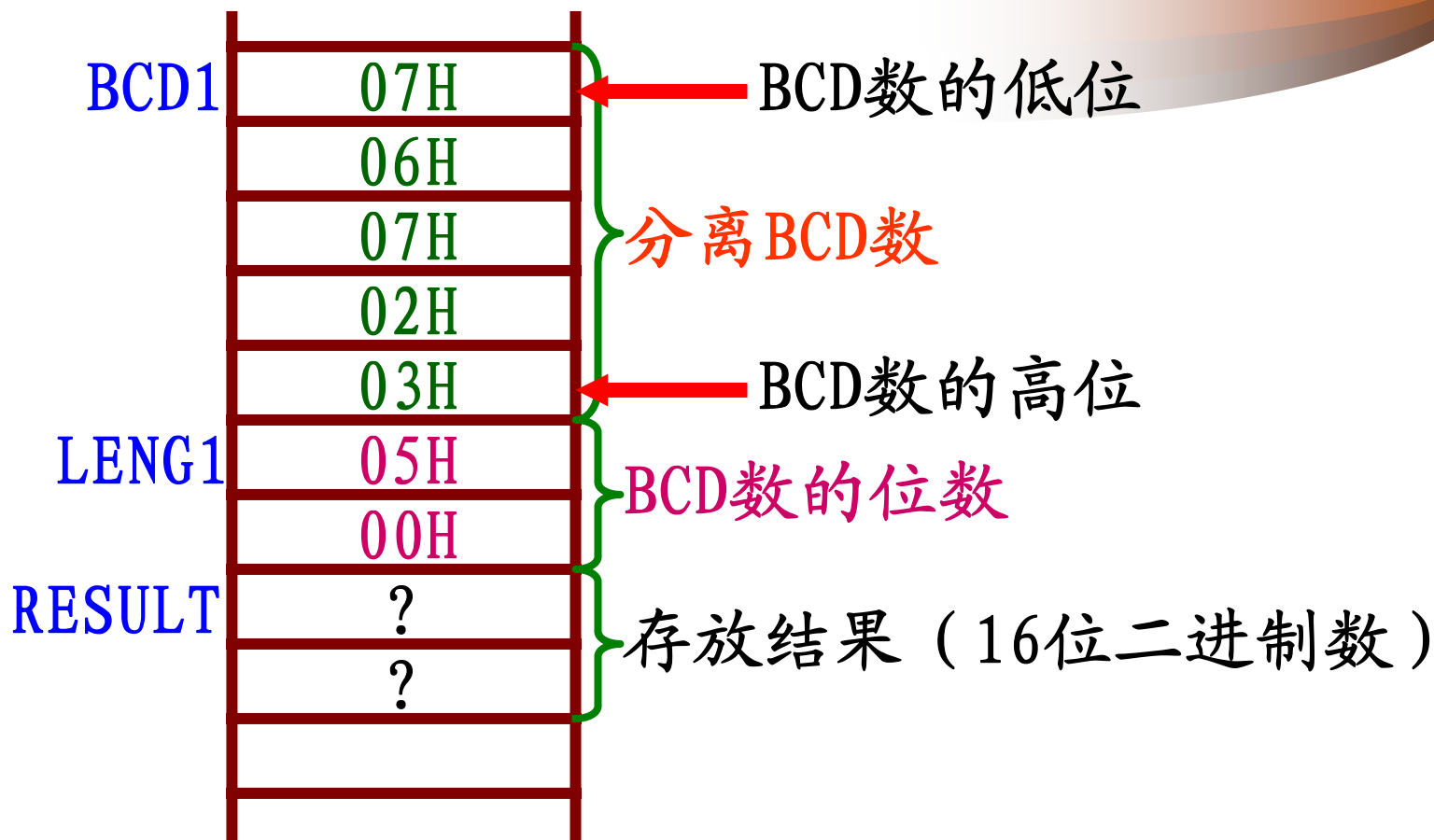
——子程序及过程定义

## 分析题目

假设一组BCD数以分离BCD数的方式存于内存的某个区，并且BCD数的高位存于高地址端，低位存于低地址端。由于转换后的二进制数为16位，所以分离BCD数最多为5个字节。

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义



# 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

## 分析题目

∴ 32767

$$= 3 \times 10^4 + 2 \times 10^3 + 7 \times 10^2 + 6 \times 10^1 + 7 \times 10^0$$

∴ BCD数转换成16位二进制数的算法  
步骤为：

## 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

- ①  $DX=0$ ; (用DX存放结果)
- ② 取要转换的一组BCD数中的高位  $\rightarrow (AX)$
- ③  $(DX) = (DX) * 10 + (AX)$ ;
- ④ 重复②、③两步，直到BCD码的所有位都转换完为止，结果在DX中。



# 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

## 程序设计风格一：利用寄存器传递参数

```
; 主程序
STACK SEGMENT STACK
        DB 256 DUP ( ? )
STACK ENDS
DATA SEGMENT
BCD1 DB 07H, 06H, 07H, 02H, 03H
LENG1 DB 5
RESULT DW ?
DATA ENDS
```

# 4.3 汇编语言程序设计的基本方法

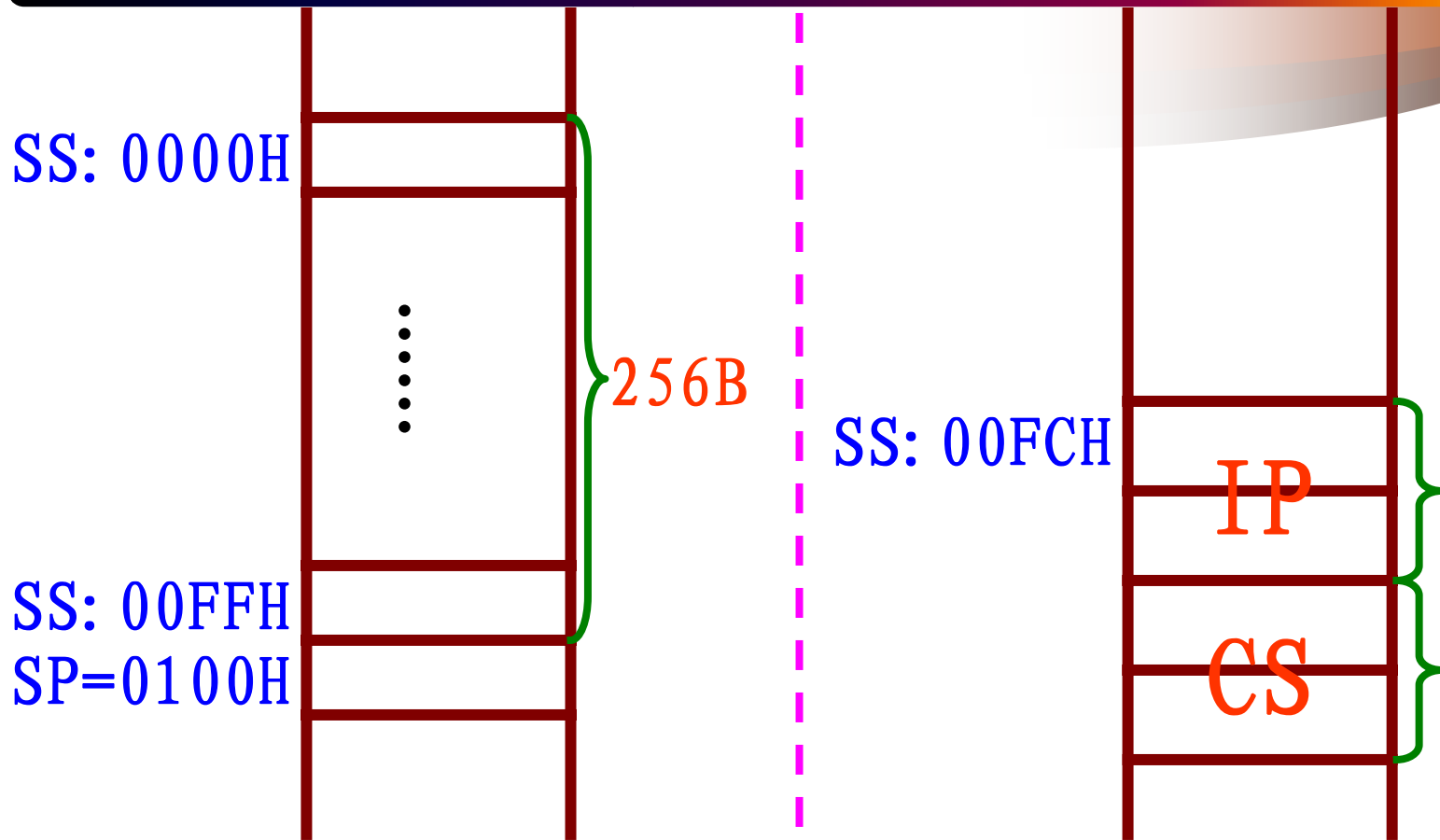
## ——子程序及过程定义

```
CODEM    SEGMENT
          ASSUME  CS: CODEM, DS: DATA, SS: STACK
START:    MOV     AX, DATA
          MOV     DS, AX
          LEA    SI, BCD1      ; SI指向BCD数首址
          MOV    CX, LENG1     ; BCD数的位数送CX
          CALL   FAR  PTR  BCD-16B
          MOV    RESULT, DX
          MOV    AH, 4CH
          INT    21H
CODEM    ENDS
```

在CALL指令前利用寄存器  
向子程序传递入口参数

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义



CALL指令执行前堆栈情况

CALL指令执行后堆栈情况

## 4.3 汇编语言程序设计的基本方法

### ——子程序及过程定义

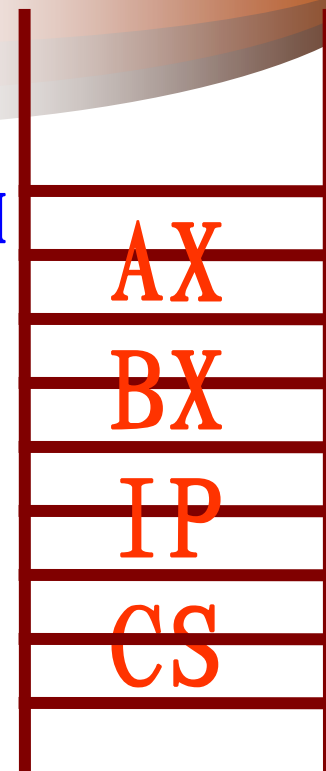
- ; 子程序（下面是子程序说明文件）
- ; ① 子程序名：BCD-16B
- ; ② 子程序完成的功能：将一组BCD数转换成16位二进制数
- ; ③ 入口参数：存放BCD数的内存首址在DS: SI，BCD数的位数在CX
- ; ④ 出口参数：转换后的16位二进制数在DX中
- ; ⑤ 用到的寄存器：无

# 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

```
CODES      SEGMENT
            ASSUME CS: CODES
BCD-16B    PROC FAR
            PUSH  BX
            PUSH  AX
            ADD   SI, CX
            MOV   DX, 0
BCDL:      DEC   SI
            PUSH  CX
            MOV   AL, [SI]
```

SP=00F8H



PUSH BX, PUSH AX指令执行后, 堆栈情况

## 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

```
AND    AL, 0FH
CBW
MOV    BX, AX
MOV    AX, DX
MOV    CX, 10
MUL   CX
MOV    DX, AX
ADD    DX, BX
POP   CX
```

# 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

```
        LOOP    BCDL
        POP     AX
        POP     BX
        RET
BCD-16B ENDP
CODES  ENDS
        END    START
```

SP=00F8H

AX

SP=00FAH

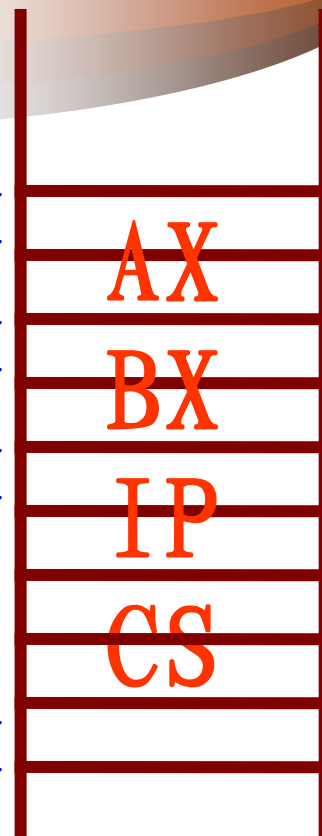
BX

SP=00FCH

IP

CS

SP=0100H



# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

### 程序设计风格二：利用存储器传递参数

```
    ; 主程序
    STACK SEGMENT STACK
        DB 256 DUP ( ? )
    STACK ENDS
    DATA SEGMENT
        BCD1 DB 07H, 06H, 07H, 02H, 03H
        ADSEG DW ? ; 存放BCD数所在段的段地址
        ADOFST DW ? ; 存放BCD数的段内偏移地址
        LENG1 DW ? ; 存放BCD数的位数
        RESULT DW ? ; 存放结果
    DATA ENDS
```



# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

```
CODEM    SEGMENT
          ASSUME  CS: CODEM, DS: DATA, SS: STACK
START:   MOV     AX, DATA
          MOV     DS, AX
          MOV     AX, SEG BCD1
          MOV     ADSEG, AX
          MOV     AX, OFFSET BCD1
          MOV     ADOFST, AX
          MOV     LENG1, 5
          CALL   FAR PTR BCD-16B
          MOV     AH, 4CH
          INT    21H
CODEM    ENDS
```

入口参数传递

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

； 子程序（下面是子程序说明文件）

； ① 子程序名：BCD-16B

； ② 子程序完成的功能：将一组BCD数转换  
； 成16位二进制数

； ③ 入口参数：被转换的BCD数所在单元的  
； 段地址在ADSEG单元中；  
； 被转换的BCD数所在单元的  
； 偏移地址在ADOFST单元中；  
； BCD数的位数在LENG1单元中

## 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

; ④ 出口参数: 转换后的16位二进制  
数在RESULT单元中

; ⑤ 用到的寄存器: 无

# 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

```
CODES SEGMENT
        ASSUME CS: CODES
BCD-16B PROC FAR
        PUSH BX
        PUSH AX
        PUSH SI
        PUSH DS
        MOV DS, ADSEG
        MOV SI, ADOFST
        MOV CX, LENG1
```

## 4.3 汇编语言程序设计的基本方法

### ——子程序及过程定义

```
XOR    DX, DX
ADD    SI, CX
BCDL:  DEC    SI
        PUSH  CX
        MOV   AL, [SI]
        AND  AL, 0FH
        CBW
```

## 4.3 汇编语言程序设计的基本方法

### ——子程序及过程定义

MOV BX, AX

MOV AX, DX

MOV CX, 10

MUL CX

MOV DX, AX

ADD DX, BX

POP CX

LOOP BCDL

# 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

POP DS

POP SI

POP AX

POP BX

RET

BCD-16B ENDP

CODES ENDS

END START

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

### 程序设计风格三：利用堆栈传递参数

```
; 主程序
STACK SEGMENT STACK
        DB 256 DUP ( ? )

STACK ENDS

DATA SEGMENT
BCD1 DB 07H, 06H, 07H, 02H, 03H
LENG1 DW 5 ; 存放BCD数的位数
RESULT DW ? ; 存放结果
DATA ENDS
```



# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

CODEM SEGMENT

ASSUME CS: CODEM, DS: DATA, SS: STACK

START: MOV AX, DATA

MOV DS, AX

PUSH DS

LEA SI, BCD1

PUSH SI

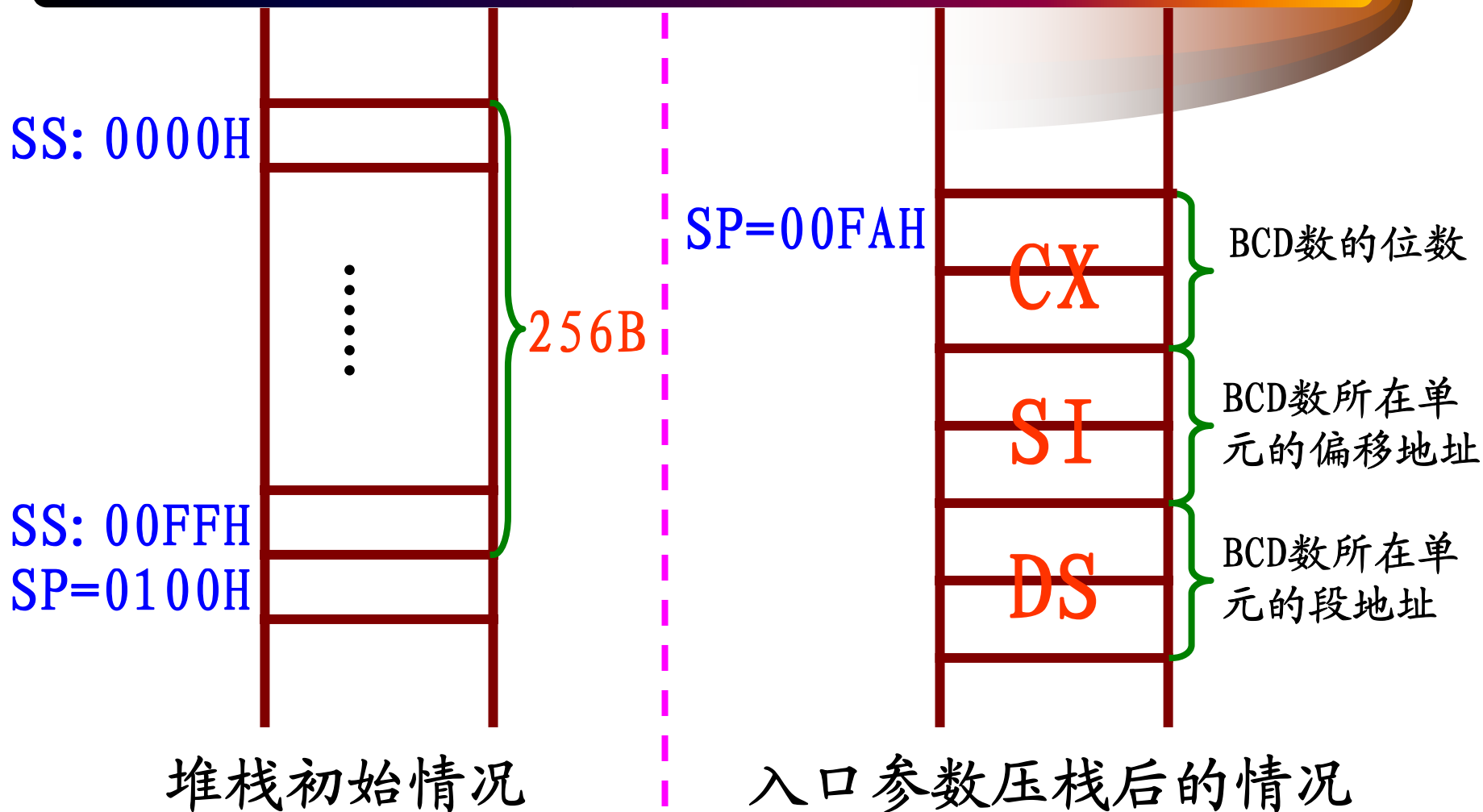
MOV CX, LENG1

PUSH CX

将被转换的BCD数所在单元  
段地址、偏移地址、BCD数的  
位数依次压栈，传递给子程序

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义



# 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

```
CALL FAR PTR BCD-16B
```

```
POP DX
```

```
MOV RESULT, DX
```

```
MOV AH, 4CH
```

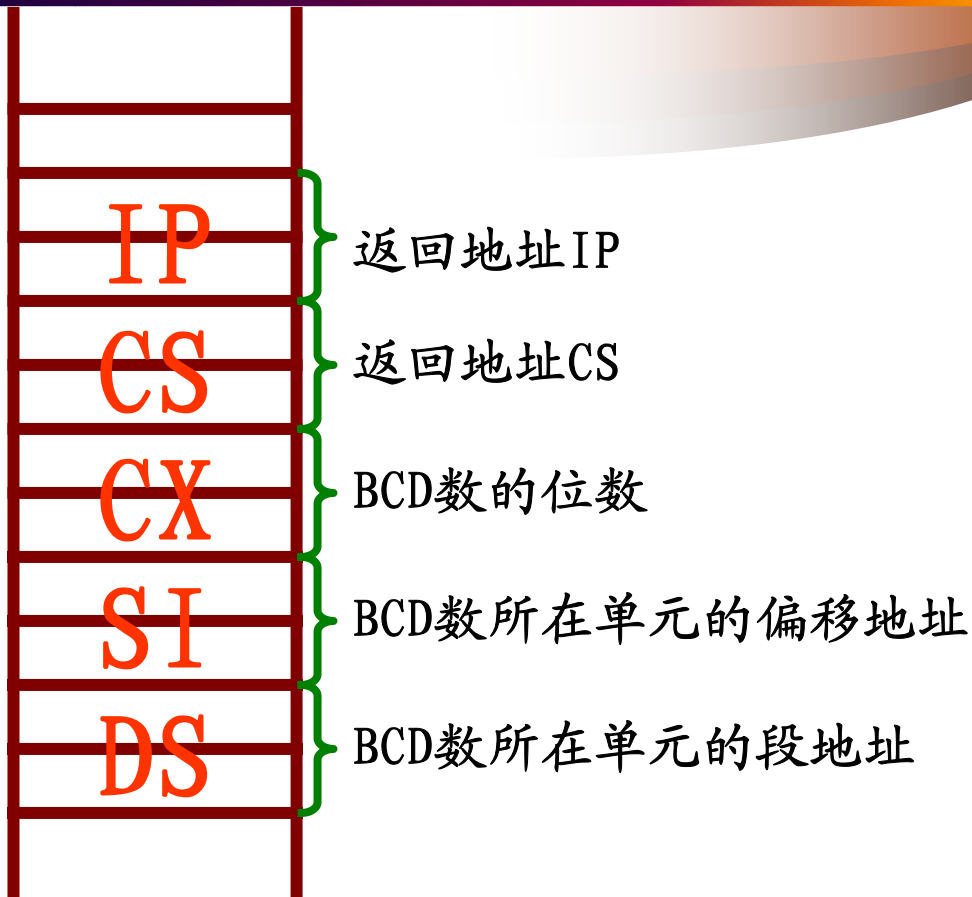
```
INT 21H
```

```
CODEM ENDS
```

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

SP=00F6H



CALL指令后堆栈的情况

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

; 子程序（下面是子程序说明文件）

; ① 子程序名：BCD-16B

; ② 子程序完成的功能：将一组BCD数转换  
; 成16位二进制数

; ③ 入口参数：被转换的BCD数所在单元  
; 的段地址、EA以及BCD数的  
; 位数依次压在堆栈中

## 4.3 汇编语言程序设计的基本方法

### ——子程序及过程定义

- ; ④ 出口参数: 转换后的16位二进制数在  
; 堆栈中。
- ; ⑤ 用到的寄存器: 无

# 4.3 汇编语言程序设计的基本方法

——子程序及过程定义

```
CODES      SEGMENT
            ASSUME  CS: CODES
BCD-16B    PROC  FAR
            PUSH   BP
            MOV    BP, SP
            PUSH   DS
            PUSH   SI
            PUSH   CX
            PUSH   BX
            PUSH   AX
```

SP=00EAH

AX

BX

CX

SI

DS

(BP) + 0

BP

+2

IP

返回地址IP

+4

CS

返回地址CS

+6

CX

BCD数的位数

+8

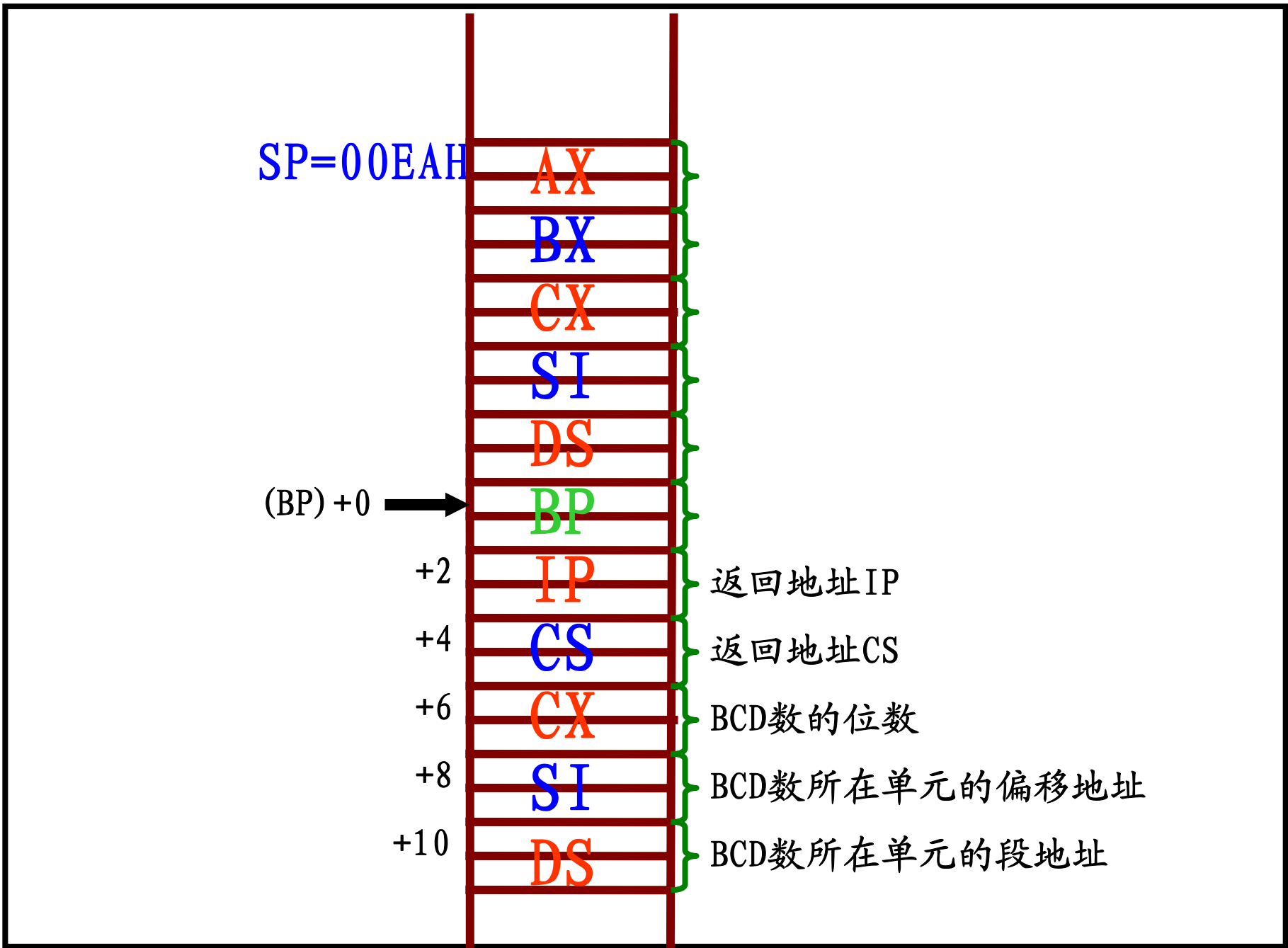
SI

BCD数所在单元的偏移地址

+10

DS

BCD数所在单元的段地址





## 4.3 汇编语言程序设计的基本方法

### ——子程序及过程定义

```
MOV    CX, [BP] + 6
MOV    SI, [BP] + 8
MOV    DS, [BP] + 10
ADD    SI, CX
MOV    DX, 0
BCDL:  DEC    SI
        PUSH  CX
        MOV   AL, [SI]
```

## 4.3 汇编语言程序设计的基本方法

### ——子程序及过程定义

```
AND    AL, 0FH
CBW
MOV    BX, AX
MOV    AX, DX
MOV    CX, 10
MUL   CX
MOV    DX, AX
ADD    DX, BX
POP    CX
LOOP  BCDL
```

# 4.3 汇编语言程序设计的基本方法

## ——子程序及过程定义

```
MOV    [BP]+10, DX
POP    AX
POP    BX
POP    CX
POP    SI
POP    DS
POP    BP
RET    4
BCD-16B ENDP
CODES  ENDS
END    START
```

SP=00EAH

AX

BX

CX

SI

DS

(BP) + 0

BP

+2

IP

返回地址IP

+4

CS

返回地址CS

+6

CX

BCD数的位数

+8

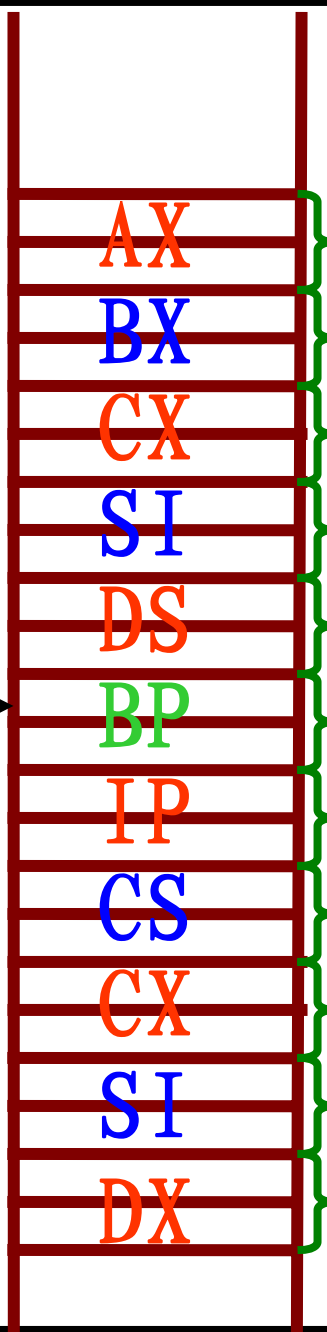
SI

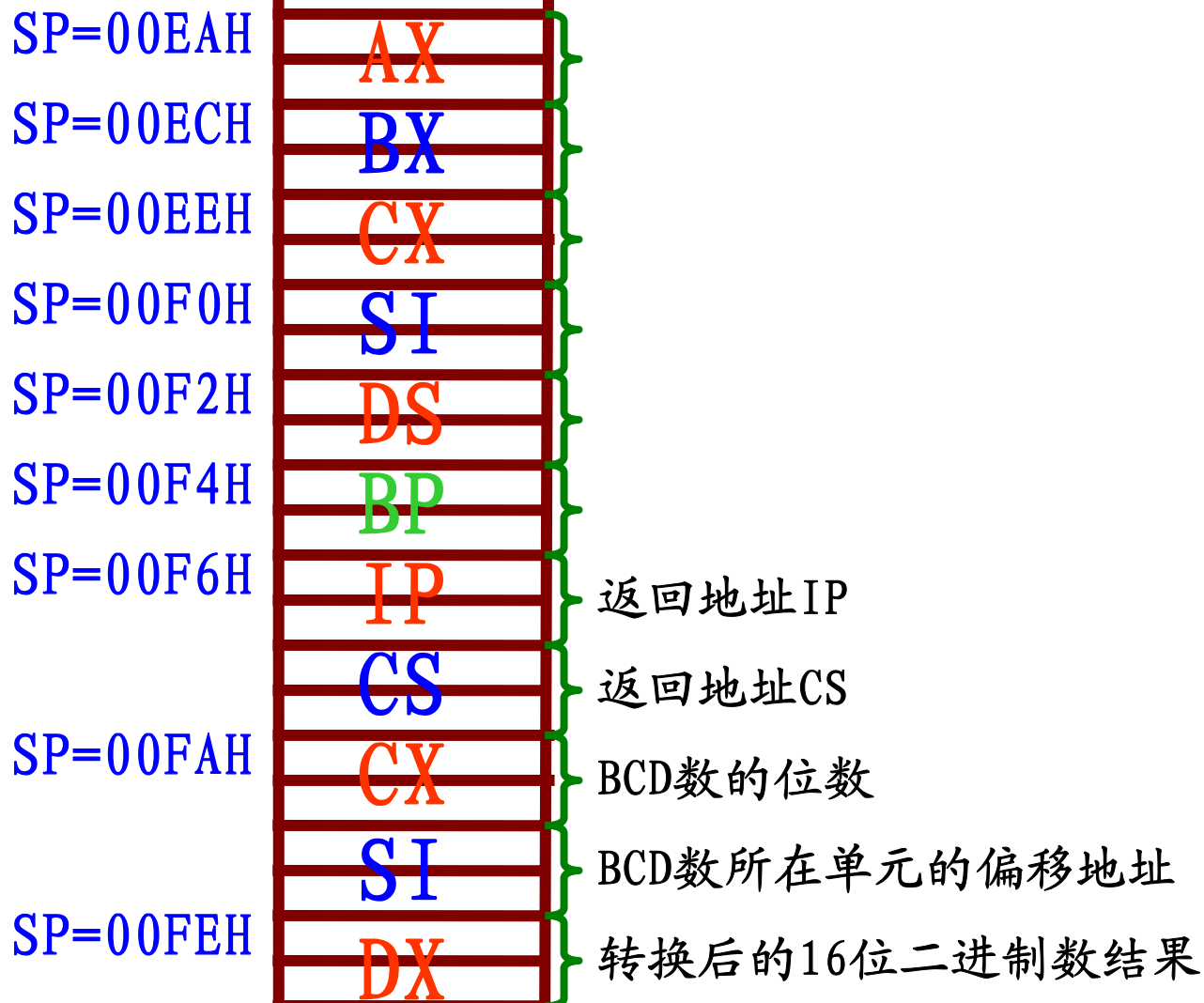
BCD数所在单元的偏移地址

+10

DX

转换后的16位二进制数结果





# 4.3 汇编语言程序设计的基本方法

——子程序及过程定义



作业:

P189

17.

20.

21.