Unit 8 8086 Interfacing – 1

Structure:

8.1 Introduction

In the previous unit you studied about the interfacing devices used to interface memory and I/O devices with microprocessor 8086. You also studied about Programmable Peripheral Interface 8255. The key board and display units are the two main components used in microprocessor based system. We use keyboard to send data and instructions and display device to receive information. In systems where a large amount of data needs to be displayed a CRT is used to display the data. In system where only a small amount of data needs to be displayed, simple digit-type displays are often used. In this unit, you will be introduced to the methods of interfacing key board and display units to 8086 microprocessor. You will also study about the 8279 Programmable Keyboard/Display Controller and its interfacing. Also you will also learn about interfacing of 8086 microprocessor with Stepper Motor.

Objectives:

After studying this unit, you should be able to:

- explain the interfacing keyboard to microprocessor
- explain the interfacing display device to microprocessor
- explain the block diagram of 8279

- list and explain the modes of operation of 8279
- Interfacing and Programming of 8279
- discuss the Interfacing of stepper motor.

8.2 Interfacing Key Board to 8086

For interfacing to the microprocessor system, usually push buttons keys are used. Whenever a key is pressed, there are small mechanical vibrations that cause noise on the input, which can cause the microprocessor to detect several keypresses instead of just one. Bouncing happens because of the tendency of any two metal contacts in an electronic device to generate multiple signals as the contacts close or open. You can solve this problem using software or hardware debouncing.

The hardware approach is shown in figure 8.1. It uses a cross-coupled latch formed with two Nand gates.

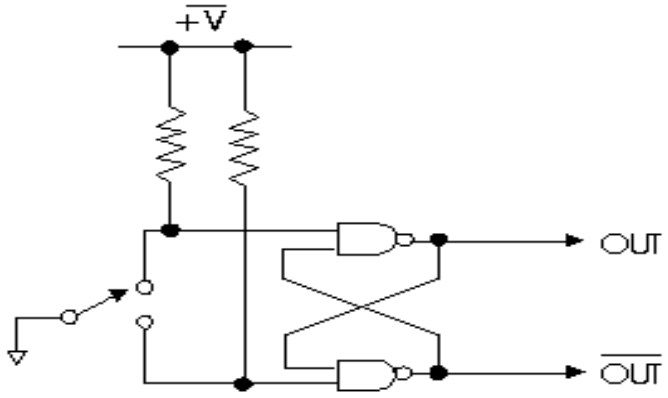


Figure 8.1: Key Debouncing with Hardware approach

You can also use an SR flip flop. The advantage of using a latch is that you get a clean debounce without a delay limitation. It will respond as fast as the contacts can open and close.

But in the software approach, when a key is pressed, the microprocessor waits for atleast 10 ms before it accepts the key as an input. The figure 8.2 shows the flow chart of key debounce with software approach.

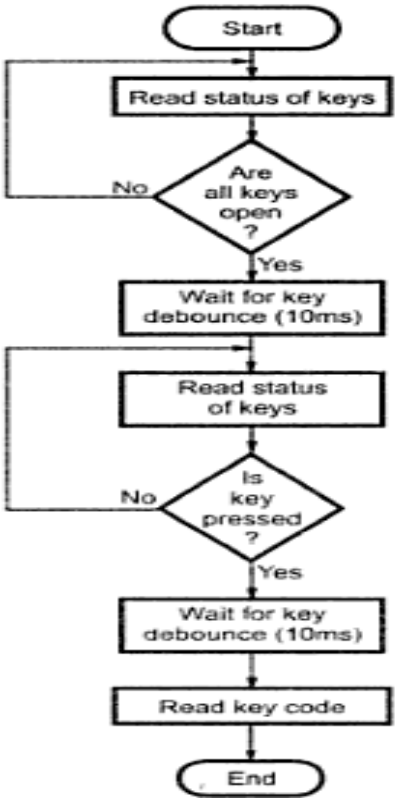


Figure 8.2: Flow chart of key debounce with software approach.

Simple Keyboard Interface:

The figure 8.3 shows the simple keyboard interface.

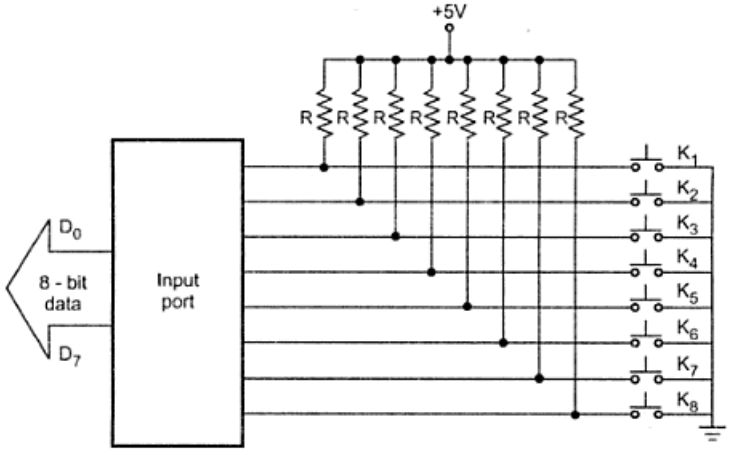


Figure 8.3: Simple keyboard interface

When port pin is logic 1,key is open,otherwise key is closed. The software routine to get key code with key debounce is:

```
START :  IN   AL,IN_PORT        ; Read key status
         CMP  AL,FFH            ; check if keys are open
         JNZ  START             ; if no, goto start otherwise
                                ; continue
         CALL DEBOUNCE_DELAY    ; call debounce delay
AGAIN :  IN   AL, IN_PORT       ; Read key status
         CMP  AL, FFH           ; check if any key is pressed
         JZ AGAIN               ; if no, goto AGAIN; otherwise
                                ; continue
         CALL DEBOUNCE_DELAY    ; call debounce delay
         IN   AL, IN_PORT       ; Get key code
         RET                    ; Return from subroutine
```

In the technique shown in figure 8.3, each key requires separate interface.This means, to interface one key,one input line is required.This is the disdavantage as it requires many line as many keys. This number of lines required can be reduced if keys are put in matrix form.

The figure 8.4 shows the interface of matrix key board.

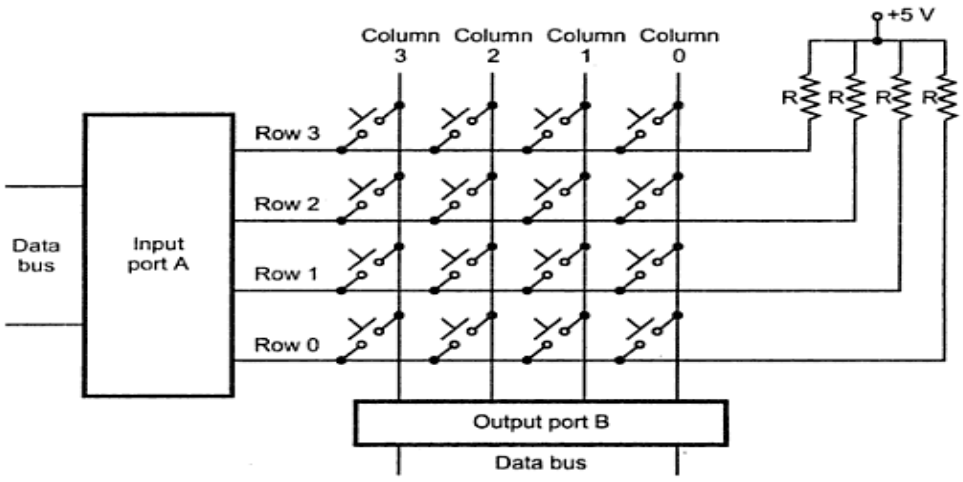


Figure 8.4: Matrix Key Board Interface

The 16 keys are arranged in four rows and four columns. The connection will be made such that when a key is pressed, it shotrs the corresponding

one row and one column. Two ports are required, input port for connecting rows and and output port for connecting columns. The lines connected to rows are called returned lines and the lines connected to columns are called scan lines.

Figure 8.5 shows the interfacing of 8086 with 4 × 4 Keyboard**.**

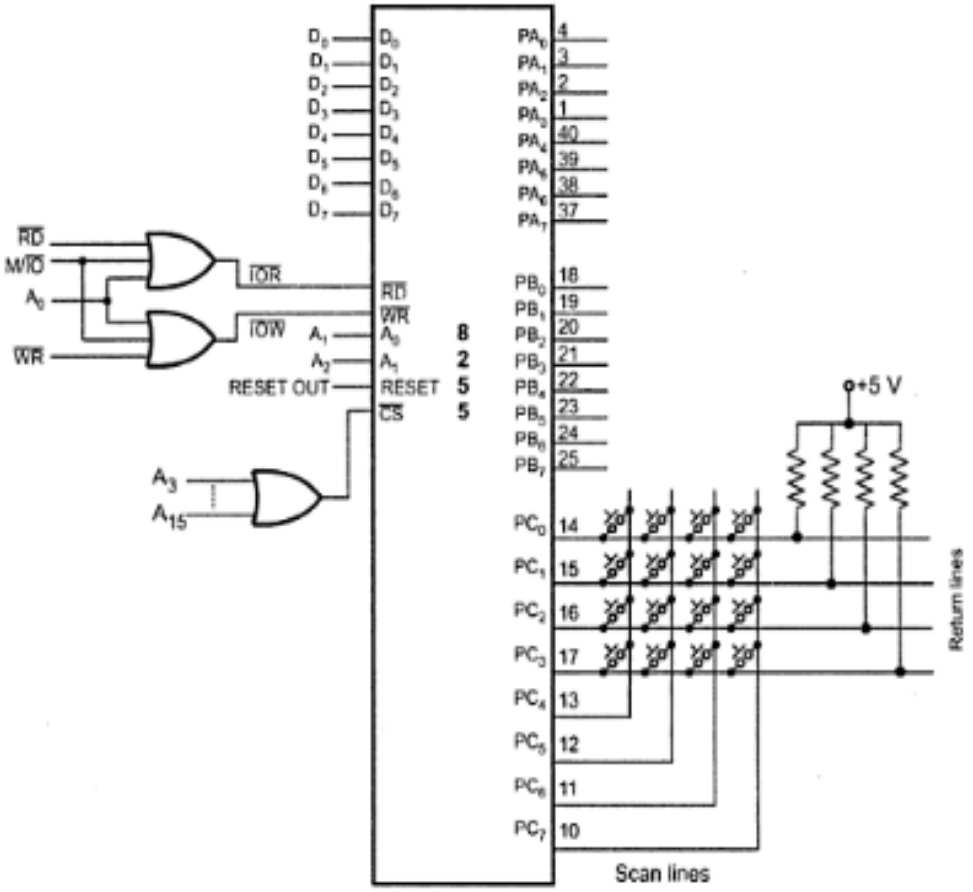


Figure 8.5: interfacing of 8086 with 4 × 4 Keyboard

When all a key is pressed it shorts the corresponding row and column. If the output line of this columnn is low, then it makes the corresponding row line low,other wise the status of row line will be high. The pressed key will be identified by the data sent on the oputput port and the input code received from the input port. Figure 8.6 shows the flow chart for interfacing of 8086.

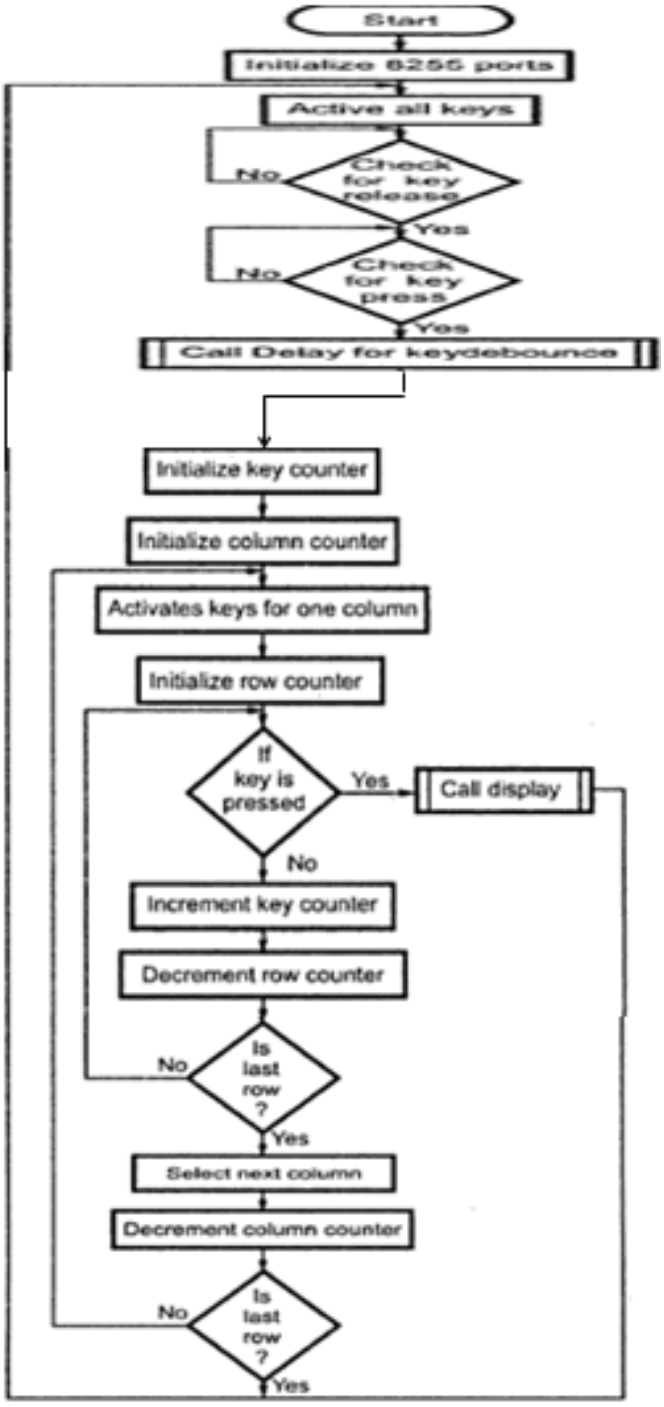


Figure 8.6: Flow chart for interfacing of 8086

The figure 8.7 shows the interfacing of 8086 with an 8 × 8 Key board.

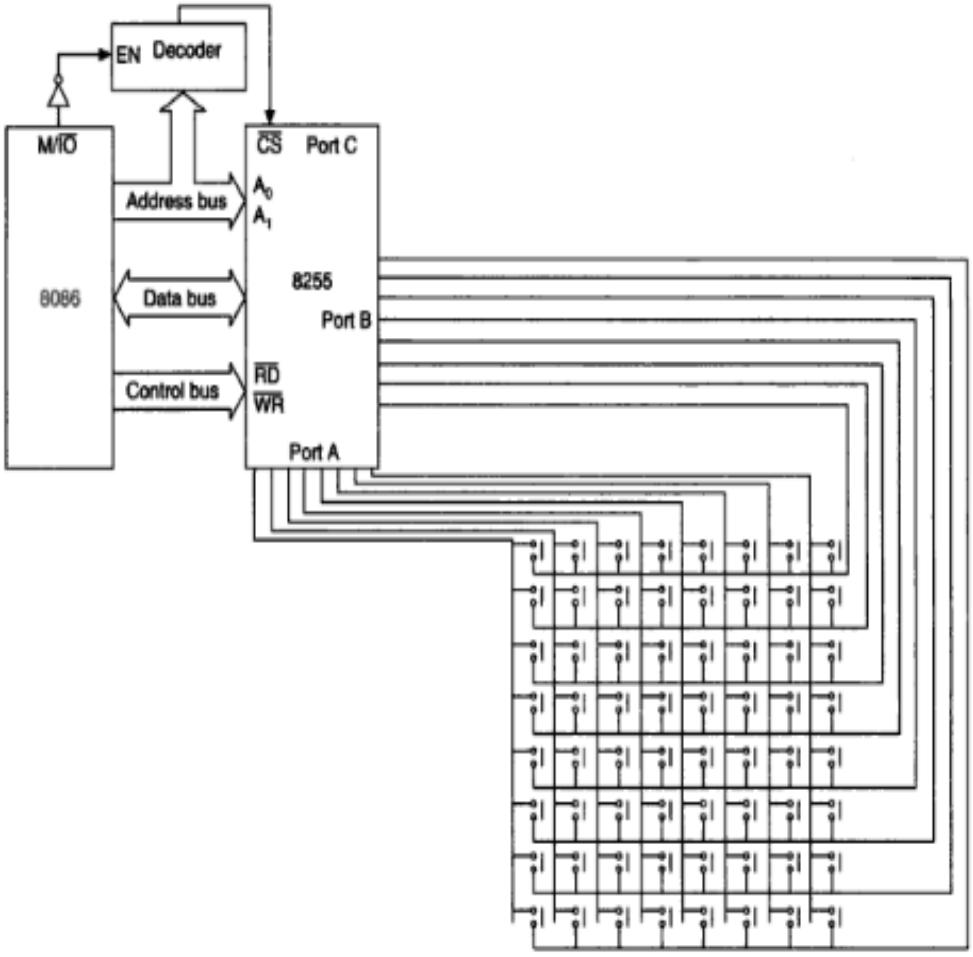


Figure 8.7: Interfacing of 8086 with Keyboard

As shown in the figure 8.7, the interfacing is done using 8255 PPI. The 8086 is being used in maximum mode and port A of 8255 is used for columns and port B for rows. By making use of the lookup table stored in the memory, the 8086 microprocessor will determine the code of the depressed key, then it will initiate the action. The figure 8.8 shows the flow chart for keyboard interface.

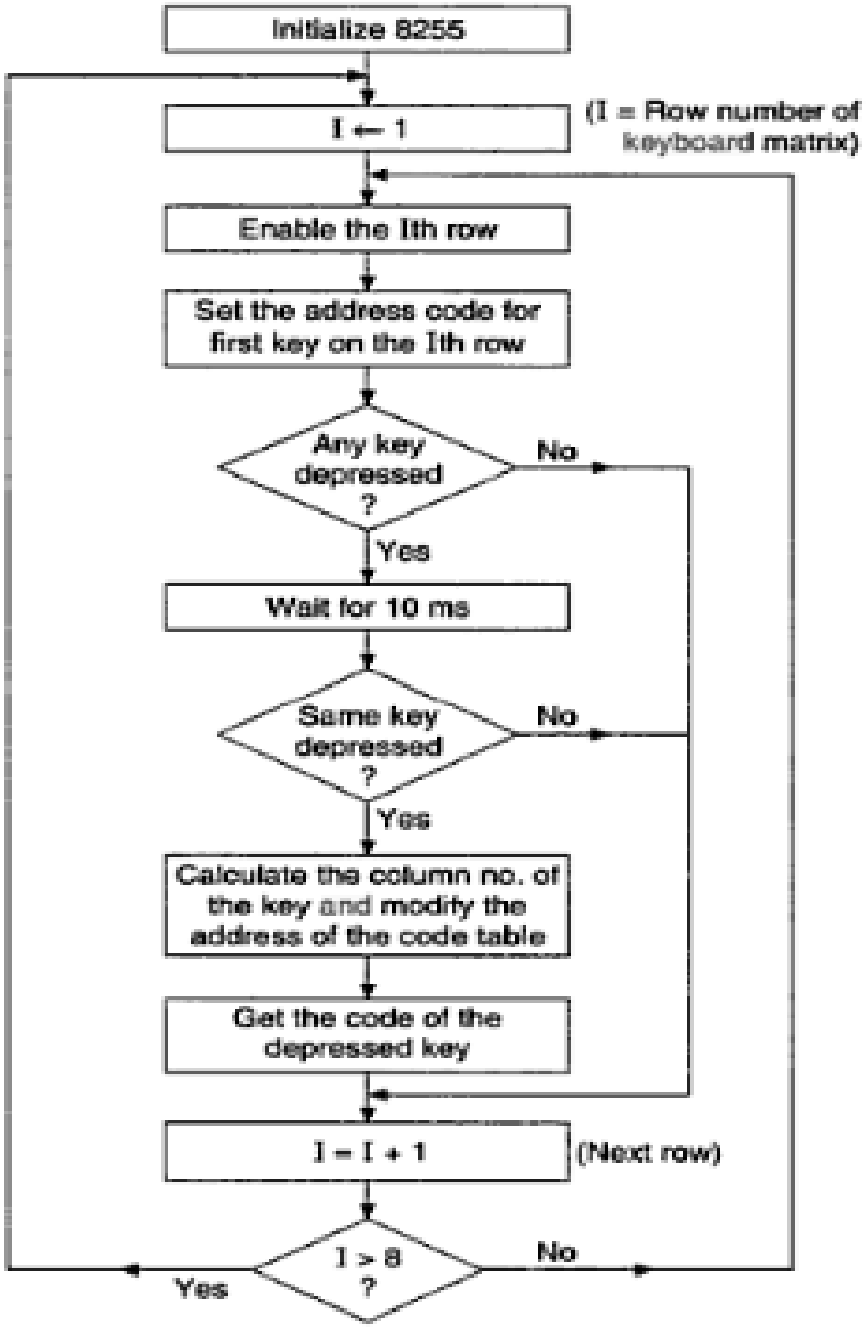


Figure 8.8: Flow chart for 8 × 8 interface

The software program is:

```
;
; Intel 8255 Control Word = 1 0 0 1 0 0 0 0 = 90H , Port A = Input, Port B = Output,
; Mode = 0
; Port Nos—Port A = 00, Port B = 01, Port C = 02, Control Register Port = 03
;
                DATA  SEGMENT
                      ROW       DW  0
                      COL       DB  0
                      XXXX      DB  –, –, –, –, –, –, –, ; XXXX is Code Table
                DATA  ENDS

                CODE  SEGMENT
                      ASSUME CS: CODE, DS: DATA
                      MOV AX, DATA
                      MOV DS, AX
;
; Initialize Intel 8255.
;
                      MOV AL, 90H
                      OUT 03

     LOOP1:           MOV CX, 0001H   ; Row Counter
     LOOP2:           MOV ROW, CX

; Set address XXXX of Code Table (First Row) in BX.
;
```

```
                    LEA  BX, XXXX
                    DEC  CX ; (CX) = ROW No. – 1
                    SAL  CX, 1
                    SAL  CX, 1
                    SAL  CX, 1 ; (X) = (ROW No. – 1) × 8
                    ADD  BX, CX ; (BX) = Address of first element of row
;
; Scan for key depression.
;
                    MOV  CX, ROW
                    MOV  AL, CL
                    OUT  01
                    IN   00
                    AND  AL, FFH
                    JZ   LOOP3   ; No key depressed. Scan the next row
;
; Key depressed.
;
                    MOV  DL, AL

; Delay the loop for debouncing.
;
                    MOV  CX, FFFFH
  DD:               NOP
                    LOOP DD
 ;
 ; Again scan for key depression.
 ;
                      IN   00
                      CMP  AL, DL
                      JNZ  LOOP3   ; False key depression
 ;
 ; Key depression validated. Calculate the Column No.
 ;
                      MOV  CL, 00
                      CLC ; Clear carry flag
   RR:                RCR  AL, 1
                      JC   CODEIS
                      INC  CL
                      JMP  RR
 ;
 ; Find the key code.
 ;
   CODEIS:            ADD  BX, CX ; (BX) = Address of key code
                      MOV  AL, (BX)
```

```
                    CALL USE   ; Use the code for processing
                    JMP  ENDIS
;
; Increment the Row No. and repeat.
;
  LOOP3:            MOV  CX, ROW
                    INC  CX
                    CMP  CX, 0009H
                    JZ   LOOP1
                    JMP  LOOP2
  ENDIS:            NOP
          CODE      ENDS
```

Self Assessment Questions

1. _____ happens because of the tendency of any two metal contacts in an electronic device to generate multiple signals as the contacts close or open.
2. Simple key board debouncing with hardware approach uses _____ gates.
3. In matrix keyboard interface, the lines connected to rows are called _____ lines and the lines connected to columns are called _____ lines.

8.3 LED Display Interfacing

Display are used for displaying letters,characters,numbers or information to the outside world/users. When only small information or data has to be displyed,then you can use Light Emmitting Diode(LED) or Liquid Crystal Display(LCD) displys. When you want to display large amount of data, you can use CRT. Seven segment display is an example of LED displays. LEDs are very popular display devices. They are easy to use and give very bright and pleasing display which can be viewed equally well from any viewing angle, unlike LCDs. The only drawback with LED displays is the high amount of current they need, unlike LCDs which need very low power.This displays are generally used to diplay numbers from 0 through 9 and seven LEDs are arranged in a seven segment fashion as shown in figure 8.9. The figure 8.10 shows the segments that are used to display numbers.

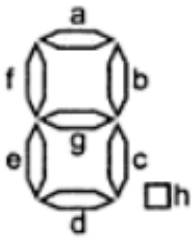


Figure 8.9: of seven segment display

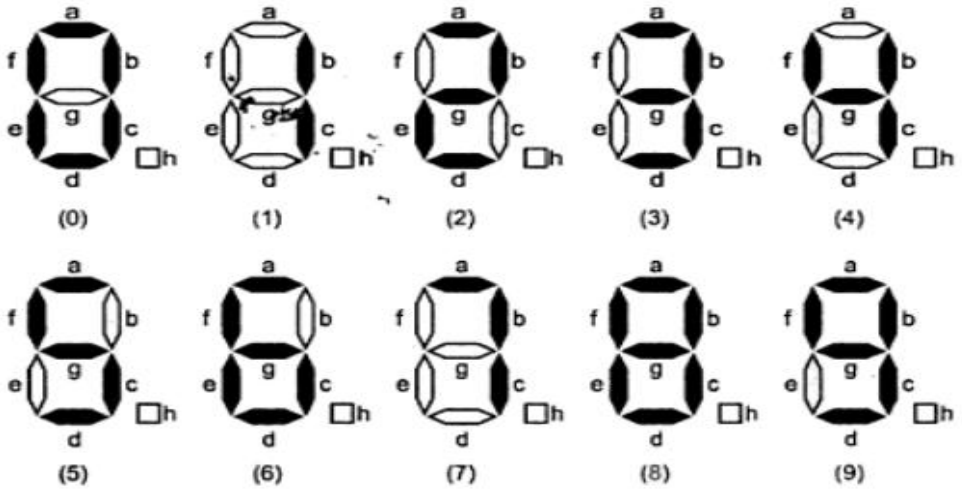


Figure 8.10: Segments used to diplay number in seven segment display

Types of Seven Segment Displays:

There are two types of seven segment displays. They are:

Common Anode type

Common Cathode type

These two typse of displays are shown in figure 8.11.

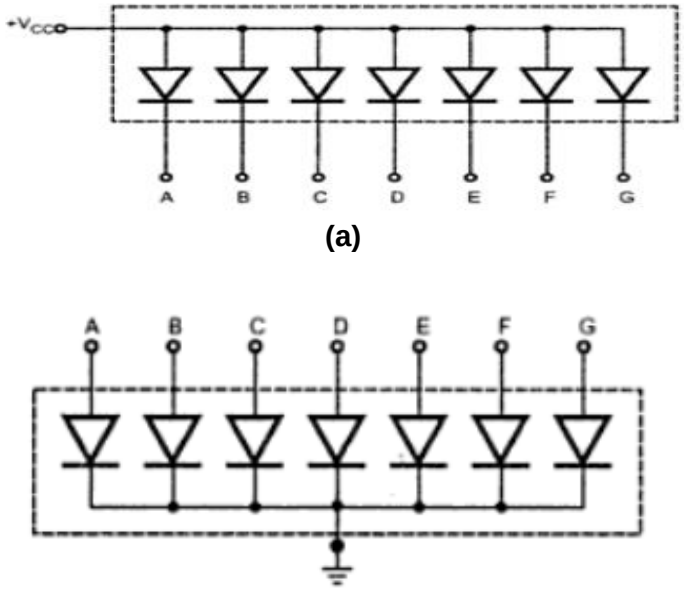


Figure 8.11: Seven segment display types(a) Common Anode. (b) Common Cathode

Suppose we need a number of LEDs for displays, we usually use only one power source for all of them. In that case, they are connected together either as 'common anode' or 'common cathode' LED displays. In figure 8.9(a), which is a common anode, the anodes of all the LEDs are connected together to a Vcc. If we want to light up any of the LEDs, then we have to apply a '0' to cathode of that LED. In figure 8.9(b), which is 'a common cathode', the cathodes of all the LEDs are connected together to ground (0). If we want to light up any of the LEDs, then apply a '1' to anode of that LED. More important applications of LEDs are as alphanumeric displays. In that case, seven segment LEDs are used, in which seven LEDs are arranged as the segments of a display arranged in a particular shape which when selectively lighted up, give the display of alphanumeric characters. By lighting up all the segments, we get '8' displayed. We can have one more segment in this display and it is for the decimal point.

The figure 8.12 shows the interfacing of LED display in which anode type of display is used.

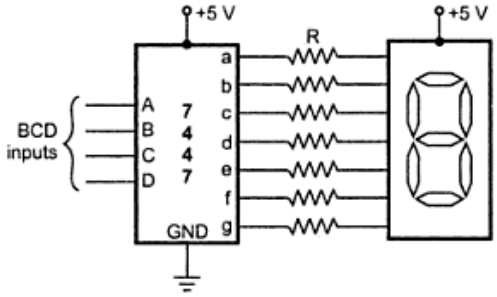


Figure 8.12: Interfacing of LED

As shown in the figure 8.12, a BCD to seven segment decoder IC 7447 is used to apply low voltages at all the cathodes of LEDs according to BCD inputs applied to 7447. Resistors R are connected to limit the current through LEDs. This type of arrangment is not suitbale incase if you use more number of LEDs, let say 8. This is because the circuit works well for one or two LEDs, But for more LEDs it posess the problem of more power consumption,separate BCD to seven segment decoder for each display. To solve this problem, a multiplexed display is used and is shown in figure 8.13.

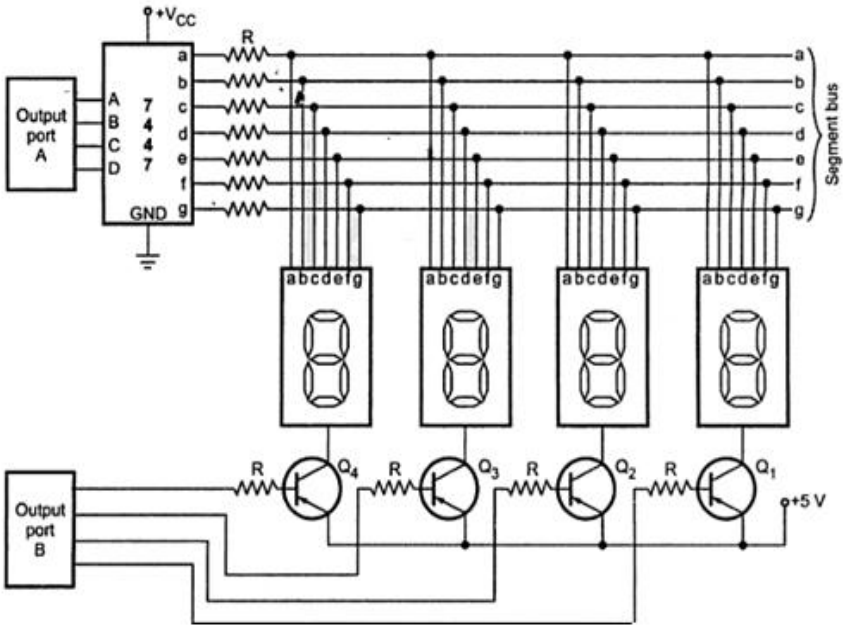


Figure 8.13: Multiplxed Seven segment display

The figure 8.14 shows the interfacing of 8086 with seven segment LED display.

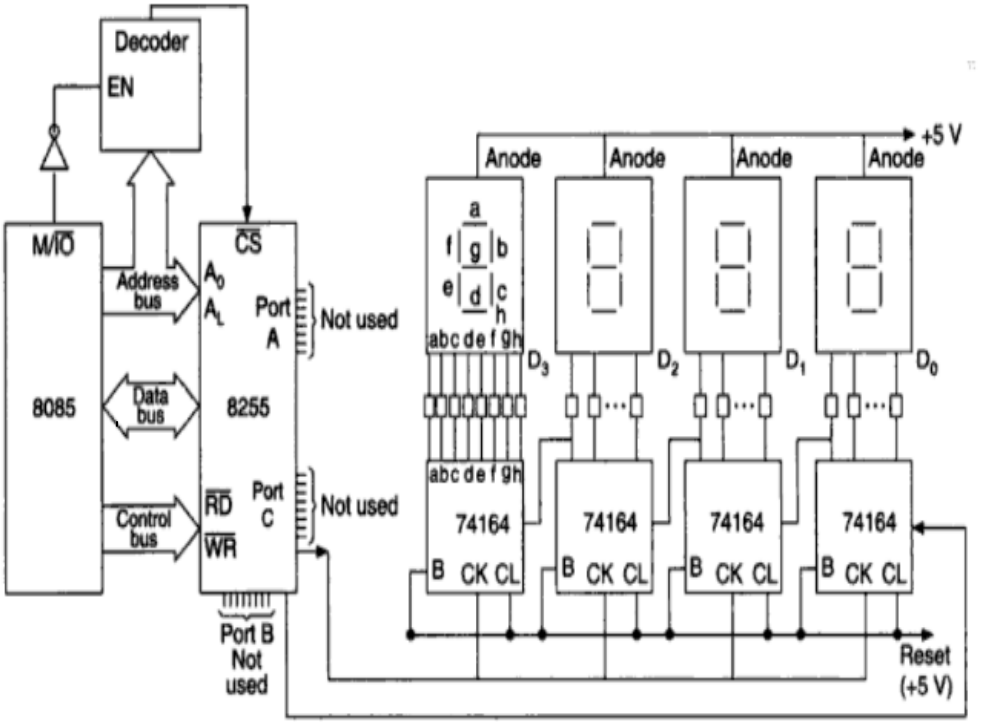


Figure 8.14: Interfacing of 8086 with Seven Segment LED display

Only two port lines of the 8255 have been used for clock and data. Other lines are free for interfacing with other circuits of the system.

The flow chart for seven segment LED display interface is shown in figure 8.15.

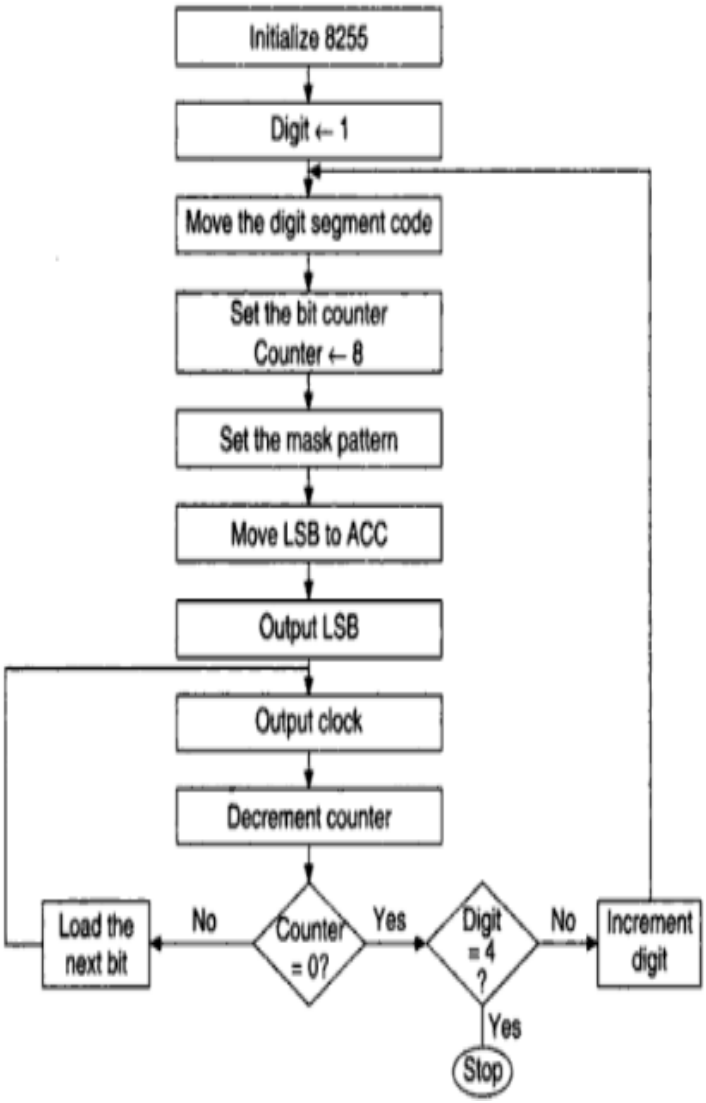


Figure 8.15: The flow chart for seven segment LED display interface

The software program is:

```
;
; Intel 8086 software for seven-segment display
;
; Intel 8255 initialization
;
; Ports A, B, C are output ports.
; Control Word = 1 0 0 0 0 0 0 0 = 80H
; Port Nos.—A = 00, B = 01, C = 02, Control Register = 03.
; Circuit connections
; C0 (0th bit of port C) connected to clock, B0 (0th bit of port B) connected to code input
;

                DATA SEGMENT
                     DISP-CODE  DB –, –, –, –, –, –, –, —; Display codes
                     DIGIT      DB 0
                DATA ENDS

                CODE SEGMENT
                     ASSUME CS: CODE, DS: DATA
                     MOV AX, DATA
                     MOV DS, AX
;
; Initialize Intel 8255.
;
                     MOV AL, 80H
                     OUT 03
                     LEA  BX, DISP-CODE   ; (BX) = Address of DISP-CODE
                     MOV DX, 00   ; Digit No.
  LOOP1:             ADD BX, DX   ; (BX) = Address of digit code to be displayed
                     MOV AL, (BX)
                     MOV DIGIT, AL
```

```
;
; Output the digit code bit-by-bit with clock.
;
                    MOV  CL, 08H
   LOOP2:           AND  AL, 01H
                    OUT  01
;
; Output Clock.
;
                    MOV  AL, 00
                    OUT  02
                    MOV  AL, 01
                    OUT  02
                    MOV  AL, 00
                    OUT  02
;
; Repeat for the next segment.
;
                    DEC  CL
                    JZ   MSB
                    MOV  AL, DIGIT
                    RCR  AL, 1
                    MOV  DIGIT, AL
                    JMP  LOOP2
;
; Repeat for the next digit.
;
   MSB:             CMP  DX, 0003H
                    JZ   FINISH
                    INC  DX
                    JMP  LOOP1
   FINISH:          NOP
             CODE   ENDS
```

Self Assessment Questions

4. Seven segment display is an example of _____ displays.
5. Which IC is used as BCD to seven segment decoder?

8.4 8279 Programmable Keyboard/Display Controller

8279 is an intel's general purpose Keyboard/Display controller. It simultaneously drives the display of a system and interfaces a Keyboard with the CPU. The Keyboard display interface scans the Keyboard to identify if any key has been pressed and sends the corresponding code of the key to the CPU. It also transmits the data received from the CPU, to the display device. The controller does both of these functions in in repetitive

fashion without involving the processor. You can interface the Keyboard to the processor either in the interrupt or the polled mode. In the interrupt mode, the processor is requested service only if any key is pressed, otherwise the processor can proceed with its main task. In the polled mode, the processor periodically reads an internal flag of 8279 to check whether any key has been pressed.

The figure 8.16 shows the functional block diagram of 8279.

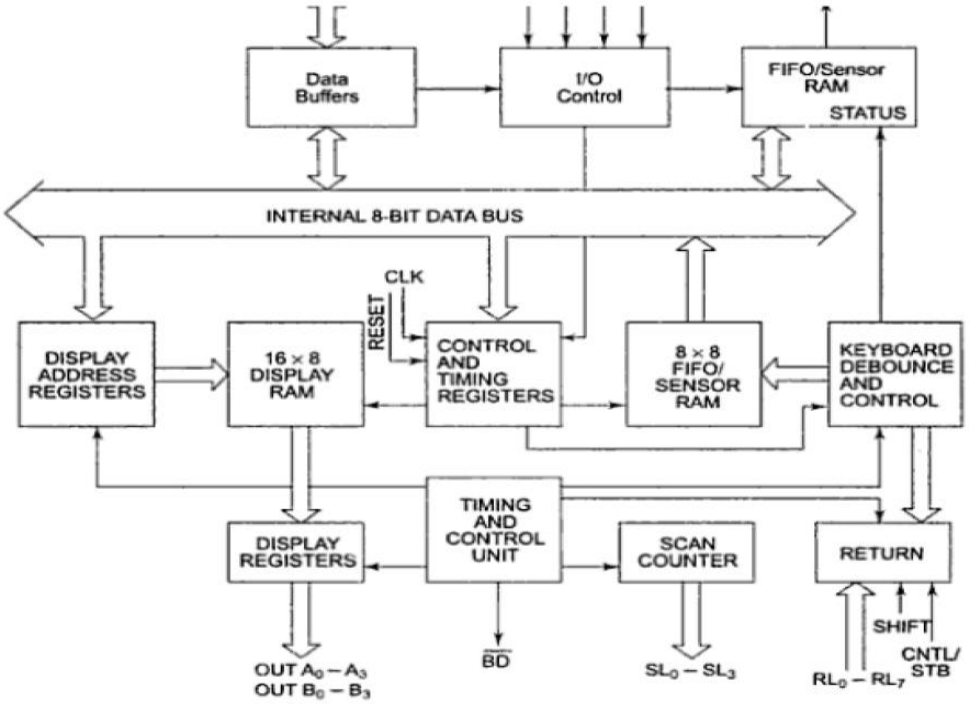


Figure 8.16: Functional Block diagram of 8279.

The Keyboard section can interface an array of a maximum of 64 keys. The Keyboard entries are debounced and stored in an 8-byte FIFO RAM and the processor reads the key codes. If more than eight characters are entered in the FIFO (i.e. more that eight keys are pressed), before any FIFO read operation, the overrun status is set. If a FIFO contains a valid key entry, the processor is interrupted (in interrupt mode) or the processor(CPU) checks the status (in polling) to read the entry.

The 8279 normally provides a maximum of sixteen 7-segment display interface with CPU. It has 16-byte display RAM that can be used either as

an integrated block of 16x8-bits or two 16x4-bit block of RAM. The CPU controls the data entry into RAM block using the control word of 8279.

The Keyboard display controller chip 8279 provides

1. A set of four scan lines and eight return lines for interfacing keyboards.
2. A set of eight output lines for interfacing display.

I/O Control and Data Buffer

The I/O control section controls the flow of data to/from the 8279. The data buffer interface the external bus of the system with internal bus of 8279. The I/O section is enabled only if D is low.

The pin A0, RD and WR select the command, status or data read/write operations carried out by the processor with 8279.

Control and Timing Register and Timing Control

These registers store the keyboard and display modes. The registers are written with Ao=1 and WR =0. The timing and control unit controls the basic timings for the operation of the circuit.

Scan Counter

It divides the operating frequency of 8279 to generate scan keyboard and scan display frequencies. The Scan Counter has two modes to scan the key matrix and refresh the display.

Return Buffers and Keyboard Debounce and Control

This block scans for a Key closure row-wise. If it is detected, the Keyboard debounce unit debounces the key entry (i.e. wait for 10 ms). After the debounce period, if the key continues to be detected. The code of the Key is directly transferred to the sensor RAM along with SHIFT and CONTROL key status.

FIFO/Sensor RAM and Status Logic

In Keyboard or strobed input mode, this block acts as 8-byte first-in-first-out (FIFO) RAM. Each pressed key code is entered in the order and read by the processor, till the RAM becomes empty. The status logic generates an interrupt request after each FIFO read operation till the FIFO is empty. In scanned sensor matrix mode, this unit acts as sensor RAM.

Display Address Registers and Display RAM

The Display address registers hold the addresses of the word currently being written or read by the processor to or from the display RAM. The contents of the registers are automatically updated by 8279 to accept the next data entry by CPU. The 16-byte display RAM contains the 16-byte of data to be displayed on the sixteen 7-segment displays in the encoded scan mode.

Pin Diagram of 8279

The figure 8.17 shows the pin diagram of 8279.

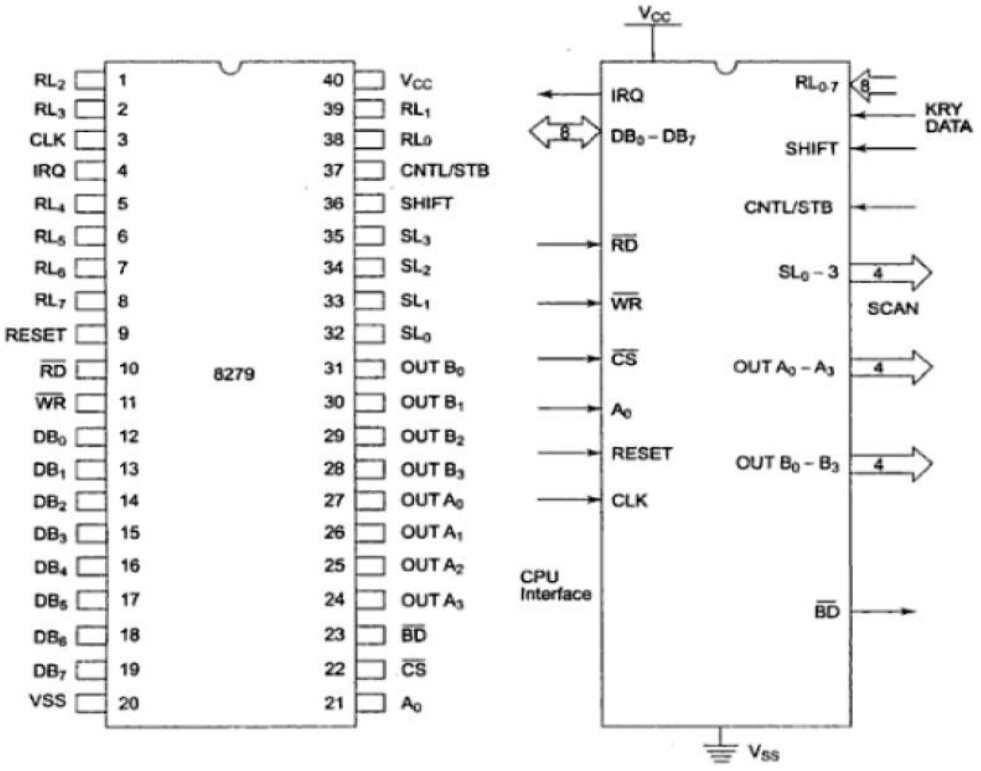


Figure 8.17: Pin Diagram of 8279

DB0 - DB7: These are bidirectional data bus lines. The data and command words to and from the CPU are transferred on these lines.

CLK: This is a clock input used to generate internal timings required by 8279.

RESET: A high on pin resets 8279. After resetting 8279, it will be in sixteen 8-bit display, left entry encoded scan, 2-key lock out mode. The clock prescaler is set to 31.

CS chip select: A low on this line enables 8279 for normal read or write operations. Otherwise this pin should be high.

A0: A high on the A0 line indicates the transfer of a command or status information. A low on this line indicates the transfer of data. This is used to select one of the internal registers of 8279.

RD, WR : (Input/Output) READ/WRITE) input pins enable the data buffer to receive or send data over the data bus.

IRQ: This interrupt output line goes high when there is data in the FIFO sensor RAM. The interrupt line goes low with each FIFO RAM read operation. If the FIFO RAM further contains any Key-code entry to be read by the CPU, this pin again goes high to generate an interrupt to the CPU.

Vss, Vcc: These are the ground and power supply lines for the circuit.

SL0-SL3 – Scan Lines: These lines are used to scan the keyboard matrix and display digits.

RL0-RL7 – Return Lines: These are the input lines which are connected to one terminal of keys, while the other terminal of the keys are connected to the decoded scan lines. These are normally high, but will be pulled to low when a key is pressed.

SHIFT: The status of the Shift input line is stored along with each key code in FIFO in the scanned keyboard mode. Till it is pulled low with a key closure it is pulled up internally to keep it high.

CNTL/STB-CONTROL/STROBED I/P Mode: This line is used as a control input and stored in FIFO on a key closure in the Keyboard mode. The line is a strobe line that enters the data into FIFO RAM, in the strobed input mode.

BD – Blank Display: This output pin is used to blank the display during digit switching or by a blanking command.

OUT A0 – OUT A3 and OUT B0 – OUT B3: These are the output ports for two 16x4 (or one 16 x 8) internal display refresh registers. The data from these lines is synchronized with the scan lines to scan the display and keyboard. The two 4-bit ports may also be used as one 8-bit port.

8.4.1 Modes of Operation of 8279

8279 operates in two modes. They are:

i) Input (Keyboard) modes
ii) Output (Display) modes

Input (Keyboard) modes: 8279 provides three input modes, they are:

1. **Scanned Keyboard Mode:** This mode allows a key matrix to be interfaced using either encoded or decoded scans. In the encoded scan, an 8 x 8 keyboard or in decoded scan, a 4 x 8 Keyboard can be interfaced. The code of key pressed with SHIFT and CONTROL status is stored into the FIFO RAM.
2. **Scanned Sensor Matrix:** In this mode, a sensor array can be interfaced with 8279 using either encoder or decoder scans. With encoder scan 8 x 8 sensor matrix or with decoder scan 4 x 8 sensor matrix can be interfaced. The sensor codes are stored in the CPU addressable sensor RAM.
3. **Strobed Input:** In this mode, if the control line goes low, the data on return lines is stored in the FIFO byte by byte.

Output (Display) Modes:

8279 provides two output modes for selecting the display options.

1. **Display Scan:** In this mode, 8279 provides 8 or 16 character multiplexed displays those can be organized as dual 4-bit or single 8-bit display units.
2. **Display Entry:** The Display data is entered for display either from the right side or from the left side.

Details of Modes of Operation

1. Scanned Keyboard Mode with 2 Key Lockout

In this mode of operation, when a key is pressed, a debounce logic comes into operation. The Key code of the identified key is entered into the FIFO with SHIFT and CNTL status, provided the FIFO is not full.

2. Scanned Keyboard with N-key Rollover

In this mode, each key depression is treated independently. When a key is pressed, the debounce circuit waits for 2 keyboard scans and then checks whether the key is still depressed. If it is still depressed, the code is entered in FIFO RAM.

3. Scanned Keyboard Special Error Mode

This mode is valid only under the N-Key rollover mode. This mode is programmed using end interrupt/error mode set command. If during a single debounce period (two Keyboard scan) two keys are found pressed, this is considered a simultaneous depression and an error flag is set. This flag, if set, prevents further writing in FIFO but allows generation of further interrupts to the CPU for FIFO read.

4. Sensor Matrix Mode

In the Sensor Matrix mode, the debounce logic is inhibited the 8-byte memory matrix. The status of the sensor switch matrix is fed directly to sensor RAM matrix.

Display Modes

There are various options of data display. The first one is known as left entry mode or type writer mode. Since in a type writer the first character typed appears at the left-most position, while the subsequent characters appears successively to the right of the first one. The other display format is known as right entry mode, or calculator mode, since the calculator the first character entered appears at the right-most position and this character is shifted one position left when the next character is entered.

1. Left Entry Mode

In the Left entry mode, the data is entered from the left side of the display unit. Address 0 of the display RAM contains the leftmost display character and address 15 of the RAM contains the rightmost display character.

2. Right Entry Mode

In the right entry mode, the first entry to be displayed is entered on the rightmost display. The next entry is also placed in the right most display but after the previous display is shifted left by one display position.

8.4.2 Command Words of 8279

All the Command words or status words are written or read with A0 = 1 and CS = 0 to or from 8279.

a) Keyboard Display mode set

The format of the command word to select different modes of operation of 8279 is shown in figure 8.18 with its bit definitions. Table 8.1 gives the Command word to select different modes

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_0
0	0	0	D	D	K	K	K	1

Figure 8.18: Format of the command word

Table 8.1: Command word to select different modes

D	*D*	*Display modes*
0	0	Eight 8-bit character Left entry
0	1	Sixteen 8-bit character Left entry (Default after reset)
1	0	Eight 8-bit character Right entry
1	1	Sixteen 8-bit character Right entry

K	*K*	*K*	*Keyboard modes*
0	0	0	Encoded scan, 2 key lockout (Default after reset)
0	0	1	Decoded scan, 2 key lockout
0	1	0	Encoded scan N-Key roll over
0	1	1	Decoded scan N-Key roll over
1	0	0	Encoded scan sensor matrix
1	0	1	Decoded scan sensor matrix
1	1	0	Strobed Input Encoded scan
1	1	1	Strobed Input Decoded scan

b. Programmable Clock

The clock for operation of 8279 is obtained by dividing the external clock input signal by a programmable constant called prescaler. The format of this is shown in figure 8.19.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_0
0	0	1	P	P	P	P	P	1

Figure 8.19: Command word for clock operation

c. Read FIFO/Sensor RAM

The format of this command is given as shown in figure 8.20.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_0
0	1	0	AI	X	A	A	A	1

X - don't care
AI - Auto increment flag
AAA - Address pointer to 8 bit FIFO RAM

Figure 8.20: Command Word for Read FIFO/Sensor RAM operation

This word is written to set up 8279 for reading FIFO/Sensor RAM. In scanned keyboard mode, AI and AAA bits are of no use. The 8279 will automatically drive data bus for each subsequent read, in the same sequence, in which the data was entered.

d. Read Display RAM

This command (refer figure 8.21) enables a programmer to read the display RAM data.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_0
0	1	1	AI	A	A	A	A	1

Figure 8.21: Command Word for Read Display RAM

The CPU writes this command word to 8279 to prepare it for display RAM read operation. AI is auto incremented flag and AAAA, the 4-bit address, points to the 16-byte display RAM that is to be read. If AI = 1, the address will be automatically, incremented after each read or write to the display RAM.

e. Write Display RAM

The format of this command is shown in figure 8.22.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_0
1	0	0	AI	A	A	A	A	1

AI - Auto increment flag
AAAA - 4-bit address for 16-bit display RAM to be written

Figure 8.22: Command Word for Write Display RAM

Other details of this command are similar to the Read Display RAM Command.

f. Display Write Inhibit/Blanking

The IW (Inhibit write flag) bits are used to mask the individual nibble. Here D0 and D2 correspond to OUTB0– OUTB3 while D1 and D3 correspond to OUTA0-OUTA3 for blanking and masking respectively. Figure 8.23 shows the command word for display Write Inhibit/Blanking.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_0
1	0	1	X	IW	IW	BL	BL	1

Output nibbles → A B A B

Figure 8.23: Command Word for Display Write Inhibit/Blanking

g. Clear Display RAM

The figure 8.24 shows the command word format for Clear Display RAM

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_0
1	1	0	CD_2	CD_1	CD_0	CF	CA	1

Figure 8.24: Command Word for Clear Display RAM

The CD2, CD1, CD0 is a selectable blanking code to clear all the rows of the display RAM as given below. The characters A and B represents the output nibbles.

CD	CD1	CDo	
1	0	x	All Zeros (x don't care) AB = 00
1	1	0	A3-Ao = 2(0010) and B3-Bo = 00(0000)
1	1	1	All ones (AB = FF), i.e. clear RAM

Here, CA represents clear All and CF represents Clear FIFO RAM

End Interrupt/Error Mode Set

The figure 8.25 shows the command word for this operation.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_0
1	1	1	E	x	x	x	x	1

x—do not care

Figure 8.25: Command Word for End Interrupt/Error Mode Set

For the sensor matrix mode, this command lowers the Interrupt request (IRQ) line and enables further writing into the RAM. Otherwise, if a charge in sensor value is detected, IRQ goes high that inhibits writing in the sensor RAM.

8.4.3 Key-code and Status Data Formats

In this section, you will study various formats of the Key-code/Sensor data in their respective modes of operation and the FIFO Status Word formats of 8279.

Key-code Data Formats:

After a valid Key closure, the key code is entered as a byte code into the FIFO RAM, in the following format, in scanned keyboard mode. The Key code format contains 3-bit contents of the internal row counter, 3-bit contents of the column counter and status of the SHIFT and CNTL Keys The data format of the Key code in scanned keyboard mode is shown in figure 8.26.

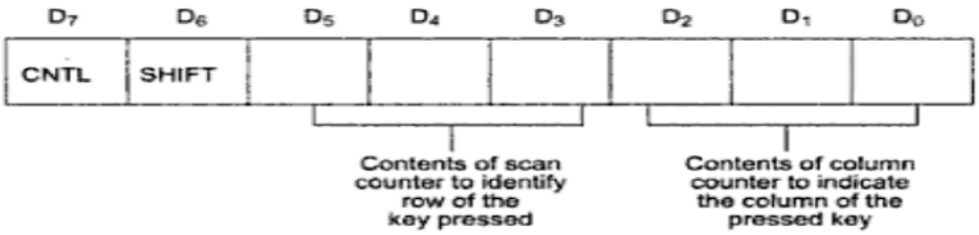


Figure 8.26: Data format of the Key code in scanned keyboard mode

In the sensor matrix mode, the data from the return lines is directly entered into an appropriate row of sensor RAM that identifies the row of the sensor that changes its status. The SHIFT and CNTL Keys are ignored in this mode. RL bits represent the return lines. Rn represents the sensor RAM row number that is equal to the row number of the sensor array in which the status change was detected. Data format of the sensor code in sensor matrix mode is shown in figure 8.27.

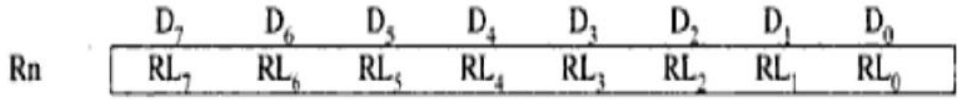


Figure 8.27: Data format of the sensor code in sensor matrix mode.

FIFO Status Word:

The FIFO status word is used in keyboard and strobed input mode to indicate the error. Overrun error occurs, when an already full FIFO is attempted an entry, under run error occurs when an empty FIFO read is attempted. FIFO status word also has a bit to show the unavailability of FIFO RAM because of the ongoing clearing operation.

In sensor matrix mode, a bit is reserved to show that at least one sensor closure indication is stored in the RAM. The S/E bit shows the simultaneous multiple closure error in special error mode. In sensor matrix mode, a bit is reserved to show that at least one sensor closure indication is stored in the RAM, The S/E bit shows the simultaneous multiple closure error in special error mode. The FIFO status word format is as shown in figure 8.28.

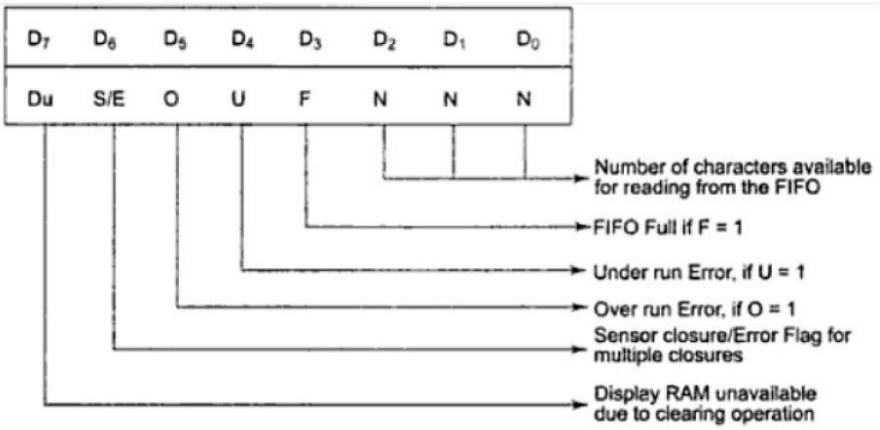


Figure 8.28: FIFO status word format

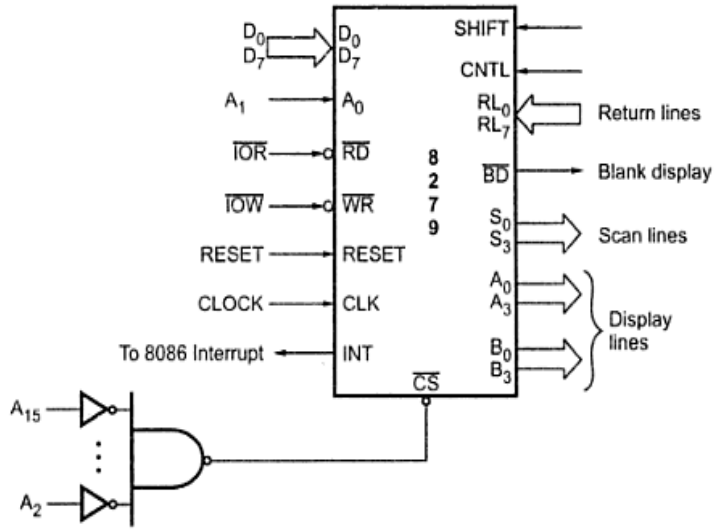
Self Assessment Questions

6. Select Intel's general purpose Keyboard/Display controller.
   a) 8279
   b) 8253
   c) 8255
   d) 8269
7. IRQ interrupt output line goes low when there is data in the FIFO sensor RAM. (True/False)
8. _____ divides the operating frequency of 8279 to generate scan keyboard and scan display frequencies.

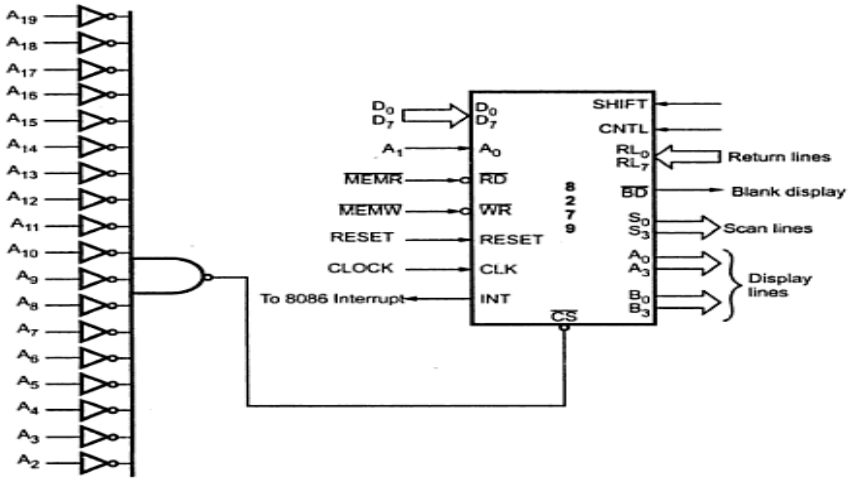
9. The clock for operation of 8279 is obtained by dividing the external clock input signal by a programmable constant called _____.
10. The _____ status word is used in keyboard and strobed input mode to indicate the error.

8.5 Interfacing and Programming 8279

The figure 8.29(a) shows the interfacing of 8279 in I/O mapped I/O and figure 8.29(b) shows the interfacing of 8279 in memory mapped I/O.



(a)



(b)

Figure 8.29: interfacing of 8279 in (a) I/O mapped I/O (b) Memory mapped I/O.

The I/O map is shown in figure 8.30.

Data/ Control Register	Address lines																			Address
	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Address
Data Register	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00H
Control Register	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	02H

Figure 8.30: I/O map

Problem:

Interface keyboard and display controller 8279 with 8086 at address 0080H. Write an ALP to set up 8279 in scanned keyboard mode with encoded scan, N-Key rollover mode. Use a 16 character display in right entry display format. Then clear the display RAM with zeros. Read the FIFO for key closure. If any key is closed, store its code to register CL. Then write the byte 55 to all the displays, and return to DOS. The clock input to 8279 is 2MHz, operate it at 100 KHz.

Solution:

The 8279 is interfaced with lower byte of the data bus, i.e. D0-D7. Hence the A0 input of 8279 is connected with address line A1.

The data register of 8279 is to be addressed as 0080H, i.e. A0=0.

For addressing the command or status word A0 input of 8279 should be 1. The next step is to write all the required command words for this problem.

Figure 8.31 shows the interfacing schematic.

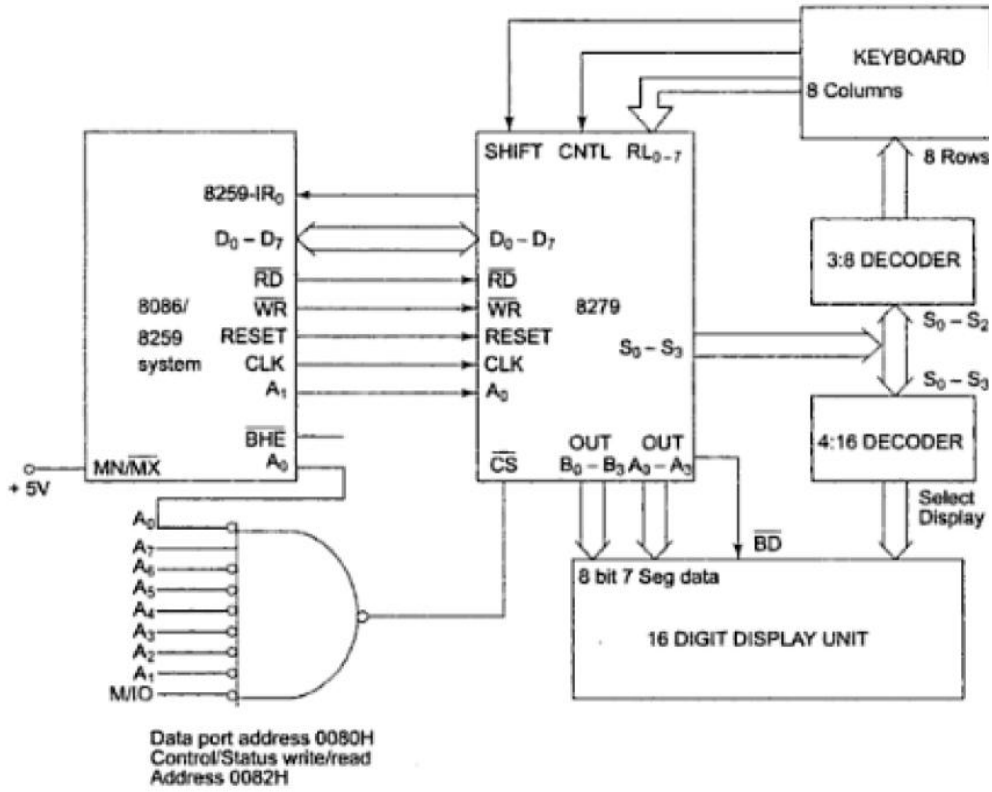
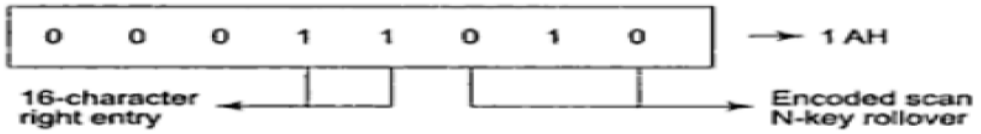


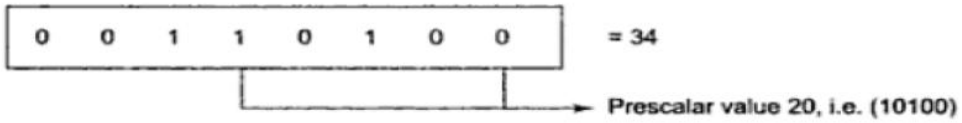
Figure 8.31: Interfacing schematic with 8086

Keyboard/Display Mode Set Command Word (CW): This command byte sets the 8279 in 16-character right entry and encoded scan N-Key rollover mode.



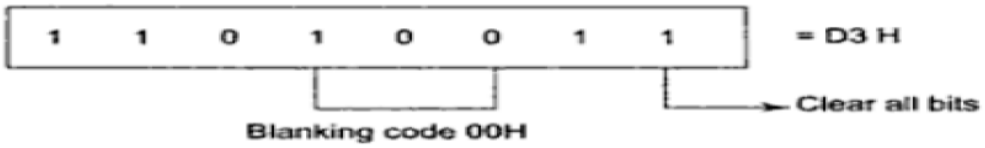
Program clock selection:

The clock input to 8279 is 2MHz, but the operating frequency is to be 100KHz, i.e. the clock input is to be divided by 20 (10100). Thus the prescalar value is 10100 and the command byte is set as:



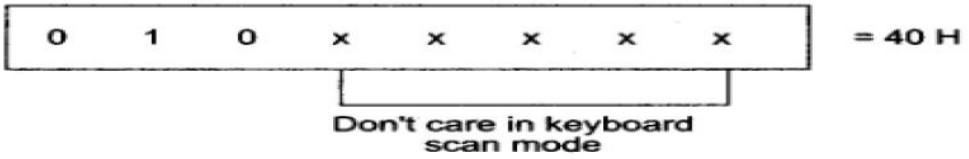
Clear Display RAM:

This command clears the display RAM with the programmable blanking code:



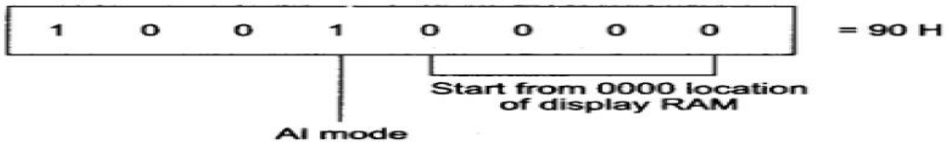
Read FIFO:

This command byte enables the programmer to read a key code from the FIFO RAM



Write Display RAM:

This command enables the programmer to write the addressed display locations of the RAM as presented below.



Program gives the ALP required to initialize the 8279 as required:

```
    Assume CS           : Code
    Code Segment

Start:  MOV AL, 1AH     ; Set 8279 in Encoded scan,
        OUT 82H, AL     ; N Key rollover, 16 display, Right entry mode.
        MOV AL, 34H     ; Set clock prescalar to
        OUT 82H, AL     ; 100 KHz
        MOV AL, 0D3H    ; Clear display ram
        OUT 82H, AL     ; command
        MOV AL, 40H     ; Read FIFO command
        OUT 82H, AL     ; for checking display RAM
```

```
Wait:   IN AL, 82H          ; wait for clearing of
        AND AL, 80H         ; Display RAM by reading
        CMP AH, 80H         ; FIFO Du bit of the status word i.e.
        JNZ Wait            ; If Du bit is not set wait, else proceed.
        MOV AH, 40H         ; Read FIFO command
        OUT 82H, AL         ; for check key closure
        IN AL, 82H          ; Read FIFO status
        AND AH, 07H         ; Mask all bits except the
        CMP AH, 00          ; number of characters bits
        JNZ Key code        ; if any key is pressed, take
Warm: MOV AL, 90H           ; required action, otherwise
        OUT 82H, AL         ; Proceed to write display
        MOV AL, 55H         ; RAM by using write display
        MOV CL, 10H         ; command. Write the byte
Next:   OUT 80H, AL         ; 55H to all display RAM
        DEC CL              ; Locations
        JNZ Next;
        JMP Stop;
Key code: Call Read code    ; Call routine to read the key
JMP Warm                    ; Code of the pressed key is assumed
                              available
Stop:   MOV AH, 4CH ; stop
        INT 21H
  Code ENDS
END START
```

8.6 Stepper Motor Interfacing

A stepper motor is device used to obtain an accurate position control of rotating shafts. In applications such as disk drives, dot matrix printers, and robotics, the stepper motor is used for position control. Stepper motor divides a full rotation into a number of equal steps. The motor's position can then be commanded to move and hold at one of these steps without any feedback sensor, as long as the motor is carefully sized to the application. The figure 8.32 shows the diagram of stepper motor.

Figure 8.32: Stepper Motor

A stepper motor is stepped from one position to the next by changing the currents through the fields in the motor. The two common field connections are referred to as two phase or four phase. There are three main areas of applications for stepper motor.

i. Instrumentation
ii. Computer peripherals
iii. Machine drives.

They are used in floppy drives, dot-matrix printers, X-Y plotters, digital watches etc. to rotate things in steps of small angles. The step size in typical stepper motor varies from 0.9^0 to 30^0. A stepper motor is a device used to obtain an accurate position control of rotating shafts. A stepper motor employs rotation of its shaft in terms of steps, rather than continuous rotation as in case of AC or DC motors. If you want to rotate the shaft of the stepper motor, then a sequence of pulses are required to be applied to the windings of the stepper motor, in proper sequence. The number of pulses required for one complete rotation of the shaft of the stepper motor are equal to its number of internal teeth on its rotor. The stator teeth and the rotor teeth lock with each other to fix a position of the shaft. With a pulse applied to the winding input, the rotor rotates by one teeth position or an angle x.

The angle x may be calculated as.

x =360° /no. of rotor teeth.

After the rotation of the shaft through angle x the rotor locks itself with the next tooth in the sequence on the internal surface of stator. The internal schematic of a typical stepper motor with four windings is shown in figure 8.33.

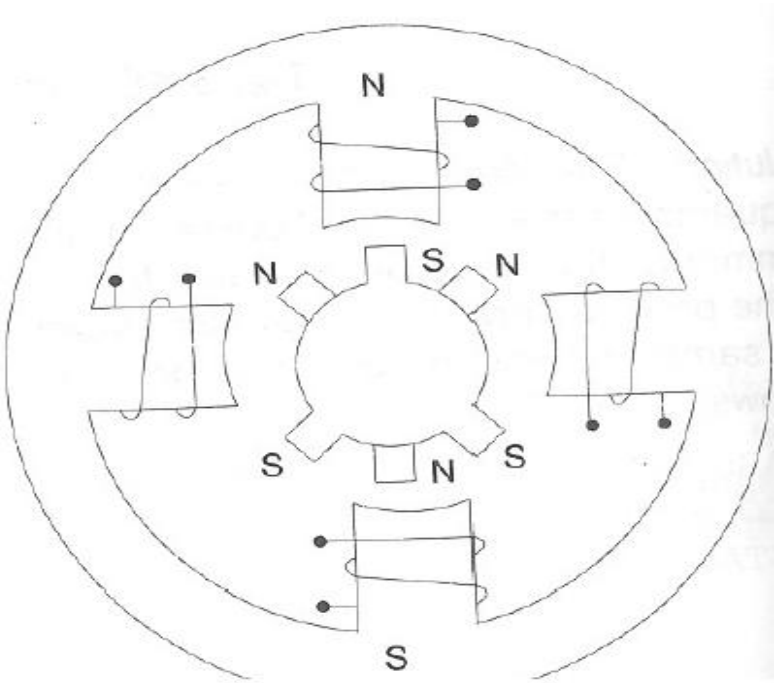


Figure 8.33: Internal Schematic of a Four Winding Stepper Motor

The stepper motors have been designed to work with digital circuits. Binary level pulses of 0-5V are required at its winding inputs to obtain the rotation of shafts. The sequence of the pulses can be decided, depending upon the required motion of the shaft. Figure 8.34 shows a typical winding arrangement of the stepper motor. Figure 8.35 shows conceptual positioning of the rotor teeth on the surface of rotor, for a six teeth rotor.

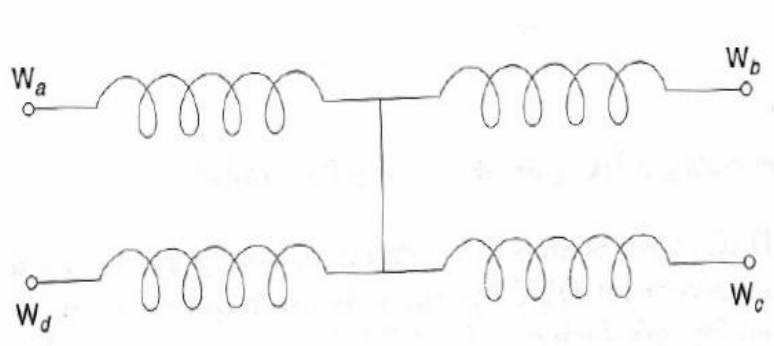


Figure 8.34: Winding Arrangement of a Stepper Motor

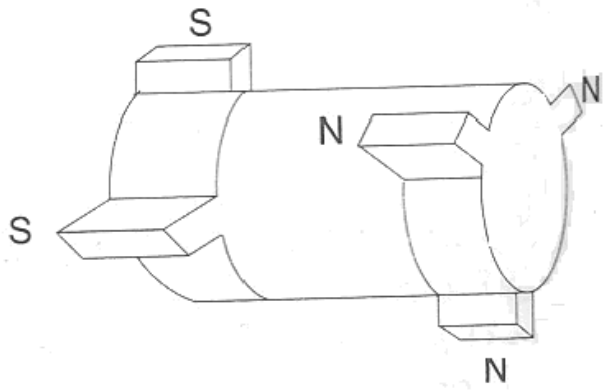


Figure 8.35: A Stepper Motor Ration

A typical stepper motor may have parameters like operating voltage 12V, current rating 1.2A and a step angle 1.80, i.e. 200 steps/revolution (number of rotor teeth).

A simple scheme for rotating the shaft of a stepper motor is called as wave scheme. In this scheme, the windings Wa, Wb, We and Wd are applied with the required voltage pulses, in a cyclic fashion. By reversing the sequence of excitation, the direction of rotation of the stepper motor shaft may be reversed.

Table 8.2 shows the excitation sequences for clockwise and anticlockwise rotations. Another popular scheme for rotation of a stepper motor shaft applies pulses to two successive windings at a time but these are shifted only by one position at a time. This scheme for rotation of stepper motor shaft is shown in table 8.2.

Table 8.2: Excitation Sequences of a Stepper Motor Using Wave Switching Scheme

Motion	Step	A	B	C	D
Clockwise	1	1	0	0	0
	2	0	1	0	0
	3	0	0	1	0
	4	0	0	0	1
	5	1	0	0	0
Anticlock wise	1	1	0	0	0
	2	0	0	0	1
	3	0	0	1	0
	4	0	1	0	0
	5	1	0	0	0

Figure 8.36 shows the Interfacing Stepper Motor winding.

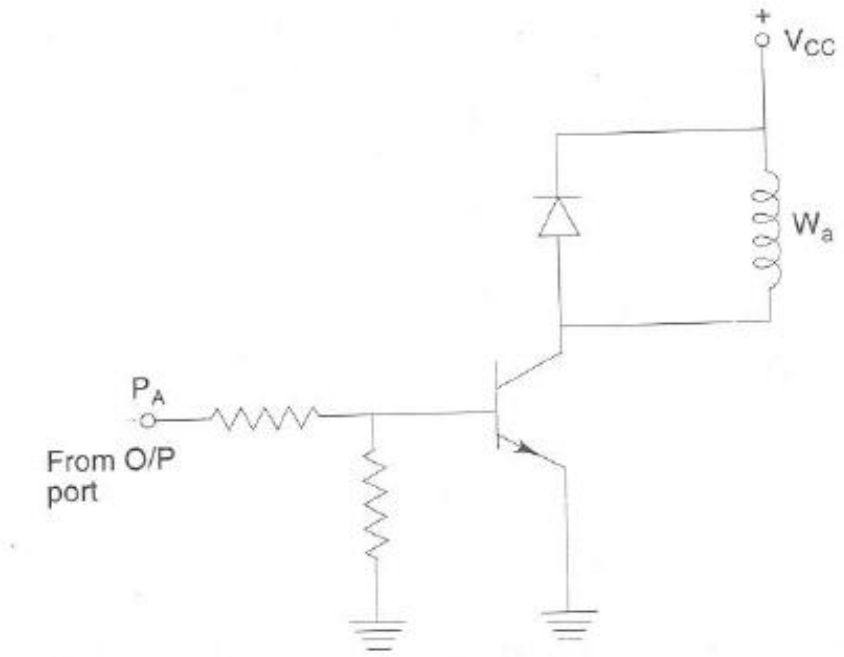


Figure 8.36: Interfacing Stepper Motor winding

```
DATA SEGMENT
        PORTA EQU 0C800H
        PORTB EQU 0C801H
        PORTC EQU 0C802H
        CWR EQU 0C803H
        PHA EQU 077H
        PHB EQU 0BBH
        PHC EQU 0DDH
        PHD EQU 0EEH
DATA ENDS

CODE SEGMENT
        ASSUME CS  : CODE, DS: DATA
START:  MOV AX, DATA
        MOV DS, AX
        MOV DX, CWR
        MOV AL, 80H
        OUT DX, AL
AGAIN:  MOV AL, PHA
        CALL STEP
        MOV AL, PHB
        CALL STEP
        MOV AL, PHC
        CALL STEP
        MOV AL, PHD
        CALL STEP
        MOV BL, 0FFH
X:      MOV CX, 0FFFFH
X1:     LOOP X1
        DEC BL
        JNZ X
        MOV AH, 0BH
        INT 21H
        OR AL, AL
        JZ AGAIN
        MOV AH, 4CH
        INT 21H
```

```
STEP PROC NEAR
            MOV DX, PORTC
            OUT DX, AL
            MOV BL, 60H
K1:         MOV CX, 0FFFFH
K2:         LOOP K2
            DEC BL
            JNZ K1
            RET
STEP ENDP
CODE ENDS
END START
```

Self Assessment Questions

11. _____ command byte sets the 8279 in 16-character right entry and encoded scan N-Key rollover mode.
12. A stepper motor is a device used to obtain an accurate position control of rotating shafts. (True/False)
13. A stepper motor is stepped from one position to the next by changing the _____ through the fields in the motor.
14. A simple scheme called wave scheme, the windings Wa, Wb, We and Wd are applied with the required voltage pulses, in a _____ fashion.

8.7 Summary

Let us recapitulate the important concepts discussed in this unit:

- Key bouncing problem can be solved by using software or hardware debouncing.
- In the software approach, when a key is pressed, the microprocessor waits for at least 10 ms before it accepts the key as an input.
- In matrix keyboard interface, the lines connected to rows are called returned lines and the lines connected to columns are called scan lines.
- In keyboard interfacing using 8255 PPI, the 8086 is being used in maximum mode and port A of 8255 is used for columns and port B for rows.

- Seven segment display is an example of LED displays. LEDs are very popular display devices.
- 8279 is an Intel's general purpose Keyboard/Display controller.
- In 8279 the Keyboard entries are debounced and stored in an 8-byte FIFO RAM and the processor reads the key codes.
- In Display Scan mode, 8279 provides 8 or 16 character multiplexed displays those can be organized as dual 4-bit or single 8-bit display units.
- The clock for operation of 8279 is obtained by dividing the external clock input signal by a programmable constant called prescaler.
- Stepper motor is a brushless DC electric motor that divides a full rotation into a number of equal steps.
- Stepper motors are used in floppy drives, dot-matrix printers, X-Y plotters, digital watches etc to rotate things in steps of small angles.

8.8 Terminal Questions

1. Why key bouncing happens? Explain how you can solve this problem.
2. Explain the concept of simple keyboard interfacing.
3. Explain the interfacing of keyboard with 8086.
4. Discuss about interfacing LED display
5. Draw the functional block diagram of 8279 and explain the functions of various blocks.
6. Discuss about the interfacing of stepper motor with 8086

8.9 Answers

Self Assessment Questions

1. Bouncing
2. NAND
3. Returned, scan
4. LED
5. IC 7447
6. a) 8279
7. False
8. Scan Counter
9. Prescaler

10. FIFO
11. 1AH
12. True
13. Currents
14. Cyclic

Terminal Questions

1. Key Bouncing happens because of the tendency of any two metal contacts in an electronic device to generate multiple signals as the contacts close or open. Refer to section 8.2 for details.
2. In simple keyboard interface, when port pin is logic 1, key is open, otherwise key is closed. Refer to section 8.2 for details.
3. The interfacing is done using 8255 PPI. The 8086 is being used in maximum mode and port A of 8255 issued for columns and port B for rows. Refer to section 8.2 for details.
4. Display is used for displaying letters, characters, numbers or information to the outside world/users. Refer to section 8.3 for details.
5. 8279 is an Intel's general purpose Keyboard/Display controller. It simultaneously drives the display of a system and interfaces a Keyboard. Refer to section 8.4 for details.
6. A stepper motor is stepped from one position to the next by changing the currents through the fields in the motor. Refer to section 8.6 for details.