计算机原理夏季学期实验报告

张蔚桐 *2015011493* 自 *55*

1 第六次实验

1.1 实验目的

1. 学习 D/A 及 A/D 转换的基本原理
2. 掌握转换器 DAC0832 及 ADC0809 的使用方法

1.2 任务一

用 DAC0832 实现 D/A 转换，使产生的模拟电压波形分别为锯齿波，三角波和正弦波并符合题目的要求完成相关功能。汇编代码如下所示：

```
01
02
03 DATA SEGMENT
04     SAW_WAVE DB 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,↩
05     14, 15
06         DB 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, ↩
07         28, 29, 30, 31
08         DB 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, ↩
09         44, 45, 46, 47
10         DB 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, ↩
11         60, 61, 62, 63
12         DB 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, ↩
13         76, 77, 78, 79
14         DB 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, ↩
15         92, 93, 94, 95
16         DB 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106,↩
17         107, 108, 109, 110, 111
18         DB 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, ↩
19         122, 123, 124, 125, 126, 127
20         DB 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, ↩
```

```
21          138, 139, 140, 141, 142, 143
22          DB 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, ↩
23          154, 155, 156, 157, 158, 159
24          DB 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, ↩
25          170, 171, 172, 173, 174, 175
26          DB 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, ↩
27          186, 187, 188, 189, 190, 191
28          DB 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, ↩
29          202, 203, 204, 205, 206, 207
30          DB 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, ↩
31          218, 219, 220, 221, 222, 223
32          DB 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, ↩
33          234, 235, 236, 237, 238, 239
34          DB 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, ↩
35          250, 251, 252, 253, 254, 255
36      TRIANGLE_WAVE DB 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, ↩
37      23, 25, 27, 29, 31
38          DB 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, ↩
39          57, 59, 61, 63
40          DB 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, ↩
41          89, 91, 93, 95
42          DB 97, 99, 101, 103, 105, 107, 109, 111, 113, 115, ↩
43          117, 119, 121, 123, 125, 127
44          DB 129, 131, 133, 135, 137, 139, 141, 143, 145, 147, ↩
45          149, 151, 153, 155, 157, 159
46          DB 161, 163, 165, 167, 169, 171, 173, 175, 177, 179, ↩
47          181, 183, 185, 187, 189, 191
48          DB 193, 195, 197, 199, 201, 203, 205, 207, 209, 211, ↩
49          213, 215, 217, 219, 221, 223
50          DB 225, 227, 229, 231, 233, 235, 237, 239, 241, 243, ↩
51          245, 247, 249, 251, 253, 255
52          DB 255, 253, 251, 249, 247, 245, 243, 241, 239, 237, ↩
53          235, 233, 231, 229, 227, 225
54          DB 223, 221, 219, 217, 215, 213, 211, 209, 207, 205, ↩
55          203, 201, 199, 197, 195, 193
56          DB 191, 189, 187, 185, 183, 181, 179, 177, 175, 173, ↩
```

```
57          171, 169, 167, 165, 163, 161
58          DB 159, 157, 155, 153, 151, 149, 147, 145, 143, 141, ↩
59          139, 137, 135, 133, 131, 129
60          DB 127, 125, 123, 121, 119, 117, 115, 113, 111, 109, ↩
61          107, 105, 103, 101, 99, 97
62          DB 95, 93, 91, 89, 87, 85, 83, 81, 79, 77, 75, 73, ↩
63          71, 69, 67, 65
64          DB 63, 61, 59, 57, 55, 53, 51, 49, 47, 45, 43, 41, ↩
65          39, 37, 35, 33
66          DB 31, 29, 27, 25, 23, 21, 19, 17, 15, 13, 11, 9, 7, ↩
67          5, 3, 1
68      SINE_WAVE DB 128, 131, 134, 137, 140, 143, 146, 149, 152,↩
69      155, 158, 162, 165, 167, 170, 173
70          DB 176, 179, 182, 185, 188, 190, 193, 196, 198, 201, ↩
71          203, 206, 208, 211, 213, 215
72          DB 218, 220, 222, 224, 226, 228, 230, 232, 234, 235, ↩
73          237, 238, 240, 241, 243, 244
74          DB 245, 246, 248, 249, 250, 250, 251, 252, 253, 253, ↩
75          254, 254, 254, 255, 255, 255
76          DB 255, 255, 255, 255, 254, 254, 254, 253, 253, 252, ↩
77          251, 250, 250, 249, 248, 246
78          DB 245, 244, 243, 241, 240, 238, 237, 235, 234, 232, ↩
79          230, 228, 226, 224, 222, 220
80          DB 218, 215, 213, 211, 208, 206, 203, 201, 198, 196, ↩
81          193, 190, 188, 185, 182, 179
82          DB 176, 173, 170, 167, 165, 162, 158, 155, 152, 149, ↩
83          146, 143, 140, 137, 134, 131
84          DB 128, 124, 121, 118, 115, 112, 109, 106, 103, 100, ↩
85          97, 93, 90, 88, 85, 82
86          DB 79, 76, 73, 70, 67, 65, 62, 59, 57, 54, 52, 49, ↩
87          47, 44, 42, 40
88          DB 37, 35, 33, 31, 29, 27, 25, 23, 21, 20, 18, 17, ↩
89          15, 14, 12, 11
90          DB 10, 9, 7, 6, 5, 5, 4, 3, 2, 2, 1, 1, 1, 0, 0, 0
91          DB 0, 0, 0, 0, 1, 1, 1, 2, 2, 3, 4, 5, 5, 6, 7, 9
92          DB 10, 11, 12, 14, 15, 17, 18, 20, 21, 23, 25, 27, ↩
```

```
93          29, 31, 33, 35
94          DB 37, 40, 42, 44, 47, 49, 52, 54, 57, 59, 62, 65, ♪
95          67, 70, 73, 76
96          DB 79, 82, 85, 88, 90, 93, 97, 100, 103, 106, 109, ♪
97          112, 115, 118, 121, 124
98      WAVE_TYPE DB 00h
99  DATA ENDS
100
101 STACKS SEGMENT STACK
102     DB 100 dup(?)
103 STACKS ENDS
104
105 CODE SEGMENT
106     ASSUME CS:CODE, DS:DATA, SS:STACKS
107
108
109 MAIN PROC FAR
110
111 START:
112     MOV AX, DATA
113     MOV DS, AX
114     MOV AX, 0
115     MOV DI, AX
116
117 MAIN_LOOP:
118     MOV AH, 1
119     INT 16h
120     JNZ JUDGE
121
122 START_CONVERSION:
123     MOV AX, DI
124     CMP AX, 0ffh
125     JNZ P1
126     MOV AX, 0h
127     MOV DI, AX
128 P1:
```

```
129     MOV AL, WAVE_TYPE
130     CMP AL, 00h
131     JZ WAVE_SAW
132     CMP AL, 01h
133     JZ WAVE_TRIANGLE
134     CMP AL, 02h
135     JZ WAVE_SINE
136
137 FLAG:
138
139     PUSH AX
140     PUSH BX
141     PUSH CX
142 DELAY:
143     MOV CX, 03ffh
144 P2:
145     LOOP P2
146     POP CX
147     POP BX
148     POP AX
149
150     INC DI
151     JMP MAIN_LOOP
152
153 JUDGE:
154     MOV AH, 1
155     INT 21h
156
157     CMP AL, 34h
158     JZ EXIT
159
160     SUB AL, 31h
161     MOV WAVE_TYPE, AL
162     JMP START_CONVERSION
163
164 EXIT:
```

```
165     MOV AH, 4CH
166     INT 21h
167
168 WAVE_SAW:
169     MOV DX, 280h
170     MOV AL, SAW_WAVE[DI]
171     OUT DX, AL
172     JMP FLAG
173
174 WAVE_TRIANGLE:
175     MOV DX, 280h
176     MOV AL, TRIANGLE_WAVE[DI]
177     OUT DX, AL
178     JMP FLAG
179
180
181 WAVE_SINE:
182     MOV DX, 280h
183     MOV AL, SINE_WAVE[DI]
184     OUT DX, AL
185     JMP FLAG
186
187 MAIN ENDP
188 CODE ENDS
189
190 END START
```

1.3 任务二

用 ADC0809 实现 A/D 转换，用汇编语言程序自动对一个模拟信号重复采集 20 组不同的数据，在 CRT 上将每组数据对应显示成要求的形式，代码如下所示：

```
01 DATA SEGMENT
02     HEAD DB 0DH, 0AH, 'D/A          A/D', 0DH, 0AH, '$'
03     INFO DB 'Please input c to get the datas, e to exit:'
04     LINE DB 0DH, 0AH, '$'
05     SPACE DB '          $'
```

```
06 DATA ENDS
07
08 STACKS SEGMENT
09 STACKS ENDS
10
11 CODE SEGMENT
12     ASSUME CS:CODE,DS:DATA,SS:STACKS
13 MAIN PROC FAR
14
15 START:
16     MOV AX,DATA
17     MOV DS,AX
18
19     MOV BX, 0
20
21 DISPLAY:
22
23     MOV BL,BH
24     INC BH
25     MOV CX, 20
26     MOV AH, 9
27     LEA DX, HEAD
28     INT 21H
29 ONE_LINE:
30     MOV AL, BL
31     MOV DX, 280H
32     OUT DX, AL
33
34     CALL SHOW
35     CALL DELAY
36
37     MOV AH, 9
38     LEA DX, SPACE
39     INT 21H
40
41     MOV DX, 289H
```

```
42      OUT DX, AL
43
44      CALL DELAY
45
46      MOV DX, 289H
47      IN AL, DX
48
49      CALL SHOW
50
51      MOV AH, 9
52      LEA DX, LINE
53      INT 21H
54
55      ADD BL, 0FH
56      LOOP ONE_LINE
57
58      MOV AH, 9
59      LEA DX, INFO
60      INT 21H
61
62  ENTER:
63      MOV AH, 1
64      INT 21H
65      CMP AL, 'C'
66      JZ DISPLAY
67      CMP AL, 'c'
68      JZ DISPLAY
69      CMP AL, 'E'
70      JZ EXIT
71      CMP AL, 'e'
72      JZ EXIT
73      JMP ENTER
74
75  EXIT:
76      MOV AH,4CH
77      INT 21H
```

```
78
79 DELAY PROC
80     PUSH AX
81     PUSH BX
82     PUSH CX
83     MOV CX, 0FFFH
84 WAIT:
85     LOOP WAIT
86
87     POP CX
88     POP BX
89     POP AX
90     RET
91 DELAY ENDP
92
93
94 SHOW PROC
95     PUSH AX
96     AND AL, 0F0H
97     SHR AL, 1
98     SHR AL, 1
99     SHR AL, 1
100     SHR AL, 1
101     CMP AL, 09H
102     JBE DIG2
103     ADD AL, 07H
104 DIG2:
105     ADD AL, 30H
106     MOV DL, AL
107     MOV AH, 2
108     INT 21H
109     POP AX
110
111     AND AL, 0FH
112     CMP AL, 09H
113     JBE DIG1
```

```
114     ADD AL, 07H
115 DIG1:
116     ADD AL, 30H
117     MOV DL, AL
118     MOV AH, 2
119     INT 21H
120     MOV DL, 'H'
121     MOV AH, 2
122     INT 21H
123     RET
124 SHOW ENDP
125
126 MAIN ENDP
127 CODE ENDS
128     END START
```

1.4 完成情况及心得体会

本次实验使用了 AD/DA 转换，通过具体代码的编写了解了计算机 IO 速度和 CPU 运算速度的差异，并提升了自己的汇编编程水平

2 第七次实验

2.1 实验目的

综合汇编语言编程及 I/O 接口的知识，提高实际应用的能力

2.2 任务一

将 TPC 实验台上的 8255 电路 A 口设置成方式 0 输入，检测 8 只开关的状态；将 C 口设置成方式 0 输出，控制 8 只 LED 灯。程序运行后不断地读入 8 只开关的状态，送往对应的 LED 灯显示，直至在计算机键盘上敲入空格键退回 DOS

2.2.1 必做任务

代码如下所示 01 DATA **SEGMENT**
02 DATA **ENDS**

```
03
04 STACKS SEGMENT
05 STACKS ENDS
06
07 CODE SEGMENT
08     ASSUME CS:CODE,DS:DATA,SS:STACKS
09
10 MAIN PROC FAR
11 START:
12     MOV AX,DATA
13     MOV DS,AX
14
15     MOV DX, 283H
16     MOV AL, 10010000B
17     OUT DX, AL
18
19 NEXT:
20     MOV DX, 280H
21     IN AL, DX
22     MOV DX, 282H
23     OUT DX, AL
24     MOV AH, 1
25     INT 16H
26     JZ NEXT
27     MOV AH, 0
28     INT 16H
29     CMP AL, 20H
30     JZ EXIT
31     JMP NEXT
32
33 EXIT:
34     MOV AH,4CH
35     INT 21H
36
37 MAIN ENDP
38 CODE ENDS
```

```
39      END START
```

2.2.2 选做任务

A 口仍保持方式 0 输入开关状态，C 口仍以方式 0 输出 LED 灯显示，满足要求的条件，代码如下所示：

```
01 DATA SEGMENT
02 DATA ENDS
03
04 STACKS SEGMENT
05 STACKS ENDS
06
07 CODE SEGMENT
08      ASSUME CS:CODE,DS:DATA,SS:STACKS
09
10 MAIN PROC FAR
11 START:
12      MOV AX,DATA
13      MOV DS,AX
14
15      MOV DX, 283H
16      MOV AL, 10010000B
17      OUT DX, AL
18 SCAN:
19      MOV DX, 280H
20      IN AL, DX
21
22      CMP AL, 11000000B
23      JZ TOFLASH
24
25      CMP AL, 10000000B
26      JZ L_MOVE
27
28      CMP AL, 01000000B
29      JZ R_MOVE
30
31      MOV DX, 282H
```

```
32      OUT DX, AL
33
34      MOV AH, 1
35      INT 16H
36      JZ SCAN
37
38      MOV AH, 0
39      INT 16H
40      CMP AL, 20H
41      JZ TOEXIT
42      JMP SCAN
43
44  L_MOVE:
45      MOV DX, 282H
46      MOV BL, 10000000B
47      MOV AL, BL
48      OUT DX, AL
49  R_LEFT:
50      ROL BL, 1
51      MOV AL, BL
52      MOV DX, 282H
53      OUT DX, AL
54
55      CALL DELAY
56
57      MOV DX, 280H
58      IN AL, DX
59      CMP AL, 10000000B
60      JNZ SCAN
61
62      MOV DX, 282H
63      OUT DX, AL
64      MOV AH, 1
65      INT 16H
66      JZ R_LEFT
67
```

```
68      MOV AH, 0
69      INT 16H
70      CMP AL, 20H
71      JZ EXIT
72
73      JMP R_LEFT
74
75  TOFLASH:
76      JMP FLASH
77  TOEXIT:
78      JMP EXIT
79
80  R_MOVE:
81      MOV DX, 282H
82      MOV BL, 10000000B
83      MOV AL, BL
84      OUT DX, AL
85
86  R_RIGHT:
87      ROR BL, 1
88      MOV AL, BL
89      MOV DX, 282H
90
91      CALL DELAY
92
93      MOV DX, 280H
94      IN AL, DX
95      CMP AL, 01000000B
96      JNZ SCAN
97
98      MOV DX, 282H
99      OUT DX, AL
100      MOV AH, 1
101      INT 16H
102      JZ R_RIGHT
103      MOV AH, 0
```

```
104     INT 16H
105     CMP AL, 20H
106     JZ EXIT
107
108     JMP R_RIGHT
109
110 TOSCAN:
111     JMP SCAN
112 FLASH:
113     MOV DX, 282H
114     MOV AL, 0
115     OUT DX, AL
116
117     CALL DELAY
118
119     MOV DX, 282H
120     MOV AL, 0FFH
121     OUT DX, AL
122
123     MOV DX, 280H
124     IN AL, DX
125     CMP AL, 11000000B
126     JNZ TOSCAN
127
128     CALL DELAY
129
130     MOV DX, 282H
131     OUT DX, AL
132     MOV AH, 1
133     INT 16H
134     JZ FLASH
135     MOV AH, 0
136     INT 16H
137     CMP AL, 20H
138     JZ EXIT
139
```

```
140     JMP FLASH
141
142 EXIT:
143     MOV AH,4CH
144     INT 21H
145
146 DELAY PROC
147     PUSH BX
148     PUSH CX
149     MOV BX, 0FH
150 WAITB:
151     MOV CX, 0FFFFH
152 WAITC:
153     DEC CX
154     JNZ WAITC
155     DEC BX
156     JNZ WAITB
157     POP CX
158     POP BX
159     RET
160 DELAY ENDP
161
162 MAIN ENDP
163 CODE ENDS
164     END START
```

2.3 任务二

实验中每按一次单脉冲按键，通过 8255 电路发一次中断请求。CRT 上显示一个 A 口的 ASCII 码字符，直到 A 口数据为 FFH 退出。

2.3.1 必做任务

代码如下所示：

```
01 DATA SEGMENT
02 KEEP_IP DW 0
03 KEEP_CS DW 0
04 FLAG DB 0
```

```
05 DATA ENDS
06
07 STACKS SEGMENT
08 STACKS ENDS
09
10 CODE SEGMENT
11     ASSUME CS:CODE,DS:DATA,SS:STACKS
12
13 MAIN PROC FAR
14 START:
15     MOV AX,DATA
16     MOV DS,AX
17
18
19     MOV DX, 283H
20     MOV AL, 10110000B
21     OUT DX, AL
22
23     MOV DX, 283H
24     MOV AL, 00001001B
25     OUT DX, AL
26
27
28     MOV AH, 35H
29     MOV AL, 0BH
30     INT 21H
31     MOV KEEP_IP, BX
32     MOV KEEP_CS, ES
33
34
35     PUSH DS
36     MOV DX, OFFSET INTR
37     MOV AX, SEG INTR
38     MOV DS, AX
39     MOV AH, 25H
40     MOV AL, 0BH
```

```
41      INT 21H
42      POP DS
43
44      MOV AL, 0F7H
45      OUT 21H, AL
46
47  WAIT_FOR:
48      MOV BL, FLAG
49      CMP BL, 1
50      JZ ISINT
51      JMP WAIT_FOR
52
53  ISINT:
54      CMP CL, 0FFH
55      JZ EXIT
56      MOV DL, CL
57      MOV AH, 2
58      INT 21H
59      MOV FLAG, 0
60      JMP WAIT_FOR
61
62  EXIT:
63
64      MOV AL, 0FFH
65      OUT 21H, AL
66
67
68      PUSH DS
69      MOV DX, KEEP_IP
70      MOV AX, KEEP_CS
71      MOV DS, AX
72      MOV AH, 25H
73      MOV AL, 0BH
74      INT 21H
75      POP DS
76      MOV AH,4CH
```

```
77      INT 21H
78
79  INTR PROC
80      MOV FLAG, 1
81      MOV DX, 280H
82      IN AL, DX
83      MOV CL, AL
84      MOV AL, 20H
85      OUT 20H, AL
86      IRET
87  INTR ENDP
88
89  MAIN ENDP
90  CODE ENDS
91      END START
```

2.3.2 选做任务一

修改主程序实现密码检测功能，连续两次从 A 口拨入数据，与计算机内部事先存放的两字节数比较，相符则在 CRT 上显示“OK”，否则重新输入，代码如下所示：

```
01 DATAS SEGMENT
02 KEEP_IP DW 0
03 KEEP_CS DW 0
04 FLAG DB 0
05 PASSWORD DB 0F0H, 0FH
06 OK DB 'OK',0DH, 0AH, '$'
07 NO DB 'NO',0DH, 0AH, '$'
08 DATAS ENDS
09
10 STACKS SEGMENT
11 STACKS ENDS
12
13 CODES SEGMENT
14     ASSUME CS:CODES,DS:DATAS,SS:STACKS
15
16 MAIN PROC FAR
```

```
START:
    MOV AX,DATAS
    MOV DS,AX


    MOV DX, 283H
    MOV AL, 10110000B
    OUT DX, AL

    MOV DX, 283H
    MOV AL, 00001001B
    OUT DX, AL


    MOV AH, 35H
    MOV AL, 0BH
    INT 21H
    MOV KEEP_IP, BX
    MOV KEEP_CS, ES


    PUSH DS
    MOV DX, OFFSET INTR
    MOV AX, SEG INTR
    MOV DS, AX
    MOV AH, 25H
    MOV AL, 0BH
    INT 21H
    POP DS

    MOV AL, 0F7H
    OUT 21H, AL

WAIT_FOR1:
    MOV BL, FLAG
    CMP BL, 1
```

```
53      JZ WAIT_FOR2
54      JMP WAIT_FOR1
55
56  WAIT__FOR2:
57      CMP CL, 0FFH
58      JZ EXIT
59      MOV DL, CL
60      MOV AH, 2
61      INT 21H
62      MOV BH,     CL
63      MOV FLAG, 0
64      JMP WAIT2
65  WAIT2:
66      MOV BL, FLAG
67      CMP BL, 1
68      JZ CHECK
69      JMP WAIT2
70
71  CHECK:
72
73      CMP CL, 0FFH
74      JZ EXIT
75      MOV DL, CL
76      MOV AH, 2
77      INT 21H
78      CMP BH,PASSWORD
79      JNZ ERROR
80      CMP CL,PASSWORD+1
81      JNZ ERROR
82
83      MOV AH, 9
84      LEA DX, OK
85      INT 21H
86      JMP WAIT_FOR1
87
88  ERROR:
```

```
89      MOV AH, 9
90      LEA DX, NO
91      INT 21H
92      JMP WAIT_FOR1
93
94  EXIT:
95
96      MOV AL, 0FFH
97      OUT 21H, AL
98
99
100      PUSH DS
101      MOV DX, KEEP_IP
102      MOV AX, KEEP_CS
103      MOV DS, AX
104      MOV AH, 25H
105      MOV AL, 0BH
106      INT 21H
107      POP DS
108      MOV AH,4CH
109      INT 21H
110
111 INTR PROC
112      MOV FLAG, 1
113      MOV DX, 280H
114      IN AL, DX
115      MOV CL, AL
116      MOV AL, 20H
117      OUT 20H, AL
118      IRET
119 INTR ENDP
120
121 MAIN ENDP
122 CODES ENDS
123      END START
```

2.3.3 选做任务二

将 8255 电路 A 口改成方式 1 输出（仅将 PA7 接一只 LED 示范即可），修改前面的程序实现每次中断后，通过 A 口输出数据控制 LED 状态在 0,1 之间翻转，代码如下所示。

```
01 DATA SEGMENT
02 SIG DB 0
03 KEEP_IP DW 0
04 KEEP_CS DW 0
05 FLAG DB 0
06 DATA ENDS
07
08 STACKS SEGMENT
09 STACKS ENDS
10
11 CODE SEGMENT
12     ASSUME CS:CODE,DS:DATA,SS:STACKS
13
14 MAIN PROC FAR
15 START:
16     MOV AX,DATA
17     MOV DS,AX
18
19
20     MOV DX, 283H
21     MOV AL, 10101000B
22     OUT DX, AL
23
24     MOV DX, 283H
25     MOV AL, 11001000B
26     OUT DX, AL
27
28
29     MOV AH, 35H
30     MOV AL, 0BH
31     INT 21H
32     MOV KEEP_IP, BX
33     MOV KEEP_CS, ES
```

```
34
35
36
37      PUSH DS
38      MOV DX, OFFSET INTR
39      MOV AX, SEG INTR
40      MOV DS, AX
41      MOV AH, 25H
42      MOV AL, 0BH
43      INT 21H
44      POP DS
45
46      MOV AL, 0F7H
47      OUT 21H, AL
48
49  WAIT_FOR:
50      MOV BL, FLAG
51      CMP BL, 1
52      JZ ISINT
53      JMP WAIT_FOR
54
55  ISINT:
56      MOV FLAG, 0
57      JMP WAIT_FOR
58
59  EXIT:
60
61      MOV AL, 0FFH
62      OUT 21H, AL
63
64
65      PUSH DS
66      MOV DX, KEEP_IP
67      MOV AX, KEEP_CS
68      MOV DS, AX
69      MOV AH, 25H
```

```
70      MOV AL, 0BH
71      INT 21H
72      POP DS
73      MOV AH,4CH
74      INT 21H
75
76  INTR PROC
77      MOV FLAG, 1
78      XOR SIG,0FFH
79      MOV AL, SIG
80      MOV DX, 280H
81      OUT DX,AL
82      MOV AL, 20H
83      OUT 20H, AL
84      IRET
85  INTR ENDP
86
87  MAIN ENDP
88  CODE ENDS
89      END START
```

2.4 任务三

8255 电路 A 口以方式 0 输出，C 口也初始化成方式 0 输出且仅用其最低两位：PC1 接数码管位码输入端 S1，PC0 接位码输入端 S0。程序实现当 A 口输出字形“0”的段码时，C 口输出 01H，第一个数码管显示“0”，当 A 口输出字形“1”的段码时，C 口输出 02H，于是第二个数码管显示“1”。每一位显示之后调用一段延时程序，选择恰当的延时程序，使“01”几乎同时显示在两位数码管上。

2.4.1 必做任务

代码如下所示：
```
01 DATAS SEGMENT
02 DATAS ENDS
03
04 STACKS SEGMENT
05 STACKS ENDS
```

```
06
07 CODES SEGMENT
08     ASSUME CS:CODES,DS:DATAS,SS:STACKS
09
10 MAIN PROC FAR
11 START:
12     MOV AX,DATAS
13     MOV DS,AX
14
15     MOV DX, 283H
16     MOV AL, 10000000B
17     OUT DX, AL
18
19 NEXT:
20     MOV DX, 282H
21     MOV AL, 01H
22     OUT DX, AL
23     MOV DX, 280H
24     MOV AL, 3FH
25     OUT DX, AL
26
27     CALL DELAY
28
29     MOV DX, 280H
30     MOV AL, 06H
31     OUT DX, AL
32     MOV DX, 282H
33     MOV AL, 02H
34     OUT DX, AL
35
36     CALL DELAY
37
38     MOV AH, 1
39     INT 16H
40     JZ NEXT
41     CMP AL, 20H
```

```
42      JZ EXIT
43      JMP NEXT
44
45  EXIT:
46      MOV AH,4CH
47      INT 21H
48
49  DELAY PROC
50      MOV CX, 0FFH
51  D_WAIT:     LOOP D_WAIT
52      RET
53  DELAY ENDP
54
55  MAIN ENDP
56  CODES ENDS
57      END START
```

2.4.2 选做任务一

当程序运行后，从计算机键盘上输入两位十进制数，分别在两个数码管上显示。若继续输入数字则更新显示。若发现输入了非数字键则退回 DOS，代码如下所示。

```
01 DATAS SEGMENT
02 NUM DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H,7FH,6FH
03 DATAS ENDS
04
05 STACKS SEGMENT
06    DB 100 DUP(?)
07 STACKS ENDS
08
09 CODES SEGMENT
10      ASSUME CS:CODES,DS:DATAS,SS:STACKS
11 START:
12      MOV AX,DATAS
13      MOV DS,AX
14
15      MOV DX,283H
```

```
16      MOV AL,10000000B
17      OUT DX,AL
18
19      MOV BX,0
20      MOV CX,0
21  MAIN:
22      MOV AH,0BH
23      INT 21H
24      INC AL
25      JNE NEXT
26
27      MOV AH,1
28      INT 21H
29      CMP AL,'0'
30      JS EXIT
31      CMP AL,'9'+1
32      JNS EXIT
33      SUB AL,30H
34      MOV BL,AL
35
36      MOV AH,1
37      INT 21H
38      CMP AL,'0'
39      JS EXIT
40      CMP AL,'9'+1
41      JNS EXIT
42      SUB AL,30H
43      MOV CL,AL
44
45  CALL DELAY
46  CALL DELAY
47  CALL DELAY
48
49  NEXT:
50
51
```

```
52      MOV DX,280H
53      PUSH BX
54      MOV BX,CX
55      MOV AL,[NUM+BX]
56      OUT DX,AL
57      MOV DX,282H
58      MOV AL,01H
59      OUT DX,AL
60      POP BX
61
62      CALL DELAY
63
64
65      MOV DX,282H
66      MOV AL,00H
67      OUT DX,AL
68
69      MOV DX,280H
70      MOV AL,[NUM+BX]
71      OUT DX,AL
72      MOV DX,282H
73      MOV AL,02H
74      OUT DX,AL
75
76      CALL DELAY
77
78
79      MOV DX,282H
80      MOV AL,00H
81      OUT DX,AL
82      JMP MAIN
83
84  EXIT:
85      MOV AH,4CH
86      INT 21H
87
```

```
88
89 DELAY  PROC
90       PUSH    CX
91       PUSH    AX
92       MOV     AX,000FH
93 X1:    MOV     CX,0FFFH
94 X2:    DEC     CX
95       JNE     X2
96       DEC     AX
97       JNE     X1
98       POP     AX
99       POP     CX
100       RET
101 DELAY  ENDP
102
103
104 CODES ENDS
105     END START
106
107
108
109
```

2.4.3 选做任务二

使用 TPC 实验台上的 8253 定时计数电路来代替前面的软件延时。8253 定时器自动重复工作，每工作一个周期发出一次中断请求信号，在中断服务程序里同步更换段码和位码，实现扫描显示，代码如下所示。

```
01 DATA SEGMENT
02 KEEPIP DW 0
03 KEEPCS DW 0
04 DATA ENDS
05
06 STACK SEGMENT
07     DB 100 DUP(?)
08 STACK ENDS
```

```
09
10 CODE SEGMENT
11     ASSUME CS:CODE,DS:DATA,ES:DATA,SS:STACK
12 START:
13
14     MOV AX,DATA
15     MOV DS,AX
16     MOV ES,AX
17
18     MOV DX,293H
19     MOV AL,00110111B
20     OUT DX,AL
21
22     MOV DX,290H
23     XOR AL,AL
24     OUT DX,AL
25     MOV AL,50
26     OUT DX,AL
27
28     MOV DX,283H
29     MOV AL,10000000B
30     OUT DX,AL
31
32
33     MOV AH,35H
34     MOV AL,0BH
35     INT 21H
36     MOV KEEPIP,BX
37     MOV KEEPCS,ES
38
39     PUSH DS
40     MOV DX,OFFSET INTR
41     MOV AX,SEG INTR
42     MOV DS,AX
43     MOV AH,25H
44     MOV AL,0BH
```

```
45      INT 21H
46      POP DS
47
48
49      IN AL,21H
50      AND AL,011110111B
51      OUT 21H,AL
52
53      MOV BL,0
54  MAIN:
55      HLT
56      MOV AH,1
57      INT 16H
58      JNZ EXIT
59      JMP MAIN
60
61  EXIT:
62
63      IN   AL,21H
64      OR   AL,00001000B
65      OUT 21H,AL
66
67
68      PUSH DS
69      MOV DX,KEEPIP
70      MOV AX,KEEPCS
71      MOV DS,AX
72      MOV AH,25H
73      MOV AL,0BH
74      INT 21H
75      POP DS
76
77      MOV AH,4CH
78      INT 21H
79
80
```

```
81
82 INTR PROC
83
84
85      MOV DX,282H
86      MOV AL,00H
87      OUT DX,AL
88
89      CMP BL,0
90      JNZ OUT1
91 OUT0:
92      MOV DX,280H
93      MOV AL,3FH
94      OUT DX,AL
95      MOV DX,282H
96      MOV AL,01H
97      OUT DX,AL
98      MOV BL,1
99      JMP END_INTR
100 OUT1:
101
102      MOV DX,280H
103      MOV AL,06H
104      OUT DX,AL
105      MOV DX,282H
106      MOV AL,02H
107      OUT DX,AL
108      MOV BL,0
109      JMP END_INTR
110
111 END_INTR:
112      MOV AL,20H
113      OUT 0A0H,AL
114      OUT 20H,AL
115 IRET
116 INTR ENDP
```

```
117
118
119 CODE ENDS
120 END START
121
122
123
124
125
126
127
128
```

2.5 完成情况及心得体会

本次实验使用了并口等元件完成了 CPU 对外设的控制，提升了汇编语言学习能力。