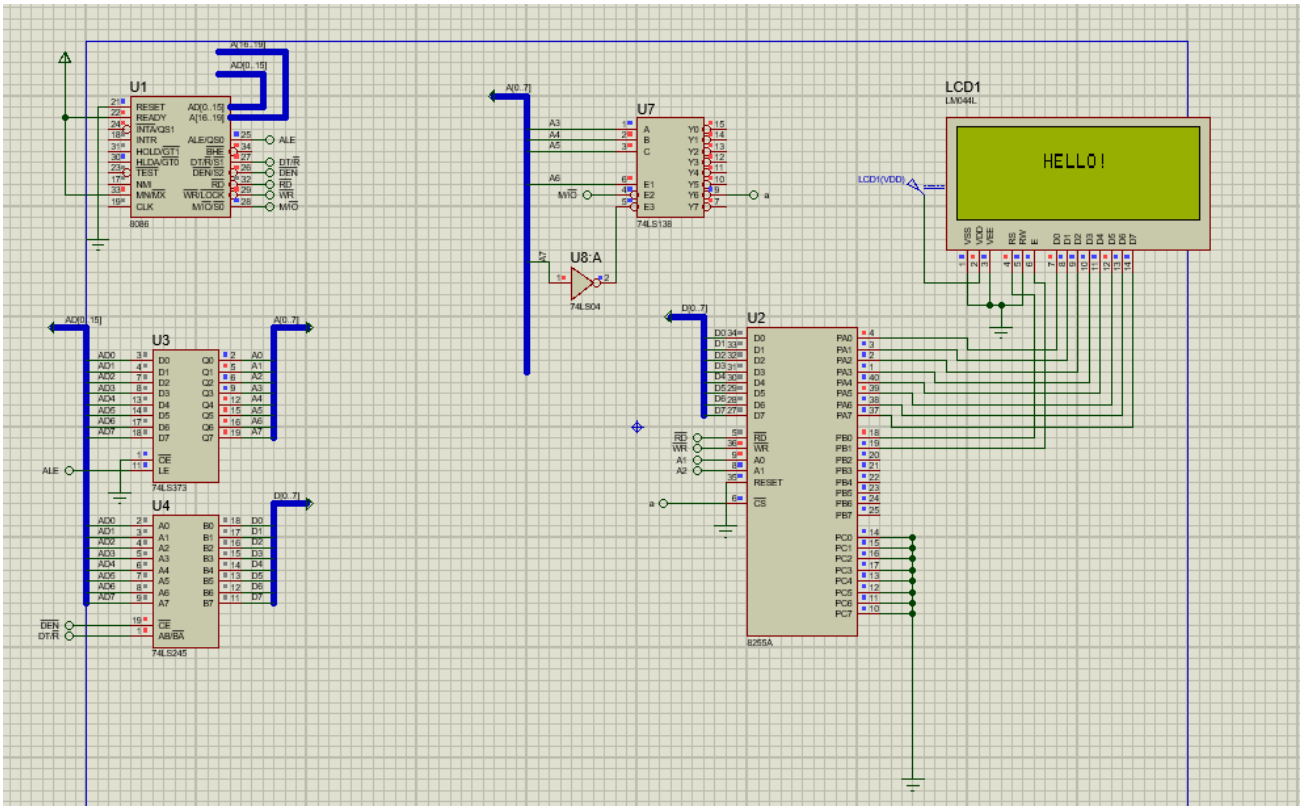
Laboratory Report

Laboratory Exercise No.:	6	**Date Performed:**	November 12, 2023
Laboratory Exercise Title:	Parallel I/O Devices Interfacing		
Name of Student:	Diaz, John Ivan Domingo, Niklas Goc-Ong, Craig	**Document Version:**	1.2.2

Activity #6-1

Proteus configuration:



Assembly code:

```
;===================================================================
                ; Group 4 CPE-3104
                ; LE6-1
                ; John Ivan Diaz
                ; Niklas Domingo
                ; Craig Joseph Goc-ong
;===================================================================

DATA SEGMENT
   PORTA EQU 0F0H
   PORTB EQU 0F2H
   PORTC EQU 0F4H
   PORT_CON EQU 0F6H
DATA ENDS

CODE    SEGMENT PUBLIC 'CODE'
        ASSUME CS:CODE

START:
       MOV DX, PORT_CON
       MOV AL, 89H
       OUT DX, AL

       CALL INIT_LCD

       MOV AL, 0C7H
       CALL INST_CTRL

       MOV AL, 'H'
       CALL DATA_CTRL
       MOV AL, 'E'
       CALL DATA_CTRL
       MOV AL, 'L'
       CALL DATA_CTRL
       MOV AL, 'L'
```

main.asm

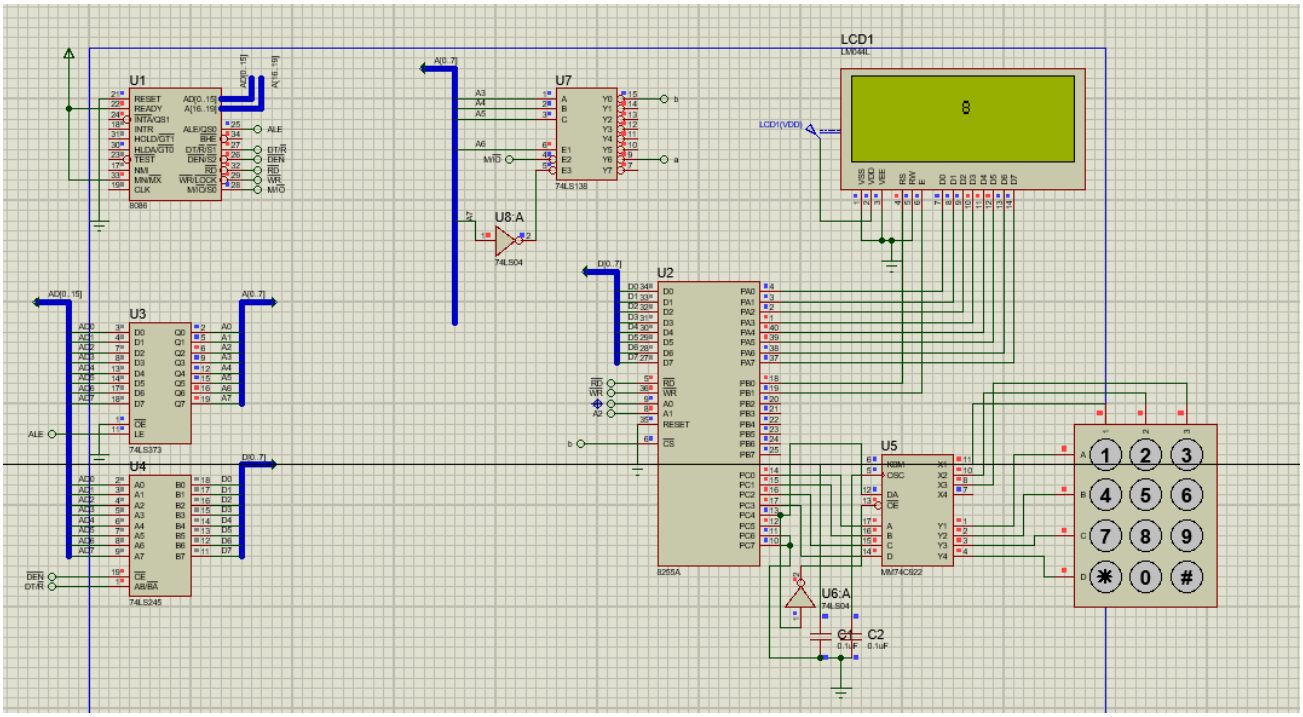
```
       MOV AL, 'O'
       CALL DATA_CTRL
       MOV AL, '!'
       CALL DATA_CTRL



ENDLESS:
        JMP ENDLESS

DELAY_1MS:
   MOV BX, 02CAH
L1:
   DEC BX
   NOP
   JNZ L1
   RET

INST_CTRL:
   PUSH AX
   MOV DX, PORTA
   OUT DX, AL
   MOV DX, PORTB
   MOV AL, 02H
   OUT DX, AL
   CALL DELAY_1MS
   MOV DX, PORTB
   MOV AL, 00H
   OUT DX, AL
   POP AX
   RET

INIT_LCD:
   MOV AL, 38H
   CALL INST_CTRL
```

main.asm

```
63    MOV DX, PORTB
64    MOV AL, 00H
65    OUT DX, AL
66    POP AX
67    RET
68
69  INIT_LCD:
70    MOV AL, 38H
71    CALL INST_CTRL
72    MOV AL, 08H
73    CALL INST_CTRL
74    MOV AL, 01H
75    CALL INST_CTRL
76    MOV AL, 06H
77    CALL INST_CTRL
78    MOV AL, 0CH
79    CALL INST_CTRL
80    RET
81
82  DATA_CTRL:
83    PUSH AX
84    MOV DX, PORTA
85    OUT DX, AL
86    MOV DX, PORTB
87    MOV AL, 03H
88    OUT DX, AL
89    CALL DELAY_1MS
90    MOV DX, PORTB
91    MOV AL, 01H
92    OUT DX, AL
93    POP AX
94    RET
95
96  CODE    ENDS
97          END START
```

Activity #6-2

Proteus configuration:



Assembly code:

```
;===================================================================
                ; Group 4 CPE-3104
                ; LE6-2
                ; John Ivan Diaz
                ; Niklas Domingo
                ; Craig Joseph Goc-ong
;===================================================================


DATA SEGMENT

   PORTA EQU 0C0H
   PORTB EQU 0C2H
   PORTC EQU 0C4H
   PORT_CON EQU 0C6H

   MYSTR    DB    "HELLO$"


DATA ENDS

CODE    SEGMENT PUBLIC 'CODE'
        ASSUME CS:CODE


START:

        MOV DX, PORT_CON
        MOV AL, 89H
        OUT DX, AL
BEGIN:
       CALL INIT_LCD

CHECK_DAVBL:
   MOV DX, PORTC ; set port of DAVBL(PORTC)
```

```
   TEST AL, 10H ; check if DAVBL is high
   JZ CHECK_DAVBL ; if low then check again
   IN AL, DX ; read 4-bit keypad data
   AND AL, 0FH ; mask upper nibble


         CMP AL, 00H ; check if key pressed is 1 (00H)
         JE D1 ; display 1
         CMP AL, 01H ; check if key pressed is 2 (01H)
         JE D2 ; display 2
         CMP AL, 02H ; check if key pressed is 3 (02H)
         JE D3 ; display 3
         CMP AL, 04H ; check if key pressed is 4 (04H)
         JE D4 ; display 4
         CMP AL, 05H ; check if key pressed is 5 (05H)
         JE D5 ; display 5
         CMP AL, 06H ; check if key pressed is 6 (06H)
         JE D6 ; display 6
         CMP AL, 08H ; check if key pressed is 7 (08H)
         JE D7 ; display 7
         CMP AL, 09H ; check if key pressed is 8 (09H)
         JE D8 ; display 8
         CMP AL, 0AH ; check if key pressed is 9 (0AH)
         JE D9 ; display 9
         CMP AL, 0CH ; check if key pressed is 9 (0AH)
         JE D10 ; display *
         CMP AL, 0EH
         JE D11
         CMP AL, 0DH ; check if key pressed is 0 (0DH)
         JE D0 ; display 0

         CALL DELAY_1MS
         JMP CHECK_DAVBL

D0:
```

main.asm

```
71  D0:
72
73    PUSH AX
74    MOV AL, 0CAH
75    CALL INST_CTRL
76    MOV AL, '0' ; display '1'
77    CALL DATA_CTRL
78    POP AX
79    JMP CONT
80
81  D1:
82    PUSH AX
83    MOV AL, 0CAH
84    CALL INST_CTRL
85    MOV AL, '1' ; display '1'
86    CALL DATA_CTRL
87    POP AX
88    JMP CONT
89
90
91  D2:
92    PUSH AX
93    MOV AL, 0CAH
94    CALL INST_CTRL
95    MOV AL, '2' ; display '1'
96    CALL DATA_CTRL
97    POP AX
98    JMP CONT
99  D3:
100   PUSH AX
101   MOV AL, 0CAH
102   CALL INST_CTRL
103   MOV AL, '3' ; display '1'
104   CALL DATA_CTRL
105   POP AX
```

```
104   CALL DATA_CTRL
105   POP AX
106   JMP CONT
107 D4:
108   PUSH AX
109   MOV AL, 0CAH
110   CALL INST_CTRL
111   MOV AL, '4' ; display '1'
112   CALL DATA_CTRL
113   POP AX
114   JMP CONT
115 D5:
116   PUSH AX
117   MOV AL, 0CAH
118   CALL INST_CTRL
119   MOV AL, '5' ; display '1'
120   CALL DATA_CTRL
121   POP AX
122   JMP CONT
123 D6:
124   PUSH AX
125   MOV AL, 0CAH
126   CALL INST_CTRL
127   MOV AL, '6' ; display '1'
128   CALL DATA_CTRL
129   POP AX
130   JMP CONT
131 D7:
132   PUSH AX
133   MOV AL, 0CAH
134   CALL INST_CTRL
135   MOV AL, '7' ; display '1'
136   CALL DATA_CTRL
137   POP AX
138   JMP CONT
```

```
D8:
 PUSH AX
 MOV AL, 0CAH
 CALL INST_CTRL
 MOV AL, '8' ; display '1'
 CALL DATA_CTRL
 POP AX
 JMP CONT
D9:
 PUSH AX
 MOV AL, 0CAH
 CALL INST_CTRL
 MOV AL, '9' ; display '1'
 CALL DATA_CTRL
 POP AX
 JMP CONT

D10:
 PUSH AX
 MOV AL, 0CAH
 CALL INST_CTRL
 MOV AL, '*'
 CALL DATA_CTRL
 POP AX
 JMP CONT

 D11:
 PUSH AX
 MOV AL, 0CAH
 CALL INST_CTRL
 MOV AL, '#'
 CALL DATA_CTRL
 POP AX
 JMP CONT

```

main.asm

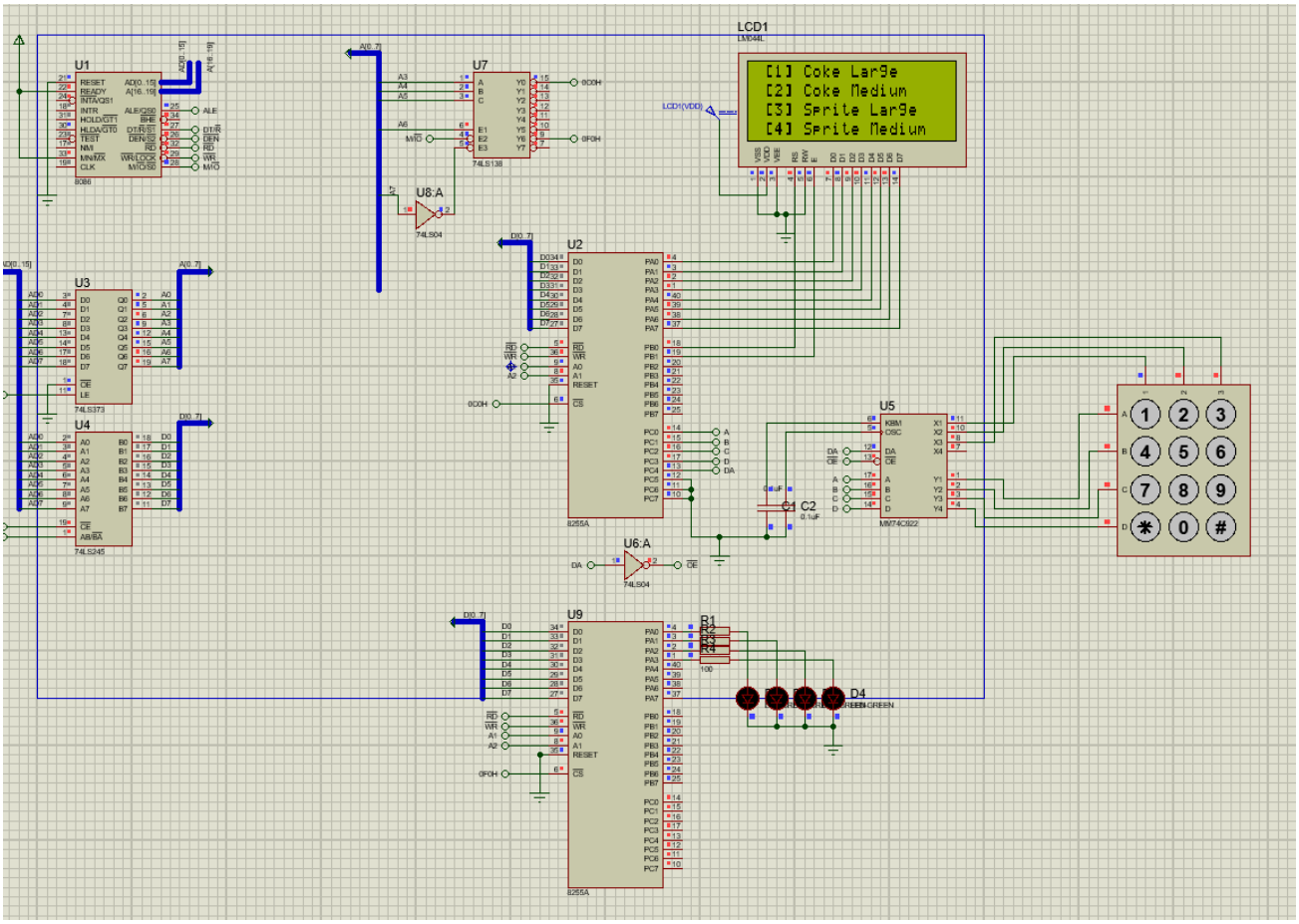
```
CONT:
   CALL DELAY_1MS
   JMP  CHECK_DAVBL

DELAY_1MS:
   MOV BX, 02CAH
   L1:
   NOP
   DEC BX
   JNZ L1
   RET


INST_CTRL:
   PUSH AX
   MOV DX, PORTA
   OUT DX, AL
   MOV DX, PORTB
   MOV AL, 02H
   OUT DX, AL
   CALL DELAY_1MS
   MOV DX, PORTB
   MOV AL, 00H
   OUT DX, AL
   POP AX
   RET

INIT_LCD:
   MOV AL, 38H
   CALL INST_CTRL
   MOV AL, 08H
   CALL INST_CTRL
   MOV AL, 01H
   CALL INST_CTRL
   MOV AL, 06H
```

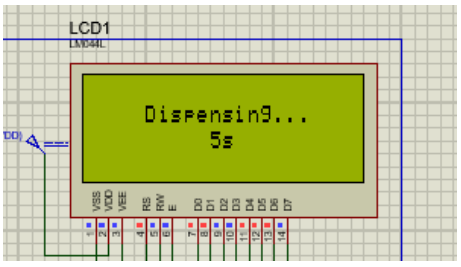
main.asm

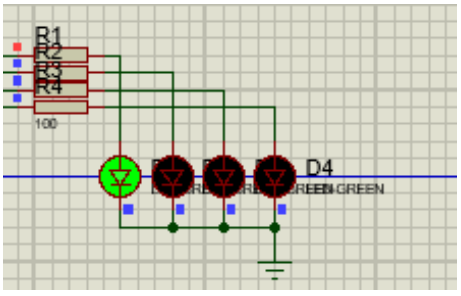
```
197     OUT DX, AL
198     POP AX
199     RET
200
201  INIT_LCD:
202     MOV AL, 38H
203     CALL INST_CTRL
204     MOV AL, 08H
205     CALL INST_CTRL
206     MOV AL, 01H
207     CALL INST_CTRL
208     MOV AL, 06H
209     CALL INST_CTRL
210     MOV AL, 0CH
211     CALL INST_CTRL
212     RET
213
214  DATA_CTRL:
215     PUSH AX
216     MOV DX, PORTA
217     OUT DX, AL
218     MOV DX, PORTB
219     MOV AL, 03H
220     OUT DX, AL
221     CALL DELAY_1MS
222     MOV DX, PORTB
223     MOV AL, 01H
224     OUT DX, AL
225     POP AX
226     RET
227
228  ENDLESS:
229          JMP ENDLESS
230  CODE     ENDS
231          END START
```

Activity #6-3

Proteus configuration:







main.asm

```
;===================================================================
                ; Group 4 CPE-3104
                ; LE6-3
                ; John Ivan Diaz
                ; Niklas Domingo
                ; Craig Joseph Goc-ong
;===================================================================


DATA    SEGMENT
      PORTA     EQU 0C0h
      PORTB     EQU 0C2h
      PORTC     EQU 0C4h
      COM_REG1  EQU 0C6h

      PORTA2    EQU 0F0h
      COM_REG2  EQU 0F6h

      CLEAN     DB " Initializing.","$"
      OPTION1   DB "[1] Coke Large","$"
      OPTION2   DB "[2] Coke Medium","$"
      OPTION3   DB "[3] Sprite Large","$"
      OPTION4   DB "[4] Sprite Medium","$"
      MESSAGE1  DB "Dispensing...", "$"
      OUTPUT1   DB "Drink is Ready!!!","$"

DATA    ENDS


CODE    SEGMENT PUBLIC 'CODE'
        ASSUME CS:CODE
        MOV AX, DATA
        MOV DS, AX
        org 0000h
```

main.asm

```
        org 0000h

START:

   ; Initialize 8255 command registers
      MOV DX, COM_REG1
      MOV AL, 10001001b
      OUT DX, AL

      MOV DX, COM_REG2
      MOV AL, 10001001b
      OUT DX, AL

    ; Initialize LCD
      CALL INIT_LCD

    ; ===== START =====

  MAIN_MENU:
      CALL CLEAR_SCREEN

      LEA SI, CLEAN
      MOV AL, 081h      ; LCD position
      CALL INST_CTRL
      CALL PRINT_STRING

      LEA SI, OPTION1
      MOV AL, 081h      ; LCD position
      CALL INST_CTRL
      CALL PRINT_STRING

      LEA SI, OPTION2
      MOV AL, 0C1h      ; LCD position
      CALL INST_CTRL
      CALL PRINT_STRING
```

main.asm

```
70      LEA SI, OPTION3
71      MOV AL, 095h      ; LCD position
72      CALL INST_CTRL
73      CALL PRINT_STRING
74
75      LEA SI, OPTION4
76      MOV AL, 0D5h      ; LCD position
77      CALL INST_CTRL
78      CALL PRINT_STRING
79
80
81      CHECK_DAVBL:
82
83         MOV DX, PORTC
84         IN AL, DX
85         TEST AL, 10h
86         JZ CHECK_DAVBL
87         IN AL, DX
88         AND AL, 0Fh
89         PUSH AX
90            CHECK_INPUT:
91               CMP AL, 00h
92               JE INPUT_OPTION1
93               CMP AL, 01h
94               JE INPUT_OPTION2
95               CMP AL, 02h
96               JE INPUT_OPTION3
97               CMP AL, 04h
98               JE INPUT_OPTION4
99            JMP CHECK_DAVBL
100
101
102        INPUT_OPTION1:
103            CALL CLEAR_SCREEN
104            LEA SI, MESSAGE1
```

main.asm

```
106            CALL INST_CTRL
107            CALL PRINT_STRING
108            MOV CL, 07h
109
110            MOV DX, PORTA2
111            MOV AL, 00000001b
112            OUT DX, AL
113            CALL DISPLAY_COUNT
114            CALL FINISH
115            POP AX
116            JMP CHECK_DAVBL
117
118        INPUT_OPTION2:
119            CALL CLEAR_SCREEN
120            LEA SI, MESSAGE1
121            MOV AL, 0C4h
122            CALL INST_CTRL
123            CALL PRINT_STRING
124            MOV CL, 04h
125
126            MOV DX, PORTA2
127            MOV AL, 00000010b
128            OUT DX, AL
129            CALL DISPLAY_COUNT
130            CALL FINISH
131            POP AX
132            JMP CHECK_DAVBL
133
134        INPUT_OPTION3:
135            CALL CLEAR_SCREEN
136            LEA SI, MESSAGE1
137            MOV AL, 0C4h
138            CALL INST_CTRL
139            CALL PRINT_STRING
140            MOV CL, 07h
```

main.asm

```
142             MOV DX, PORTA2
143             MOV AL, 00000100b
144             OUT DX, AL
145             CALL DISPLAY_COUNT
146             CALL FINISH
147             POP AX
148             JMP CHECK_DAVBL
149
150          INPUT_OPTION4:
151             CALL CLEAR_SCREEN
152             LEA SI, MESSAGE1
153             MOV AL, 0C4h
154             CALL INST_CTRL
155             CALL PRINT_STRING
156             MOV CL, 04h
157
158             MOV DX, PORTA2
159             MOV AL, 00001000b
160             OUT DX, AL
161             CALL DISPLAY_COUNT
162             CALL FINISH
163             POP AX
164             JMP CHECK_DAVBL
165
166
167
168          FINISH:
169             CALL CLEAR_SCREEN
170             LEA SI, OUTPUT1
171             MOV AL, 0C2h
172             CALL INST_CTRL
173             CALL PRINT_STRING
174             MOV DX, PORTA2
175             MOV AL, 00h
176             OUT DX, AL
```

main.asm

```
183         CLEAR_SCREEN:
184             MOV AL, 01h
185             CALL INST_CTRL
186
187         PRINT_STRING:
188
189             MOV AX, [SI]
190             CMP AL, '$'
191             JE DELAY2
192             CALL DATA_CTRL
193             INC SI
194             JMP PRINT_STRING
195
196         DISPLAY_COUNT:
197
198             MOV AL, 09Dh
199             CALL INST_CTRL
200             MOV AL, 030h
201             ADD AL, CL
202             CALL DATA_CTRL
203             MOV AL, 's'
204             CALL DATA_CTRL
205             CALL DELAY
206             DEC CL
207             CMP CL, 00h
208             JNE DISPLAY_COUNT
209
210
211
212         INST_CTRL:
213          PUSH AX ; preserve value of AL
214          MOV DX, PORTA ; set port of LCD data bus (PORTA)
215          OUT DX, AL ; write data in AL to PORTA
216          MOV DX, PORTB ; set port of LCD control lines (PORTB)
217          MOV AL, 02H ; E=1, RS=0 (access instruction reg)
```

main.asm

```
216         MOV DX, PORTB ; set port of LCD control lines (PORTB)
217         MOV AL, 02H ; E=1, RS=0 (access instruction reg)
218         OUT DX, AL ; write data in AL to PORTB
219         CALL DELAY2 ; delay for 1 ms
220         MOV DX, PORTB ; set port of LCD control lines (PORTB)
221         MOV AL, 00H ; E=0, RS=0
222         OUT DX, AL ; write data in AL to PORTB
223         POP AX ; restore value of AL
224
225         RET
226
227
228      DATA_CTRL:
229         PUSH AX ; preserve value of AL
230         MOV DX, PORTA ; set port of LCD data bus (PORTA)
231         OUT DX, AL ; write data in AL to PORTA
232         MOV DX, PORTB ; set port of LCD control lines (PORTB)
233         MOV AL, 03H ; E=1, RS=1 (access data register)
234         OUT DX, AL ; write data in AL to PORTB
235         CALL DELAY2 ; delay for 1 ms
236         MOV DX, PORTB ; set port of LCD control lines (PORTB)
237         MOV AL, 01H ; E=0, RS=1
238         OUT DX, AL ; write data in AL to PORTB
239         POP AX ; restore value of AL
240
241         RET
242
243
244
245      INIT_LCD:
246
247         MOV AL, 38H ; 8-bit interface, dual-line display
248         CALL INST_CTRL ; write instruction to LCD
249         MOV AL, 08H ; display off, cursor off, blink off
250         CALL INST_CTRL ; write instruction to LCD
```

main.asm

```
245      INIT_LCD:
246
247         MOV AL, 38H ; 8-bit interface, dual-line display
248         CALL INST_CTRL ; write instruction to LCD
249         MOV AL, 08H ; display off, cursor off, blink off
250         CALL INST_CTRL ; write instruction to LCD
251         MOV AL, 01H ; clear display
252         CALL INST_CTRL ; write instruction to LCD
253         MOV AL, 06H ; increment cursor, display shift off
254         CALL INST_CTRL ; write instruction to LCD
255         MOV AL, 0CH ; display on, cursor off, blink off
256         CALL INST_CTRL ; write instruction to LCD
257
258         RET
259
260      DELAY:
261         MOV BX, 07FFFh
262
263         LOOP1:
264            DEC BX
265            NOP
266            JNZ LOOP1
267
268            RET
269
270      DELAY2:
271         MOV BX, 02CAh
272
273         LOOP2:
274            DEC BX
275            NOP
276            JNZ LOOP2
277
278            RET
279
```

```
279
280
281   ENDLESS:
282           JMP ENDLESS
283   CODE    ENDS
284           END START
```