

《微机原理及应用》第一章习题答案

习题与思考题

1.1、 在计算机中为什么都采用二进制数而不采用十进制数？二进制数有哪两种缩写形式？

[解] 二进制数只有两个状态，而十进制数有十个状态，……

有八进制和十六进制两种缩写形式：xxxxQ, xxxxH。

1.2、 将下列十进制数转换为二进制数：50, 0.83, 24.31, 79.75, 199, 73.25

[解] $50 \rightarrow 00110010B$; $0.83 \rightarrow 0.1101010001\cdots B$; $24.31 \rightarrow 11000.01001111\cdots B$

$79.75 \rightarrow 01001111.11B$; $199 \rightarrow 11000111B$; $73.25 \rightarrow 1001001.01B$

1.3、 将下列十进制数转换为八进制和十六进制数：39, 99.735, 54.625, 127, 119

[解] $39 \rightarrow 47Q \rightarrow 27H$; $99.735 \rightarrow 123.5702Q \rightarrow 63.BC28H$; $54.625 \rightarrow 66.5Q \rightarrow 36.AH$

$127 \rightarrow 177Q \rightarrow 7FH$; $119 \rightarrow 167Q \rightarrow 77H$

1.4、 将下列二进制数转换为十进制数：

11 1101.101B, 10 0101.11B, 1001 1001.001B, 110 0110.011B

1101 1010.1101B

[解] $11\ 1101.101B \rightarrow 61.625$; $10\ 0101.11B \rightarrow 37.75$; $1001\ 1001.001B \rightarrow 153.125$

$110\ 0110.011B \rightarrow 102.375$; $1101\ 1010.1101B \rightarrow 218.8125$

1.5、 完成下列转换：

(1) 10 110.10 111B 转换为十六进制数；

(2) 34.97H 转换为八进制数和十进制数；

(3) 0BA.7FH 转换为二进制数和八进制数；

(4) 43.27Q 转换为二进制数和十六进制数；

[解] (1) $10\ 110.10\ 111B \rightarrow 16.B8H$; (2) $34.97H \rightarrow 64.456Q \rightarrow 52.59$

(3) $0BA.7FH \rightarrow 10111010.01111111B \rightarrow 272.376Q$

(4) $43.27Q \rightarrow 100011.010111B \rightarrow 23.5CH$

1.6、 设机器字长为 8 位，写出下列用真值表示的二进制数的原码、补码和反码：

+0010101, +1111111, +1000000, -0010101, -1111111, -1000000

[解] +0010101 的原码、补码和反码均为 00010101B;

+1111111 的原码、补码和反码均为 01111111B;

+1000000 的原码、补码和反码均为 01000000B;

-0010101 的原码为 10010101B, 补码为 11101011B, 反码为 11101010B;

-1111111 的原码为 11111111B, 补码为 10000001B, 反码为 10000000B;

-1000000 的原码为 11000000B, 补码为 11000000B, 反码为 10111111B。

1.7、 设机器字长为 8 位，最高位为符号位，用二进制补码运算法则对下列各式进行运算：

(1) $17+7$; (2) $8+18$; (3) $9+(-7)$; (4) $-26+6$; (5) $8-18$;

(6) $19-(-17)$; (7) $-25-6$; (8) $87-15$

[解] (1) $17+7$ (2) $8+18$ (3) $9+(-7)$

$[17]_{补} \rightarrow 00010001B$

$[8]_{补} \rightarrow 00001000B$

$[9]_{补} \rightarrow 00001001B$

$+ [7]_{补} \rightarrow 00001111B$

$+ [18]_{补} \rightarrow 00010010B$

$+ [-7]_{补} \rightarrow 11111001B$

$00011000B \rightarrow 24$

$00011010B \rightarrow 26$

$00000010B \rightarrow 2$

(4) $-26+6$	(5) $8-18$	(6) $19-(-17)$
$[-26]_{\text{补}} \rightarrow 11100110\text{B}$	$[8]_{\text{补}} \rightarrow 00001000\text{B}$	$[19]_{\text{补}} \rightarrow 00010011\text{B}$
$+) \quad [6]_{\text{补}} \rightarrow 00000110\text{B}$	$+) \quad [-18]_{\text{补}} \rightarrow 11101110\text{B}$	$+) \quad [-(-17)]_{\text{补}} \rightarrow 00010001\text{B}$
$11101100\text{B} \rightarrow [-20]_{\text{补}}$	$11110110\text{B} \rightarrow [-10]_{\text{补}}$	$00100100\text{B} \rightarrow 36$

(7) $-25-6$;	(8) $87-15$
$[-25]_{\text{补}} \rightarrow 11100111\text{B}$	$[87]_{\text{补}} \rightarrow 01010111\text{B}$
$+) \quad [-6]_{\text{补}} \rightarrow 11111010\text{B}$	$+) \quad [-15]_{\text{补}} \rightarrow 11110001\text{B}$
$1 \downarrow 11100001\text{B} \rightarrow [-31]_{\text{补}}$	$1 \downarrow 01001000\text{B} \rightarrow 72$

1.8、 已知下列各数均为二进制补码:

$a=0011\ 0010\text{B}$; $b=0100\ 1010\text{B}$; $c=1110\ 1001\text{B}$; $d=1011\ 1010\text{B}$ 。

试求: (1) $a+b$; (2) $a+c$; (3) $c+b$; (4) $c+d$; (5) $a-b$; (6) $c-a$; (7) $d-c$; (8) $a+d-c$;
(9) $b+c-d$; (10) $d-c-a$

[解] (1) $a+b=0011\ 0010\text{B} + 0100\ 1010\text{B}=01111100\text{B}$
 (2) $a+c=0011\ 0010\text{B} + 1110\ 1001\text{B}=00011011\text{B}$
 (3) $c+b=1110\ 1001\text{B} + 0100\ 1010\text{B}=00110011\text{B}$
 (4) $c+d=1110\ 1001\text{B} + 1011\ 1010\text{B}=10100011\text{B}$
 (5) $a-b=a+[-b]_{\text{补}}=0011\ 0010\text{B} + 10110110\text{B}=11101000\text{B}$
 (6) $c-a=c+[-a]_{\text{补}}=1110\ 1001\text{B} + 11001110\text{B}=10110111\text{B}$
 (7) $d-c=d+[-c]_{\text{补}}=1011\ 1010\text{B} + 00010111\text{B}=11010001\text{B}$
 (8) $a+d-c=a+d+[-c]_{\text{补}}=0011\ 0010\text{B} + 1011\ 1010\text{B} + 00010111\text{B}=00000011\text{B}$
 (9) $b+c-d=b+c+[-d]_{\text{补}}=0100\ 1010\text{B} + 1110\ 1001\text{B} + 01000110\text{B}=11111010\text{B}$
 (10) $d-c-a=d+[-c]_{\text{补}}+[-a]_{\text{补}}=1011\ 1010\text{B} + 00010111\text{B} + 11001110\text{B}=10011111\text{B}$

1.9、 设机器字长为8位, 最高位为符号位, 用双高位法判别下述各二进制运算是否产生溢出, 并说明是正溢出还是负溢出。

(1) $43+8$; (2) $-52+7$; (3) $50+84$; (4) $72-8$; (5) $(-33)+(-47)$;
(6) $(-90)+(-75)$; (7) $-127+60$

[解] (1) $43+8$	(2) $-52+7$	(3) $50+84$
$[43]_{\text{补}} \rightarrow 00101011\text{B}$	$[-52]_{\text{补}} \rightarrow 11001100\text{B}$	$[50]_{\text{补}} \rightarrow 00110010\text{B}$
$+) \quad [8]_{\text{补}} \rightarrow 00001000\text{B}$	$+) \quad [7]_{\text{补}} \rightarrow 00000111\text{B}$	$+) \quad [84]_{\text{补}} \rightarrow 01010100\text{B}$
00110011B	11010011B	10000110B
$C_s=0, C_p=0$	$C_s=0, C_p=0$	$C_s=0, C_p=1$
$V=C_s \oplus C_p=0$	$V=C_s \oplus C_p=0$	$V=C_s \oplus C_p=1$
无溢出	无溢出	有溢出, 正溢出
(4) $72-8$	(5) $-33+(-47)$	(6) $(-90)+(-75)$
$[72]_{\text{补}} \rightarrow 01001000\text{B}$	$[-33]_{\text{补}} \rightarrow 11001100\text{B}$	$[-90]_{\text{补}} \rightarrow 10100110\text{B}$
$+) \quad [-8]_{\text{补}} \rightarrow 11111000\text{B}$	$+) \quad [-47]_{\text{补}} \rightarrow 11010001\text{B}$	$+) \quad [-75]_{\text{补}} \rightarrow 10110101\text{B}$
$1 \downarrow 11000000\text{B}$	$1 \downarrow 10110000\text{B}$	$1 \downarrow 11011011\text{B}$
$C_s=1, C_p=1$	$C_s=1, C_p=1$	$C_s=1, C_p=0$
$V=C_s \oplus C_p=0$	$V=C_s \oplus C_p=0$	$V=C_s \oplus C_p=1$
无溢出	无溢出	有溢出, 负溢出

(7) $-127+60$

$[-127]_{\text{补}} \rightarrow 10000001\text{B}$

$+ [60]_{\text{补}} \rightarrow 00111100\text{B}$

10111101B

$C_s=0, C_p=0$

$V=C_s \oplus C_p=0$

无溢出

1. 10、a, b 均为用十六进制形式表示的 8 位带符号数补码, 按下列给定的 a, b 之值进行 $a+b$ 和 $a-b$ 的运算, 并用双高位法判断是否产生溢出:

(1) $a=37, b=57$; (2) $a=0\text{B}7\text{H}, b=0\text{D}7\text{H}$; (3) $a=0\text{F}7\text{H}, b=0\text{D}7\text{H}$; (4) $a=37\text{H}, b=0\text{C}7\text{H}$

[解] (1) $a=37, b=57$

$a+b$

$[37]_{\text{补}} \rightarrow 00100101\text{B}$

$+ [57]_{\text{补}} \rightarrow -00111001\text{B}$

01011110B

$C_s=0, C_p=0$

$V=C_s \oplus C_p=0$

无溢出

$a-b$

$[37]_{\text{补}} \rightarrow 00100101\text{B}$

$+ [-57]_{\text{补}} \rightarrow -11000111\text{B}$

11101011B

$C_s=0, C_p=0$

$V=C_s \oplus C_p=0$

无溢出

(2) $a=0\text{B}7\text{H}, b=0\text{D}7\text{H}$

$a+b$

10110111B

$+ 11010111\text{B}$

1 : 10001110B

$C_s=1, C_p=1$

$V=C_s \oplus C_p=0$

无溢出

$a-b$

10110111B

$+ 00101001\text{B}$

11100000B

$C_s=0, C_p=0$

$V=C_s \oplus C_p=0$

无溢出

(3) $a=0\text{F}7\text{H}, b=0\text{D}7\text{H}$

$a+b$

11110111B

$+ 11010111\text{B}$

1 : 11001110B

$C_s=1, C_p=1$

$V=C_s \oplus C_p=0$

无溢出

$a-b$

11110111B

$+ 00101001\text{B}$

1 : 00100000B

$C_s=1, C_p=1$

$V=C_s \oplus C_p=0$

无溢出

(4) $a=37\text{H}, b=0\text{C}7\text{H}$

$a+b$

00110111B

$+ 11000111\text{B}$

11111110B

$C_s=0, C_p=0$

$V=C_s \oplus C_p=0$

无溢出

$a-b$

00110111B

$+ 00111001\text{B}$

01110000B

$C_s=0, C_p=0$

$V=C_s \oplus C_p=0$

无溢出

1. 11、 将下列十进制数变为 8421BCD 码:

- (1) 8609; (2) 5254; (3) 2730; (4) 1998

[解] (1) 8609→8609H 或 1000, 0110, 0000, 1001B

(2) 5254→5254H 或 0101, 0010, 0101, 0100B

(3) 2730→2730H 或 0010, 0111, 0011, 0000B

(4) 1998→1998H 或 0001, 1001, 1001, 1000B

1. 12、 将下列 8421BCD 码表示成十进制数和二进制数:

- (1) 01111001; (2) 01010111; (3) 10000011; (4) 10011001

[解] (1) 01111001→79→01001111B

(2) 01010111→57→00111001B

(3) 10000011→83→01010011B

(4) 10011001→99→01100011B

1. 13、 将下列数值或字符串表示成相应的 ASCII 代码:

- (1) 51; (2) 7FH; (3) ABH; (4) C6H; (5) SAM; (6) JONS; (7) Hello;

(8) how are you?

[解] (1) 51→35H, 31H

(2) 7FH→37H, 46H, 48H

(3) ABH→41H, 42H, 48H

(4) C6H→43H, 36H, 48H

(5) SAM→53H, 41H, 4DH

(6) JONS→4AH, 4FH, 4EH, 53H

(7) Hello→48H, 65H, 6CH, 6CH, 6FH

(8) how are you? →68H, 6FH, 77H, 20H, 61H, 72H, 65H, 20H, 79H, 6FH, 75H, 3FH

1. 14、 有一个 16 位的数值 0101, 0000, 0100, 0011

(1) 如果它是一个二进制数, 和它等值的十进制数是多少?

(2) 如果它们是 ASCII 码字符, 则是些什么字符?

(3) 如果是压缩的 BCD 码, 它表示的数是什么?

[解] (1) 20547

(2) PC

(3) 5043H

《微机原理及应用》第二章习题答案

2-10. [解] 逻辑地址为: 0FFFFH:0000H, 物理地址 PA=0FFFF0H

2-11. [解]

12340H	7B
	41
	42
	24
	30
	39
	30
	21
	78
	56
	34
	12

2-12. 有一个由 20 个字组成的数据区, 其起始地址为 610AH: 1CE7H, 试写出该数据区首末单元的实际地址 PA。

[解] 首地址 PA=62D87H, 末地址 PA=62DAEH。

2-13. 存储器中每一个段最多为 64K 字节, 当程序 routadf 运行后, 用 DEBUG 命令显示出当前各寄存器的内容如下, 要求: (1) 画出此时存储器分段示意图; (2) 写出各状态标志的值。

B>C:debug routadf.exe

~r

AX=0000 BX=0000 CX=006D DX=0000 SP=00C8 BP=0000 SI=0000 DI=0000
DS=53A7 ES=11A7 SS=21BE CS=31B8 IP=0000 NV UP EI PL NZ NA PO NC

[解] (1) 数据段: 53A70H~63A6FH ; 附加段: 11A70H~21A6FH
堆栈段: 21BE0H~31BDFH ; 代码段: 31B80H~41B7FH
只有堆栈段和代码段之间有地址重叠, 其它段之间均无重叠。

(2) OF=0 DF=0 IF=1 SF=0 ZF=0 AF=0 PF=0 CF=0

2-14. 已知 SS=20A0H, SP=0032H, AX=0FF42H, SI=537AH, BL=5CH, 将 CS=0A5BH, IP=0012H 推入堆栈后, 再执行下列指令,

```
PUSH AX
PUSH BX ; 位置 1
PUSH SI ; 位置 2
POPF ; 位置 3
```

(1) 画出堆栈存放示意图;

(2) 画出指令执行到位置 1、位置 2 和位置 3 时堆栈区及 SP 指针内容的变化示意图。

图中应标出存储单元的实际地址 PA。

[解]

SP=0028H	②	20A28H	7 A	SI
		20A29H	5 3	
SP=002AH	①	20A2AH	5 C	BX
③		20A2BH	X X	
		20A2CH	4 2	AX
		20A2DH	F F	
		20A2EH	1 2	IP
		20A2FH	0 0	
		20A30H	5 B	CS
		20A31H	0 A	
SP=0032H		20A32H	X X	

《微机原理及应用》第三章习题答案

3-1

序号	目的操作数	源操作数
(1)	寄存器寻址	立即寻址
(2)	寄存器寻址	存储器寻址(直接寻址)
(3)	寄存器寻址	存储器寻址 (基址变址寻址)
(4)	存储器寻址 (寄存器间接寻址)	寄存器寻址
(5)	寄存器寻址	存储器寻址 (寄存器相对寻址)
(6)	隐含寻址	寄存器寻址

3-2 (1) 错

改: MOV AL, [SI]
MOV [DI], AL

或: MOV AX, [SI]
MOV [DI], AX

(2) 错, CS 不能做目的操作数。

(3) 错, -25 为有符号数, 应用 IMUL 指令, 此外, 乘法指令中不能采用立即寻址,。

改: MOV BL, -25
IMUL BL

(4) 错, 目的操作数不能用立即寻址。

(5) 错, 两个段寄存器之间不能直接进行数据传送。

改: MOV AX, CS
MOV DS, AX

(6) 错, SI 和 DI 不能同时出现在一个操作数的寻址方式中。

(7) 错, 将标志寄存器的内容推入堆栈, 用专门的指令: PUSHF

(8) 错, 寄存器间接寻址不能用 AX 寄存器。

改: MOV AX, [BX]

(9) 错, 操作数位数不匹配。

改: MOV AX, WORD PTR[SI]
或: MOV AL, BYTE PTR[SI]

(10) 错, 操作数位数不匹配。

改: MOV CX, AX
或: MOV CL, AL

(11) 错, OUT 指令中指令用 DX 提供端口地址。

改: OUT DX, AL

(12) 错, 同一操作数的寻址方式中不能同时出现 BX 和 BP。

(13) 正确。

(14) 正确。

(15) 错, 移位次数不为 1, 不能直接在指令中写出。

改: MOV CL, 5
SAR AX, CL

3-3 本题已知条件不够。

若 P、Q 都视为无符号数, 分别填: JA、JB、JA

若 P、Q 都视为有符号数, 分别填: JG、JL、JG

3-6 依次填: 2000H、63H (或 99)、JGE、L1、[2100H]

- 3-8 (1) AX=06FAH, CF、ZF、OF 保持不变。
 (2) AL=66H, CF=ZF=OF=0。
 (3) BX=083EH, CF=ZF=OF=0。
 (4) AX=1B25H, CF=ZF=OF=0。
 (5) BX=06F9H, CF=1 (不变), ZF=OF=0。
 (6) CX=0FEBDH, CF=1, ZF=OF=0。
 (7) BL=0FBH, CF=1 (不变), ZF=OF=0。
 (8) DX=01CDH, AX=0E5C2H, CF=OF=1, ZF 保持不变。
 (9) AX=41FCH, CF、ZF、OF 都保持不变。

3-11 题目要求的 3 个功能 (3 个小题) 可用一段程序全部实现。

```

    LEA    SI, ARRAY
    MOV    CX, 7
    LEA    DI, NEW
    MOV    WORD PTR SUM, 0    ; 总分单元清零
    CLD
NEXT: LODSB
    ADD    AL, 5                ; 每人加 5 分
    STOSB
    ADD    BYTE PTR SUM, AL    ; 累加成绩 (结果为 16 位)
    ADC    BYTE PTR SUM+1, 0
    LOOP   NEXT
    MOV    AX, SUM
    MOV    BL, 7
    DIV    BL                    ; 求平均分 (商和余数)
    MOV    AVERAGE, AX
  
```

注: 每个分数不超过 100, 所以为字节数据。但总数可能达到 700>256, 所以总分应为 16 位。

- 3-13 (1) LEA SI, STRING
 LEA DI, GET_CHAR
 MOV CX, 26
 CLD
 REP MOVSB
- (2) 假设字符串 'The computer' 放在附加段中偏移地址为 STR1 的单元。
 LEA SI, STRING
 LEA DI, STR1
 MOV CX, 26
 CLD
 MOV BL, CL
 REPZ CMPSB ; 相同则继续比较
 JZ SAME 相同则至 SAME
 XOR AL, AL ; 否则将 AL 清零
 JMP NUM
 SAME: MOV AL, 1
 NUM: SUB BL, CL ; 计算比较次数
 HLT
- (3) LEA DI, STRING
 CLD
 MOV CX, 26


```

MOV AL, '&'
REPZ SCASB
JNZ DONE
MOV BYTE PTR -1[DI], 20H
; 空格字符 (ASCII 码为 20H) 送原&字符的位置。
DONE: .....
(4) LEA SI, STRING
LEA DI, CAPS
LEA BX, CHART
CLD
MOV CX, 26
LP: LODSB ; 取一个字符
CMP AL, 'A' ; 判断是否大写字母
JB LC ; 否, 跳转
CMP AL, 'Z'
JA LC
STOSB ; 是, 存入大写字母单元
JMP NEXT
LC: MOV [BX], AL ; 存入其他字符单元
INC BX ; 修改地址指针
NEXT: MOV BYTE PTR [SI-1], 0 ; 原字符单元清零
LOOP LP ; 未完循环

```

3-14 题目要求: 假设 AX=1100001110100110
则执行后 DX=0110010111000011, 且执行后 AX 中的内容保持不变。

```

MOV CX, 16
MOV BL, 0 ; 为 1 的位数清零
NEXT: ROR AX, 1 ; 小循环右移, 最低位移入 CF
JNC NO1 ; CF=0, 即原最低位为 0, 跳转
INC BL ; 否则, 位数加 1
NO1: RCL DX, 1 ; CF 移入 DX 最低位
LOOP NEXT ; 未完循环

```

- 3-15 (1) 目标地址: 1200H; 0278H (逻辑地址) 或 12278H (物理地址)
 (2) 1200; 0300H 或 12300H
 (3) 1200H; 4800H 或 16800H
 (4) 3400H; 0ABCH 或 34ABCH
 (5) 00FFH; 4800H 或 057F0H

3-16

	AX	BX	CX	DX
填 LOOP	5	10H	0	0
填 LOOPE	2	4	3	1
填 LOOPNE	3	7	2	0

3-17 中断向量表用于存放中断向量, 即各中断源中断服务程序的入口地址。
 中断向量表位于内存的最低 1KB, 地址范围为: 00000H-003FFH。
 由于 $3 \times 4 = 12 = 0000CH$, 因此类型 3 的中断向量所在地址为: 0000CH-0000FH

00008H	16H
00009H	00H
0000AH	85H
0000BH	04H

3-18 $2 \times 4 = 00008H$, 因此类型 2 的中断向量 0485H:0016H 从 00008H 开始存放, 存放格式如图所示。

3-19 (1) $0040H/4 = 0010H = 16$, 因此中断类型为 16。

(2) 中断服务程序的起始地址为 D169H: 240BH, 物理地址为

$$D169H \times 16 + 240BH = D3A9BH$$

3-22 (1) $DX = 2006H, BX = 0004H$

(2) $SP = 005AH, (SP) = 0FFH$

(3) $CX = 69FBH, BX = 0004H$

(4) $AX = 0094H, ZF = 1$

(5) $SI = 000AH, (SI) = 0C6H$ 无法将 WORD 转换为 PDF

(6) $AH = 0, CF = 0$

$AL = 1BH, CF = 1, AF = 0$

$AL = 81H, AH = 1, CF = 1, AF = 1$

(7) $SI = 000BH$

RAM	执行前	执行后
20506H	06H	06H
20507H	20H	20H
20508H	87H	87H
20509H	15H	15H
2050AH	37H	C6H
50A5AH	C5H	FFH
50A5BH	2FH	69H

4-1、XOR AX ,AX 改为XOR AH,AH

SAL AX,2改为

MOV CL,2

SAL AX,CL

4-2、(1)、(5)常数不能做目的操作数，其他类型不匹配

4-3 、 2, 16, 40, 18

4-4 、 R1=6

R2=12

4-5、

VAR1	09H
	00H
VAR2	XX
	XX
	XX
	XX
	XX
	XX
	XX
	XX
VAR3	XX
	XX
	XX
	XX
	XX
	XX
	XX
	XX
	XX
	XX
	XX
	XX
VAR4	00H
	00H
	41H
	42H
	XX
	00H
	00H
	41H
	42H

4-6 AL=12H,

BX=0003H,

CX=0FFFFH,

SI=0FFFFH,

DX=4112H,

4-7 (1) MOV AX,0FFFFH

(2) ADD CX,0040H

(3) OR AL,03H

(4) AND BL,0A8H
(5) MOV AL,12H
(6) MOV AL,17H
4-8 VAR1: 0010H,
VAR2: 002AH

第五章 存储器

5-4 (1) 2^{14} 字节 = 16KB

(2) $16K \times 8 / 1K \times 1 = 128$ 片

(3) $\log_2^{1024} = 10$

5-5 (1) $16K \times 8 / 1K \times 4 = 32$ 片

(2) $2/64 = 0.03125$ s

(3) 64, 100%。

5-6 系统连接图如下：

		A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	$A_{10} \sim A_0$	地址范围
2#6116	1#6116	0	0	1	0	0	0	0	0	0	0~0	20000H
		0	0	1	0	0	0	0	0	0	1~1	20FFFH
4#6116	3#6116	0	0	1	0	0	0	0	1	0	0~0	21000H
		0	0	1	0	0	0	0	1	1	1~1	21FFFH

5-7 $16K \times 8 / 1K \times 1 = 128$ 片,地址线中有 $\log_2^{1024} = 10$ 位参与片内寻址, 6 位组成片选信号。

5-8 地址分配不连续, 相当数量的地址不能使用, 当 $A_{15} \sim A_{12}$ 同时有两个及以上出现低电平时会造成片选混乱。

芯片地址分配如下: $E000H \sim E3FFH$

$D000H \sim D3FFH$

$B000H \sim B3FFH$

$7000H \sim 73FFH$

5-9

5-11 参考教材 P220 例题。

5-15 有地址重叠, 地址范围如下:

$0000H \sim 07FFH$

$0800H \sim 0FFFH$

$1000H \sim 17FFH$

$1800H \sim 1FFFH$

第六章 中断系统

6-1 中断是一个过程, 是指 CPU 正常运行时, 由于某种随机发生的外部事件而使它暂停执行当前程序, 转而执行另外一段程序, 完成后再返回暂停处继续执行原来的

程序。

中断是计算机技术的重要组成部分，使计算机功能更为完善。主要作用如下：

- 1) 使 CPU 和外设同时工作。
- 2) 实现实时处理
- 3) 故障处理

6-2 参考教材 P 229

6-4 参考教材 P233~P234

6-5 保护现场可以使中断处理结束时，CPU 能正确地返回断点处继续执行下一条指令。

恢复现场能使得程序顺利回到断点处，执行下一条指令。

6-7 中断向量表的设置方法有三种，举例如下：

1.在程序设计时定义一个如下数据段

```
VECDATA    SEGMENT AT 0  
  
            ORG  N*4  
  
INTSUB :    DW  SUBOFFSET, SUBSEG  
  
VECDATA    ENDS
```

其中 N 为要用到的中断类型号；SUBSEG 表示中断服务程序入口的段基址；SUBOFFSET 表示段内偏移地址。

2. 在程序的初始化部分设置几条传送指令，把中断服务程序的入口地址置入中断向量表中

```

        VECDATA    SEGMENT AT 0

                        ORG  N*4

        INTSUB     DW  2DUP(?)

        VECDATA    ENDS

VECSET      SEGMENT

                ASSUME  CS:VECSET,DS: VECDATA

VECINT      PROC  FAR

                MOV  AX,  VECDATA

                MOV  DS,  AX

                MOV  INTSUB,  SUBOFFSET

                MOV  INTSUB+2,  SUBSEG

                RET

VECINT      ENDP

VECSET      ENDS

                END VECINT

```

3. 借助 DOS 功能调用，把中断服务程序的入口地址置入中断向量表中。利用功能调用指令 INT 21H。

6-10

```

IN      AL,  INTPORT

TEST    AL,  80H

JNE     INTSVS_1

IN      AL,  INTPORT

TEST    AL,  40H

```



```

JNE      INTSVS_2
IN        AL,      INTPORT
TEST     AL,      20H
JNE      INTSVS_3
IN        AL,      INTPORT
TEST     AL,      10H
JNE      INTSVS_4
IN        AL,      INTPORT
TEST     AL,      08H
JNE      INTSVS_5
IN        AL,      INTPORT
TEST     AL,      04H
JNE      INTSVS_6
IN        AL,      INTPORT
TEST     AL,      02H
JNE      INTSVS_7
IN        AL,      INTPORT
TEST     AL,      01H
JNE      INTSVS_8

```

第七章 基本输入输出接口

7-3 并行接口在传送一个数据时，数据的各位同时传送，即 CPU 与 I/O 设备之间需多根数据线。

特点：传送速率快；但不适合长距离数据传输；所传送的信息一般不要求固定的格

式；一般在接口和外设之间需设置握手信号线，以进行查询方式的通信。

串行接口在传送数据时只能将数据的各位一位一位通过单条 1 位宽的传输线按顺序分时传送。

特点：适宜于远距离传送；抗干扰能力强；远距离传输中速率快于并行传送；通信费用低，但实现较复杂。

7-4 82C55 有三种工作方式。

方式 0 是一种基本的输入或输出方式，通常不用联络信号，不使用中断，三个端口的任何一个都提供简单的输入输出操作。

基本功能：1)两个 8 位端口：端口 A 何端口 B。两个四位端口：端口 C 的高四位和低四位。

2) 任何一端口都可作为输入/输出。

3) 输出是锁存的。

4) 输入不是锁存的。

5) 在方式 0 时各端口的输入/输出可以有 16 种组合方式。

方式 1 方式 2 参考教材 P262~267。

7-7 参考教材 P273~278

7-8

MOV DX, 343H

MOV AL, 00110100B

OUT DX, AL

MOV AX, N

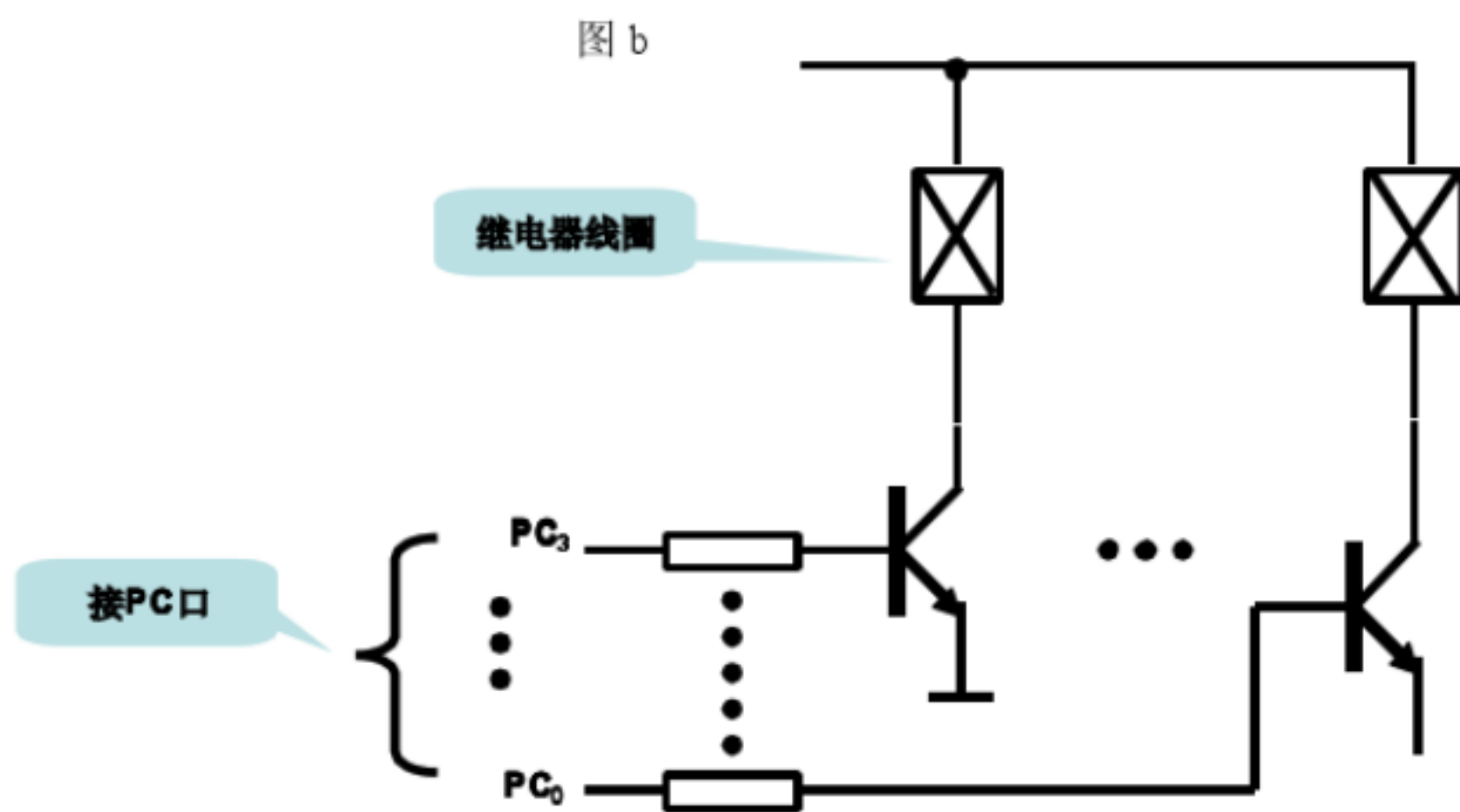
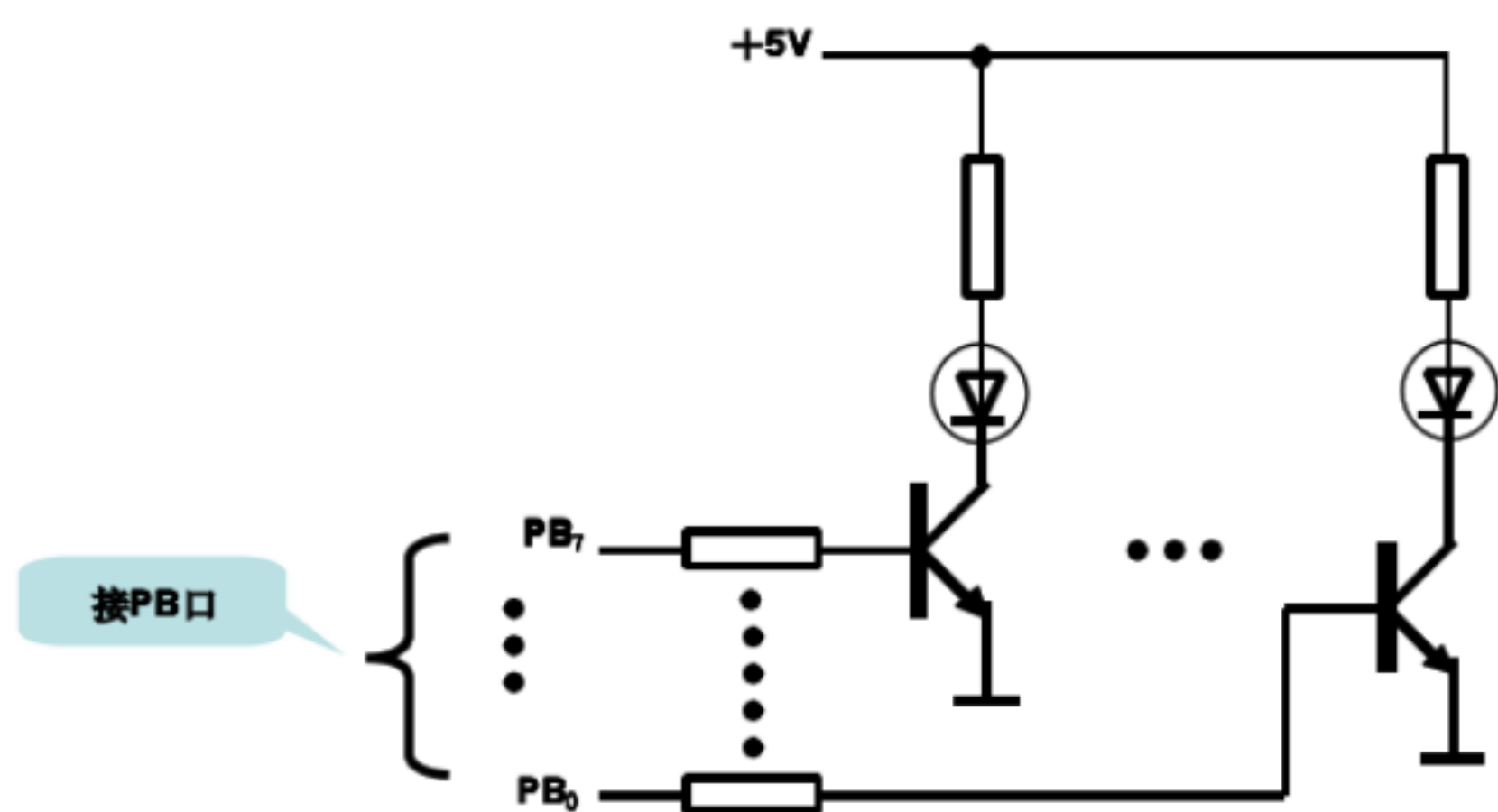
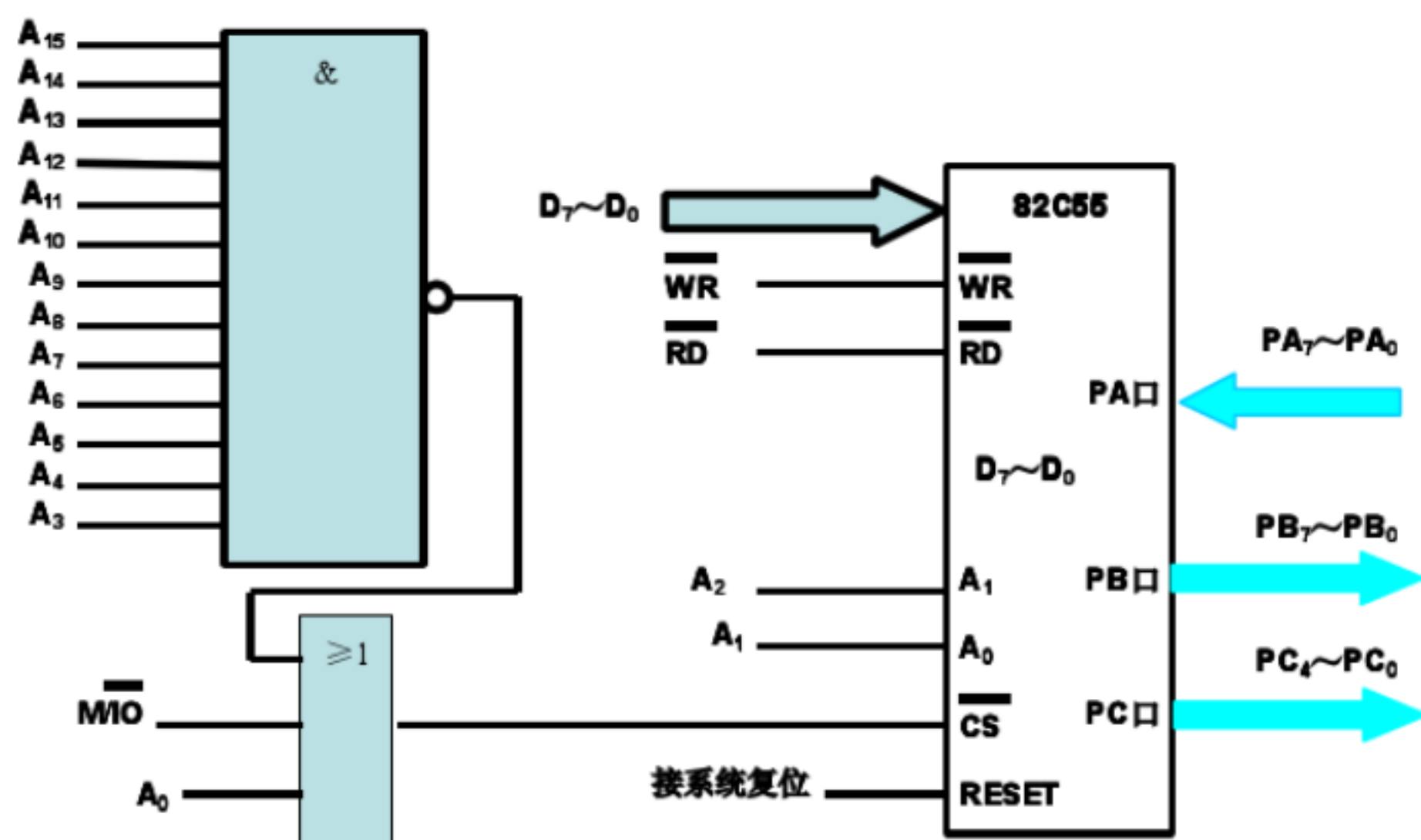
```

MOV    DX,    340H
OUT    DX,    AL
MOV    AL,    AH
OUT    DX,    AL
MOV    DX,    343H
MOV    AL,    10111010B
OUT    DX,    AL
MOV    AX,    NUM
MOV    DX,    342H
OUT    DX,    AL
MOV    AL,    AH
OUT    DX,    AL

```

7—5 解

(1) 设 82C55 与 8086CPU 连接, 占用 I/O 端口地址 0FFF8H~0FFFFH, A 口输入 8 个开关状态, 其地址为 0FFF8H, B 口输出 8 个 LED 发光二极管控制信号, 其地址为 0FFFAH, C 口低 3 位输出 3 个继电器控制信号, 其地址为 0FFFCH, 命令口地址为 0FFFEH。图 a 为 82C55 与 8086 接线简图, 图 b 为 LED 输出及驱动接线简图, 图 c 为继电器输出及驱动接线简图, 图 d 为开关信号输入接线简图。



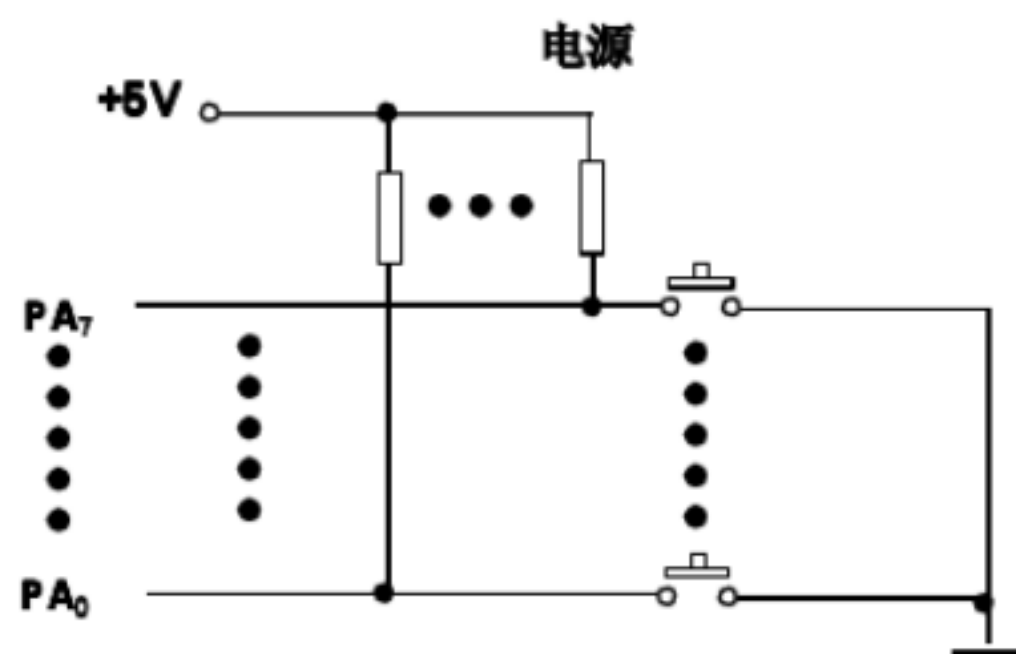


图 d

(2) 如图示, 开关闭合时, A 口输入低电平 0, 开关断开时, A 口输入高电平 1。B 口引脚输出高电平 1 时 LED 点亮, 反之熄灭。C 口输出高电平 1 时继电器得电闭合, 反之失电断开。分析控制要求可知, 82C55 所有端口应工作于方式 0, A 口输入, B 口输出, C 口下半部输出。设有开关闭合时 LED 灯点亮, 反之熄灭。程序段如下 (宏汇编语言结构定义部分略)

```

:
DATAC DB 0 ; 存放继电器控制数据
:

MOV AL, 10010000B ;A 组方式 0 输入, B 组方式 0 输出
MOV DX, 0FFFEH
OUT DX, AL ; 初始化 82C55
MOV AL, 0
MOV DX, 0FFFAH
OUT DX, AL ;LED 先熄灭
ADD DX, 2 ;指向 C 口

```

```

        OUT  DX, AL      ;继电器断开
L1:     MOV  DX, 0FFF8H

        IN   AL, DX

        NOT  AL

        MOV  DX, 0FFFAH

        OUT  DX, AL      ;LED 灯控制

        OR   AL, AL

        JZ   L1

        MOV  DX, 0FFFCH

        MOV  AL, DATAC

        NOT  AL

        OUT  DX, AL

L2:     JMP  L1

        ⋮

```