

## 4. Chapter4

### Chapter 4

4. 下列指令分别完成什么功能.

(1) `MOV AL, NOT 10001110B`  $10001110B$  取非后得值运到AL中  $AL \leftarrow 01110001B$

(2) `MOV CX, 8 GT 00011000B`  $8 < 00011000B$  小, 结束为假  $CX \leftarrow 0000H$

(3) `MOV DL, 27/5`  $27/5 = 5$   $DL \leftarrow 5$

(4) `MOV BX, $-2257` 将变量长度赋予BX (现行地址偏移量 - 2257 偏移地址) 结果送入BX.

5. 阅读下列程序段, 说明每条指令执行后结果是什么.

$X_1$  DB 15H, 78H

$X_2$  DW 06FFH, 5200H

$X_3$  DD ?

GD: `MOV AL, TYPE X1`  $AL = 01H$

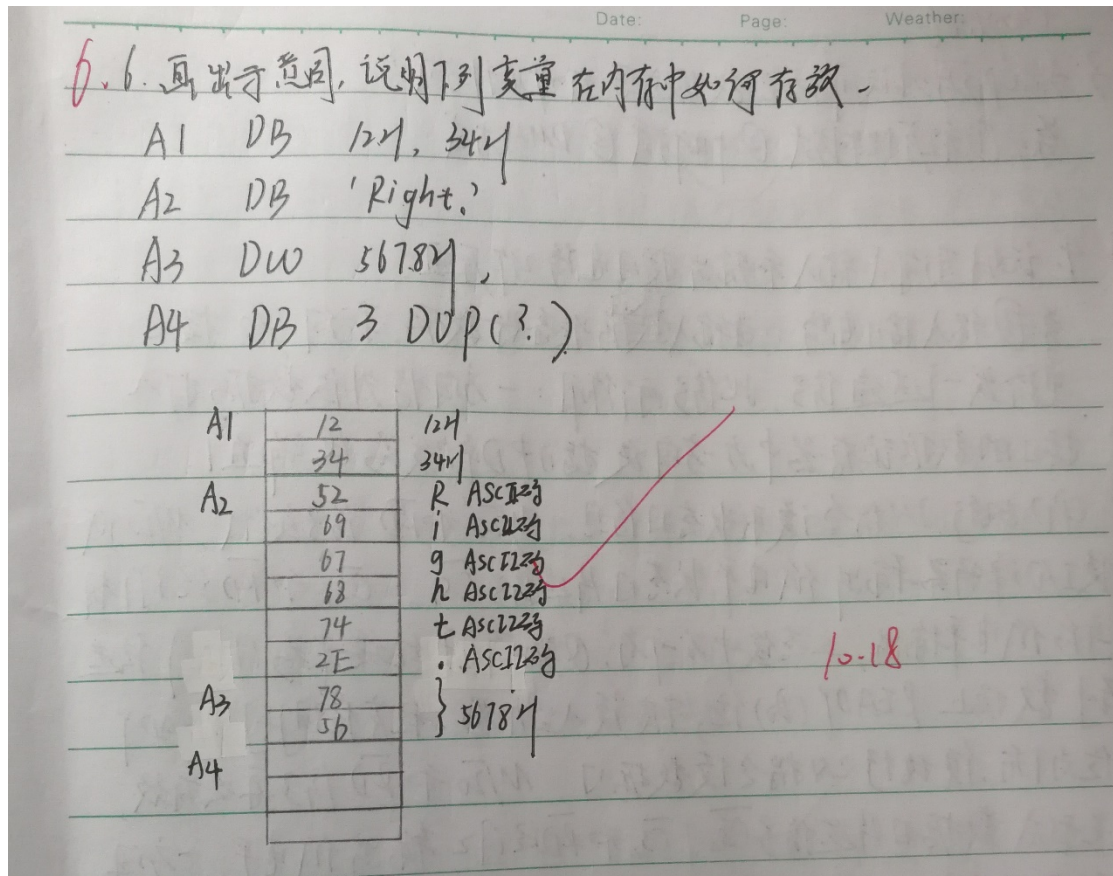
`MOV BL, TYPE X2`  $BL = 02H$

`MOV CL, TYPE X3`  $CL = 04H$

`MOV AH, TYPE GD`  $AH = 0FFH$

`MOV BH, SIZE X2`  $BH = 02H$

`MOV CH, LENGTH X3`  $CH = 01H$



第 14 题：在 TABLE 开始的内存字节单元中，存放了 12 个带符号数，编写完整的汇编语言程序统计其中的正数、负数和零的个数，分别存入 PLUS、NEG 和 ZERO 单元中。（参考例 3.89）

```
.model small
.stack 100h
```

```
.data
TABLE DW 12 DUP(?) ; 假设 TABLE 是包含 12 个带符号字的数组
PLUS DW ? ; 用于存放正数的个数
NEG DW ? ; 用于存放负数的个数
ZERO DW ? ; 用于存放零的个数
```

```
.code
start:
```

```
MOV AX, @DATA ; 初始化数据段寄存器
MOV DS, AX
```

```
MOV CX, 12 ; 设置循环计数器，因为有 12 个数
LEA SI, TABLE ; 将 TABLE 的地址加载到 SI 寄存器中
XOR AX, AX ; 清除 AX 寄存器，用于计数
```

```

MOV PLUS, AX      ; 初始化正数计数器
MOV NEG, AX       ; 初始化负数计数器
MOV ZERO, AX      ; 初始化零计数器

count_loop:
MOV AX, [SI]      ; 将当前 SI 指向的值加载到 AX 中
CMP AX, 0         ; 比较 AX 与 0
JE zero_case      ; 如果 AX 等于 0，跳转到 zero_case
JG positive_case  ; 如果 AX 大于 0，跳转到 positive_case
; 如果不满足上述条件，则 AX 小于 0，即负数
INC [NEG]         ; 增加负数计数器
JMP next_number   ; 跳转到下一个数

zero_case:
INC [ZERO]        ; 增加零计数器
JMP next_number   ; 跳转到下一个数

positive_case:
INC [PLUS]        ; 增加正数计数器

next_number:
ADD SI, 2         ; 移动到下一个数（假设每个数是 16 位的）
LOOP count_loop   ; 递减 CX 并检查是否为零，如果不为零则跳转到
count_loop

; 程序结束，可以返回 DOS 或进行其他操作
MOV AH, 4Ch       ; DOS 功能调用，结束程序
INT 21h

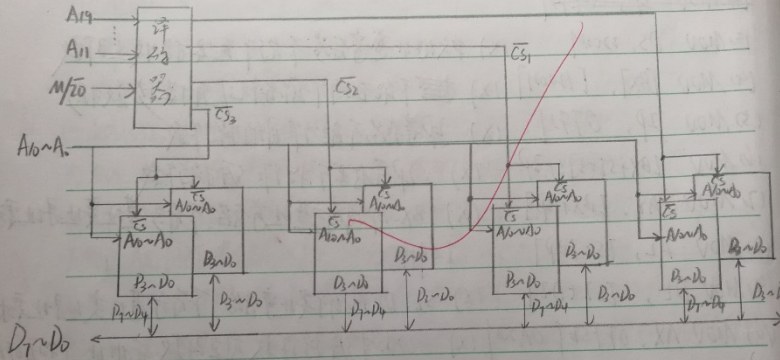
end start

```

## 5. Chapter5

chapter 3

15. 共需要 8 块 RAM 芯片, 必须将地址线  $A_0 \sim A_9$  直接连到每个存储器芯片上, 并用地址线  $A_{11} \sim A_{19}$  作为地址译码器输入, 需要译码器产生 4 个片选信号



17.  $2k \times 8 \rightarrow 8k \times 8$

