计算机组成原理 Homework12 (11.18)

中国人民大学　信息学院　崔冠宇　2018202147

1. 在 200H~20FH 处存放了 8 个 word (16 位数据):

200H: 06H, 00H, 08H, 00H, 0CH, 00H, 1FH, 00H

208H: A6H, 00H, 80H, 00H, 30H, 00H, 09H, 00H

要求扫描 8 个数据, 统计其中出能被 4 整除的数据放在 210H 开始处.

解: 需要用到循环结构和选择结构.

```
MOV CX, 8          ; 8次循环
MOV SI, 200        ; 源地址
MOV DI, 210        ; 目标地址
MOV BL, 4
L:                 ; 循环体(10B)
MOV AX, [SI]       ; 读入该数
DIV BL
CMP AH, 0          ; 检验能否被4整除.
JNE E              ; 不能被整除, 跳转, 不执行数据传送.
MOV AX, [SI]
MOV [DI], AX       ; 可被整除, 进行数据传送.
ADD DI, 2
E:                 ; (11B)
ADD SI, 2
DEC CX
JNZ L
INT 3
```

在 DEBUG 环境中，首先用 -e 命令将地址为 200H~20FH 的单元的数据修改为对应值，然后用 -a 键入代码并用 -g 命令执行，最后用 -d 观察执行效果. 部分截图如下所示:

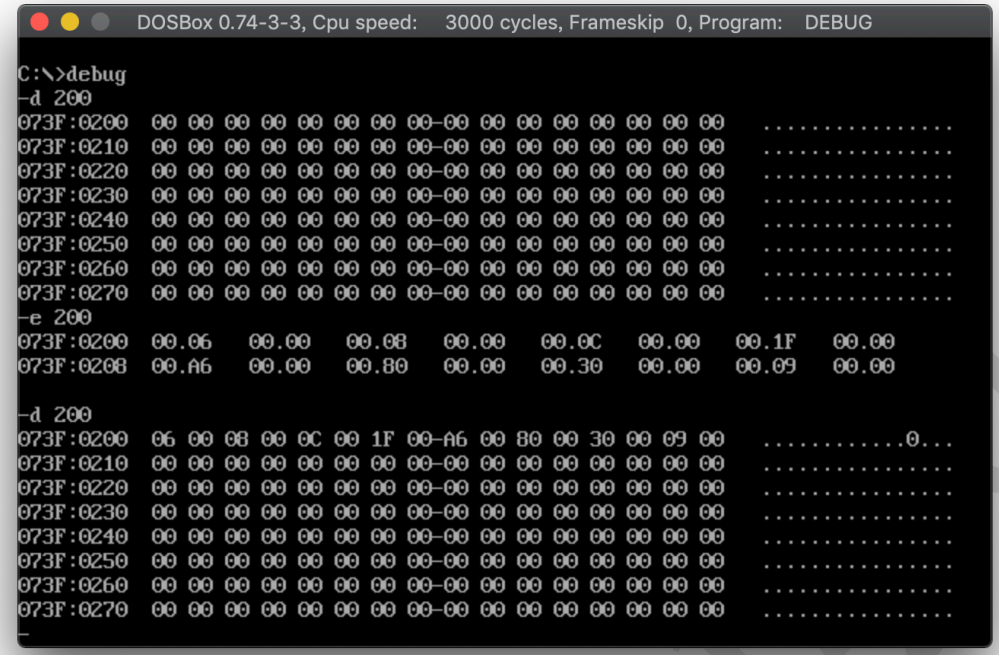

图 1: 设置内存单元的值

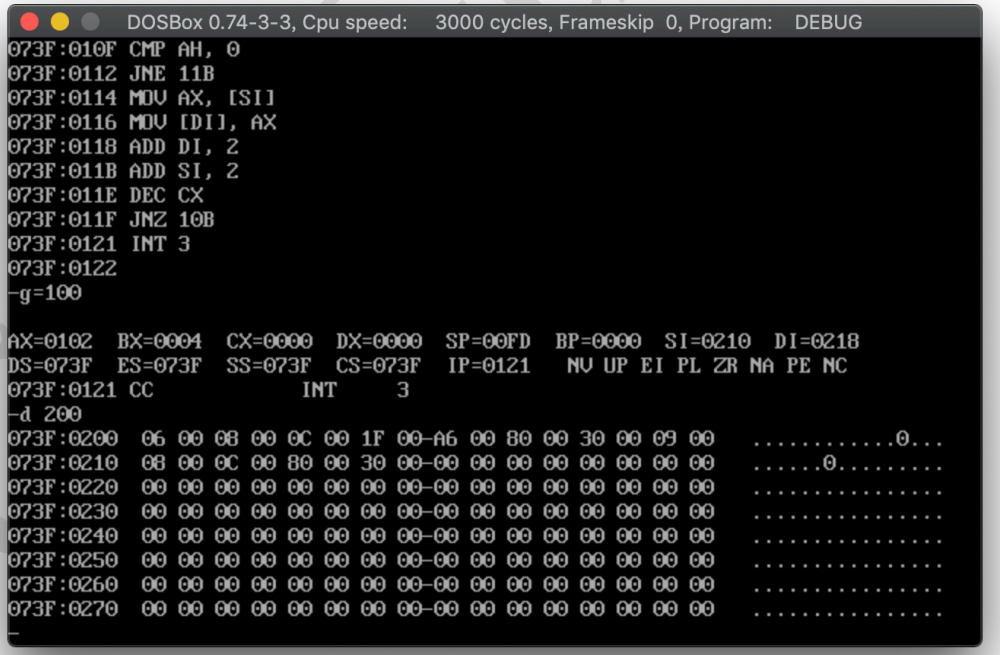

图 2: 键入代码并执行后内存单元的值

根据程序运行结果可见代码功能正确.

2. 在 200H 和 204H 分别存放了 4 个单元 BCD 码:

200H: 56H, 34H, 82H, 12H, 表示十进制数: 12823456

204H: 78H, 45H, 23H, 25H, 表示十进制数: 25234578

编程计算实现两数之和 BCD 码运算, 进位在 CY.

解: 需要用到循环结构.

```
MOV CX, 4          ; 4次循环
MOV SI, 200
MOV DI, 204
CLC                ; CF <- 0
L:
MOV AL, [SI]
ADC AL, [DI]
DAA
MOV [SI], AL
INC SI
INC DI
DEC CX
JNZ L
INT 3
```

在 DEBUG 环境中，首先用 -e 命令将地址为 200H~207H 的单元的数据修改为对应值，然后用 -a 键入代码并用 -g 命令执行，最后用 -d 观察执行效果. 部分截图如下所示:

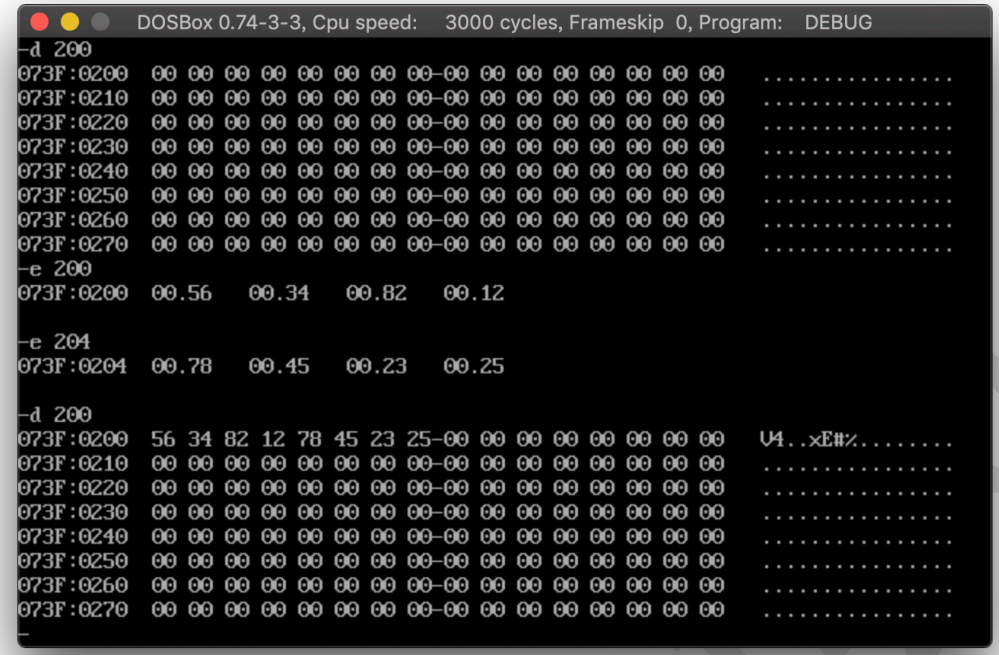

图 3: 设置内存单元的值

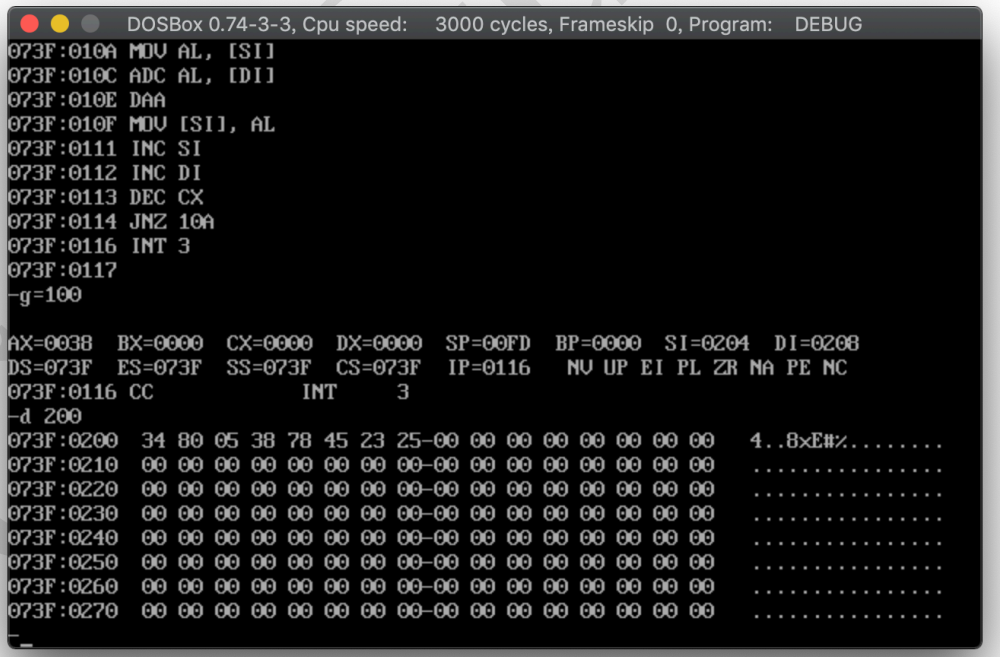

图 4: 键入代码并执行后内存单元的值

根据程序运行结果可见代码功能正确.

3. 编写程序, 在 200H 开始的数据单元存放了一个 ASCII 码的 5 位数字, 数字大小自己设定 (不要超过 65535). 编程要求转换该数字字符串为一个 2 字节的二进制数, 结果在放在 210H~211H 中.

解: 需要用到循环结构. 约定数据 ASCII 数组按大端方式存放.

```
MOV CX, 5          ; 5次循环
MOV SI, 200
MOV AX, 0          ; sum = 0
L:
MOV BX, A          ; BX <- 0xA (10)
MUL BX             ; sum = sum * 10, (DX, AX)=AX * 0xAH
MOV BL, [SI]       ; BL <- char
SUB BL, 30         ; BL <- BL - 48, ASCII 转数字 c
ADD AX, BX         ; sum += c
INC SI
DEC CX
JNZ L
MOV [210], AX
INT 3
```

在 DEBUG 环境中，首先用 -e 命令将地址为 200H~204H 的单元的数据修改 31H、32H、33H、34H、35H，然后用 -a 键入代码并用 -g 命令执行，最后用 -d 观察执行效果. 部分截图如下所示:

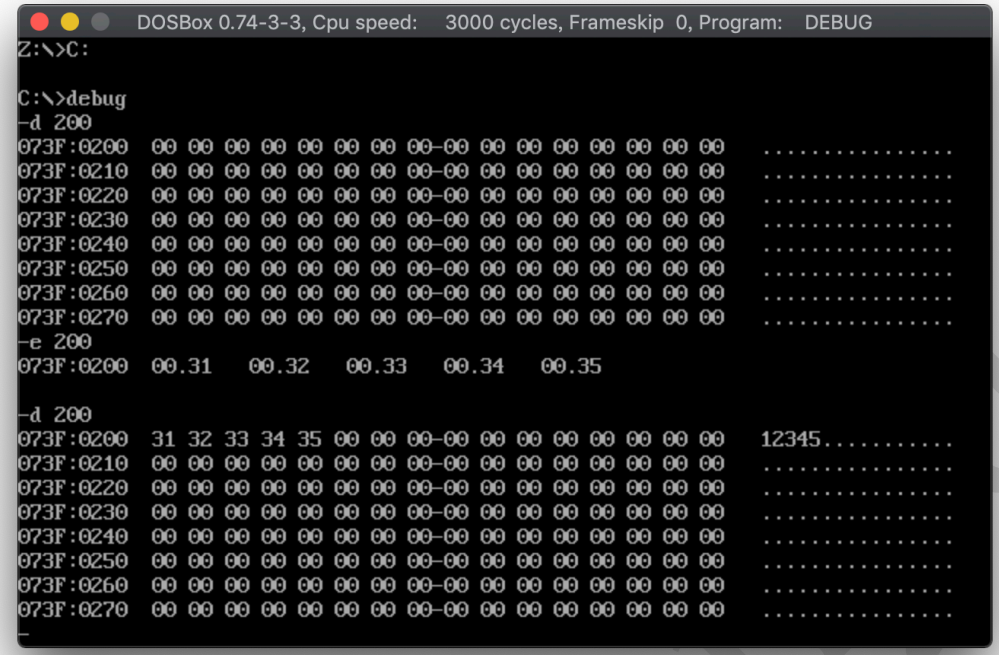

图 5: 设置内存单元的值

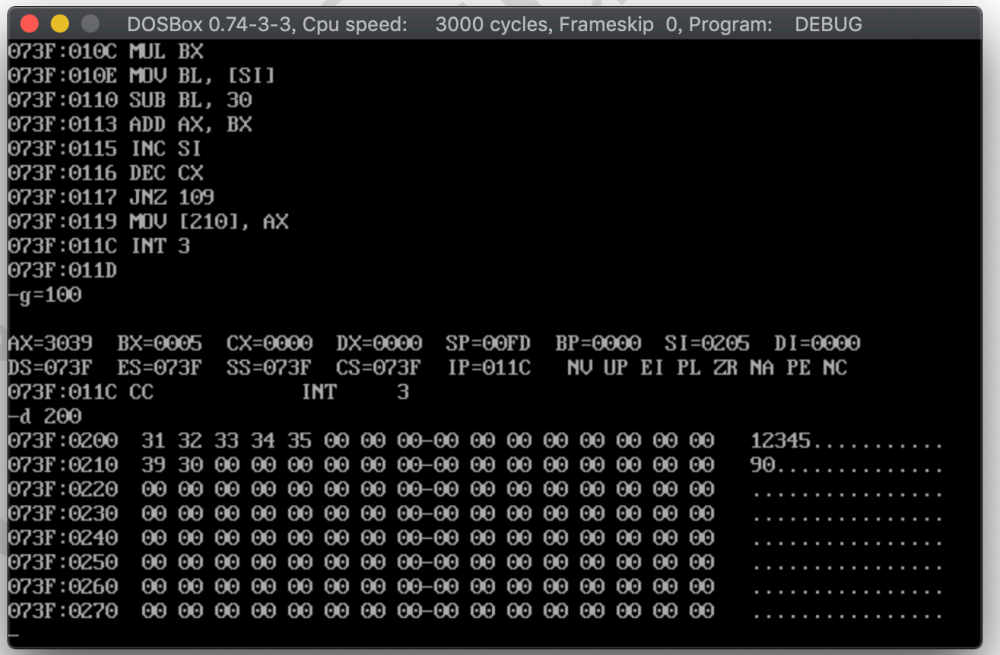

图 6: 键入代码并执行后内存单元的值

根据程序运行结果可见代码功能正确.

4. 编写程序, 在 200H 开始的数据单元存放了 2 个字节的 16 位二进制数, 编程要求将其转换为十进制数 ASCII 码表示的字符串, 结果放在 210H 起始的单元中.

解: 需要用到循环结构. 约定数据 ASCII 数组按小端方式存放.

```
MOV DI, 210
MOV AX, [200]    ; num <- [200]
MOV BX, A        ; BX <- 0xA (10)
L:
MOV DX, 0        ; DX <- 0, 32位除法准备
DIV BX           ; c = num % 10, num /= 10
ADD DL, 30       ; c = c + 30H, 转成 ASCII
MOV [DI], DL     ; [DI] <- c
INC DI
CMP AX, 0        ; num = 0, 结束
JNE L
INT 3
```

在 DEBUG 环境中，首先用 -e 命令将地址为 200H~201H 的单元的数据修改 39H、30H，然后用 -a 键入代码并用 -g 命令执行，最后用 -d 观察执行效果. 部分截图如下所示:

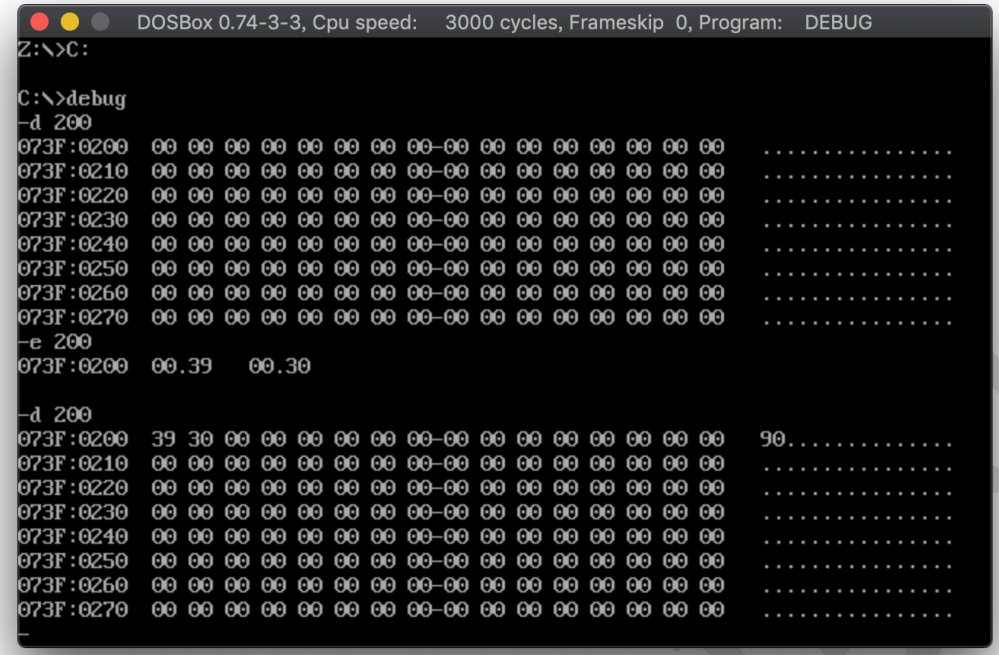

图 7: 设置内存单元的值

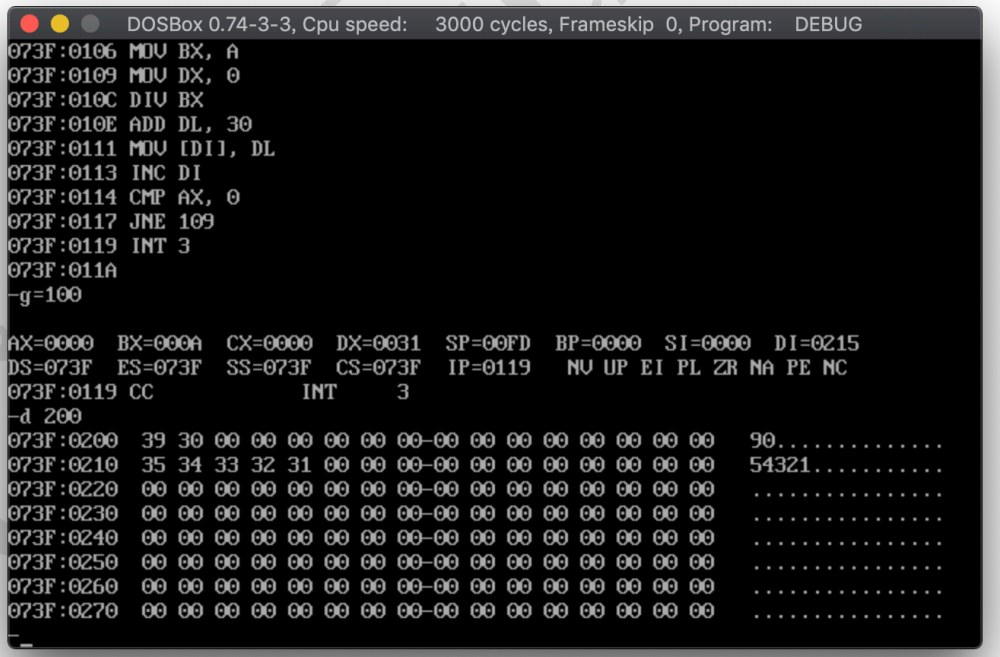

图 8: 键入代码并执行后内存单元的值

根据程序运行结果可见代码功能正确.