EEE 410 – Microprocessors I

Spring 04/05 – Lecture Notes # 17

Outline of the Lecture

- **I/O Programming in Assembly Language**

PROGRAMMING THE PPI (MODE 0)

Ex: Configure the ports of 8255 (PPI) as follows: port A=input,
port B=output,
port CH=output, port CL=input

(CL=Lower 4 bits of Port C, CH (CU) =Upper 4 bits of Port C)
(Assume that the 8255 PPI is located at 300H)

Soln:

Control register:

			A	CH		B	CL
1	0	0			0		

1 : input
0 : output

			A	CH		B	CL
1	0	0	1	0	0	0	1

```
;This program initializes the PPI located at 300H as follows:
;       port    A    : input
;       port    B    : output
;       port    CH : output
;       port    CL  : input

        .MODEL      SMALL
        .CODE
        MAIN:       MOV  AL,10010001B
                    MOV  DX,303H            ;If PPI is at 300H, then Port A=300H,
                    OUT  DX,AL              ;Port B =301, Port C=302 and Control reg:303H
                    …
                    …
                    …
                    MOV  AH,4CH
                    INT  21H
                    END  MAIN
```

I/O INTERFACE APPLICATIONS

Application 1: Stepper Motor Control

Stepper motors are used for position control applications, such as for the control of the disk drives and in robotics.

- The most common stepper motors have four stator windings that are paired with a center tapped common shown in Figure below (Figure 12.38-39 of the textbook).
- While the conventional motor shaft runs freely, the stepper motor shaft moves in a fixed repeatable increment, which allows one to move it to a precise position.

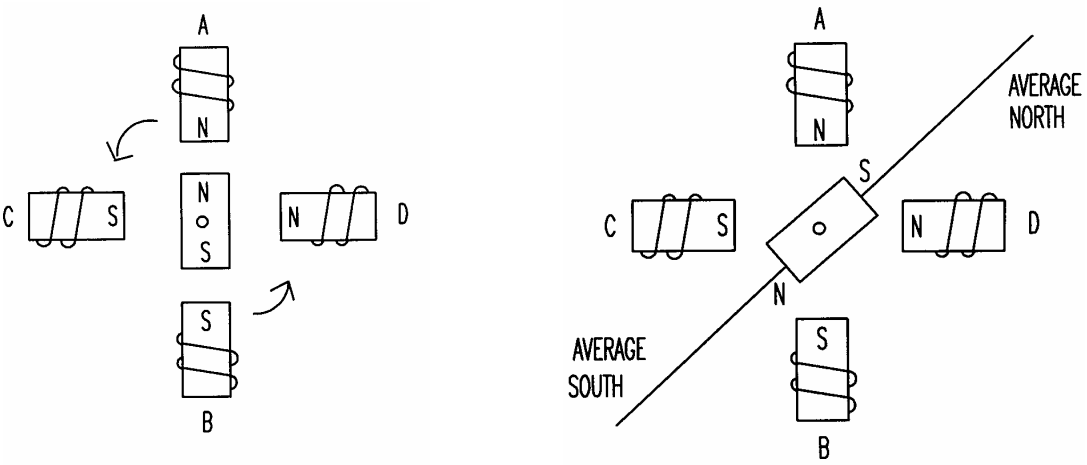


Figure 12-38. Rotor alignment of the stepper motor.

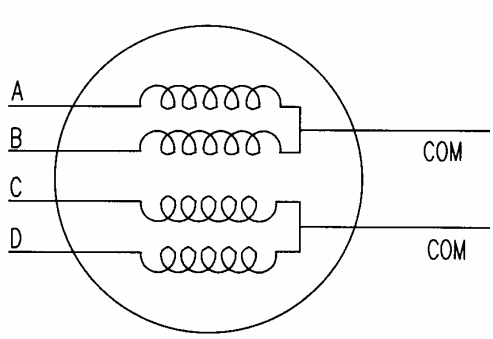


Figure 12-38. Stator windings configuration

- The table below shows a normal 4-step Sequence. Note that once, we start with any of the sequences in Table below (i.e. step 3 [0110]) we must continue in the sequence of steps, 4, 1, 2, and so on.

Clockwise	Step #	Winding A	Winding B	Winding C	Winding D	Counter Clockwise
|	1	**1**	**0**	**0**	**1**	/ \
|	2	**1**	**1**	**0**	**0**	|
|	3	**0**	**1**	**1**	**0**	|
\ /	4	**0**	**0**	**1**	**1**	

Normal 4-Step Sequence

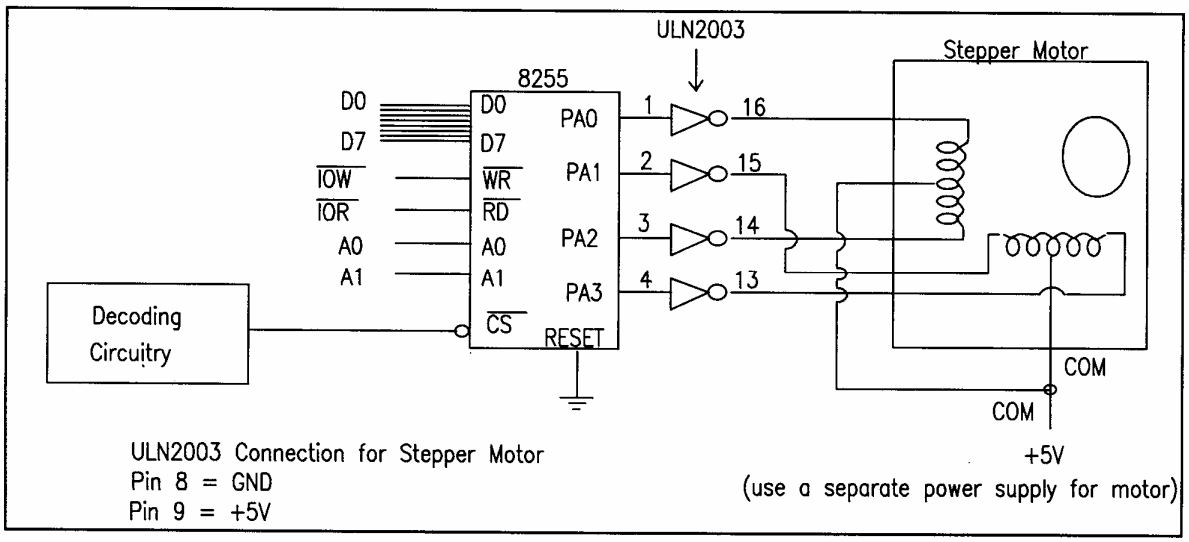


Figure 12-40:8255 Connection to Stepper Motor

Ex: Given the 8255 connection to the stepper motor of Figure 12-40,
- code a program to rotate it continuously.
- press any key on the IBM PC keyboard to stop it.

(Assume that the 8255 PPI is located at 300H)
Use Port A as the output port, then the Control register will contain:

			A	CH		B	CL
1	0	0			0		

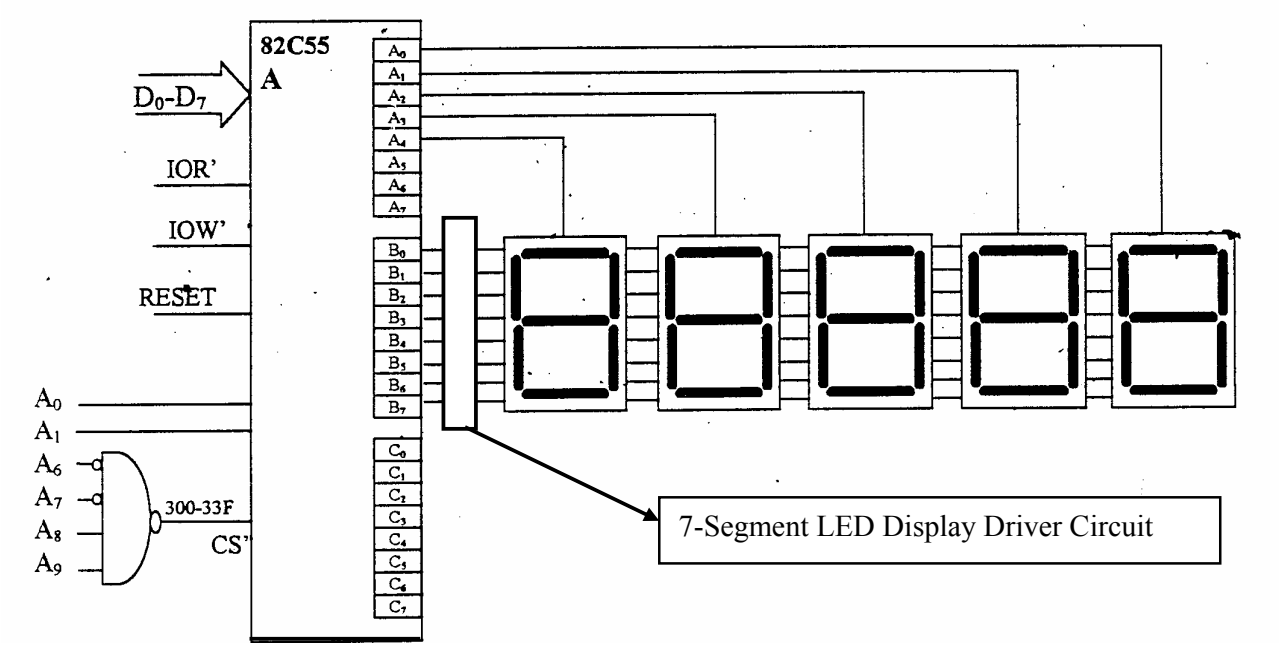
1 : input
0 : output

			A	CH		B	CL
1	0	0	0	0	0	0	0

```
            ............
            MOV  AL,80H          ;control word for all 8255 ports as out
            MOV  DX,303H         ;control register address of 8255
            OUT  DX,AL           ;to control register
            MOV  BL,66H          ;or BL=33H, BL=99H or BL=0CCH
AGAIN:      MOV  AH,01           ;check the key press
            INT  16H             ;using INT 16
            JNZ  EXIT            ;stop if any key pressed
            MOV  AL,BL           ;otherwise send pulse to the stepper motor
            MOV  DX,300H         ;port A address of 8255
            OUT  DX,AL
            MOV  CX,0FFFFH       ;(change this value to see rotation speed)
HERE:       SUB  AH,AH
            LOOP HERE            ;wait for delay
            ROR  BL,1            ;rotate for the next step
            JMP  AGAIN           ;and continue until a key is pressed
```

EXIT:

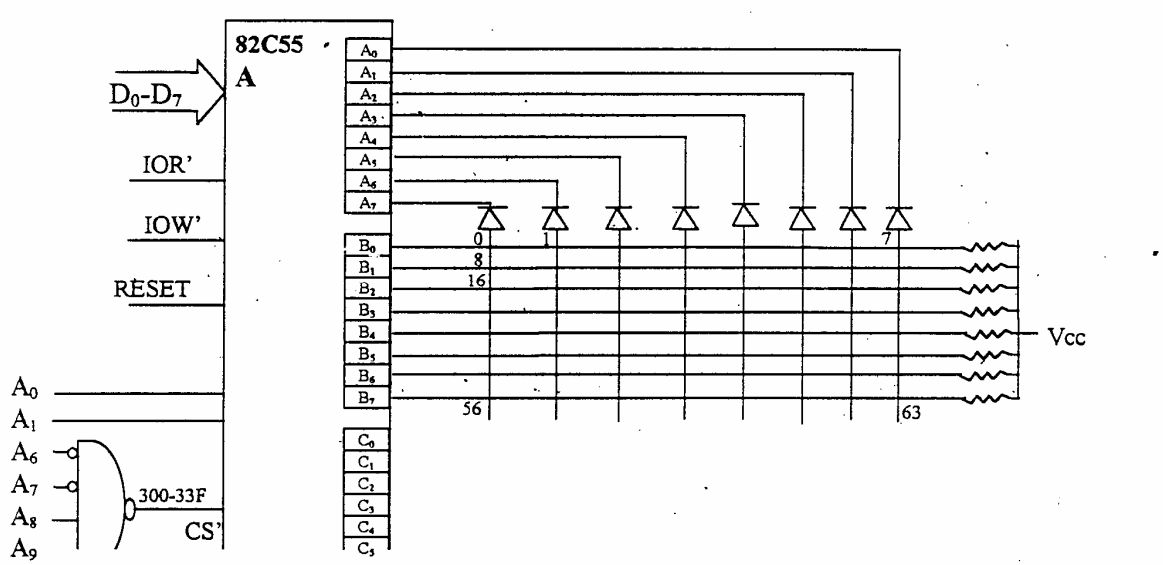
Application 2: 7 Segment Display Control



This program displays 5 hexadecimal digits on the 7-segment LED display units connected to the 8255 PPI as shown above.

```
;------ data segment ------
.DATA
MSG             DB      0EH,0EH,4,1,1
;------ code segment ------
.CODE
MAIN:           MOV     AX,@DATA
                MOV     DS,AX
                MOV     DX,303H
                MOV     AL,10000000B
                OUT     DX,AL
START:          MOV     SI,OFFSET MSG
                MOV     DI,0
                MOV     BL,00010000B
NEXT:           MOV     BH,[SI+DI]
                CALL    DISPLAY         ;CALL THE DISPLAY SUBROUTINE
                SHR     BL,1
                INC     DI
                CMP     DI,4
                JNE     NEXT
                JMP     START
DISPLAY:        MOV     DX,300H         ;DISPLAY SUBROUTINE
                MOV     AL,BL
                OUT     DX,AL
                MOV     DX,301H
                MOV     AL,BH
                OUT     DX,AL
                RET
                END     MAIN
```

Application 3: Keyboard Control



This program reads a keystroke from the 64-key keyboard connected to the PPI as shown above. (The program ignores the debounce effect)

```
;------ data segment ------
.DATA
LETTERS        DB       '0123456789ABCDEFGHIJKLMNOPRSTUVWXYZabcdefghijklmnopqrstuvwxyz.?'
;------ code segment ------
.CODE
MAIN:             MOV    AX,@DATA
                  MOV    DS,AX
                  MOV    DX,303H
                  MOV    AL,10000010B
                  OUT    DX,AL
                  MOV    BL,11111110B
SCAN:             ROR   BL
                  MOV   AL,BL
                  MOV   DX,300H
                  OUT    DX,AL
                  CMP    AL,11111111B
                  JE       SCAN
                  PUSH  BX
                  MOV      DI,0
NEXT:           SHL     BL,1
                  JNC     FINISHC
                  INC      DI
                  JMP     NEXT
FINISHC:        SHL     AL,1
                  JNC     FINISHR
                  ADD     DI,8
                  JMP     NEXT
FINISHR:        MOV    SI,OFFSET LETTERS
                  MOV    DL,[SI+DI]
                  MOV     AH,2
                  INT      21H
                  POP     BX
                  JMP    SCAN
                  END     MAIN
```